Smart Innovation, Systems and Technologies 20

For further volumes:
http://www.springer.com/series/8767

Ruay-Shiung Chang, Lakhmi C. Jain,
and Sheng-Lung Peng (Eds.)

Advances in Intelligent Systems and Applications – Volume 1

Proceedings of the International Computer Symposium ICS 2012 Held at Hualien, Taiwan, December 12–14, 2012

Springer

Editors

Prof. Ruay-Shiung Chang
Department of Computer Science and
Information Engineering
National Dong Hwa University
Hualien
Taiwan
Republic of China

Prof. Sheng-Lung Peng
Department of Computer Science and
Information Engineering
National Dong Hwa University
Hualien
Taiwan
Republic of China

Dr. Lakhmi C. Jain
Adjunct Professor
University of Canberra
ACT 2601
Australia

ISSN 2190-3018
e-ISSN 2190-3026
ISBN 978-3-642-35451-9
e-ISBN 978-3-642-35452-6
DOI 10.1007/978-3-642-35452-6
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012953385

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The field of Intelligent Systems and Applications has expanded enormously during the last two decades. Theoretical and practical results in this area are growing rapidly due to many successful applications and new theories derived from many diverse problems. This book is dedicated to the proceedings of International Computer Symposium (ICS). ICS is a biennial event and is one of the largest joint international IT symposiums held in Taiwan. Founded in 1973, its aim was to provide a forum for researchers, educators, and professionals to exchange their discoveries and practices, and to explore future trends and applications in computer technologies. ICS 2012 consists of twelve workshops. Totally, we received 257 submissions. The Program Committee finally selected 150 papers for presentation at the symposium. This volume contains papers from the following workshops. We would like to express our gratitude to all of the authors, the reviewers, and the attendees for their contributions and participation.

- Workshop on Algorithms, Bioinformatics, and Computation Theory
- Workshop on Artificial Intelligence and Fuzzy Systems
- Workshop on Computer Networks and Web Service/Technologies
- Workshop on Database, Data Mining, and Information Retrieval
- Workshop on Information Literacy, e-Learning, and Social Media
- Workshop on Mobile Computing, Wireless Communications, and Vehicular Technologies

In ICS 2012, we are very pleased to have the following four distinguished invited speakers, who delivered state-of-the-art information on the conference topics:

- Professor Fedor V. Fomin from University of Bergen, Norway
- Professor L. Harn from University of Missouri-Kansas City, USA
- Professor C.-C. Jay Kuo from University of Southern California, USA
- Mr. Michael Wang, an Enterprise Architect, from Oracle, USA

ICS 2012 would not have been possible without the support of many people and organizations that helped in various ways to make it a success. In particular, we would like to thank the Ministry of Education of ROC (especially, the Computer Center of the MOE), National Science Council of ROC, Computer Audit Association of ROC, and Taiwan Association of Cloud Computing for their assistance and financial supports.

December 2012

Ruay-Shiung Chang
Lakhmi C. Jain
Sheng-Lung Peng

Organization

Honorary Chairs

Wei-ning Chiang	Ministry of Education, Taiwan
Cyrus Chin-yi Chu	National Science Council, Taiwan
Han-Chieh Chao	Computer Audit Association, Taiwan
Maw-Kuen Wu	National Dong Hwa University, Taiwan

Conference Chairs

Rong-Guey Ho	Computer Center of Ministry of Education, Taiwan
Wei-Pang Yang	National Dong Hwa University, Taiwan
Ruay-Shiung Chang	National Dong Hwa University, Taiwan

Conference Co-Chairs

Shin-Feng Lin	National Dong Hwa University, Taiwan
Chenn-Jung Huang	National Dong Hwa University, Taiwan

Program Chairs

Ching-Nung Yang	National Dong Hwa University, Taiwan
Shiow-Yang Wu	National Dong Hwa University, Taiwan
Sheng-Lung Peng	National Dong Hwa University, Taiwan

Publicity Chairs

Cheng-Chin Chiang	National Dong Hwa University, Taiwan
Han-Ying Kao	National Dong Hwa University, Taiwan
Chung Yung	National Dong Hwa University, Taiwan

Local Arrangement Chairs

Wen-Kai Tai	National Dong Hwa University, Taiwan
Shi-Jim Yen	National Dong Hwa University, Taiwan
Min-Xiou Chen	National Dong Hwa University, Taiwan
Shou-Chih Lo	National Dong Hwa University, Taiwan
Chih-Hung Lai	National Dong Hwa University, Taiwan

Publication Chairs

Chia-Chen Lin	Providence University, Taiwan
Chang-Hsiung Tsai	National Dong Hwa University, Taiwan
I-Cheng Chang	National Dong Hwa University, Taiwan
Pao-Lien Lai	National Dong Hwa University, Taiwan

Registration Chairs

Guanling Lee	National Dong Hwa University, Taiwan
Mau-Tsuen Yang	National Dong Hwa University, Taiwan

Web Chairs

Chenn-Jung Huang	National Dong Hwa University, Taiwan
Hsin-Chou Chi	National Dong Hwa University, Taiwan
Tao-Ku Chang	National Dong Hwa University, Taiwan

Workshop on Algorithms, Bioinformatics, and Computation Theory

Chairs

Jimmy J.M. Tan	National Chiao Tung University, Taiwan
Lih-Hsing Hsu	Providence University, Taiwan

Co-Chairs

Chang-Hsiung Tsai	National Dong Hwa University, Taiwan
Pao-Lien Lai	National Dong Hwa University, Taiwan

Program Committee

Tatsuya Akutsu	Kyoto University, Japan
Jou-Ming Chang	National Taipei College of Business, Taiwan
Maw-Shang Chang	Hungkuang University, Taiwan
Kun-Mao Chao	National Taiwan University, Taiwan
Francis Y.L. Chin	University of Hong Kong, Hong Kong
Jianxi Fan	Soochow University, China
Van Bang Le	Universität Rostock, Germany
Rey-Long Liu	Tzu Chi University, Taiwan
Chin Lung Lu	National Tsing Hua University, Taiwan
Sun-Yuan Hsieh	National Cheng Kung University, Taiwan
Tsan-sheng Hsu	Academia Sinica, Taiwan
Iain A. Stewart	University of Durham, U.K.
Chuan Yi Tang	Providence University, Taiwan
Yue-Li Wang	National Taiwan University of Science and Technology, Taiwan
Xiaofan Yang	Chongqing University, China

Workshop on Artificial Intelligence and Fuzzy Systems

Chair

Chang-Shing Lee	National University of Tainan, Taiwan

Co-Chairs

Shi-Jim Yen	National Dong Hwa University, Taiwan
Wen-Cheng Lin	Tzu Chi University, Taiwan

Program Committee

Hussein Abbass	University of New Wales, Australia
Keh-Hsun Chen	University of North Carolina at Charlotte, USA
Remi Coulom	University Charles de Gaulle, France
Damien Ernst	University Liege, Belgium
Hani Hagras	University of Essex, UK
Tzung-Pei Hong	National University of Kaohsiung, Taiwan
Shun-Chin Hsu	Chang Jung Christian University, Taiwan
Shinho Hsu	Northeastern University, China
June-Jei Kuo	National Chung Hsing University, Taiwan
Yue-Shi Lee	Ming-Chung University, Taiwan
Vincenzo Loia	University of Salerno, Italy
Hideki Kato	University of Tokyo, Japan
Masakazu Muramatsu	The University of Electrocommunications, Japan
Hiroshi Tsuji	Osaka Prefecture University, Japan
Marek Reformat	University of Albert, Canada
Olivier Teytaud	INRIA, France
I-Chen Wu	National Chiao Tung University, Taiwan

Workshop on Database, Data Mining, and Information Retrieval

Chair

Arbee L.P. Chen	National Chengchi University, Taiwan

Co-Chairs

Guanling Lee	National Dong Hwa University, Taiwan
Han-Ying Kao	National Dong Hwa University, Taiwan

Program Committee

Chia-Hui Chang	National Central University, Taiwan
Shu-Yuan Chen	Yuan Ze University, Taiwan
Yi-Shin Chen	National Tsing Hua University, Taiwan
Li-Chen Cheng	Soochow University, Taiwan
Been-Chian Chien	National University of Tainan, Taiwan
Tuan-Fang Fan	National Penghu University of Science and Technology, Taiwan
Chia-Hui Huang	National Taipei College of Business, Taiwan
Yu-Ling Hsueh	National Chung Cheng University, Taiwan
Chih-Kun Ke	National Taichung University of Science and Technology, Taiwan
Jia-Ling Koh	National Taiwan Normal University, Taiwan
Chin-Hui Lai	Chung Yuan Christian University, Taiwan
Chiang Lee	National Cheng Kung University, Taiwan
Yuh-Jye Lee	National Taiwan University, Taiwan
Hsuan-Tien Lin	National Taiwan University,Taiwan
Chuan-Ming Liu	National Taipei University of Technology, Taiwan
Yu-Chin Liu	Shin Hsin University, Taiwan
Man-Kwan Shan	National Cheng Chi University, Taiwan
Wen-Chih Peng	National Chiao Tung University, Taiwan
Yi-Hung Wu	Chung Yuan Christian University, Taiwan

Workshop on Information Literacy, e-Learning, and Social Media

Chair

An-Chi Liu	Feng Chia University, Taiwan

Co-Chairs

Kun-Huang Huarng	Feng Chia University, Taiwan
Chih-Hung Lai	National Dong Hwa University, Taiwan
Sheng-Lung Peng	National Dong Hwa University, Taiwan

Program Committee

Chih-Ming Chen	National Chengchi University, Taiwan
Lin Ching Chen	National Chiayi University, Taiwan
Chien Chou	National Chiao Tung University, Taiwan
Chih-Yueh Chou	Yuan Ze University, Taiwan
Raymond Greenlaw	The United States Naval Academy, USA
Tsukasa Hirashima	Hiroshima University, Japan
Pao-Nuan Hsieh	National Taiwan University, Taiwan
Sanpawat Kantabutra	Chiang Mai University, Thailand
Holin Lin	National Taiwan University, Taiwan
Yi-Chia Tsai	Tamkang University, Taiwan
Lung Hsiang Wong	Nanyang Technological University, Singapore
Bang Ye Wu	National Chung Cheng University, Taiwan
Mei-Mei Wu	National Taiwan Normal University, Taiwan

Workshop on Computer Networks and Web Service/Technologies

Chair

Kwei-Jay Lin	University of California, Irvine, USA

Co-Chairs

Tao-Ku Chang	National Dong Hwa University, Taiwan
Min-Xiou Chen	National Dong Hwa University, Taiwan
Der-Jiunn Deng	National Changhua University of Education, Taiwan

Program Committee

Ben-Jye Chang	National Yunlin University of Science and Technology, Taiwan
Chin-Chih Chang	Chung Hua University, Taiwan
Ing-Chau Chang	National Changhua University of Education, Taiwan
I-Fen Chao	Yuan Ze University, Taiwan
Yen-Cheng Chen	National Chi Nan University, Taiwan
Ling-Jyh Chen	Institute of Information Science Academia Sinica, Taiwan
Yu-Liang Chi	Chung Yuan Christian University, Taiwan
Cheng-Fu Chou	National Taiwan University, Taiwan
Gwan-Hwan Hwang	National Taiwan Normal University, Taiwan
Kuo-Yu Liu	Providence University, Taiwan
Hung-Chi Su	Arkansas State University, USA
Tzu-Chieh Tsai	National Chengchi University, Taiwan
Judy C.R. Tseng	Chung Hua University, Taiwan
Hsiao-Kuang Wu	National Central University, Taiwan
Chun-Chuan Yang	National Chi Nan University, Taiwan

Workshop on Mobile Computing, Wireless Communications, and Vehicular Technologies

Chair

Sy-Yen Kuo	National Taiwan University, Taiwan

Co-Chairs

Jiann-Liang Chen	National Taiwan University of Science and Technology, Taiwan
Shou-Chih Lo	National Dong Hwa University, Taiwan

Program Committee

Chih-Yung Chang	Tamkang University, Taiwan
Yao-Chung Chang	National Taitung University, Taiwan
Hsi-Lu Chao	National Chiao Tung University, Taiwan
Jen-Yeu Chen	National Dong Hwa University, Taiwan
Jyh-Cheng Chen	National Chiao Tung University, Taiwan
Yuh-Shyan Chen	National Taipei University, Taiwan
Cheng-Fu Chou	National Taiwan University, Taiwan
Li-Der Chou	National Central University, Taiwan
Kien A. Hua	University of Central Florida, USA
Jeng-Ji Huang	National Taiwan Normal University, Taiwan
Ren-Hung Hwang	National Chung Cheng University, Taiwan
Wei-Kuang Lai	National Sun Yat-sen University, Taiwan
Kun-Chan Lan	National Cheng Kung University, Taiwan
Victor Leung	University of British Columbia, Canada
Phone Lin	National Taiwan University, Taiwan
Jin Mitsugi	Keio University, Japan
Jang-Ping Sheu	National Tsing Hua University, Taiwan
Chih-Cheng Tseng	National Ilan University, Taiwan
Shun-Ren Yang	National Tsing Hua University, Taiwan

Additional Reviewers

Bae-Muu Chang
Chia-Hui Chang
Chien-Ping Chang
Chih-Yung Chang
Huei-Chu Chang
Chien-Liang Chen
Jen-Yeu Chen
Y-Chuang Chen
Yen Hung Chen
Rung-Shiang Cheng
Hsin-Hung Chou
Tung-Yang Ho
Hong-Chun Hsu
Chun Nan Hung
Ping Sheu Jang
Jia-Ling Koh
Wen Chung Kuo
Yuang-ling Lai
Wen-Hsiang Lai
Chihfeng P. Lin
Feng-Jyh Lin
Shun-Shii Lin
Jin Mitsugi
Lun-Min Shih
Yuan-Kang Shih
Mei Ling Wang
Chu-Sing Yang
C. Eugene Yeh
Li-Hsing Yen

Contents

Keynote Speeches

Track 1: Graph Theory and Algorithms

Track 2: Interconnection Networks and Combinatorial Algorithms

Track 3: Artificial Intelligence and Fuzzy Systems

Track 4: Database, Data Mining, and Information Retrieval

Track 5: Information Literacy, e-Learning, and Social Media

Track 6: Computer Networks and Web Service/Technologies

Track 7: Wireless Sensor Networks

Track 8: Wireless Network Protocols

Track 9: Wireless Data Processing

Kernelization Algorithms

Fedor V. Fomin[*]

Department of Informatics, University of Bergen, N-5020 Bergen, Norway
fomin@ii.uib.no

1 Introduction

Preprocessing or data reductions means reducing the input to something simpler by solving an easy part of the input and this is the type of algorithms used in almost every application. In spite of wide practical applications of preprocessing, a systematic theoretical study of such algorithms remains elusive. The framework of parameterized complexity can be used as an approach to analyse preprocessing algorithms. Input to parameterized algorithms include a parameter (in addition to the input) which is likely to be small, and this resulted in a study of preprocessing algorithms that reduce the size of the input to a pure function of the parameter (independent of the input size). Such type of preprocessing algorithms are called kernelization algorithms.

In the talk we discuss some of the classical and recent algorithmic techniques for obtaining kernels. We do not try to give a comprehensive overview of all significant results in the area—doing this will require at least a book. We refer to the surveys of Fellows [4] and Guo and Niedermeier [4, 6] for further reading on kernelization algorithms. We also do not discuss here techniques for deriving lower bounds on the sizes of the kernels. We refer to fairly comprehensive survey of Misra et al. [7] on the kernelization intractability.

Examples of kernelization: In parameterized complexity each problem instance comes with a parameter k. As a warm-up, let us consider the following parameterized examples. Our first example is about vertex cover. A set of vertices S in a graph is a vertex cover if every edge of the graph contains at least one vertex from S. In the parameterized version of vertex cover, we call it p-VERTEX COVER, we use $p-$ to emphasise that this is the parameterized problem, the parameter is integer k and we ask if the given graph has a vertex cover of size k. Another problem, p-LONGEST PATH asks if a given graph contains a path of length at least k. And finally, p-DOMINATING SET is to decide if a given graph has a dominating set of size k, i.e. a set of vertices such that every vertex of the input graph is either in this set or is adjacent to some vertex from the set.

The parameterized problem is said to admit a *kernel* if there is an algorithm that reduces the input instance down to an instance with size bounded by some function $h(k)$ of k only, while preserving the answer. The running time of this

[*] Supported by the European Research Council (ERC) advanced grant PREPROCESSING, reference 267959.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 1–5.
DOI: 10.1007/978-3-642-35452-6_1

algorithm should be polynomial in the input size and the degree of polynomial is independent of the parameter k. Such an algorithm is called a *kernelization* algorithm. If function $h(k)$ is polynomial in k, then we say that the problem admits a polynomial kernel.

In our examples, p-Vertex Cover admits a polynomial kernel—there is a polynomial time algorithm that for any instance (G, k) of the problem outputs a new instance (G', k') such that G' has at most $2k$ vertices and G has a vertex cover at most k if and only if G' has a vertex cover of size at most k' [2]. The second example, p-Longest Path, admits a kernel but the bounding function $h(k)$ is exponential. It is possible to show that up to some assumptions from complexity theory, the problem does not admit a polynomial kernel [1]. The problem does not admit a polynomial kernels even when the input graph G is planar. Finally, p-Dominating Set admits no kernel unless FPT=W[2], the collapse of several levels in parameterized complexity hierarchy [3]. However, on planar graph p-Dominating Set admits kernel with function $h(k) = \mathcal{O}(k)$, i.e. a linear kernel.

2 Basic Definitions

Here we mainly follow notations from the book of Flum and Grohe [5]. We describe decision problems as languages over a finite alphabet Σ.

Definition 1. *Let Σ be a finite alphabet.*

(1) A parameterization *of Σ^* is a polynomial time computable mapping $\kappa : \Sigma^* \to \mathbb{N}$.*
(2) A parameterized problem *(over Σ) is a pair (Q, κ) consisting of a set $Q \subseteq \Sigma^*$ of strings over Σ and a parameterization κ of Σ^*.*

For a parameterized problem (Q, κ) over alphabet Σ, we call the strings $x \in \Sigma^*$ the *instances* of Q or (Q, κ) and the number of $\kappa(x)$ the corresponding *parameters*. We usually represent a parameterized problem in the form

Instance: $x \in \Sigma^*$.
Parameter: $\kappa(x)$.
Problem: Decide whether $x \in Q$.

Very often the parameter is also a part of the instance. For example, consider the following parameterized version of the minimum feedback vertex set problem, where the instance consists of a graph G and a positive integer k, the problem is to decide whether G has a feedback vertex set, a set of vertices which removal destroys all cycles in the graph, of k elements.

p-Feedback Vertex Set
Instance: A graph G, and a non-negative integer k.
Parameter: k.
Problem: Decide whether G has a feedback vertex set
with at most k elements.

In this problem the instance is the string (G, k) and $\kappa(G, k) = k$. When the parameterization κ is defined as $\kappa(x, k) = k$, the paramerized problem can be defined as subsets of $\Sigma^* \times \mathbb{N}$. Here the parameter is the second component of the instance. In this survey we use both notations for parameterized problems.

The notion of kernelization is intimately linked with the notion of fixed-parameter tractability. Fixed-parameter tractable algorithms are a class of exact algorithms where the exponential blowup in the running time is restricted to a small parameter associated with the input size. That is, the running time of such an algorithm on an input of size n is of the form $\mathcal{O}(f(k)\, n^c)$, where k is a parameter that is typically small compared to n, $f(k)$ is a (typically super-polynomial) function of k that does not involve n, and c is a constant. Formally,

Definition 2. *A parameterized problem (Q, κ) is* fixed-parameter tractable *if there exists an algorithm that decides in $f(\kappa(x)) \cdot n^{\mathcal{O}(1)}$ time whether $x \in Q$, where $n := |x|$ and f is a computable function that does not depend on n. The algorithm is called a* fixed parameter algorithm *for the problem. The complexity class containing all fixed parameter tractable problems is called FPT.*

There is also a hierarchy of intractable parameterized problem classes above FPT, the main ones are:

$$FPT \subseteq M[1] \subseteq W[1] \subseteq M[2] \subseteq W[2] \subseteq \cdots \subseteq W[P] \subseteq XP$$

The principal analogue of the classical intractability class NP is $W[1]$, which is a strong analogue, because a fundamental problem complete for $W[1]$ is the k-Step Halting Problem for Nondeterministic Turing Machines (with unlimited nondeterminism and alphabet size) — this completeness result provides an analogue of Cook's Theorem in classical complexity. A convenient source of $W[1]$-hardness reductions is provided by the result that p-Clique is complete for $W[1]$. Other highlights of the theory include that p-Dominating Set, by contrast, is complete for $W[2]$. Another highlight is that $FPT = M[1]$ if and only if the *Exponential Time Hypothesis* fails [5]. The classical reference on Parameterized Complexity is the book of Downey and Fellows [3]. For more updated material we refer to books of Flum and Grohe [5] and Niedermeier [8].

The notion of *kernelization* is formally defined as follows.

Definition 3. *Let (Q, κ) be a parameterized problem over a finite alphabet Σ. A* kernelization algorithm*, or in short, a* kernelization*, for (Q, κ) is an algorithm that for any given $x \in \Sigma^*$ outputs in time polynomial in $|x| + \kappa(x)$ a string $x' \in \Sigma^*$ such that*

$$(x \in Q \iff x' \in Q) \text{ and } |x'|, |\kappa(x')| \leq h(\kappa(x)),$$

where h is an arbitrary computable function. If K is a kernelization of (Q, κ), then for every instance x of Q the result of running K on input x is called the kernel of x (under K). The function h is referred to as the size *of the kernel. If h is a polynomial function then we say the kernel is polynomial.*

We often say that a problem (Q, κ) admits a kernel of size h, meaning that every instance of Q has a kernel of size h. We also often say that (Q, κ) admits a kernel with property Π, meaning that every instance of Q has a kernel with property Π. For example, by saying p-Vertex Cover admits a kernel with $\mathcal{O}(k)$ vertices and $\mathcal{O}(k^2)$ edges, we mean that there is a kernelization algorithm K, such that for every instance (G, k) of the problem, there is a kernel with $\mathcal{O}(k)$ vertices and $\mathcal{O}(k^2)$ edges.

It is easy to see that if a decidable problem admits kernelization for some function f, then the problem is FPT—for every instance of the problem we run polynomial time kernelization algorithm and then use the decision algorithm to identify if this is the valid instance. Since the size of the kernel is bounded by some function of the parameter, the running time of the decision algorithm depends only on the parameter. Interestingly, the converse also holds, that is, if a problem is FPT then it admits a kernelization. The proof of this fact is quite simple, and we present it here.

Lemma 1 (Folklore, [5, 8]). *If a parameterized problem* (Q, κ) *is* FPT *then it admits kernelization.*

Proof. Suppose that there is an algorithm deciding if $x \in Q$ in time $f(\kappa(x))|x|^c$ time for some function f and constant c. If $|x| \geq f(\kappa(x))$, then we run the decision algorithm on the instance in time $f(\kappa(x))|x|^c \leq |x|^{c+1}$. If the decision algorithm outputs YES, the kernelization algorithm outputs a constant size YES instance, and if the decision algorithm outputs NO, the kernelization algorithm outputs a constant size NO instance. On the other hand, if $|x| < f(\kappa(x))$, then the kernelization algorithm outputs x. This yields a kernel of size $f(\kappa(x))$ for the problem.

Lemma 1 shows that kernelization can be seen as an alternative definition of fixed parameter tractable problems. However, we are interested in kernels that are as small as possible, and a kernel obtained using Lemma 1 has size that equals the dependence on k in the running time of the best known FPT algorithm for the problem. The question is—can we do better? The answer is that quite often we can. In fact, for many problems we can polynomial kernels. In this talk we survey some of the old and new techniques for showing that problems admit polynomial kernels.

References

[1] Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, J.: On problems without polynomial kernels. J. Comput. Syst. Sci. 75, 423–434 (2009)

[2] Chen, J., Kanj, I.A., Jia, W.: Vertex cover: further observations and further improvements. Journal of Algorithms 41, 280–301 (2001)

[3] Downey, R.G., Fellows, M.R.: Parameterized complexity. Springer, New York (1999)

[4] Fellows, M.R.: The Lost Continent of Polynomial Time: Preprocessing and Kernelization. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 276–277. Springer, Heidelberg (2006)

[5] Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)

[6] Guo, J., Niedermeier, R.: Invitation to data reduction and problem kernelization. SIGACT News 38, 31–45 (2007)

[7] Misra, N., Raman, V., Saurabh, S.: Lower bounds on kernelization. Discrete Optim. 8, 110–128 (2011)

[8] Niedermeier, R.: Invitation to fixed-parameter algorithms. Oxford Lecture Series in Mathematics and its Applications, vol. 31. Oxford University Press, Oxford (2006)

Characterizing and Recognizing Probe Block Graphs

Van Bang Le[1] and Sheng-Lung Peng[2,⋆]

[1] Universität Rostock, Institut für Informatik, Germany
le@informatik.uni-rostock.de
[2] Department of Computer Science and Information Engineering,
National Dong Hwa University, Shoufeng, Hualien 97401, Taiwan
slpeng@mail.ndhu.edu.tw

Abstract. Block graphs are graphs in which every block (biconnected component) is a clique. A graph $G = (V, E)$ is said to be an (unpartitioned) probe block graph if there exists an independent set $\mathbb{N} \subseteq V$ and some set $E' \subseteq \binom{\mathbb{N}}{2}$ such that the graph $G' = (V, E \cup E')$ is a block graph; if an independent set $\mathbb{N}$ is given, G is called a partitioned block graph. In this note we give good characterizations for probe block graphs, in both unpartitioned and partitioned cases. As a result, partitioned and unpartitioned probe block graphs can be recognized in linear time.

Keywords: Probe graph, block graph, probe block graph.

1 Introduction

Given a graph class $\mathcal{C}$, a graph $G = (V, E)$ is called a *probe* $\mathcal{C}$ *graph* if there exists an independent set $\mathbb{N} \subseteq V$ (of *non-probes*) and a set $E' \subseteq \binom{\mathbb{N}}{2}$ such that the graph $G' = (V, E \cup E')$ is in the class $\mathcal{C}$, where $\binom{\mathbb{N}}{2}$ stands for the set of all 2-element subsets of $\mathbb{N}$. A graph $G = (V, E)$ with a *given* independent set $\mathbb{N} \subseteq V$ is said to be a *partitioned probe* $\mathcal{C}$ *graph* if there exists a set $E' \subseteq \binom{\mathbb{N}}{2}$ such that the graph $G' = (V, E \cup E')$ is in the class $\mathcal{C}$. Recognizing partitioned probe $\mathcal{C}$ graphs is a special case of the $\mathcal{C}$-GRAPH SANDWICH problem (cf. [7]).

Probe graphs have been investigated for various graph classes; see [3] for more information. The case $\mathcal{C}$ is the class of block graphs has been addressed in [4], where the authors provided an $O(n^3)$ time recognition algorithm for unpartitioned probe block graphs and an $O(n^2)$ time recognition algorithm for partitioned probe block graphs; n is the number of vertices of the input graph. They also noted that they have not found a characterization of probe block graphs by forbidden induced subgraphs.

In this note we characterize partitioned and unpartitioned probe block graphs. Our characterizations are good in the sense that they imply linear-time recognition algorithms for probe block graphs, both in partitioned and unpartitioned cases. The main results of this note consist of

[⋆] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 7–13.
DOI: 10.1007/978-3-642-35452-6_2

- characterizations of partitioned and unpartitioned probe block graphs by forbidden induced subgraphs,
- a characterization of partitioned probe graphs by a certain "enhanced" graph, saying that G is a partitioned probe block graph if and only if the enhanced graph G^* is a block graph.

While the first result settles a question on characterizing probe block graphs in [4], the second one is analogous to some interesting results in the literature: [10] shows that a certain enhanced graph of an interval partitioned probe graph must be chordal. [5] shows that a bipartite graph without induced cycles of length > 6 is partitioned probe chordal-bipartite if a certain enhanced graph is chordal bipartite; [8] (see also [5]) shows that a graph without chordless cycles of length at least five is partitioned probe chordal if and only if a certain enhanced graph is chordal. [1] shows that a graph is a partitioned probe threshold (trivially perfect) graph if and only if a certain enhanced graph is a threshold (trivially perfect) graph, and [9] shows that a graph is a partitioned probe chain graph if and only if a certain enhanced graph is a chain graph; see [2,6] for graph classes not defined here.

Some Notion

A (connected or not) graph is a *block graph* if each of its maximal 2-connected components, *i.e.*, its blocks, is a clique. A *chordal graph* is one in which every cycle of length at least 4 has a chord. A *diamond* is the complete graph on four vertices minus an edge. It is well-known that block graphs are exactly the chordal graphs without induced diamond.

A *split graph* is one whose vertex set can be partitioned into a clique and an independent set. It is well-known that split graphs are exactly the chordal graphs without induced $2K_2$ (the 1-regular graph with four vertices). A *complete split graph* is a split graph $G = (V, E)$ admitting a partition $V = Q \cup S$ into clique Q and independent set S such that every vertex in Q is adjacent to every vertex in S. Such a partition is also called a *complete split partition* of a split graph. Note that if the complete split graph $G = (V, E)$ is not a clique, then G has exactly one complete split partition $V = Q \cup S$.

Let $G = (V, E)$ be a graph. For a vertex $v \in V$ we write $N(v)$ for the set of its neighbors in G. For a subset $U \subseteq V$ we write $G[U]$ for the subgraph of G induced by U and $G - U$ for the graph $G[V - W]$; for a vertex v we write $G - v$ rather than $G[V \setminus \{v\}]$.

2 Partitioned Probe Block Graphs

Let $G = (V, E)$ be a graph with a given independent set $\mathbb{N} \subseteq V$. Suppose there exists a set $E' \subseteq \binom{\mathbb{N}}{2}$ such that the graph $G = (V, E \cup E')$ is a block graph, that is, G is a partitioned probe block graph with respect to the given independent set $\mathbb{N}$. Then, clearly, any induced diamond in G must have exactly two vertices x, y in $\mathbb{N}$ and $\{x, y\}$ must belong to E'.

In what follows, given a graph $G = (V, E)$ together with an independent set $\mathbb{N} \subseteq V$, the *enhanced graph* $G^* = (V, E^*)$ is obtained from G by adding all edges between two vertices in $\mathbb{N}$ that are two vertices of an induced diamond in G.

Partitioned probe block graphs can be characterized as follows; see Fig. 1 for the graphs D_1, D_2 and D_3.

Theorem 1. *Let $G = (V, E)$ be a graph with a partition $V = \mathbb{P} \cup \mathbb{N}$, where $\mathbb{N}$ is an independent set. Then the following statements are equivalent.*

(i) *$G = (\mathbb{P} \cup \mathbb{N}, E)$ is a partitioned probe block graph;*
(ii) *G is a $\{D_1, D_2, D_3\}$-free chordal graph;*
(iii) *Every block B of G is a complete split graph with $B = (B \cap \mathbb{P}) \cup (B \cap \mathbb{N})$ a complete split partition;*
(iv) *G^* is a block graph.*

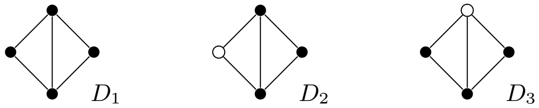


Fig. 1. Forbidden induced subgraphs for partitioned probe block graphs; white vertices are in $\mathbb{N}$

Proof. (i) ⇒ (ii): Let $G' = (V, E \cup E')$ be a block graph for some $E' \subseteq \binom{\mathbb{N}}{2}$. Consider a cycle $C : v_1 v_2 \ldots v_q v_1$ of length $q \geq 4$ in G. Then C is contained in a block of G', hence $G'[C]$ is a clique. Therefore, $v_1 v_3 \in E \cup E'$ and $v_2 v_4 \in E \cup E'$. Hence $v_1 v_3 \in E$ or $v_2 v_4 \in E$ because edges in E' have both endvertices in the independent set $\mathbb{N}$. Thus, $v_1 v_3$ or $v_2 v_4$ is a chord of C, showing that G is chordal. Furthermore, it is obvious that the graphs D_1, D_2 and D_3 depicted in Fig. 1 are not partitioned probe block graphs with respect to the black/white partition. Thus, G is $\{D_1, D_2, D_3\}$-free.

(ii) ⇒ (iii): We note first, that $G[\mathbb{P}]$ is a block graph because G is chordal and has no diamond D_1. Hence every block of $G[\mathbb{P}]$ is a clique. Next, we show that

$$\text{for every vertex } v \in \mathbb{N}, \text{ every non-trivial connected component of } N(v) \text{ is a block of } G[\mathbb{P}]. \tag{1}$$

To see (1), let A be a non-trivial connected component of $N(v)$. If A is not a clique, then, by connectedness, A contains an induced path $a_1 a_2 a_3$, which together with v induce a D_3, a contradiction. Thus, A is a clique in $G[\mathbb{P}]$. If A is not a maximal clique in $G[\mathbb{P}]$, then there is a vertex in $\mathbb{P} \setminus A$ adjacent to all vertices in A. This vertex together with v and two vertices in A induce a D_2, a contradiction. Thus, A is a maximal clique in $G[\mathbb{P}]$. Therefore, A is a block of the block graph $G[\mathbb{P}]$, hence (1).

Consider now a block B of G. We have

$$\text{for every } v \in B \cap \mathbb{N},\ N(v) \cap B \text{ is a clique.} \tag{2}$$

Otherwise, as $B-v$ is connected, there is a (chordless) path W in $B-v$ connecting two non-adjacent vertices in $N(v) \cap B$. Since G is chordal, v must be adjacent to all vertices on W. In particular, $W \subseteq B \cap \mathbb{P}$. Hence G has an induced D_3, a contradiction, and (2) follows. Moreover,

$$B \cap \mathbb{P} \text{ is a block of } G[\mathbb{P}]. \tag{3}$$

This is because of (2), if $B \cap \mathbb{P}$ were disconnected, then B was also disconnected, and if w was a cutvertex of $B \cap \mathbb{P}$, then w was also a cutvertex of B. That is, (3) holds.

Note that we may assume that B is not a clique (if B is a clique, $(B \cap \mathbb{P}) \cup (B \cap \mathbb{N})$ is clearly a complete split partition of B). Hence, for all $v \in B \cap \mathbb{N}$, $|N(v) \cap B| \geq 2$ (as B is 2-connected). Thus, by (2) and (1), the (non-trivial) connected component A of $N(v)$ containing $N(v) \cap B$ is a block of $G[\mathbb{P}]$. Therefore, the blocks $B \cap \mathbb{P}$ (by (3)) and A of $G[\mathbb{P}]$ are the same because they have $|N(v) \cap B| \geq 2$ vertices in common.

It follows, for all $v \in B \cap \mathbb{N}$, $N(v) \cap B = B \cap \mathbb{P}$. Since $B \cap \mathbb{P}$ is a clique, $(B \cap \mathbb{P}) \cup (B \cap \mathbb{N})$ is therefore the complete split partition of B.

(iii) $\Rightarrow$ (iv): If x and y are two vertices in $\mathbb{N}$ belonging to a diamond D then, as D is 2-connected, D is clearly contained in a block of G. Conversely, if B is a block of G with $|B \cap \mathbb{N}| \geq 2$, then any pair of vertices $x, y \in B \cap \mathbb{N}$ belongs to a diamond which is contained in B. Thus, G^* is obtained from G by adding all edges between vertices in $B \cap \mathbb{N}$, for each block B of G. Since, for block B of G, $(B \cap \mathbb{P}) \cup (B \cap \mathbb{N})$ is a complete split partition, each block of G^* is a clique. That is, G^* is a block graph.

(iv) $\Rightarrow$ (i): This implication is obvious. □

As the blocks of a graph can be computed in linear time (well-known), as well as checking if a given partition is a complete split partition can be done in linear time (obvious), Theorem 1 (iii) implies:

Theorem 2. *Partitioned probe block graphs can be recognized in linear time.*

3 Unpartitioned Probe Block Graphs

Probe block graphs can be characterized as follows; see Fig. 2 for the graphs F_1, F_2, F_3 and F_4.

Theorem 3. *G is a probe block graph if and only if G is an $\{F_1, F_2, F_3, F_4\}$-free chordal graph.*

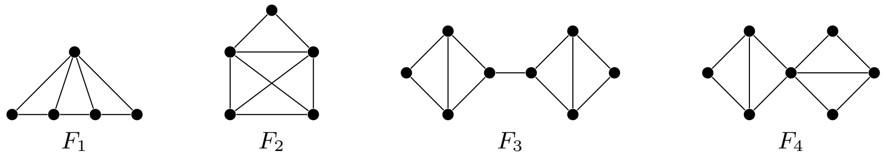


Fig. 2. Forbidden induced subgraphs for unpartitioned probe block graphs

Proof. Necessity: The proof of the chordality of G is the same as the one of Theorem 1, part (i) ⇒ (ii). Also, it is easy to check that the graphs $F_1, \ldots, F_4$ are not probe block graphs.

Sufficiency: We first show that

$$\text{every block of } G \text{ is a complete split graph.} \tag{4}$$

To see (4), let B be a block of G. We claim that, for any edge xy and any vertex v in B, v is adjacent to x or y. Note hat by the 2-connectedness, there exist in B two paths $W(v,x)$ and $W(v,y)$ connecting v and x, respectively, v and y, and having only the vertex v in common.

Now, suppose the contrary that B has an edge xy and a vertex v such that v is non-adjacent to both x and y. Consider such an edge xy and vertex v with the property that the paths $W(v,x)$ and $W(v,y)$ as short as possible. In particular, $W(v,x)$ and $W(v,y)$ are chordless. Write $W(v,x) = va_1a_2 \ldots a_rx$, $W(v,y) = vb_1b_2 \ldots b_sy$, for some $r \geq 1$ and $s \geq 1$. As v is non-adjacent to a_i, b_j, x, and y, $i \geq 2, j \geq 2$, and as G is chordal, a_1b_1 must be an edge.

Now, if $r = s = 1$, then $a_1y \in E$ or $b_1x \in E$ because G is chordal, hence v, a_1, b_1, x, y induce an F_1 or an F_2. So, let $r \geq 2$, say. Then a_1 must be adjacent to y, otherwise the vertex a_1 would violate the choice of v. Thus, by chordality, a_1 is adjacent to all b_j and y is adjacent to all a_i, implying $s = 1$ and $r = 2$ (otherwise G would have an induced F_1). But then v, a_1, a_2, b_1 and y induce an F_1 (if $a_2b_1 \notin E$) or an F_2 (if $a_2b_1 \in E$), a contradiction.

We have shown that B is $(K_2 \cup K_1)$-free ($K_2 \cup K_1$ is the graph with three vertices and one edge). In particular, B is a split graph. Let $B = Q \cup S$ be a partition into a maximal clique Q and an independent set S. As B is 2-connected, $|N(v) \cap B| \geq 2$ for all $v \in S$, and as B has no induced $K_2 \cup K_1$ and Q is maximal, every vertex $v \in S$ is non-adjacent to exactly one vertex in Q. It follows that *all* vertices $v \in S$ are non-adjacent to the same vertex in Q, say w (otherwise, G would have an F_1). Now, $B = Q_B \cup S_B$ with $Q_B = Q - w$ and $S_B = S + w$ is the complete split partition of B, and (4) is proved.

Let $\mathcal{B}$ denote the set of all blocks of G that are not cliques. For each block $B \in \mathcal{B}$ let $B = Q_B \cup S_B$ be the complete split partition of B with clique Q_B and independent set S_B. Note that $|Q_B| \geq 2$ and $|S_B| \geq 2$. We now show that

$$\mathbb{N} = \bigcup_{B \in \mathcal{B}} S_B \text{ is an independent set.} \tag{5}$$

Suppose, by contradiction, $\mathbb{N}$ contains two adjacent vertices x and y. Then $x \in S_B, y \in S_{B'}$ for some distinct blocks $B, B' \in \mathcal{B}$. Let w be a vertex in $S_B - x$, z be a vertex in $S_{B'} - y$. We consider two cases. If $xy \notin B \cup B'$, then w, x, y, z together two vertices in Q_B and two vertices in $Q_{B'}$ induce an F_3, a contradiction. If $xy \in B \cup B'$, say $xy \in B$, then $y \in Q_B$. Now w, x, y, z together with a vertex in $Q_B - y$ and two vertices in $Q_{B'}$ induce an F_4, a contradiction again, hence (5).

Write $\mathbb{P} = V - \mathbb{N}$. By (4) and (5), G with the partition $V = \mathbb{P} \cup \mathbb{N}$ satisfies Theorem 1 (iii), hence G is a probe block graph. □

Theorem 3 implies that probe block graphs can be recognized in polynomial time. Indeed, a brute-force algorithm that first tests if a graph G is chordal in linear time, then runs in time $O(n^8)$ to check if G contains an F_i, $1 \leq i \leq 4$, and finally to decide if G is a probe block graph. We can improve this complexity significantly by following the steps of the proof of the theorem:

1. Check if G is chordal; if not: stop with `no`
2. Compute the blocks of B, and let $\mathcal{B}$ be the set of all non-complete blocks
3. If some $B \in \mathcal{B}$ is not a complete split graph: stop with `no`
4. Put $\mathbb{N} := \bigcup_{B \in \mathcal{B}} S_B$, where $B = Q_B \cup S_B$ is the complete split partition of B
5. Check if $\mathbb{N}$ is an independent set; if not: stop with `no`; otherwise stop with `yes`

It is well-known that each of steps 1, 2 and 5 can be performed in linear time. Also, both steps 3 and 4 can be done easily: Take any maximal clique Q of B. B is a complete split graph if and only if there exists exactly one vertex $w \in Q$ such that $B = Q_B \cup S_B$ is the complete split partition of B, where $Q_B = Q - w$ and $S_Q = S + w$.

Thus, we obtain:

Theorem 4. *Probe block graphs can be recognized in linear time.*

References

1. Bayer, D., Le, V.B., de Ridder, E.: Probe Threshold Graphs and Probe Trivially Perfect Graphs. Theoretical Computer Science 410, 4812–4822 (2009)
2. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (1999)
3. Chandler, D.B., Chang, M.-S., Kloks, T., Peng, S.-L.: Probe Graphs (2009) (manuscript)
4. Chang, M.-S., Hung, L.-J., Kloks, T., Peng, S.-L.: Block-graph Width. Theoretical Computer Science 412, 2496–2502 (2011)
5. Cohen, E., Golumbic, M.C., Lipshteyn, M., Stern, M.: On the Bi-enhancement of Chordal-bipartite Probe Graphs. Information Processing Letters 110, 193–197 (2010)
6. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980); Annals of Discrete Math., 2nd edn, vol. 57. Elsevier, Amsterdam (2004)
7. Golumbic, M.C., Kaplan, H., Shamir, R.: Graph Sandwich Problems. Journal of Algorithms 19, 449–473 (1995)

8. Golumbic, M.C., Lipshteyn, M.: Chordal Probe Graphs. Discrete Applied Mathematics 143, 221–237 (2004)
9. Le, V.B.: Two Characterizations of Chain Partitioned Probe Graphs. Annals of Operations Research 188, 279–283 (2011)
10. Zhang, P.: Probe Interval Graphs and Their Application to Physical Mapping of DNA (1994) (manuscript)

An Upper Bound of the Rainbow Connection Number in RTCC Pyramids

Fu-Hsing Wang*, Ze-Jian Wu, and Yann-Jong Hwang

Department of Information Management, Chinese Culture University,
Taipei, Taiwan, R.O.C.
wang.fuhsing@gmail.com

Abstract. Rainbow connection number of a connected graph G is the minimum number of colors needed to color the edges of G, so that every pair of vertices is connected by at least one path whose edges have distinct colors. In this paper, we propose an upper bound to the size of the rainbow connection number in Recursive Transpose-Connected 4-Cycles (RTCC) pyramids.

Keywords: Graph theory, rainbow connection number, rainbow coloring, RTCC pyramids.

1 Introduction

Edge coloring of a graph is a function from its edge set to the set of natural numbers. A $u - v$ path in an edge colored graph with no two edges sharing the same color is called a *rainbow* $u - v$ *path.* An edge-colored graph G is *rainbow connected* if any two vertices are connected by a rainbow path. In this case, the edge coloring of G is called a *rainbow coloring* of G. A *rainbow* $r-$*coloring* of a graph is a rainbow coloring that uses r colors. The *rainbow connection number* of a connected graph G, denoted by $rc(G)$, is the smallest number of colors that are needed to make G rainbow connected.

The problem of rainbow connection in graphs was introduced by Chartrand et al. in [5] and has application in secure transfer of classified information between various agencies which may have other agencies as intermediaries by assigning passwords between agencies [6]. Every pair of agencies with one or more secure paths along with distinct passwords reveals the rainbow connection and is prohibitive to intruder [6].

Chakraborty et al. showed that computing the rainbow connection number of a general graph is NP-hard [4]. In fact, even deciding whether $rc(G) = 2$ holds for a graph G is an NP-complete problem [4]. An easy observation that $rc(G) = 1$ if and only if G is a complete graph, that $rc(G) = |V(G)| - 1$ if and only if G is a tree. In [3], Caro et al. gave sufficient conditions that guarantee $rc(G) = 2$ and determined a threshold function for a random graph to have $rc(G) = 2$. Also notice that $rc(G) \leq |V(G)| - 1$ for a general graph G, since one may color the

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 15–23.
DOI: 10.1007/978-3-642-35452-6_3

edges of a given spanning tree with distinct colors (and color the remaining edges with one of the already used colors). Clearly, $rc(G) \geq \mathrm{diam}(G)$ where $\mathrm{diam}(G)$ denotes the diameter of G. Most recent research has been devoted to study the bounds of the rainbow connection numbers on 3-connected graphs [9], connected bridgeless graphs [2], strongly regular graphs [1], triangular pyramid networks [12], etc. Chartrand et al. computed the precise rainbow connection number for certain special graphs, e.g., Peterson graphs and complete multi-partite graphs [5]. A good survey of rainbow connection problem can be found in [10].

We consider the problem for a new modular topology for interconnection networks, namely, Recursive Transpose-Connected 4-Cycles (RTCC) network. The RTCC pyramid is based on RTCC network instead of the two-dimensional square mesh employed by traditional pyramids. The RTCC network has a recursive definition quite similar to that of fractal graphs having interesting topological characteristics, making it suitable for utilization as the base topology of large-scale multicomputer interconnection networks. The RTCC network is superior to conventional topologies such as the mesh and cube [7]. In this paper, we propose a linear time algorithm for finding a rainbow path for every pair of vertices of a RTCC pyramid. As far as we know, no minimum rainbow coloring algorithm exists for RTCC pyramids.

The remaining part of this paper is organized as follows. In Section 2, we give the definition of RTCC network and RTCC pyramids. Section 3 shows a rainbow coloring of RTCC pyramids. Finally, some concluding remarks and future research are given in the last section.

2 Preliminaries

An interconnection network usually can be modeled as a simple graph whose vertices represent processing nodes of the system and edges represent communication links. The *$n-$dimensional RTCC network*, denoted by R_n, is a $2^n \times 2^n$ network. The vertex set of R_n is the set of all quaternary strings of length n. That is, $V(R_n) = \{(x_1, x_2, \ldots, x_n) \,|\, 0 \leq x_i < 4\}$. For a vertex $v = (x_1, x_2, \ldots, x_n) \in V(R_n)$, x_1 is called the *first index* of v. We also use $(x_1, x_2, \ldots, x_{i-1}, (x_i)^j, x_{i+j}, x_{i+j+1}, \ldots, x_n)$ to represent the vertex $(x_1, x_2, \ldots, x_n)$ if $x_i = x_{i+1} = \cdots = x_{i+j-1}$.

An R_n is consisted of four subnetworks of size $2^{n-1} \times 2^{n-1}$. We let $R^n(0)$, $R^n(1)$, $R^n(2)$ and $R^n(3)$ to denote the top left subnetwork, top right subnetwork, bottom left subnetwork and bottom right subnetwork of R_n, respectively, where $R^n(i)$, for each $i = 0, 1, 2, 3$, is isomorphic to an $(n-1)-$dimensional RTCC network. The structure of an R_n is based on a 4-cycle. Each 4-cycle of R_n has four vertices numbered 0 through 3 clockwise around the cycle and has two h-edges and two v-edges. The two edges incident on vertices numbered 0, 1 and 2, 3 are called *h-edges* of a 4-cycle, while the other two edges of the 4-cycle are said to be *v-edges*. We label the vertices of $R^n(i)$ with initial index i and get $V(R^n(i)) = \{(i, x_2, \ldots, x_n) \,|\, 0 \leq x_i < 4\}$. A vertex with labeling $(x_1, x_2, \ldots, x_n)$ is connected to vertices $(x_1, x_2, \ldots, (x_n \pm 1) \bmod 4)$ and connected to a vertex with address schema $(x_1, x_2, \ldots, x_{n-j}, x_{n-j+2}, (x_{n-j+1})^j)$, where $1 \leq x_{n-j+1} \leq 3$, if there exists one j such that $x_n = x_{n-1} = \cdots = x_{n-j+1}$ and $x_{n-j} \neq x_{n-j+1}$.

The number of vertices and edges in a R_n is equal to 4^n and $3 \times 2^{2n-1} - 2$, respectively. The diameter of R_n is equal to $3 \times 2^{n-1} - 1$. For example, Figure 1 depicts an instance of a RTCC network R_2. The R_2 consists of subnetworks $R^2(0), R^2(1), R^2(2)$ and $R^2(3)$. The vertices of $V(R^2(1)) = \{(1,0), (1,1), (1,2), (1,3)\}$ have the same initial index 1 and are connected as a 4-cycle. The vertex $(1,3)$ has neighbors $(1,0)$ and $(1,2)$ according to the modification rule at the rightmost index 3. The vertices $(1,3)$ and $(3,1)$ are adjacent due to the other neighboring rule.

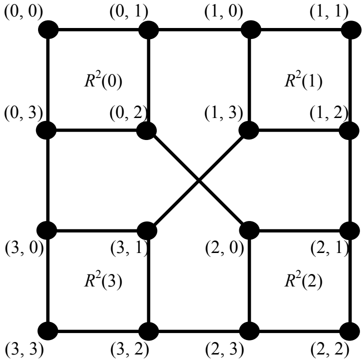


Fig. 1. The R_2

The *n-layered RTCC pyramid*, denoted by RP_n, has the vertex set $V(RP_n) = V_0 \cup V_1 \cup \ldots \cup V_n$, where $V_k = \{(x_1, x_2, \ldots, x_k) \,|\, 0 \leq x_i < 4, 1 \leq i \leq k\}$ is the set of vertices on layer k and $0 \leq k \leq n$. Each V_k is connected as a R_k and an edge of R_k is also called an *internal edge* in RP_n. We use $< (x_1, x_2, \ldots, x_k), (y_1, y_2, \ldots, y_k) >$ to denote an internal edge incident on vertices $(x_1, x_2, \ldots, x_k)$ and $(y_1, y_2, \ldots, y_k)$ of RP_n. Each vertex $(x_1, x_2, \ldots, x_k) \in V_k$ is adjacent to exactly four vertices $(x_1, x_2, \ldots, x_k, t)$ of V_{k+1} where $1 \leq k \leq n-1$ and $t = 0, 1, 2, 3$. And the edges connecting layers k and $k+1$ are said to be *layer edges*. There are a total of $1 + 4 + 4^2 + \cdots + 4^n = (4^{n+1} - 1)/3$ vertices, $(3 \times 2 - 2) + (3 \times 2^3 - 2) + (3 \times 2^5 - 2) + \cdots + (3 \times 2^{2n-1} - 2) = 2^{2n+1} - 2n - 2$ internal edges and $\frac{4(4^n-1)}{3}$ layer edges in an RP_n. The diameter of RP_n is $2n$. Figure 2 illustrates an example of a 3–layered RTCC pyramid RP_2. Layer 0 has only one vertex which is called *Apex*. Layer 1 is a 4-cycle having solid vertices with labels $(0), (1), (2)$ and (3) around the cycle. The dash lines indicate layer edges, while the solid lines are internal edges. The vertex (0) connects to the vertex (1) by the internal edge $< (0), (1) >$ and has layer edges connecting to vertices $(0,0), (0,1), (0,2)$ and $(0,3)$ in layer 2.

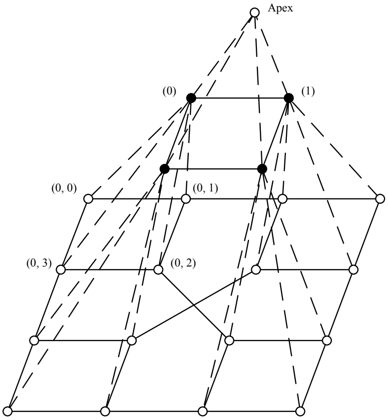


Fig. 2. The RP_2

3 A Rainbow Coloring on RTCC Networks

Note that the edge set of a RTCC pyramid RP_n is classified into internal edges and layer edges. We now present the edge coloring χ under a partition of the vertices on each layer of RP_n. For the layer k, we shall classify all the $2^k \times 2^k$ vertices into 2^{k-1} clusters. A *cluster* on the layer k is an induced subgraph of R_k such that any two vertices of the cluster have a rainbow path consisting of the edges in this cluster. There are two types of clusters in RP_n depending on the parity of k of layer k where the cluster belongs. If k is odd, we let the clusters in R_k of the form square cluster. Otherwise, the clusters are of the form buddy cluster in those layers of even layer numbers. Each *square cluster* in R_k is of size $2^{\frac{k+1}{2}} \times 2^{\frac{k+1}{2}}$ and thus a subgraph induced by the set of vertices with the same leftmost $\frac{k-1}{2}$ indices. We label the square cluster by the first $\frac{k-1}{2}$ indices following $(X)^{\frac{k+1}{2}}$. A *buddy cluster* in R_k is a union of two adjacent square clusters which have the same parity in their second index, and we address a buddy cluster by the label of the square cluster with smaller index. We use Figure 3 to illustrate the definition of clusters. Among the 2^4 square clusters of R_5, the solid lines show the two square clusters with the leftmost two indices 01, and 03, respectively. In fact, the clusters labeled $(01XXX)$ and $(03XXX)$ will be joined to be a buddy cluster in R_6 as those second indices 1 and 3 have the same parity. Note that the union of the clusters labeled $(00XXXX)$ and $(02XXXX)$ is a buddy cluster in R_6. And we name this buddy cluster as $(00XXXX)$ or $(00(X)^4)$ in R_6.

We first introduce the coloring for internal edges in χ. The internal edges are composed of cycle edges and bridge edges. A *cycle edge* is an internal edge which belongs to some 4-cycle, while the *bridge edges* incident on two distinct 4-cycles. Since the colorings of two clusters on the same layer is independent to each other in χ, the bridge edges connecting any two clusters can be assigned an arbitrary color. Let C_k be a square cluster on layer k. Note that k is odd and C is of size

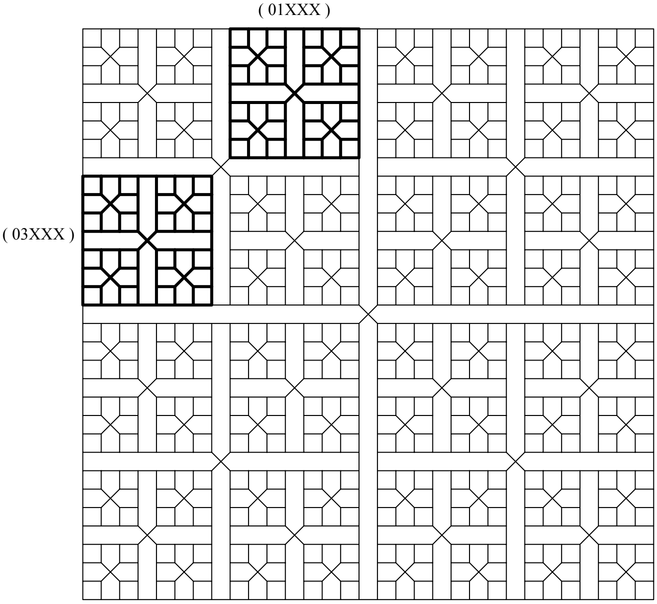


Fig. 3. Square clusters $(01XXX)$ and $(03XXX)$ in R_5

$2^{(k+1)/2} \times 2^{(k+1)/2}$. We decompose C_k into four subclusters $C^k(0), C^k(1), C^k(2)$ and $C^k(3)$. There are six bridge edges between $C^k(i), i = 0, 1, 2, 3$, and each of the six bridge edges is assigned the same color, say color 1. The colors for the coloring of edges in $C^k(i), i = 0, 1, 2, 3$ are forbidden to use the color 1. We then color the edges of each subcluster by using a recursive construction. This means that each subcluster $C^k(i), 0 \leq i \leq 3$, will be further decomposed into four smaller subclusters and then the other six bridge edges will be colored by using a new color. It can be seen that we now only need to decide the colors of the cycle edges. The two v-edges (resp. h-edges) are assigned the same color in a 4-cycle and the color is different to the color of the h-edges (resp. v-edges). Every 4-cycle is colored by two colors, and none of the two colors can be assigned to other edges in this cluster. We then assign the color set of C_k to other clusters on the layer k.

To color the edges of a buddy cluster on layer x. Let C_x be a buddy cluster in the layer x. Notice that C_x is constituted by two square subnetworks of equal size, say A and B, and a bridge edge. We assign color 1 to the bridge edge and color A and B by using two disjoint color sets. The coloring of A proceeds as the coloring of a square cluster mentioned in the last paragraph without using the color 1. Similarly, we regard B as a square cluster and assign a new color set to color B. So, we need one more than twice the number of colors used to color A to color C_x in the edge coloring χ. Figure 4 shows the edge coloring χ on a square cluster of R_5.

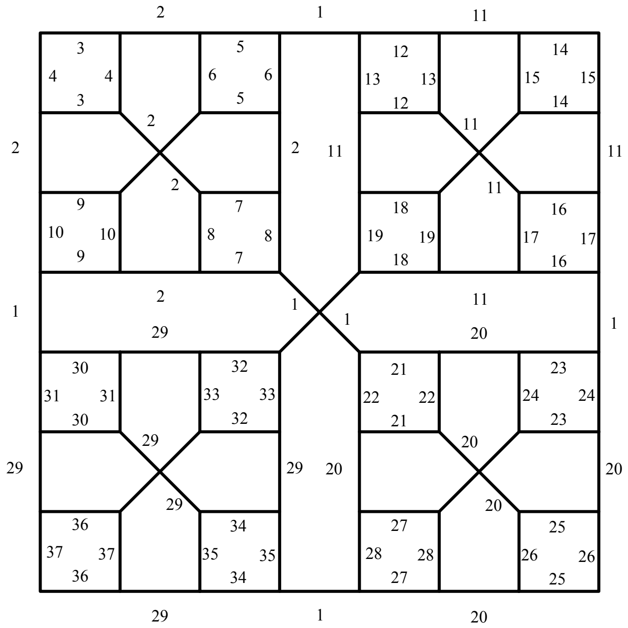


Fig. 4. The edge coloring χ on a cluster of R_5

We want to show that the edge coloring of every cluster is a rainbow coloring by the next two results.

Lemma 1. *Let C_k be a square cluster of RP_k where k is odd. C_k has a rainbow $(\frac{7\times 2^{k-1}-1}{3})-$coloring under the edge coloring χ.*

Proof. We first show by induction that any two vertices s and t of C_k have a rainbow path under the edge coloring χ. The square cluster C_1 is indeed a 4-cycle. By the definition of χ, C_1 is interleaved coloring, verifying the Basis Step. We assume that C_{k-2} has a rainbow $(\frac{7\times 2^{k-2}-1}{3})-$coloring under the edge coloring χ. In R_k, by the definition of square cluster, each square cluster of R_k is of size $2^{(k+1)/2} \times 2^{(k+1)/2}$ and can be decomposed into four square subcluster of size $2^{(k-1)/2} \times 2^{(k-1)/2}$. Let S be the set of the above four square subclusters. By the definition of χ, the edges between any two square clusters of S are assigned the same color but distinct to the colors used in S. Note that each square cluster in S, by the definition of χ, has a color set which has no intersection to the color sets of other square clusters in S. If s and t are in the same square cluster of S, by hypothesis, s and t have a rainbow path under the edge coloring χ. Otherwise, s and t belong to two distinct square clusters of S. It can be seen that there is an $s-t$ path containing exactly one edge e which connects the two square clusters. The $s-e$ path and $e-t$ path, by hypothesis, are rainbow colored. Furthermore, by the definition of χ, the two paths $s-e$ and $e-t$ have no common color. Since e is assigned a color new to the colors of paths $s-e$ and $e-t$. Vertices s and t of C_k have a rainbow path under the edge coloring χ.

We now show that C_k needs $\frac{7\times 2^{k-1}-1}{3}$ under the edge coloring χ. Let a_k be the number of colors used in a square cluster of R_k. When $k=1$. R_1 is indeed a 4-cycle and has the square cluster including itself. So, $a_1=2$ as an initial value. Therefore, by the definition of χ, we have the recursive formula: $a_k=4a_{k-2}+1$. Then, by iteration, we get $a_k=\frac{7\times 2^{k-1}-1}{3}$, completing the proof. Q. E. D.

Lemma 2. *Let C_k be a buddy cluster of RP_k where k is even. C_k has a rainbow $(\frac{7\times 2^{k-1}+1}{3})-$coloring under the edge coloring χ.*

Proof. By the construction of a buddy cluster, each buddy cluster in R_k, k is even, is established by two square clusters of size $2^{k/2}\times 2^{k/2}$ and a connecting edge e. Let s and t be two vertices of C_k. We want to show that there exists an $s-t$ rainbow path under the edge coloring χ. Two cases are considered depending on the positions of s and t.

Case 1. s and t are on the same square clusters.
By Lemma 1, there is an $s-t$ rainbow path using at most $\frac{7\times 2^{k-2}-1}{3}$ colors, completing the proof of this case.

Case 2. s and t are on different square clusters.
Each of the two paths $s-e$ and $e-t$, by Lemma 1, is a rainbow path and assigned by using at most $\frac{7\times 2^{k-2}-1}{3}$ colors. Therefore, the $s-t$ path is colored by less than or equal to $(\frac{7\times 2^{k-2}-1}{3})+(\frac{7\times 2^{k-2}-1}{3})+1=\frac{7\times 2^{k-1}+1}{3}$ colors, completing the proof. Q. E. D.

We now consider the number of colors assigned to the layer edges of RP_n under the edge coloring χ. Since the layer edges incident on the cluster C and the parent vertices of the vertices in C are assigned the same color, the number of colors assigned to the layer edges of RP_n is equal to the sum of number of clusters in RP_n. By the definition of clusters, the layer k of RP_n has exactly 2^{k-1} clusters. Then we have the number of clusters in RP_n as: $1+2+2^2+\cdots+2^{n-1}=2^n-1$. Thus, the next result follows.

Lemma 3. *Let RP_n be an $n-$layered RTCC pyramid. The layer edges of RP_n are assigned totally 2^n-1 colors under the edge coloring χ.*

We now show that χ is a rainbow coloring of a RTCC pyramid.

Theorem 1. *Let RP_n be an $n-$layered RTCC pyramid where n is odd (respectively, even). Then $rc(RP_n)\leq\frac{7\times 2^{n-1}-1}{3}$ (respectively, $rc(RP_n)\leq\frac{7\times 2^{n-1}+1}{3}$).*

Proof. Let s and t be two vertices of RP_n. If one of the two vertices is the apex of RP_n, then we get a shortest $s-t$ path T, consisting of only layer edges, of length less than n. Since the layer edges in distinct layers are assigned different colors, T is a rainbow path under the edge coloring χ. Now, we suppose neither s nor t is the apex. If s and t are in the same cluster C_k, then, by Lemmas 1 and 2, s and t have a rainbow path of length at most $\frac{7\times 2^{k-1}-1}{3}$ (respectively,

$\frac{7\times 2^{k-1}+1}{3}$) for odd k (respectively, even k). Otherwise, we let s' (respectively, t') be an ancestor of s (respectively, t) such that s' and t' are in the same cluster to complete the proof. Q. E. D.

4 Concluding Remarks

For any two distinct vertices u and v of a connected graph G, a *rainbow* $u - v$ *geodesic* in G is a rainbow $u - v$ path of shortest length. A *strong rainbow coloring* of G is a rainbow coloring in which every pair of vertices G is connected by a rainbow $u - v$ geodesic. In this paper, we define a rainbow $u - v$ path P for a pair of distinct vertices u, v of a RTCC pyramid. For algorithmic purposes, an efficient algorithm is demonstrated for determining a rainbow path for every pair of vertices of a RTCC pyramid. The algorithm considers balancing characteristic of a RTCC pyramid and avoids a possible bottleneck in the apex. When the network contains multiple pairs of source and destination vertices between which messages are transmitted along their path, the algorithm is applicable.

Acknowledgments. This work was supported by the National Science Council, Republic of China, under grants NSC99-2221-E-034-008-MY2 and NSC101-2221-E-034-016-.

References

1. Ahadi, A., Dehghan, A.: On Rainbow Connection of Strongly Regular Graphs. Arxiv preprint arXiv:1001.3413v1 [math.CO] (2010)
2. Basavaraju, M., Chandran, L.S., Rajendraprasad, D., Ramaswamy, A.: Rainbow Connection Number and Radius. Arxiv preprint arXiv:1011.0620v1 [math.CO] (2010)
3. Caro, Y., Lev, A., Roditty, Y., Tuza, Z., Yuster, R.: On Rainbow Connection. The Electronic Journal of Combinatorics 15, #R57 (2008)
4. Chakraborty, S., Fischer, E., Matsliah, A., Yuster, R.: Hardness and Algorithms for Rainbow Connection. Journal of Combinatorial Optimization 21(3), 330–347 (2011)
5. Chartrand, G., Johns, G.L., McKeon, K.A., Zhang, P.: Rainbow Connection in Graphs. Mathematica Bohemica 133(1), 85–98 (2008)
6. Ericksen, A.: A Matter of Security. Graduating Engineer & Computer Careers, 24–28 (2007)
7. Kourdy, R., Rad, M.R.N.: RTCC-Pyramid-NOC: Scalable, Regular and Symmetric Network-on-chip Topology. The Journal of Computing 4, 58–64 (2012)
8. Krivelevich, M., Yuster, R.: The Rainbow Connection of a Graph is (at most) Reciprocal to Its Minimum Degree. Journal of Graph Theory 63(3), 185–191 (2009)
9. Li, X., Shi, Y.: Rainbow Connection in 3-connected Graphs. Arxiv preprint arXiv:1010.6131v1 [math.CO] (2010)

10. Li, X., Sun, Y.: Rainbow Connection of Graphs–A Survay, arXiv:1101.5747v1 [math.CO] (2011)
11. Razavi, S., Sarbazi-Azad, H.: The RTCC Pyramid: Routing and Topological Properties. Information Sciences 180, 2328–2339 (2010)
12. Wang, F.H., Sung, G.S.: Rainbow Connection Number in Triangular Pyramids. In: 29th Workshop on Combinatorial Mathematics and Computation Theory, Taipei, Taiwan (2012)

On the Hamiltonian-Connectedness for Graphs Satisfying Ore's Theorem*

Yuan-Kang Shih[1], Hsun Su[2], and Shin-Shin Kao[3,**]

[1] Intel-NTU Connected Context Computing Center,
National Taiwan University,
Taipei, Taiwan 10617, R.O.C.
[2] Department of Public Finance and Taxation,
Takming University of Science and Technology,
Taipei City 11451, Taiwan, R.O.C.
[3] Department of Applied Mathematics,
Chung Yuan Christian University,
Chung-Li City 32023, Taiwan, R.O.C.
shin2kao@gmail.com

Abstract. Consider any undirected and simple graph $G = (V, E)$, where V and E denote the vertex set and the edge set of G, respectively. Let $|G| = |V| = n \geq 3$. The well-known Ore's theorem states that if $deg_G(u) + deg_G(v) \geq n$ holds for each pair of nonadjacent vertices u and v of G, then G is hamiltonian. A similar theorem given by Erdös is as follows: if $deg_G(u) + deg_G(v) \geq n + 1$ holds for each pair of nonadjacent vertices u and v of G, then G is hamiltonian-connected. In this paper, we improve both theorems by showing that any graph G satisfying the condition in Ore's theorem is hamiltonian-connected unless G belongs to two exceptional families.

1 Introduction

For the graph definitions and notations we follow [1], and we consider finite, undirected and simple graphs only. $G = (V, E)$ is a *graph* if V is a finite set and $E \subseteq \{(u, v) \mid (u, v)$ is an unordered pair of $V\}$, where V is called the *vertex set* of G and E the *edge set* of G. We use $|G|$ or $|V|$ for the number of distinct vertices in G. Two vertices u and v are *adjacent*, denoted by $u \sim v$, if $(u, v) \in E$. Given a vertex u of G, the *neighborhood* of u, denoted by $N_G(u)$, is the set $\{v \mid (u, v) \in E\} \subseteq V$. The *degree* of u, denoted by $deg_G(u)$, is the total number of elements of $N_G(u)$. Let S be a subgraph of G. Define two symbols $N_S(u) = N_G(u) \cap S$, and $deg_S(u) = |N_S(u)|$. A *path* P is represented by $\langle v_0, v_1, v_2, \cdots, v_k \rangle$. A path is called a *hamiltonian path* if its vertices are distinct and span V. We say that G is a *hamiltonian-connected* graph if there exists a hamiltonian path between any pair of distinct vertices of G. A *cycle* is a path of at least three vertices such that

* This research was partially supported by the National Science Council of the Republic of China under contract NSC 101-2115-M-033-003-.

** Correspondence author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 25–31.
DOI: 10.1007/978-3-642-35452-6_4

the first vertex is the same as the last vertex. A cycle is called a *hamiltonian cycle* if its vertices are distinct except for the first vertex and the last vertex, and if they span V. We say that G is a *hamiltonian* graph if G contains a hamiltonian cycle.

The following results are well-known.

Theorem 1. *(Ore [4], 1960) A simple graph $G = (V, E)$ with $|G| = |V| = n \geq 3$ is hamiltonian if, for each pair of nonadjacent vertices u and v in V, $deg_G(u) + deg_G(v) \geq n$.*

Theorem 2. *(Erdös [2], 1959) A simple graph $G = (V, E)$ with $|G| = |V| = n \geq 3$ is hamiltonian-connected if, for each pair of nonadjacent vertices u and v in V, $deg_G(u) + deg_G(v) \geq n + 1$.*

In this paper, we intend to establish the following theorem.

Theorem 3. *Let $G = (V, E)$ be a simple graph with $|G| = |V| = n \geq 3$ such that $deg_G(u) + deg_G(v) \geq n$ holds for each pair of nonadjacent vertices u and v in V. Then either G is hamiltonian-connected or G belongs to one of the two families $\mathcal{G}_1$ and $\mathcal{G}_2$.*

Graphs in $\mathcal{G}_i$, where $i \in \{1, 2\}$, are illustrated in Figure 1 and shall be defined in the next section.

2 Proof of Theorem 3

We first define some operators on graphs. Let G_1 and G_2 be two graphs. We say that G_1 and G_2 are *disjoint* if G_1 and G_2 have no vertex in common, and that G_1 and G_2 are *edge-disjoint* if G_1 and G_2 have no edge in common. The *union* of two disjoint graphs G_1 and G_2, denoted by $G_1 + G_2$, is a graph with $V(G_1 + G_2) = V(G_1) \cup V(G_2)$ and $E(G_1 + G_2) = E(G_1) \cup E(G_2)$. The *join* of two disjoint subgraphs G_1 and G_2, denoted by $G_1 \vee G_2$, is the graph obtained from $G_1 + G_2$ by joining each vertex of G_1 to each vertex of G_2.

Definition 1. *Let H_i be any simple graph with i vertices. Let s and t be positive integers, and n be an integer with $n \geq 3$. Define two families of graphs as follows. $\mathcal{G}_1 = \{H_2 \vee (K_s + K_t) \mid s + t \geq 2$ and $s + t = n - 2\}$; $\mathcal{G}_2 = \{H_s \vee sK_1 \mid 2s = n\}$.*

It is easy to see that graphs in $\mathcal{G}_1 \cup \mathcal{G}_2$ satisfy the degree-sum condition in Ore's theorem, so they are hamiltonian. However, for $G \in \mathcal{G}_1$, taking $x, y \in V(H_2)$, there exists no hamiltonian path between x and y. And for $G \in \mathcal{G}_2$, taking $x, y \in V(sK_1)$ or $x, y \in V(H_s)$, there exists no hamiltonian path between x and y. Thus any graph of $\mathcal{G}_1 \cup \mathcal{G}_2$ is not hamiltonian-connected.

Lemma 1. *For $i \in \{1, 2\}$, any graph in $\mathcal{G}_i$ is not hamiltonian-connected.*

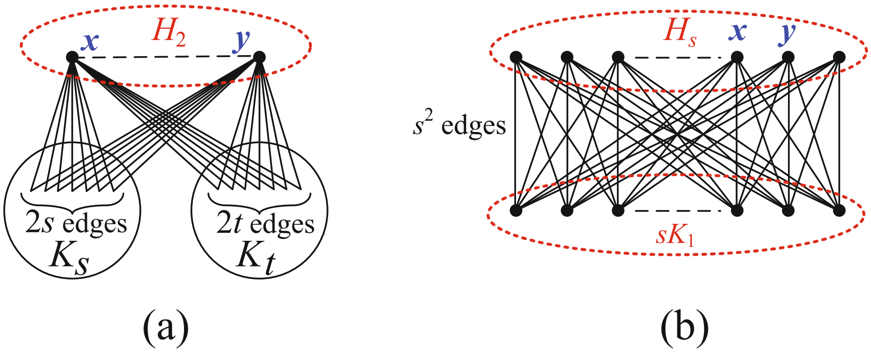


Fig. 1. An illustration of graphs in $\mathcal{G}_i$ for $i \in \{1, 2\}$

To prove Theorem 3, we have the following assumptions. Let $G = (V, E)$ be a simple graph with $|G| = n \geq 3$ and G satisfy the degree-sum condition in Theorem 3. Let u and v be two arbitrary vertices in G and P the longest path between u and v in G. If $|P| = n$, then P is a hamiltonian path. Thus it suffices to verify the case when $|P| \leq n - 1$. Let $P = \langle u = u_1, u_2, \ldots, u_k = v\rangle$ with $k \leq n - 1$, and $S = G - P$. For any vertex x of V, by $x \sim S$, we mean there exists some vertex y in S such that $(x, y) \in E$. By $x \nsim S$, we mean there exists no vertex y in S such that $(x, y) \in E$.

Property 1. $|N_P(S)| \geq 2$.

Proof. Suppose that $|N_P(S)| \leq 1$. Consider any vertex x of S and any vertex y of P, where $y \nsim S$. It is easy to see x and y are nonadjacent, $deg_G(x) = deg_S(x) + deg_P(x) \leq (|V(S)| - 1) + 1 = |V(S)|$, and $deg_G(y) \leq |V(P)| - 1$. Thus $deg_G(x) + deg_G(y) \leq |V(S)| + |V(P)| - 1 = n - 1$, a contradiction to the degree-sum condition of Theorem 3. Consequently, it must be $|N_P(S)| \geq 2$.

Property 2. Let $1 \leq i \leq k - 1$. For any vertex u_i in P, if $u_i \sim x$ for some $x \in V(S)$, then $u_{i+1} \nsim x$.

Proof. Assume the opposite by letting $u_{i+1} \sim x$, then $\langle u = u_1, u_2, \ldots, u_i, x, u_{i+1}, \ldots, u_k = v\rangle$ is a path longer than P. It is impossible since P is the longest path between u and v. It must be $u_{i+1} \nsim x$.

Property 3. S is connected. That is, S consists of exactly one component.

Proof. Suppose that S is not connected and S consists of two components, denoted by S_1 and S_2. Let x and y be two vertices of S with $x \in S_1$ and $y \in S_2$. It is obvious that $deg_{S_1}(x) \leq |V(S_1)| - 1$ and $deg_{S_2}(y) \leq |V(S_2)| - 1$. By Proposition 2, $deg_P(x) \leq \lceil\frac{|V(P)|}{2}\rceil$ and $deg_P(y) \leq \lceil\frac{|V(P)|}{2}\rceil$. Therefore,

$$\begin{aligned} deg_G(x) + deg_G(y) &\leq |V(S_1)| + |V(S_2)| + 2\lceil\frac{|V(P)|}{2}\rceil - 2 \\ &\leq |V(S)| + |V(P)| - 1 \\ &= n - 1. \end{aligned}$$

Since $x \nsim y$, it is contradictory to the degree-sum assumption of Theorem 3. Thus S must be connected and consist of exactly one component.

Property 4. For $1 \leq i \leq k-1$, if $u_i \sim S$, then $u_{i+1} \nsim S$.

Proof. Suppose $u_{i+1} \sim S$. Let $x, y \in S$ such that $u_i \sim x$ and $u_{i+1} \sim y$, respectively. With Proposition 3, S is connected and there is a path Q in S between x and y. Thus $\langle u = u_1, u_2, \ldots, u_i, x, Q, y, u_{i+1}, \ldots, u_k = v\rangle$ is a path longer than P in G, which is impossible since P is the longest path between u and v in G. It must be $u_{i+1} \nsim S$.

Property 5. Let u_i and u_j be two distinct vertices in P such that $u_i \sim S$ and $u_j \sim S$, where $1 \leq i \leq j-2$ and $i+2 \leq j \leq k$. If $j \leq k-1$, then $u_{i+1} \nsim u_{j+1}$. Similarly, if $i \geq 2$, then $u_{i-1} \nsim u_{j-1}$.

Proof. Let $x, y \in S$ such that $u_i \sim x$ and $u_j \sim y$, respectively. For the case with $j \leq k-1$, suppose $u_{i+1} \sim u_{j+1}$. By Proposition 3, S is connected and there is a path Q in S between x and y. Thus $\langle u = u_1, u_2, \ldots, u_i, x, Q, y, u_j, u_{j-1}, \ldots, u_{i+1}, u_{j+1}, u_{j+2}, \ldots, u_k = v\rangle$ is a path longer than P, which leads to a contradiction. The proof for the case with $i \geq 2$ is similar. Hence the proposition is proved.

Property 6. Both u and v are adjacent to S. That is, $u \sim S$ and $v \sim S$.

Proof. To prove $v \sim S$, we assume the opposite by letting $v \nsim S$. Let $x \in S$ be an arbitrary vertex. Obviously, $deg_S(x) \leq |V(S)| - 1$ and $deg_P(x) \leq |N_P(S)|$. Let $u_i \sim S$ for some $1 \leq i \leq k-1$. By Proposition 4, $u_{i+1} \nsim S$. And by Proposition 5, $deg_P(u_{i+1}) \leq |V(P)| - |N_P(S)|$. We have

$$\begin{aligned} deg_G(x) + deg_G(u_{i+1}) &\leq \big(|V(S)| - 1 + |N_P(S)|\big) + \big(|V(P)| - |N_P(S)|\big) \\ &= |V(S)| + |V(P)| - 1 \\ &= n - 1. \end{aligned}$$

Since $x \nsim u_{i+1}$, it contradicts the degree-sum assumption of Theorem 3. Consequently, it must be $v \sim S$. With the similar argument, it can be shown that $u \sim S$.

Property 7. S is a complete graph.

Proof. Suppose there exist two distinct vertices x and y of S such that $x \nsim y$. It is obvious that $deg_S(x) \leq |V(S)| - 2$ and $deg_P(x) \leq |N_P(S)|$. By Proposition 6, $u = u_1 \sim S$ and $v = u_k \sim S$. Consider u_2 of P. By Proposition 2, $u_2 \nsim S$. And by Proposition 5, $deg_P(u_2) \leq |V(P)| - (|N_P(S)| - 1)$.

$$\begin{aligned} deg_G(x) + deg_G(u_2) &\leq \big(|V(S)| - 2 + N_P(S)\big) + \big(|V(P)| - N_P(S) + 1\big) \\ &= |V(S)| + |V(P)| - 1 \\ &= n - 1. \end{aligned}$$

Since $x \nsim u_2$, it is contradictory to the degree-sum assumption of Theorem 3. Therefore, S must be a complete graph.

With Proposition 1 and 4, we know that $2 \leq |N_p(S)| \leq \lceil \frac{|V(P)|}{2} \rceil$. Moreover, $|N_p(S)| = \frac{|V(P)|+1}{2}$ occurs only when $|V(P)|$ is an odd integer. We shall complete the proof of Theorem 3 in the following three propositions.

Property 8. If $|N_P(S)| = 2$, then $G \in \mathcal{G}_1$.

Proof. With Proposition 6, $N_P(S) = \{u, v\}$ and $u_i \nsim S$ for all $2 \leq i \leq k-1$. Let $x \in V(S)$ be arbitrary. It is obvious that $deg_S(x) \leq |V(S)|-1$ and $deg_P(x) \leq 2$. Thus $deg_G(x) \leq |V(S)|+1$. Let $u_i \in P$ be an arbitrary vertex with $2 \leq i \leq k-1$. Since $x \nsim u_i$, the degree-sum condition of Theorem 3 implies that

$$deg_G(u_i) \geq n - \big(|V(S)| + 1\big) = |V(P)| - 1.$$

That is, the vertex u_i is adjacent to all vertices of P except itself. This implies that the set of vertices $\{u_2, u_3, \ldots, u_{k-1}\}$ forms a complete graph with $k-2$ vertices.

On the other hand, u_i is adjacent to at most $|V(P)| - 1$ vertices in P. So

$$deg_G(x) \geq n - \big(|V(P)| - 1\big) = |V(S)| + 1.$$

Thus x is adjacent to all vertices of S except itself, $x \sim u$ and $x \sim v$. As a result, we have shown that $G \in \mathcal{G}_1$.

Property 9. If $3 \leq |N_P(S)| \leq \lfloor \frac{|V(P)|}{2} \rfloor$, a contradiction will occur.

Proof. In this proposition, $|N_P(S)|$ might attain its maximum if $|V(P)|$ is an even integer. (The situation where $|N_P(S)| = \lceil \frac{|V(P)|}{2} \rceil$ with $|V(P)|$ being an odd integer is presented in the next proposition.) There are two cases.

Case 1. There exists at least one set of vertices $\{u_{p-1}, u_p, u_{p+1}\}$ of P, where $2 \leq p \leq k-1$, such that $u_{p-1} \sim S$, $u_p \nsim S$ and $u_{p+1} \sim S$.

Let x be an arbitrary vertex in S. Then $deg_P(x) \leq |N_P(S)|$. By Proposition 7, $deg_S(x) = |V(S)| - 1$. So, $deg_G(x) \leq |N_P(S)| + |V(S)| - 1$.

Note that $u_p \nsim x$ and $n = |V(P)| + |V(S)|$. Therefore,

$$\begin{aligned} deg_G(u_p) &\geq n - (|N_P(S)| + |V(S)| - 1) \\ &= |V(P)| - (|N_P(S)| - 1) \\ &> (|V(P) - 1) - (|N_P(S)| - 1). \end{aligned}$$

The first parenthesis is the total number of vertices of $P - \{u_p\}$, and the second parenthesis is the total number of vertices of $\{u_{i+1} | u_i \in N_P(S)\}$ or $\{u_{i-1} | u_i \in N_P(S)\}$ since it is known by Proposition 5 that $u \sim S$ and $v \sim S$. It means that either $u_p \sim u_{i-1}$ or $u_p \sim u_{i+1}$ for some $u_i \in N_P(S)$ with $i \neq p$, but it is contradictory to Proposition 5.

Case 2. There exists no set of vertices $\{u_{i-1}, u_i, u_{i+1}\}$ of P, where $2 \leq i \leq k-1$, such that $u_{i-1} \sim S$, $u_i \nsim S$ and $u_{i+1} \sim S$.**Case 2.1** $|S| \geq 2$. Let x and z be two distinct vertices of S such that $z \sim u$ and $x \sim u_{i^*}$ for some i^*.

Obviously, $i^* \geq 4$, and there exists a hamiltonian path of S, denoted by Q, between z and x by Proposition 7. By Proposition 4, $u_2 \nsim S$, so $u_2 \nsim x$. It must be $deg_G(u_2) + deg_G(x) \geq n$. Since $deg_G(x) \leq |S| - 1 + |N_P(S)|$, $deg_G(u_2) \geq n - (|S| - 1 - |N_P(S)|) = k - (|N_P(S)| - 1)$. On the other hand, with Proposition 5, $deg_G(u_2) \leq k - (|N_P(S)| - 1)$, where $(|N_P(S)| - 1)$ is the total number of vertices of $\{u_{i+1} \mid u_i \in N_P(S) \text{ and } 1 \leq i \leq k-1\}$. It implies that $deg_G(u_2) = k - (|N_P(S)| - 1)$, and $N_G(u_2) = \{u_i \mid u_{i-1} \nsim S\}$. Therefore, $u_2 \sim u_{i^*+2}$ since $u_{i^*+1} \nsim S$ by the assumption of Case 2. Now the path $\langle u, z, Q, x, u_{i^*}, u_{i^*-1}, u_{i^*-2}, \ldots, u_2, u_{i^*+2}, u_{i^*+3}, \ldots, v\rangle$ is a path longer than P as long as $|Q| \geq 2$. This leads to a contradiction.**Case 2.2** $|S| = 1$. Let $S = \{x\}$ and $x \sim u_{i^*}$. As in Case 2.1, $deg_G(u_2) = k - (|N_P(S)| - 1)$ and $N_G(u_2) = \{u_i \mid u_{i-1} \nsim S\}$. Thus $u_2 \sim u_{i^*}$. Note that $u_{i^*+1} \nsim x$, using the same argument as for u_2, it is found that $deg_G(u_{i^*+1}) = k - (|N_P(S)| - 1)$ and $N_G(u_{i^*+1}) = \{u_i \mid u_{i-1} \nsim S\}$. So $u_{i^*+1} \sim u_{i^*-1}$ since $u_{i^*-2} \nsim x$ by the assumption of Case 2. Now the path $\langle u, x, u_{i^*}, u_2, u_3, \ldots, u_{i^*-1}, u_{i^*+1}, u_{i^*+2}, \ldots, v\rangle$ is a path longer than P. This leads to a contradiction again.

Property 10. If $|N_P(S)| = \lceil\frac{|V(P)|}{2}\rceil$, the following two statements hold. (1) $|V(S)| = 1$. (2) $G \in \mathcal{G}_2$.

Proof. Here, $|V(P)| = k$ is an odd integer. Let $P_{odd} = \{u_1, u_3, \ldots, u_k\}$ and $P_{even} = \{u_2, u_4, \ldots, u_{k-1}\}$. With Proposition 2, it must be $N_P(S) = P_{odd}$. To prove (1), we assume $|V(S)| \geq 2$. Let $u_i, u_{i+2} \in P_{odd}$, $x, y \in S$ such that $u_i \sim x$ and $u_{i+2} \sim y$. With Proposition 7, S is a complete graph. There is a path Q between x and y with $|V(Q)| \geq 2$. Thus $\langle u = u_1, u_2, \ldots, u_i, x, Q, y, u_{i+2}, \ldots, u_k = v\rangle$ is a path longer than P. This is a contradiction. Consequently, $|V(S)| = 1$.

Let u_i and u_j be two vertices in P_{even} with $2 \leq i < j \leq k-1$. With Proposition 5, $u_i \nsim u_j$, and hence $deg_G(u_j) = deg_P(u_j) \leq \lceil\frac{|V(P)|}{2}\rceil$. By (1), $n = |V(P)| + 1$. Note that $|V(P)|$ is odd. Applying the degree-sum condition on $\{u_i, u_j\}$, we obtain

$$\begin{aligned} deg_G(u_i) &\geq n - \lceil\frac{|V(P)|}{2}\rceil \\ &= (|V(P)| + 1) - \lceil\frac{|V(P)|}{2}\rceil \\ &= \lceil\frac{|V(P)|}{2}\rceil. \end{aligned}$$

It implies that $deg_G(u_i) = \lceil\frac{|V(P)|}{2}\rceil$ for any $u_i \in P_{even}$. As a result, any vertex in P_{even} is adjacent to all vertices of P_{odd}. Hence $G \in \mathcal{G}_2$.

With Proposition 2-10, Theorem 3 is proved.

3 Remarks

In this paper, we prove that any graph satisfying Ore's condition are hamiltonian-connected unless it belongs to certain exceptional graph families. It is interesting to note these two exceptional families, $\mathcal{G}_1$ and $\mathcal{G}_2$, are identical with those in [5], where they proved that any graph G satisfying Ore's theorem remains hamiltonian even after one vertex or one edge is deleted, unless $G \in \mathcal{G}_1 \cup \mathcal{G}_2$. Such an observation reflects the close relationship between 1-fault-tolerant hamiltonian graphs and hamiltonian-connected graphs, as was mentioned in [3]. Moreover, it is noted that all graphs satisfying Ore's theorem are with diameter at most 2, so the condition of Theorem 3 can be replaced by checking the degree-sum of those vertices with distance 2. Consequently, the following result is established as a corollary.

Corollary 1. *Let $G = (V, E)$ be a simple graph with $|G| = |V| = n \geq 3$, and $\delta(u, v)$ be the distance between two vertices u and v. If $deg_G(u) + deg_G(v) + \delta(u, v) \geq n + 2$ holds for each pair of nonadjacent vertices u and v in V, then either G is hamiltonian-connected or G belongs to $\mathcal{G}_1 \cup \mathcal{G}_2$.*

Finally, our result corrected the similar result in [6], in which they obtained the following theorem (Theorem 3 of [6]):

Let G be a 2-connected graph with $|G| \geq 3$. If $deg_G(x) + deg_G(y) \geq n$ for each pair of nonadjacent vertices x and y with $\delta(x, y) = 2$ in G, then G is hamiltonian-connected or $G \in \{H_2 \vee^* (K_{n-m} \cup H_{m-2}), H_{\frac{n}{2}} \cup K^c_{\frac{n}{2}}\}$, where $H_2 \vee^* (K_s \cup H_t)$ is obtained from $H_2 \vee (K_s \cup H_t)$ by removing some edges connecting to H_2. Our exceptional family $\mathcal{G}_1$ is different from the above result by [6]. However, it is easy to see that the degree-sum of x and y, where $x \in K_{n-m}$ and $y \in H_{m-2}$, fails to satisfy the assumption $deg_G(x) + deg_G(y) \geq n$. Therefore, our theorem rectifies it and provides the correct version.

References

1. Bondy, J.A., Mutry, U.S.R.: Graph Theory with Applications. North-Holland, New York (1980)
2. Erdös, P., Gallai, T.: On Maximal Paths and Circuits of Graphs. Acta Mathematica Hungarica 10, 337–356 (1959)
3. Kao, S.S., Hsu, K.M., Hsu, L.H.: Cubic planar hamiltonian graphs of various types. Discrete Mathematics 309, 1364–1389 (2006)
4. Ore, O.: Note on Hamilton Circuit. American Mathematical Monthly 67, 55 (1960)
5. Su, H., Shih, Y.-K., Kao, S.-S.: On the 1-fault hamiltonicity for graphs satisfying Ore's theorem. Information Processing Letters 112, 839–843 (2012)
6. Zhao, K., Chen, D., Lin, Y., Li, Z., Zeng, K.: Some new sufficient conditions and Hamiltonian-connected graphs. Procedia Engineerig 24, 278–281 (2011)

On Total Covers of Block-Cactus Graphs*

Yu-Ting Li[1], Jia-Jie Liu[2], and Yue-Li Wang[1]

[1] Department of Information Management,
National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
[2] Department of Information Management, Shih Hsin University,
Taipei, Taiwan, R.O.C.

Abstract. Let $G = (V, E)$ be a simple graph with vertex set V and edge set E. A subset $W \subseteq V \cup E$ is a total covering set if every element $x \in (V \cup E) \setminus W$ is either adjacent to or incident to an element of W. The total covering problem is to find a total covering set of G. In this paper, we show that this problem can be solved in linear-time on block-cactus graphs.

Keywords: total cover, dominating set, mixed dominating set, block-cactus graph.

1 Introduction

All graphs $G = (V, E)$ considered in this paper are simple graphs, i.e., no loops or multiple edges, where $V(G)$ and $E(G)$ are the vertex and edge, respectively, sets of G. When context is clear, $V(G)$ and $E(G)$ are simply written as V and G, respectively. Furthermore, let $n = |V|$ and $m = |E|$ unless specified otherwise. A set $S \subseteq V \cup E$ is called a *total covering set* if every element in $V \cup E \setminus S$ is adjacent or incident to at least one element in S [1]. The *total covering number* of G, denoted by $\alpha_2(G)$, is the minimum cardinality of a total covering set of G; such a set is called an α_2-set of G. The *total covering problem* is to find $\alpha_2(G)$ of a graph G.

The total covering problem was first introduced in [1]. In [4], Erdős and Meir gave the best upper and lower bounds for $\alpha_2(G) + \alpha_2(\overline{G})$, where $\overline{G}$ is the complement of G. In [6], Meir proved that $\alpha_2(G) + \beta_2(G) > 5n/4$ for any large n, where $\beta_2(G)$ is the maximum total matching number of G. In [2], Alavi et al. introduced some properties of connected graphs which have a total covering number $\lfloor n/2 \rfloor$. A practical application of total covering in electric power companies is stated in [7]. It is known that its determination is NP-complete even when restricted to split graphs [10]. Hence it makes sense to determine this invariant exactly for some interesting and non-trivial graph classes. Zhao, Kang, and Sohn provided an $O(n + m)$-time algorithm for finding a minimum total covering set on trees [10]. The reader is referred to [5] for the detail of the total

* This work was supported in part by the National Science Council of the Republic of China under contracts NSC 100–2221–E–011–067-MY3 and NSC 101–2221–E–011–038–MY3.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 33–39.
DOI: 10.1007/978-3-642-35452-6_5

covering problem. Note that, in [5,10], a total covering set is also called a *mixed dominating set.*

A vertex v is called a *cut vertex* of a connected graph G if the number of connected components is increased after removing v or only one vertex is remained. A *block* of G is a maximal connected subgraph in which no vertex is a cut vertex. A graph is called a *block graph* if and only if its blocks are complete graphs (or cliques) and the intersection of two blocks is either empty or exactly one cut vertex. The definition of cactus graphs is similar to that of block graphs except that a block can only be a cycle or a complete graph of two vertices. A graph G is called a *block-cactus* graph if each block is either a complete graph or a cycle [9]. In this paper, we solve the total covering set problem on block-cactus graphs.

The rest of this paper is organized as follows. In Section 3, the total covering numbers of paths, cycles, and cliques are introduced. In Section 4, we propose a linear time algorithm for solving the total covering problem on block-cactus graphs. Finally, concluding remarks are given in Section 5.

2 Preliminaries

First we introduce some terms which will be used in the rest of this paper. Let $G = (V, E)$ be a simple connected graph. A vertex u of G is said to cover itself, all edges incident to u, and all vertices adjacent to u. Similarly, an edge e of G covers itself, the two end vertices of e, and all edges adjacent to e. Denote by $x \sim y$ if x covers y, where x and y are two elements in $V \cup E$. The *mixed closed neighborhood* of x is the set $N_{\mathrm{m}}[x] = \{y \in V \cup E | x \sim y\}$. The *mixed open neighborhood* of x is the set $N_{\mathrm{m}}(x) = N_{\mathrm{m}}[x] \setminus \{x\}$. For a set $S \subseteq V \cup E$, let $N_{\mathrm{m}}[S] = \bigcup_{x \in S} N_{\mathrm{m}}[x]$ and $N_{\mathrm{m}}(S) = N_{\mathrm{m}}[S] \setminus S$. An element $e \in V \cup E$ is said to be covered by S if $e \in N_{\mathrm{m}}[S]$.

By using a similar technique in [3], we in fact deal with a slightly more general version of the total covering problem which can be formulated as follows. Let all elements in $V \cup E$ be partitioned into three subsets F, B, and R, where F consists of free elements, B consists of bound elements, and R consists of required elements. We use $\mathcal{P}(G)$ to denote such a partition. When dealing with more than one partition of G simultaneously, we use $\mathcal{P}_1(G)$, $\mathcal{P}_2(G)$, ..., to denote those partitions, and their corresponding subsets F, B, and R are denoted by $F(\mathcal{P}_i), B(\mathcal{P}_i)$, and $F(\mathcal{P}_i)$, respectively, for $i \geqslant 1$. A *mixed total covering set* in G is a set $M \subseteq V \cup E$ which contains all required vertices, i.e. $R \subseteq M$, and which covers all bound elements e, i.e. $e \in N_{\mathrm{m}}[M]$. Free elements need not be covered by M but may be included in M in order to cover bound elements. The mixed total covering number of G, denoted by $\alpha_{2\mathrm{m}}(\mathcal{P}(G))$, is the minimum cardinality of a total covering set of G with partition $\mathcal{P}$, and such a set is called an $\alpha_{2\mathrm{m}}$-set of G with partition $\mathcal{P}$. The *mixed total covering problem* is to find $\alpha_{2\mathrm{m}}(\mathcal{P}(G))$ of a graph G with partition $\mathcal{P}$.

Let $\mathcal{P}$ be a partition of G. For two elements $x, y \in V \cup E$, if x and y are in sets X and Y, respectively, then, for brevity, we say that x and y are in XY.

For example, if $x \in F$, $y \in B$, and $z \in F$, then we say that x, y, and z are in FBF. An adjustment on partition $\mathcal{P}$ with respect to vertex v, denoted by $A_1(\mathcal{P}, v)$, is to reset $v \in R$ and all elements in $N_{\mathrm{m}}(v)$ to F except those ones are in R, and all other elements remain unchanged as in $\mathcal{P}$. An adjustment on partition $\mathcal{P}$ with respect to edge e, denoted by $A_2(\mathcal{P}, e)$, is to reset $e \in R$ and all elements in $N_{\mathrm{m}}(e)$ to F except those ones are in R, and all other elements remain unchanged as in $\mathcal{P}$. Let $A_3(\mathcal{P}, v)$ denote the adjustment on partition $\mathcal{P}$ with respect to vertex v that all edges in $N_{\mathrm{m}}(v)$ are reset to be in F except those ones are in R, and all other elements remain unchanged as in $\mathcal{P}$.

Proposition 1. *Let G be a graph, and $\mathcal{P}_1$ and $\mathcal{P}_2$ be two partitions of G. If $R(\mathcal{P}_2) \subseteq R(\mathcal{P}_1)$, $B(\mathcal{P}_1) \subseteq B(\mathcal{P}_2)$, and $F(\mathcal{P}_1) \subseteq F(\mathcal{P}_2)$, then $\alpha_{2\mathrm{m}}(\mathcal{P}_2(G)) \leqslant \alpha_{2\mathrm{m}}(\mathcal{P}_1(G))$. Furthermore, if $R(\mathcal{P}_1) = R(\mathcal{P}_2)$, $B(\mathcal{P}_2) \subseteq B(\mathcal{P}_1)$, and $F(\mathcal{P}_1) \subseteq F(\mathcal{P}_2)$, then $\alpha_{2\mathrm{m}}(\mathcal{P}_2(G)) \leqslant \alpha_{2\mathrm{m}}(\mathcal{P}_1(G))$.*

Theorem 1. *The mixed total covering problem on paths can be solved in linear time.*

Theorem 2. *The mixed total covering problem on cycles can be solved in linear time.*

Lemma 1 ([1]). *For a clique K_n, $\alpha_2(K_n) = \lceil \frac{n}{2} \rceil$.*

Lemma 2. *For a clique K_n, if $R = \emptyset$, all edges $e \in B$, and all vertices $v \in F$, then $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$.*

Proof. Since all vertices $v \in F$ and the edges covered by v are also covered by an incident edge of v, there exists an $\alpha_{2\mathrm{m}}$-set M of G with partition $\mathcal{P}$ such that $M \cap V = \emptyset$. It clear that removing all elements in $N_{\mathrm{m}}[e]$ from K_n yields a K_{n-2} for $n \geqslant 3$. The value of $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n))$ can be computed by the following recurrence formula:

$$\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \begin{cases} 1 & \text{if } n = 2, 3, \\ \alpha_{2\mathrm{m}}(\mathcal{P}(K_{n-2})) + 1 & \text{if } n > 3. \end{cases}$$

Therefore, $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$ for $n \geqslant 2$ and the lemma follows.

Corollary 1. *For a clique K_n, if $R = \emptyset$, all edges $e \in B$, and at least one vertex is in F, then $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$.*

Proof. Let $\mathcal{P}_1(K_n)$ be the partition described in the statement of this corollary and $\mathcal{P}_2(K_n)$ be the partition described in Lemma 4. Clearly, by Lemma 4, $\lfloor \frac{n}{2} \rfloor = \alpha_{2\mathrm{m}}(\mathcal{P}_2(K_n)) \leqslant \alpha_{2\mathrm{m}}(\mathcal{P}_1(K_n))$. Then, by using a similar argument as in Lemma 4, we can obtain $\alpha_{2\mathrm{m}}(\mathcal{P}_1(K_n)) \leqslant \lfloor \frac{n}{2} \rfloor$. Therefore, $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$. This completes the proof.

Note that, in Corollary 2, finding an $\alpha_{2\mathrm{m}}$-set is equivalent to selecting $\lfloor \frac{n}{2} \rfloor$ disjoint pair of vertices. Thus an $\alpha_{2\mathrm{m}}$-set of a clique with a partition described in Corollary 2 can be found in $O(n)$ time.

3 Computing $\alpha_2(G)$ for Block-Cactus Graphs G

First we introduce some terms which will be used in the rest of this paper. Let $G = (V, E)$ be a simple connected graph. A vertex u of G is said to cover itself, all edges incident to u, and all vertices adjacent to u. Similarly, an edge e of G covers itself, the two end vertices of e, and all edges adjacent to e. Denote by $x \sim y$ if x covers y, where x and y are two elements in $V \cup E$. The *mixed closed neighborhood* of x is the set $N_{\mathrm{m}}[x] = \{y \in V \cup E | x \sim y\}$. The *mixed open neighborhood* of x is the set $N_{\mathrm{m}}(x) = N_{\mathrm{m}}[x] \setminus \{x\}$. For a set $S \subseteq V \cup E$, let $N_{\mathrm{m}}[S] = \bigcup_{x \in S} N_{\mathrm{m}}[x]$ and $N_{\mathrm{m}}(S) = N_{\mathrm{m}}[S] \setminus S$. An element $e \in V \cup E$ is said to be covered by S if $e \in N_{\mathrm{m}}[S]$.

By using a similar technique in [3], we in fact deal with a slightly more general version of the total covering problem which can be formulated as follows. Let all elements in $V \cup E$ be partitioned into three subsets F, B, and R, where F consists of free elements, B consists of bound elements, and R consists of required elements. We use $\mathcal{P}(G)$ to denote such a partition. When dealing with more than one partition of G simultaneously, we use $\mathcal{P}_1(G), \mathcal{P}_2(G), \ldots$, to denote those partitions, and their corresponding subsets F, B, and R are denoted by $F(\mathcal{P}_i), B(\mathcal{P}_i)$, and $F(\mathcal{P}_i)$, respectively, for $i \geqslant 1$. A *mixed total covering set* in G is a set $M \subseteq V \cup E$ which contains all required vertices, i.e. $R \subseteq M$, and which covers all bound elements e, i.e. $e \in N_{\mathrm{m}}[M]$. Free elements need not be covered by M but may be included in M in order to cover bound elements. The mixed total covering number of G, denoted by $\alpha_{2\mathrm{m}}(\mathcal{P}(G))$, is the minimum cardinality of a total covering set of G with partition $\mathcal{P}$, and such a set is called an $\alpha_{2\mathrm{m}}$-set of G with partition $\mathcal{P}$. The *mixed total covering problem* is to find $\alpha_{2\mathrm{m}}(\mathcal{P}(G))$ of a graph G with partition $\mathcal{P}$.

Let $\mathcal{P}$ be a partition of G. For two elements $x, y \in V \cup E$, if x and y are in sets X and Y, respectively, then, for brevity, we say that x and y are in XY. For example, if $x \in F$, $y \in B$, and $z \in F$, then we say that x, y, and z are in FBF. An adjustment on partition $\mathcal{P}$ with respect to vertex v, denoted by $A_1(\mathcal{P}, v)$, is to reset $v \in R$ and all elements in $N_{\mathrm{m}}(v)$ to F except those ones are in R, and all other elements remain unchanged as in $\mathcal{P}$. An adjustment on partition $\mathcal{P}$ with respect to edge e, denoted by $A_2(\mathcal{P}, e)$, is to reset $e \in R$ and all elements in $N_{\mathrm{m}}(e)$ to F except those ones are in R, and all other elements remain unchanged as in $\mathcal{P}$. Let $A_3(\mathcal{P}, v)$ denote the adjustment on partition $\mathcal{P}$ with respect to vertex v that all edges in $N_{\mathrm{m}}(v)$ are reset to be in F except those ones are in R, and all other elements remain unchanged as in $\mathcal{P}$.

Proposition 2. *Let G be a graph, and $\mathcal{P}_1$ and $\mathcal{P}_2$ be two partitions of G. If $R(\mathcal{P}_2) \subseteq R(\mathcal{P}_1)$, $B(\mathcal{P}_1) \subseteq B(\mathcal{P}_2)$, and $F(\mathcal{P}_1) \subseteq F(\mathcal{P}_2)$, then $\alpha_{2\mathrm{m}}(\mathcal{P}_2(G)) \leqslant \alpha_{2\mathrm{m}}(\mathcal{P}_1(G))$. Furthermore, if $R(\mathcal{P}_1) = R(\mathcal{P}_2)$, $B(\mathcal{P}_2) \subseteq B(\mathcal{P}_1)$, and $F(\mathcal{P}_1) \subseteq F(\mathcal{P}_2)$, then $\alpha_{2\mathrm{m}}(\mathcal{P}_2(G)) \leqslant \alpha_{2\mathrm{m}}(\mathcal{P}_1(G))$.*

Theorem 3. *The mixed total covering problem on paths can be solved in linear time.*

Theorem 4. *The mixed total covering problem on cycles can be solved in linear time.*

Lemma 3 ([1]). *For a clique K_n, $\alpha_2(K_n) = \lceil \frac{n}{2} \rceil$.*

Lemma 4. *For a clique K_n, if $R = \emptyset$, all edges $e \in B$, and all vertices $v \in F$, then $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$.*

Proof. Since all vertices $v \in F$ and the edges covered by v are also covered by an incident edge of v, there exists an $\alpha_{2\mathrm{m}}$-set M of G with partition $\mathcal{P}$ such that $M \cap V = \emptyset$. It clear that removing all elements in $N_{\mathrm{m}}[e]$ from K_n yields a K_{n-2} for $n \geqslant 3$. The value of $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n))$ can be computed by the following recurrence formula:

$$\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \begin{cases} 1 & \text{if } n = 2, 3, \\ \alpha_{2\mathrm{m}}(\mathcal{P}(K_{n-2})) + 1 & \text{if } n > 3. \end{cases}$$

Therefore, $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$ for $n \geqslant 2$ and the lemma follows.

Corollary 2. *For a clique K_n, if $R = \emptyset$, all edges $e \in B$, and at least one vertex is in F, then $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$.*

Proof. Let $\mathcal{P}_1(K_n)$ be the partition described in the statement of this corollary and $\mathcal{P}_2(K_n)$ be the partition described in Lemma 4. Clearly, by Lemma 4, $\lfloor \frac{n}{2} \rfloor = \alpha_{2\mathrm{m}}(\mathcal{P}_2(K_n)) \leqslant \alpha_{2\mathrm{m}}(\mathcal{P}_1(K_n))$. Then, by using a similar argument as in Lemma 4, we can obtain $\alpha_{2\mathrm{m}}(\mathcal{P}_1(K_n)) \leqslant \lfloor \frac{n}{2} \rfloor$. Therefore, $\alpha_{2\mathrm{m}}(\mathcal{P}(K_n)) = \lfloor \frac{n}{2} \rfloor$. This completes the proof.

Note that, in Corollary 2, finding an $\alpha_{2\mathrm{m}}$-set is equivalent to selecting $\lfloor \frac{n}{2} \rfloor$ disjoint pair of vertices. Thus an $\alpha_{2\mathrm{m}}$-set of a clique with a partition described in Corollary 2 can be found in $O(n)$ time.

4 Computing $\alpha_2(G)$ for Block-Cactus Graphs G

An *end block* of a block-cactus graph is a block containing exactly one cut vertex which is called the *hanging vertex* of the end block. A path P in G is called an *end path* if every vertex in P has degree 2 in G except that the degree of one end vertex is 1 and the degree of the other end vertex is greater than 2 in G. In an end path P, the end vertex of degree greater than 2 in G is also called the *hanging vertex* of P. Hereafter, if an end block of G is a K_2 with no hanging vertex, then we use the end path containing it to replace the K_2 as an end block. Let $G \ominus S$ denote the resulting graph by removing end block S from G except the hanging vertex in S.

Let $\mathcal{P}$ be a partition of G, v a hanging vertex in end block S, and $G' = G \ominus S$. Let $\mathcal{P}_{\mathrm{f}}$, $\mathcal{P}_{\mathrm{b}}$, and $\mathcal{P}_{\mathrm{r}}$ be the partitions on G with the only difference from $\mathcal{P}$ on $v \in F, B$, and R, respectively. Furthermore, let $\mathcal{P}_{\mathrm{fr}}(S) = A_2(\mathcal{P}_{\mathrm{f}}, vw)$ (respectively, $\mathcal{P}_{\mathrm{br}}(S) = A_2(\mathcal{P}_{\mathrm{f}}, vw)$), where vw is an incident edge of v in S. Conversely, we use $\mathcal{P}_{\mathrm{f}\bar{\mathrm{r}}}(S)$ and $\mathcal{P}_{\mathrm{b}\bar{\mathrm{r}}}(S)$ to denote the partitions with $vw \notin R$ for $v \in F$ and $v \in B$, respectively. Note that, in $\mathcal{P}_{\mathrm{b}\bar{\mathrm{r}}}(S)$, w must be in the resulting $\alpha_{2\mathrm{m}}$-set of S, and, in $\mathcal{P}_{\mathrm{f}\bar{\mathrm{r}}}(S)$, if $vw \in B$, then either w or another incident edge of w must be in the resulting $\alpha_{2\mathrm{m}}$-set of S. Furthermore, if $vw \in R$, then $\mathcal{P}_{\mathrm{f}\bar{\mathrm{r}}}(S) = \infty$. We can define $\mathcal{P}_{\mathrm{fr}}(G)$ and $\mathcal{P}_{\mathrm{f}\bar{\mathrm{r}}}(G)$ in a similar way. Note that, For brevity, in the following, we use $\alpha_x(G)$ to denote $\alpha_{2\mathrm{m}}(\mathcal{P}_x(G))$ for $x \in \{\mathrm{f}, \mathrm{b}, \mathrm{r}, \mathrm{f}\bar{\mathrm{r}}, \mathrm{fr}, \mathrm{b}\bar{\mathrm{r}}, \mathrm{br}\}$.

Proposition 3. *For an end block S of a block-cactus graph, the following equations are true:*
(1) $\alpha_{\mathrm{f}}(S) = \min\{\alpha_{\mathrm{f}\bar{\mathrm{r}}}(S), \alpha_{\mathrm{fr}}(S)\}$,
(2) $\alpha_{\mathrm{b}}(S) = \min\{\alpha_{\mathrm{b}\bar{\mathrm{r}}}(S), \alpha_{\mathrm{br}}(S)\}$,
(3) $\alpha_{\mathrm{f}\bar{\mathrm{r}}}(S) \leqslant \alpha_{\mathrm{b}\bar{\mathrm{r}}}(S)$, *and*
(4) $\alpha_{\mathrm{fr}}(S) \leqslant \alpha_{\mathrm{br}}(S)$.

Lemma 5. *Let G be block-cactus graph with partition $\mathcal{P}$, v a hanging vertex in end block S, and $G' = G \ominus S$. If $v \in R$, then $\alpha_{2\mathrm{m}}(G) = \alpha_{\mathrm{r}}(G') + \alpha_{\mathrm{r}}(S) - 1$.*

Proof. Since v is in both $\alpha_{2\mathrm{m}}$-sets of G' and S with partition $\mathcal{P}$, the lemma follows directly.

Lemma 6. *Let G be block-cactus graph with partition $\mathcal{P}$, v a hanging vertex in end block S, and $G' = G \ominus S$. If $v \in B$, then the following statement is true:*

$$\alpha_{2\mathrm{m}}(G) = \begin{cases} \alpha_{\mathrm{b}}(G') + \alpha_{\mathrm{f}\bar{\mathrm{r}}} & \text{if } \alpha_{\mathrm{f}} < \alpha_{\mathrm{b}} \text{ and } \alpha_{\mathrm{f}\bar{\mathrm{r}}} < \alpha_{\mathrm{fr}}, \\ \alpha_{2\mathrm{m}}(\mathcal{P}_1(G')) + \alpha_{\mathrm{fr}} & \text{if } \alpha_{\mathrm{f}} < \alpha_{\mathrm{b}} \text{ and } \alpha_{\mathrm{f}\bar{\mathrm{r}}} \geqslant \alpha_{\mathrm{fr}} \\ \alpha_{\mathrm{f}}(G') + \alpha_{\mathrm{b}\bar{\mathrm{r}}} & \text{if } \alpha_{\mathrm{f}} = \alpha_{\mathrm{b}} < \alpha_{\mathrm{r}} \text{ and } \alpha_{\mathrm{b}\bar{\mathrm{r}}} < \alpha_{\mathrm{br}}, \\ \alpha_{2\mathrm{m}}(\mathcal{P}_1(G')) + \alpha_{\mathrm{br}} & \text{if } \alpha_{\mathrm{f}} = \alpha_{\mathrm{b}} < \alpha_{\mathrm{r}} \text{ and } \alpha_{\mathrm{b}\bar{\mathrm{r}}} \geqslant \alpha_{\mathrm{br}}, \\ \alpha_{\mathrm{r}}(G') + \alpha_{\mathrm{r}} - 1 & \text{if } \alpha_{\mathrm{f}} = \alpha_{\mathrm{b}} = \alpha_{\mathrm{r}}, \end{cases}$$

where $\mathcal{P}_1 = A_3(\mathcal{P}_{\mathrm{f}}, v)$ and $\alpha_x(S)$ is abbreviated as α_x.

Lemma 7. *Let G be block-cactus graph with partition $\mathcal{P}$, v a hanging vertex in end block S, and $G' = G \ominus S$. If $v \in F$, then the following equality holds:*

$$\alpha_{2\mathrm{m}}(G) = \begin{cases} \alpha_{\mathrm{f}}(G') + \alpha_{\mathrm{f}\bar{\mathrm{r}}} & \text{if } \alpha_{\mathrm{f}} \leqslant \alpha_{\mathrm{b}} < \alpha_{\mathrm{r}} \text{ and } \alpha_{\mathrm{f}\bar{\mathrm{r}}} < \alpha_{\mathrm{fr}}, \\ \alpha_{2\mathrm{m}}(\mathcal{P}_1(G')) + \alpha_{\mathrm{fr}} & \text{if } \alpha_{\mathrm{f}} \leqslant \alpha_{\mathrm{b}} < \alpha_{\mathrm{r}} \text{ and } \alpha_{\mathrm{f}\bar{\mathrm{r}}} \geqslant \alpha_{\mathrm{fr}}, \\ \alpha_{\mathrm{r}}(G') + \alpha_{\mathrm{r}} - 1 & \text{if } \alpha_{\mathrm{f}} = \alpha_{\mathrm{b}} = \alpha_{\mathrm{r}}, \end{cases}$$

where $\mathcal{P}_1 = A_3(\mathcal{P}_{\mathrm{f}}, v)$ and $\alpha_x(S)$ is abbreviated as α_x.

Theorem 5. *The total covering problem on block-cactus graphs can be solved in $O(n)$ time.*

Proof. To compute $\alpha_2(G)$ of a block-cactus graph G, we can set all elements in $V \cup E$ to be in B. Then, by Lemmas 5-7, we can compute $\alpha_{2\mathrm{m}}(G)$. Note that, by Theorems 3-4, Lemma 4, and Corollary 2, the value of $\alpha_{2\mathrm{m}}(S)$ of each end block S can be computed in linear time. Finally, only one block S' is remained. By using Theorems 3-4, Lemma 4, and Corollary 2 again, the value of $\alpha_{2\mathrm{m}}(S')$ can also be computed. Thus, by summing all obtained values according to Lemmas 5-7, $\alpha_2(G)$ can be computed and the total time-complexity on computing $\alpha_2(G)$ is linear. Note that we can also obtained an $\alpha_{2\mathrm{m}}$-set of G when computing $\alpha_{2\mathrm{m}}(G)$. This completes the proof.

5 Concluding Remarks

In this paper, we solve the total covering problem on block-cactus graphs in linear time. As a further study, it is interesting to consider the weighted total covering problem on block-cactus graphs. The *weighted total covering problem* is to find a total covering set with the minimum weight when every element in $V \cup E$ is associated with a weight.

References

1. Alavi, Y., Behzad, M., Lesniak-Foster, L.M., Nordhaus, E.A.: Total matchings and total coverings of graphs. J. Graph Theor. 1, 135–140 (1977)
2. Alavi, Y., Liu, J., Wang, J., Zhang, Z.: On total covers of graphs. Discrete Math. 100, 229–233 (1992)
3. Cockayne, E.J., Goodman, S., Hedetniemi, S.T.: A linear algorithm for the domination number of a tree. Inform. Process. Lett. 4, 41–44 (1975)
4. Erdős, P., Meir, A.: On total matching numbers and total covering numbers of complementary graphs. Discrete Math. 19, 229–233 (1977)
5. Haynes, T.W., Hedetniemi, S.T., Slater, P.J.: Fundamentals of Domination in Graphs. Marcel Dekker, New York (1998)
6. Meir, A.: On total covering and matching of graphs. J. Comb. Theory B 24, 164–168 (1978)
7. Mili, L., Baldwin, T., Phadke, A.: Phasor measurement placement for voltage and stability monitoring and control. In: Proc. of the EPRI-NSF Workshop on Application of Advanced Mathematics to Power System, San Francisco, CA (1991)
8. Natarajan, K.S., White, L.J.: Optimum domination in weighted trees. Inform. Process. Lett. 7, 261–265 (1978)
9. Rautenbach, D., Volkmann, L.: The domatic number of block-cactus graphs. Discrete Math. 187, 185–193 (1998)
10. Zhao, Y.C., Kang, L.Y., Sohn, M.Y.: The algorithmic complexity of mixed domination in graphs. Theore. Comput. Sci. 412, 2387–2392 (2011)

Randomized Self-stabilization under Distributed Daemon for 6-Coloring Planar Graph

Chi-Hung Tzeng[1], Jehn-Ruey Jiang[2], Shing-Tsaan Huang[2], and Cheng-Feng Yeh[2]

[1] Dept. of Computer Science, National Tsing Hua University, Taiwan
clark@cs.nthu.edu.tw
[2] Dept. of Computer Science and Information Engineering, National Central University, Taiwan
{jrjiang,sthuang}@csie.ncu.edu.tw, 975402006@cc.ncu.edu.tw

Abstract. Self-stabilization is a fault-tolerant mechanism that enables a distributed system to recover from transient faults. In this paper, we consider the coloring problem and propose the first self-stabilizing algorithm under the distributed daemon model to 6-color planar graphs. The algorithm is randomized, anonymous and uniform. Starting from any initial configuration, it finds a proper coloring in $O(n)$ rounds for an n-node graph.

Keywords: Distributed computing, Planar graph, Randomization, Self-stabilization, Graph Coloring.

1 Introduction

A distributed system is modeled by a connected, undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. We assume that nodes are *anonymous* and *uniform*; that is, every node has no unique ID and runs the same algorithm. Two nodes u and v are said to be neighbors if $(u, v) \in E$. The set of neighbors of a node u is denoted by $N.u$. Each node maintains a set of variables representing its local state. All local states at a specific instance form a *system state*. The computation of the system is a sequence $(c_0,c_1,...)$ of system states, where c_0 is the initial state and c_{k+1} is the successor of c_k for an integer $k \geq 0$. Note that we use $c_i \rightsquigarrow c_j$ to indicate that the computation contains the subsequence $(c_i, c_{i+1}, \ldots , c_j)$, where $0 \leq i \leq j$. System states are divided into two categories: legitimate states and illegitimate states. Ideally, the system should remain in legitimate states to work properly, but unexpected *transient faults* may bring it into an illegitimate state. Therefore, it needs to deal with such situations by some sort of fault-tolerant mechanisms, one of which is called *self-stabilization* [2].

A system is called *self-stabilizing* if its computation satisfies the two criteria: *convergence* and *closure*. The convergence criterion requires that, for any initial state c_0, there exists a legitimate state c_l such that $c_0 \rightsquigarrow c_l$, where $l \geq 0$ is a finite integer. On the other hand, the closure property requires that every configuration following a legitimate state c_l is also legitimate; that is, $c_l \rightsquigarrow c_{l+k}$ for legitimate states c_l and c_{l+k}, where $l \geq 0$ and $k \geq 1$. With this definition, self-stabilization enables a distributed system to deal with transient faults as follows. Once the system undergoes transient

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 41–48.
DOI: 10.1007/978-3-642-35452-6_6

faults, it is conceptually considered to be in an arbitrary initial system. With the convergence and closure criteria, it will return to legitimate states in finite time and remain so thereafter.

A self-stabilizing algorithm is usually presented by a set of *rules* of the form *guard → action*. The guard is a boolean function involving the local states of a node and its neighbors. If it is evaluated to be true, then the rule is said to be enabled. The action is a set of program statements that change local node states. A node is said to be *privileged* if it has an enabled rule; *unprivileged* otherwise. When a privileged node performs the program statements in the guarded action of an enabled rule; we say that the node executes the rule or makes a *move*. We may assume a *central daemon* or a *distributed daemon* to decide the execution sequence of the nodes. The central daemon selects an arbitrary node to make a move, while the distributed daemon selects a non-empty set of privileged nodes, if any, to execute rules. After every of the selected nodes execute a rule, the next period of selection and execution begins, and so on. The distributed daemon assumption is more practical than the central daemon assumption, but its correctness proof is more difficult.

In this paper, we take the distributed daemon assumption and design a randomized, self-stabilizing algorithm to 6-color planar graphs. In the algorithm, each node maintains a color for itself, and in legitimate states each node has a color different from those maintained by neighbors. The coloring in a legitimate state is said to be *proper*. A proper coloring can be directly applied to many scheduling problems. For example, it can be applied to the *local mutual exclusion problem* by allowing nodes of the same color to enter critical sections (CSs) at the same time. Nodes of different colors thus enter CSs in a round-robin manner [13]. Another application is to acquire local topology knowledge by a vertex coloring and an edge coloring [12].

Several self-stabilizing coloring algorithms [5, 6, 7, 8, 9, 11] have been proposed. They are all under the central daemon assumption to color different types of graphs with different numbers of colors. The algorithm in [5] takes 6 colors to color planar graphs by constructing *directed acyclic graphs* (DAG) with the out-degree less than 6. The algorithm in [11] takes 2 colors to color non-uniform bipartite graphs, in which there is a special node called root. By calculating the distance between itself and the root, a node receives the color 0 if the distance is even; color 1, otherwise. In [7] and [8], general graphs are colored by Δ+1 colors, where Δ is the maximum degree of graph nodes. In [9], an algorithm is proposed to 6-color planar graphs under a fair central daemon. A general approach is proposed in [6] to k-color various kinds of graphs, such as trees, series-parallel graphs and planar graphs, where k is an integer that depends on the graph considered. To the best of our knowledge, the algorithm proposed in this paper is the first self-stabilizing algorithm under the distributed daemon to color planar graphs. As we will show, the algorithm uses 6 colors to color n-node graphs with $O(n)$ rounds. The concept of the *round* is proposed in [3] to measure the time complexity of algorithms under the distributed daemon model. For a computation $\mathbb{C}$ = (c_0, c_1, ...), the first round of $\mathbb{C}$ is the shortest prefix (c_0, c_1, ... , c_f) of $\mathbb{C}$ such that every privileged node in c_0 has (1) executed at least one rule or (2) become unprivileged due to the moves of neighbors. Let state c_{f+1} be the successor of c_f. Then, the second round of $\mathbb{C}$ is the first round of the computation C' = (c_{f+1}, ...). The other rounds of computation C can be defined similarly.

The rest of this paper is organized as follows. Section 2 presents the proposed algorithm. Section 3 shows the correctness proofs and the time complexity analysis. Finally, Section 4 concludes the paper.

2 The Algorithm

Given a simple connected planar graph $G = (V, E)$, the proposed algorithm 6-colors G in $O(n)$ rounds, where $n = |V|$ is the number of nodes. Each node maintains two variables L and C to represent its local state. The variable L is the label of a node and its value is equal to or greater than 0; the variable C is the color of a node, and its value is in {0, 1, ..., 5}. We use $L.u$ and $C.u$ to denote the variables L and C maintained by a node u, respectively.

The overview of our algorithm is as follows. Nodes first decide their labels such that any node has less than 6 neighbors of the same or larger labels. Based on the labeling, the node set V is partitioned into K non-empty disjoint sets $V_0, V_1, ..., V_{K\text{-}1}$, where $V_k = \{u \mid u \in V, L.u = k\}$ is the set of nodes of label k, $0 \leq k \leq K$. Nodes are colored in an order from $V_{K\text{-}1}$ to V_0; that is, the nodes in $V_{K\text{-}1}$ determine their colors first, and then those in $V_{K\text{-}2}$, etc. A randomized technique is used when deciding the colors. As we will show, all nodes in V_k , $0 \leq k \leq K$ can determine their colors in an expected $O(1)$ rounds once all the nodes in $V_{i+1} \cup ... \cup V_{K\text{-}1}$ have determined their colors.

The labeling or the partitioning is based on the property that the total number of planar graph nodes of degrees less than 6 is at least 3, as derived below. Let $deg(u)$ denote the degree of a node u and let $\hat{V}_0$ be the set of nodes of degrees less than 6. The sum of the degrees of all nodes is at least $|\hat{V}_0| + 6(n\text{-}|\hat{V}_0|) = 6n - 5|\hat{V}_0|$. From Euler's formula, we have

$$3n - 6 \geq |E| = \frac{1}{2}\sum_{u \in V} deg(u) \geq \frac{(6n - 5|\hat{V}_0|)}{2}$$

and

$$|\hat{V}_0| \geq 3 \qquad (1)$$

Let $N^{(k)}.u = \{v \mid v \in N.u, L.v \geq k\}$ denote the set of u's neighbors whose labels are greater than or equal to k. The following rule is used to realize the label assignment.

R0: $L.u \neq \min\{k \mid k \geq 0, |N^{(k)}.u| < 6\} \rightarrow L.u = \min\{k \mid k \geq 0, |N^{(k)}.u| < 6\}$

By rule R0, each node u of a degree less than 6 sets their labels to 0 because $|N^{(k)}.u| < 6$ holds for any $k \geq 0$. Note that by Equation 1, at least 3 such nodes exist. After setting $L.u = 0$, obviously node u will not execute the rule to change its label again. The set V_0 eventually contains the nodes of degrees less than 6. Afterwards, each node u of $deg(u) < 6$ in $G - V_0$ sets its label to 1 and becomes a member of V_1. By repeating this argument, the node set V is partitioned into non-empty sets $V_0, V_1, ..., V_{K\text{-}1}$.

The next step is to decide a coloring once the labeling has completed. Because each node u has at most 5 neighbors with labels $\geq L.u$ and there are 6 colors available, it can choose a color not appearing in those neighbors. After node u recolors itself, ideally it becomes properly colored with respect to those neighbors in $N^{(L.u)}.u$. However, those neighbors may change their colors at the same time. As a result, node u could remain improperly colored. The distributed daemon could always select two neighboring nodes to recolor and such symmetric ties could always happen. To prevent this scenario from happening, we demand that any node randomly selects one of feasible colors to recolor itself.

Let $X.u = \{C.v \mid v \in N^{(L.u)}.u\}$ denote the set of colors used by u's neighbors that have the same or larger labels than u. When node u finds out that it has an improper color (i.e., $C.u \in X.u$), and that it can renew its color (i.e., $|X.u| < 6$), it randomly chooses a color available for itself. The recoloring rule is shown below.

$$\text{R1: } C.u \in X.u \wedge |X.u| < 6 \rightarrow C.u = \text{Random}(\{0, 1, ..., 5\} - X.u)$$

Rules R0 and R1 are the only rules in our algorithm, shown in Figure 1. Rule R0 is to find a labeling, whereas rule R1 is to find a proper 6-coloring. We assume that R0 has a higher priority. That is, a privileged node executes R0 if both rules are enabled.

R0: $L.u \neq \min\{k \mid k \geq 0, |N^{(k)}.u| < 6\} \rightarrow L.u = \min\{k \mid k \geq 0, |N^{(k)}.u| < 6\}$
R1: $C.u \in X.u \wedge |X.u| < 6 \rightarrow C.u = \text{Random}(\{0, 1, ..., 5\} - X.u)$

Fig. 1. The proposed algorithm

3 Correctness Proofs and Time Complexity Analysis

In this section, we prove the correctness of the algorithm and analyze its time complexity. We show that the labeling layer stabilizes in $O(n)$ rounds and the coloring layer stabilizes in another expected $O(n)$ rounds.

In this paper, we propose a randomized self-stabilizing algorithm to 6-color planar graphs. The algorithm partitions a graph into disjoint groups on the basis of the heredity property that any subgraph of a planar graph is also planar. Under the distributed daemon, the algorithm has the expected time complexity $O(n)$ rounds for an n-node graph, since it can partition nodes into $O(n)$ groups and then properly color a group of nodes in constant expected time.

We introduce some notations before starting the proofs. Let $\hat{V}_0 = \{u \mid deg(u) < 6 \text{ in } G\}$, $\hat{V}_k = \{u \mid deg(u) < 6 \text{ in } G - \bigcup_{j=0}^{k-1} \hat{V}j\ \}$ for $k \geq 1$, and K be the smallest integer such that $K \geq 1$ and $\hat{V}_K = \phi$. We use $\widehat{N}^{(k)}.u = N.u \cap \hat{V}_k$ to denote the set of u's neighbors that belong to $\hat{V}_k$. Please note that every $\hat{V}_k$ is invariant for a given G; on the other hand, V_k, the set of nodes that have label k, is variant due to the executions of rule R0.

By definition, we have the following properties:

Property 1. $N^{(k)}.u \subseteq N^{(j)}.u$ for $u \in V$ and $k \geq j \geq 0$.

Property 2. Let u $\in \hat{V}_k$. We have $|\bigcup_{j \geq k} \widehat{N}^{(j)}.u| < 6$ for $k \geq 0$. Moreover, we also have $|\bigcup_{j \geq k-1} \widehat{N}^{(j)}.u| \geq 6$ if $k \geq 1$.

We will use these properties in the proof of Lemma 1.

Lemma 1. Starting from any initial configuration, no node can execute R0 after K rounds.

Proof.
Because V can be partitioned into $\hat{V}_0$, $\hat{V}_1$, ..., $\hat{V}_{K\text{-}1}$, we use induction on $\hat{V}_\mathrm{k}$, $0 \leq k < K$, to prove this lemma. We show that, after the kth round, any node $u \in V_j$ has $L.u = j$ if $j \leq k$ and has $L.u \geq k + 1$ if $j \geq k + 1$.

(Basis) $k = 0$. First, consider a node $u \in \hat{V}_0$. Since $deg(u) < 6$, the condition $|N^{(i)}.u| < 6$ holds for any integer $i \geq 0$. If $L.u \neq 0$, node u executes R0 to set $L.u = 0$ in the 0th round. Now, consider a node $u \in \hat{V}_j$, where $j \geq k + 1 = 1$. Because node u has at least 6 neighbors and those neighbors have labels ≥ 0, the condition $|N^{(0)}.u| \geq 6$ holds. This condition makes u set $L.u \geq 1$ if $L.u = 0$ in the 0th round, so we have $L.u \geq 1$ after the 0th round.

(Hypothesis) Suppose that, after kth round, every node $u \in \hat{V}_j$ has $L.u = j$ if $j \leq k$ and has $L.u \geq k + 1$ if $j \geq k + 1$, where $0 \leq k < K$.

(Induction) Let's focus on what would happen in the $(k + 1)$th round. Consider a node $u \in \hat{V}_{k+1}$. By the hypothesis and by property 2, we have $N^{(k+1)}.u = \bigcup_{i \geq k+1} \hat{N}^{(i)}.u$ and $N^{(k)}.u = \bigcup_{i \geq k} \hat{N}^{(i)}.u$. Thus $|N^{(k+1)}.u| < 6$ and $|N^{(k)}.u| \geq 6$ hold. Combining with property 1, the minimum value i for satisfying $|N^{(i)}.u| < 6$ is $k + 1$. If $L.u \neq k + 1$, node u executes R0 to set $L.u = k + 1$ in the $(k + 1)$th round and keeps that label hereafter. Now, consider a node $u \in \hat{V}_j$, where $j \geq k + 2$. By the induction hypothesis, we have $N^{(k+1)}.u = \bigcup_{i \geq k+1} \hat{N}^{(i)}.u$. By property 2, this condition implies $|N^{(k+1)}.u| \geq 6$. Therefore, the minimum value i for satisfying $|N^{(i)}.u| < 6$ is at least k+2. If $L.u < k$+2, then node u executes R0 to set $L.u \geq k$+2 in the $(k + 1)$th round.

According to above discussion, this lemma holds. □

Based on Lemma 1 and the guard of rule R0, we have the following corollary.

Corollary 1. After K rounds, we have $|N^{(L.u)}.u| < 6$ for every node u.

Once the labeling has completed, there will be a proper coloring in another expected $O(K)$ rounds. To prove this, we say that node v has a partially proper color if $C.u \notin X.u$ (recall that $X.u = \{C.v \mid v \in N^{(L.u)}.u\}$). When every node has a partially proper color, there is a proper coloring, as will be shown in Lemma 3. In what follows, we show that every node receives a partially proper color in expected $O(n)$ rounds.

According to rule R1, node u selects a color not in $X.u$ to recolor itself. Since there are 6 colors and $|X.u| = |N^{(L.u)}.u| < 6$, node u has a non-zero probability to pick up a partially proper color, even if nodes in $N^{(L.u)}.u$ change their colors at the same time.

To be more precisely, let u be a node such that $C.u \in X.u$ and let $v \in N^{(L.u)}.u$. After node u changes its color, the probability of $C.u \neq C.v$ is 1 if node v does not change its color. On the other hand, if node v changes $C.v$ simultaneously, then the probability of $C.u \neq C.v$ becomes

$$\frac{\{\min\{6 - |X.u|, 6 - |X.v|\}\}}{(6 - |X.u|)(6 - |X.v|)} = \frac{1}{\{\max\{6 - |X.u|, 6 - |X.v|\}\}} \geq \frac{1}{6}$$

for $0 \leq |X.u|$, $|X.v| < 6$. In short, the probability of $C.u \neq C.v$ is at least 1/6 after u executes R1. Since $|N^{(L.u)}.u| < 6$, the probability that node u receives a partially proper color is $p \geq (1/6)^5$. Thus the expected number of rounds for u to receive a partially proper color is

$$\sum_{k=1}^{\infty} k(1-p)^{k-1}p = \frac{1}{p} \leq 6^5 = O(1)$$

Based on this equation, we have the following lemma.

Lemma 2. If $|N^{(L.u)}.u| < 6$ holds for every node u, node u has a partially proper color in expected $O(K)$ rounds.

Proof.
This lemma can be proven by a recursive equation. Let $T(V)$ be the expected number of rounds for each node in V to receive a partially proper color. Because $V = \bigcup_{k=0}^{K-1} V_k$ and because nodes in $V_{K\text{-}1}$ choose their colors without involving the colors in $\bigcup_{k=0}^{K-2} V_k$, we have

$$T(V) = T(\bigcup_{k=0}^{K-1} V_k) = T(\bigcup_{k=0}^{K-2} V_k) + \mathrm{O}(1) = O(K).$$

Therefore, this lemma holds. □

Lemma 3. When every node has a partially proper color, there is a proper coloring of G.

Proof.
We prove this lemma by showing that two neighboring nodes have different colors.

Consider any two neighboring nodes u and v. Without loss of generality, we assume $L.u \leq L.v$, or, equivalently, $v \in N^{(L.u)}.u$. Since $C.u$ is a partially proper color, by definition, $C.u \neq C.v$ holds.

Theorem 1. Starting from any initial configuration, the expected number of rounds for getting a proper 6-coloring is $O(n)$ rounds.

Proof.
The process of stabilization can be divided into two stages. The first stage is to get a labeling that satisfies $|N^{(L.u)}.u| < 6$ for every node u. The second stage is to receive a partially proper color for every node. It suffices to prove this theorem by showing that the first stage takes $O(n)$ rounds and that the second stage takes, expectedly, $O(n)$ rounds.

Consider the first stage. According to corollary 1, this stage ends within K rounds, where K is the smallest integer such that $\hat{V}_K = \phi$. Equation 1 suggests that the maximum value of K is $O(n)$, so the first stages ends in $O(n)$ rounds. Now, consider the second stage. From Lemma 2, Lemma 3 and the property that every node uses at most 6 colors, we can infer that there is a proper 6-coloring of G after an expected number of $O(K) = O(n)$ rounds.

According to the above discussion, this theorem holds. □

4 Concluding Remarks

In this paper, we propose a randomized self-stabilizing algorithm to 6-color planar graphs. The algorithm partitions a graph into disjoint groups on the basis of the heredity property that any subgraph of a planar graph is also planar. Under the distributed daemon, the algorithm has the expected time complexity $O(n)$ rounds for an n-node graph, since it can partition nodes into $O(n)$ groups and then properly color a group of nodes in constant expected time.

Our algorithm can stabilize in $O(\log n)$ rounds just by adding the 7th color (i.e., by replacing 6 with 7 and replacing 5 with 6 in rules R0 and R1, respectively). The reason is shown below. Let $\hat{V}_0$ be the set of nodes of degrees less than 7 in G. By definition, the condition

$$|\hat{V}_0|+7(n-|\hat{V}_0|) \leq \sum_{u\in V} deg(u) = 2|E|$$

holds, where E is the edge set. In addition, Euler's formula implies $2|E| \leq 2(3n - 6)$. Hence we have $7n-6|\hat{V}_0| \leq 6n-12$, which in turns implies n-$|\hat{V}_0| \leq 5n/6$. That is, if we remove nodes of degrees less than 7, the number of remaining nodes is at most $(5/6)^n$. Thus, if we recursively remove nodes of degrees less than 7 for k times, the number of remaining nodes is at most $(5/6)^k n$. When $(5/6)^k n < 1$ (i.e., $k > \lceil \log_{6/5} n \rceil$), all nodes are removed from the graph. According to Rule 0, nodes of the kth time of removal may be labeled with k and included in subset V_k. That is, the algorithm can partition V into $O(\log n)$ non-empty subsets, and all the nodes in V_k is expected to be colored properly in a constant number of rounds after all the nodes in V_{k+1}, ... are colored. The expected time complexity to 7-color G is thus $O(\log n)$.

The heredity property could be applied to other kinds of graphs for getting a coloring that uses fewer colors. For example, we can use it to find a Δ-coloring for an irregular graph G, where Δ is the maximum degree of G. The approach is quite simple: just replace 6 with Δ and 5 with Δ-1 in rules R0 and R1. This method partly answers the open question raised in [8], which asks whether one can design a self-stabilizing algorithm to Δ-color an irregular graph.

To the best of our knowledge, there is no self-stabilizing algorithm for finding Δ-colorings of regular graphs. It is interesting to develop one for regular graphs except for *complete graphs* or *odd cycles*, which need at least Δ+1 colors to maintain a proper coloring [1].

References

1. Brooks, R.L.: On colouring the nodes of a network. Proceedings of the Cambridge Philosophical Society, Math. Phys. Sci. 37, 194–197 (1941)
2. Dijkstra, E.W.: Self-stabilizing systems in spite of distributed control. Communications of the ACM 17, 643–644 (1974)
3. Dolev, S., Israeli, A., Moran, S.: Uniform dynamic self-stabilizing leader election. IEEE Transactions on Parallel and Distributed Systems 8(4), 424–440 (1997)

4. Frederickson, G.N.: On linear-time algorithms for five-coloring planar graphs. Information Processing Letters 19(5), 219–224 (1984)
5. Ghosh, S., Karaata, M.H.: A self-stabilizing algorithm for coloring planar graph. Distributed Computing 7, 55–59 (1993)
6. Goddard, W., Hedetniemi, S.T., Jacobs, D.P., Srimani, P.K.: Self-stabilizing algorithms for orderings and colorings. International Journal of Foundations of Computer Science 16(1), 19–36 (2005)
7. Gradinariu, M., Tixeuil, S.: Self-stabilizing vertex coloring of arbitrary graphs. In: Proceedings of the International Conference on Principles of Distributed Systems, OPODIS, pp. 55–70 (2000)
8. Hedetniemi, S.T., Jacobs, D.P., Srimani, P.K.: Linear time self-stabilizing colorings. Information Processing Letters 87(5), 251–255 (2003)
9. Huang, S.T., Hung, S.S., Tzeng, C.H.: Self-stabilizing coloration in anonymous planar networks. Information Processing Letters 95(1), 307–312 (2005)
10. Lipton, R.J., Miller, R.E.: A batching method for coloring planar graphs. Information Processing Letters 7(4), 185–188 (1978)
11. Sur, S., Srimani, P.K.: A self-stabilizing algorithm for coloring bipartite graphs. Information Sciences 69(3), 219–227 (1993)
12. Masuzawa, T., Tixeuil, S.: On bootstrapping topology knowledge in anonymous networks. ACM Transactions on Autonomous and Adaptive Systems 4(1), 1–27 (2009)
13. Yeh, H.G., Zhu, X.: Resource-sharing system scheduling and circular chromatic number. Theoretical Computer Science 332, 447–460 (2005)

An $O^*(1.4786^n)$-Time Algorithm for the Maximum Induced Matching Problem[⋆]

Maw-Shang Chang[1], Ling-Ju Hung[1,⋆⋆], and Chau-An Miau[2]

[1] Department of Computer Science and Information Engineering
HungKuang University
43302 Sha Lu, Taichung, Taiwan
{mschang,ljhung}@sunrise.hk.edu.tw

[2] Department of Computer Science and Information Engineering
National Chung Cheng University
62102 Min-Hsiung, Chiayi, Taiwan
mca99m@cs.ccu.edu.tw

Abstract. Given a graph $G = (V, E)$ the MAXIMUM r-REGULAR INDUCED SUBGRAPH problem is to find a vertex set $R \subseteq V$ of maximum cardinality such that $G[R]$ is r-regular. An *induced matching* $M \subseteq E$ in a graph $G = (V, E)$ is a matching such that no two edges in M are joined by any third edge of the graph. The MAXIMUM INDUCED MATCHING problem is to find an induced matching of maximum cardinality. By definition the maximum induced matching problem is the maximum 1-regular induced subgraph problem. Gupta *et al.* gave an $o(2^n)$ time algorithm for solving the MAXIMUM r-REGULAR INDUCED SUBGRAPH problem. This algorithm solves the MAXIMUM INDUCED MATCHING problem in time $O^*(1.6957^n)$ where n is the number of vertices in the input graph. In this paper, we show that the maximum induced matching problem can be reduced to the maximum independent set problem and we give a more efficient algorithm for the MAXIMUM INDUCED MATCHING problem running in time $O^*(1.4786^n)$.

1 Introduction

A graph $G = (V, E)$ is called *r-regular* if for any $v \in V$, the degree of v is equal to r. The problem to compute a *maximum r-regular induced subgraph* in a graph $G = (V, E)$ is to find a vertex set $R \subseteq V$ of maximum cardinality such that $G[R]$ is r-regular. We list the formal definition of the problem as follows.
MAXIMUM r-REGULAR INDUCED SUBGRAPH

Input: A graph $G = (V, E)$ and an integer $r \geq 0$.
Output: A vertex set $R \subseteq V$ of maximum cardinality such that the induced subgraph of G on R, $G[R]$, is r-regular.

[⋆] This research is partially supported by the National Science Council of Taiwan under grants NSC 99–2221–E–241–015–MY3 and NSC 101–2221–E–241–019–MY3.

[⋆⋆] Ling-Ju Hung is supported by National Science Council of Taiwan under grant NSC 101–2811–E–241–002.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 49–58.
DOI: 10.1007/978-3-642-35452-6_7

An *induced matching* $M \subseteq E$ in a graph $G = (V, E)$ is a matching such that no two edges in M are joined by any third edge of the graph. The MAXIMUM INDUCED MATCHING problem is to find an induced matching of maximum cardinality.

MAXIMUM INDUCED MATCHING [20]
Input: A graph $G = (V, E)$.
Output: A vertex set $S \subseteq V$ of maximum cardinality such that the edge set of $G[S]$ is an induced matching of G.

The problem for finding a maximum r-regular induced subgraph is NP-complete for any $r \geq 0$ [2]. Notice that when $r = 0$, the MAXIMUM r-REGULAR INDUCED SUBGRAPH problem is equivalent to the MAXIMUM INDEPENDENT SET problem. When $r = 1$, it is exactly the MAXIMUM INDUCED MATCHING problem.

Known results. The MAXIMUM INDUCED MATCHING problem was introduced by Stockmeyer and Vazirani [20]. It is NP-complete in general graphs [20]. It is even NP-complete in bipartite graphs [15], planar graphs [12], r-regular graphs, $r \geq 5$, line graphs and Hamiltonian graphs [13]. This problem can be solved in polynomial time for trees [8, 21], chordal graphs [3], weakly chordal graphs [4], HHD-free graphs [14], and $2P_3$-free graphs [16]. The MAXIMUM INDUCED MATCHING problem is W[1]-hard in general graphs [18] and is FPT in planar graphs [17,6,11]. Binkele-Raible *et al.* showed that the problem to compute the upper irredundance number can be reduced to solve the MAXIMUM INDUCED MATCHING problem in bipartite graphs [1]. They gave a fixed-parameter algorithm running in time $O^*(1.69563^k)$ for solving the MAXIMUM INDUCED MATCHING problem in bipartite graphs where k denotes the number of vertices to be deleted for obtaining an induced matching [1]. The MAXIMUM INDUCED MATCHING problem is APX-hard even on $4r$-regular graphs for $r \geq 1$ [10, 5]. Gupta *et al.* gave an $o(2^n)$ time algorithm for solving the MAXIMUM r-REGULAR INDUCED SUBGRAPH problem [9]. When $r = 1$ (Maximum Induced Matching Problem), the running time of the algorithm is $O^*(1.6957^n)$.

In this paper, we show that the MAXIMUM INDUCED MATCHING problem can be reduced to the MAXIMUM INDEPENDENT SET problem. This implies that one can take advantage of exact algorithms for solving the MAXIMUM INDEPENDENT SET problem to solve the MAXIMUM INDUCED MATCHING problem. Moreover, we design a branch-and-reduce algorithm for the MAXIMUM INDUCED MATCHING problem. For solving the problem more efficiently, we combine the branch-and-reduced algorithm and an exact algorithm for solving the MAXIMUM INDEPENDENT SET problem together to obtain an exact algorithm running in time $O^*(1.4786^n)$. This paper is organized as follows. In Section 2, we define some notation and show that the MAXIMUM INDUCED MATCHING problem can be reduced to the MAXIMUM INDEPENDENT SET problem. In Section 3, we give a branch-and-reduce algorithm running in time $O^*(1.5099^n)$ for the MAXIMUM INDUCED MATCHING problem. Finally, we present a more efficient algorithm running in time $O^*(1.4786)$ in Section 4.

2 Preliminaries

In this paper, we use a modified O-notation, O^* to bound the running time of exponential time algorithms asymptotically. For functions f and g, $f(n) = O^*(g(n))$ if $f(n) = O(g(n) \cdot poly(n))$ where $poly(n)$ is a polynomial [7].

The strategy *branch and reduce* is one of the algorithmic techniques to design exact algorithms. A typical branch-and-reduce algorithm consists of *reduction rules* and *branching rules*. When reduction rules are applied, the input size is reduced in polynomial time. When branching rules are applied, the problem can be solved recursively since all those corresponding smaller instances of the problem are solved. Search trees are often used to illustrate the execution of a branch-and-reduce algorithm. The root of a search tree represents the input of the problem, every child of the root represents a smaller instance reached by applying a branching rule associated with the instance of the root. One can recursively assign a child to a node in the search tree when applying a branching rule. Notice that we do not assign a child to a node when applying a reduction rule. The running time of a branching algorithm is usually measured by the maximum number of leaves in its corresponding search tree. Let $T(n)$ be the maximum number of leaves in any search tree where n is the input size. We analyze each branching rule and use the worst-case time complexity over all branching rules as an upper bound of the running time. Let b be any branching rule. When rule b is applied, the current instance is branched into $r \geq 2$ instances of size at most $n-t_1, n-t_2, \ldots, n-t_r$. Note that $n \geq t_i$ for $i = 1, 2, \ldots, r$. We call $\mathbf{b} = (t_1, t_2, \ldots, t_r)$ the *branching vector* of branching rule b. This can be formulated in a linear recurrence

$$T(n) \leq T(n-t_1) + T(n-t_2) + \cdots + T(n-t_r).$$

The running time of the branch-and-reduce algorithm using only the branching rule b is $O^*(\gamma^n)$ where γ is the unique positive real root of

$$x^n - x^{n-t_1} - x^{n-t_2} - \cdots - x^{n-t_r} = 0.$$

The real number γ is called the *branching number* of the branching rule b. Compute γ_i for each branching rule b_i. An upper bound of the number of leaves in the search tree is $O^*(c^n)$ where $c = \max_i \gamma_i$. If the running time of the algorithm corresponding to one node of the search tree is polynomial, the running time of a branch-and-reduce algorithm is $O^*(c^n)$. For more details about branching-and-reduce algorithms, please refer to the book [7].

Given a graph $G = (V, E)$, we use n to denote the number of vertices in G. For a vertex v in G, we use $N(v)$ to denote the *open neighborhood* of v and use $N[v] = N(v) \cup \{v\}$ to denote the *closed neighborhood* of v. Two vertices u and v are called *true twins* in G if $N[u] = N[v]$ and are called *false twins* if $N(u) = N(v)$. A vertex v is called an *isolated vertex* if $|N(v)| = 0$ and is called a *degree-one vertex* if $|N(v)| = 1$. For an edge $(u, v) \in E$, define the degree of (u, v) to be $deg(u, v) = |N(u) \cup N(v) \setminus \{u, v\}|$. An edge (u, v) is called an *isolated edge* if $deg(u, v) = 0$ and is called a *degree-one edge* if $deg(u, v) = 1$. For a vertex subset X, let $G[X]$ denote the graph induced by X in G. We use $G \setminus X$ to denote $G[V \setminus X]$. Let $W \subseteq E$ be a subset of edge set, use $V(W) = \cup_{(u,v) \in W} \{u, v\}$. For a vertex subset $U \subseteq V$, let $N(U) = \cup_{u \in U} N(u) \setminus U$ and $N[U] = N(U) \cup U$. For two vertices u and v in a graph G, we use $d_G(u, v)$ to denote the distance between u and v in G.

We close this section by showing that the MAXIMUM INDUCED MATCHING problem can be reduced to the MAXIMUM INDEPENDENT SET problem.

Definition 1. *For a graph $G = (V, E)$, $L(G) = (E, E')$, with for $(u, v), (x, y) \in E$, $(u, v) \neq (x, y)$, $((u, v), (x, y)) \in E'$ if and only if $u = x$ or $u = y$ or $v = x$ or $v = y$, is called the* line graph *of G.*

Definition 2. *Let $G = (V, E)$ be a graph and $L(G) = (E, E')$ be the line graph of G. Define $L^2(G) = (E, E'')$ to be the square graph of $L(G)$,* i.e.,

$$E'' = E' \cup \{(e_1, e_2) \mid e_1, e_2 \in E, d_{L(G)}(e_1, e_2) = 2\}.$$

Lemma 1. *Let $G = (V, E)$ be a graph and $H = (E, E') = L^2(G)$ be the square graph of the line graph of G. Then M is an induced matching of G if and only if M is an independent set of H.*

Proof. Suppose that M is an induced matching of G. Then all edges in $G[M]$ have no common endvertices and there is no edge in G incident to any two vertices in $V(M)$. Thus M forms an independent set in H. Conversely, suppose that I is an independent set in H. Notice that every vertex in H represents an edge in G. Since no two vertices in I are adjacent in H, by definition their corresponding edges, say (u, v) and (x, y), in G satisfy that $u \neq x$, $u \neq y$, $v \neq x$, and $v \neq y$, and $d_{L(G)}((u, v), (x, y)) > 2$, *i.e.,* $(u, x), (u, y), (v, x), (v, y) \notin E$. Thus I is an induced matching in G. This completes the proof. □

3 A Branch-and-Reduce Algorithm

We present a branch-and-reduce algorithm for the Maximum Induced Matching problem and analyze the running time of the algorithm using the number n of vertices in the input graph as the size of the input.

For simplicity, we denote the size of a maximum induced matching of an undirected graph G by $mim(G)$. When the algorithm puts a vertex v into the solution set, we say that it *selects* v. When the algorithm decides that a vertex v is not in the solution set, we say that it *discards* v. Before we present the algorithm, we define a rule called **Standard Branching Rule**: When the algorithm applies this rule, it selects a vertex v of degree d and recursively solves $d + 1$ Maximum Induced Matching problems on graphs $G_i = G \setminus U_i$, respectively, where $1 \leq i \leq d + 1$, $N(v) = \{u_1, u_2, \ldots, u_d\}$, $U_1 = \{v\}$ and for $i \geq 2$, $U_i = N[\{v, u_{i-1}\}]$. And then

$$mim(G) = \max\{mim(G_1), \max_{2 \leq i \leq d+1} (2 + mim(G_i))\}.$$

In the first branch of standard branching rule, it discards v. In each of the other branches, it selects v and a neighbor u and discards $N(\{v, u\})$. We give the following observations. Due to the space limitation, some proofs are omitted.

Lemma 2 (Disconnected Rule). *If G is disconnected and C is a component of G, then $mim(G) = mim(G[C]) + mim(G \setminus C)$.*

Lemma 3 (Isolated-Vertex Rule). *Let v be an isolated vertex of G. Then v is not in any solution of the* Maximum Induced Matching *problem. That is $mim(G) = mim(G \setminus \{v\})$.*

Lemma 4 (Isolated-Edge Rule). *Let (u, v) be an isolated edge of G. Then all solutions of the* MAXIMUM INDUCED MATCHING *problem contain both u and v. That is $mim(G) = 2 + mim(G \setminus \{u, v\})$.*

Lemma 5 (False-Twin Rule). *Let $G = (V, E)$ be a graph and u and v are false twins in G. Then $mim(G) = mim(G \setminus \{v\})$.*

Proof. It is easy to see that only one of u and v can be selected in an optimal solution since they are false twins. Suppose that S is an optimal solution that contains v but does not contain u. We obtain that $S \cup \{u\} \setminus v$ is also an optimal solution. This completes the proof. □

Lemma 6 (Degree-One Edge Rule). *Let $G = (V, E)$ be a graph and (u, v) be a degree-one edge of G satisfying one of the following conditions.*

1. *u is of degree two and v is of degree one;*
2. *Both u and v are of degree two, and $w \in N(u) \cap N(v)$, $deg(w) = 3$.*

Then $mim(G) = 2 + mim(G \setminus (N(u) \cup N(v)))$.

Proof. Suppose that one of u and v has degree two and the other has degree one. Let u be the endvertex of the edge (u, v) having degree one and v be the endvertex of the edge (u, v) having degree two. Suppose that S is an optimal solution not containing v. We see that $u \notin S$ since u can be matched only with v. Let w be the other neighbor of v. It is easy to see that $w \in S$ otherwise S is not maximal. Let $x \in S$ be the vertex matched with w. We may remove w and x from S and add v and u into S to obtain another optimal solution that selects both u and v. Suppose that S is an optimal solution that contains v and w and $u \notin S$. We obtain that $S \setminus \{w\} \cup \{u\}$ is also an optimal solution. Suppose that both u and v has degree two, w is their common neighbor and w has degree three. Suppose that S is an optimal solution not containing u then (v, w) is a degree-one edge of the first case, $v, w \in S$. We may remove w from S and add u into S to obtain another optimal solution that selects both u and v. This shows that $mim(G) = 2 + mim(G \setminus (N(u) \cup N(v)))$. □

Lemma 7 (Maximum-Degree-Two Rule). *If the maximum degree of G is at most two, then the* MAXIMUM INDUCED MATCHING *problem can be solved in polynomial time.*

Lemma 8 (Degree-One Vertex Rule). *Let $G = (V, E)$ be a graph, u be a degree-one vertex in G, $v \in N(u)$ and $deg(v) \geq 3$. Then*

$$mim(G) = \max\{mim(G \setminus \{u, v\}), 2 + mim(G \setminus N[v])\}.$$

Proof. If v is discarded, u is also discarded since u can be matched only with v. Let S be an optimal solution selecting v and $x \in N(v) \setminus \{u\}$ be the vertex matched with v. We see that $(S \cup \{u\}) \setminus \{x\}$ is also a solution of the same size. Thus either both u and v are discarded or both u and v are selected. This completes the proof. □

Lemma 9 (True Twin Rule). *Let $G = (V, E)$ be a graph and v and z be a pair of true twins in G. Then*

$$mim(G) = \max\{mim(G \setminus \{v, z\}), 2 + mim(G \setminus N[v])\}.$$

Proof. There are three cases, both v and z are discarded; only one of v and z is selected; or both v and z are selected. Let S be an optimal solution that selects v and discards z. It is easy to see that a vertex $x \in N(v) \setminus \{z\}$ must be selected in S, otherwise S is not optimal. Thus we obtain that $S \cup \{z\} \setminus \{x\}$ is also an optimal solution. This shows that if an optimal solution selects one of v and z, it selects both of them. Thus either both v and z are discarded or both v and z are selected. This completes the proof. □

Lemma 10 (Domination Rule). *Let v be a vertex of G and u be a neighbor of v with $N[u] \subset N[v]$. Then there is a solution of the* MAXIMUM INDUCED MATCHING *problem that either v is discarded or both v and u are selected from G. That is*

$$mim(G) = \max\{mim(G \setminus v), 2 + mim(G \setminus N[v])\}.$$

Proof. Suppose that S is an optimal solution that v is selected and vertex $x \in N(v) \setminus \{u\}$ is the vertex matched with v. We see that $S \cup \{u\} \setminus \{x\}$ is a solution of the same size. Thus there is a solution of the MAXIMUM INDUCED MATCHING problem that either v is discarded or both u and v are selected. This completes the proof. □

Lemma 11 (Degree-Two Vertex Rule 1). *There is a degree-two vertex z in G satisfying the condition that the degree of one of its two neighbors is two and the degree of the other neighbor is at least three. Let $x \in N(z)$ that $deg(x) = 2$ and $v \in N(z)$ that $deg(v) \geq 3$. Then either both x and z are selected or v and p, $p \in N(v)$, are selected. That is*

$$mim(G) = \max\{2 + mim(G \setminus (N[x] \cup N[z])), 2 + \max_{U \in \mathbb{U}}(mim(G \setminus (N[U])\}$$

where $\mathbb{U} = \{\{v, p\} \mid p \in N(v)\}$.

Proof. If v is discarded from G, then (z, x) is a degree-one edge. We may applied degree-one edge reduction rule to select both z and x. Thus, if v is selected, then there is an optimal solution that also selects one neighbor of v. This completes the proof. □

Lemma 12 (Degree-Two Vertex Rule 2). *There is a degree-two vertex z in G satisfying the condition that the degree of its two neighbors is at least three. Let $x, v \in N(z)$ that $deg(x) \leq deg(v)$. Then either all vertices in $N[z]$ are discarded, or both x and z are selected, or v and p, $p \in N(v)$, are selected. That is*

$$mim(G) = \max\{mim(G \setminus N[z]), 2 + \max_{U \in \mathbb{U}}(mim(G \setminus (N[U])\}$$

where $\mathbb{U} = \{\{x, z\}\} \cup \{\{v, p\} \mid p \in N(v)\}$.

Proof. If v is discarded from G, then z is a degree-one vertex. We may apply degree-one vertex rule to discard both z and x or select both z and x. Conversely, if v is selected, then there is an optimal solution that also selects one neighbor of v. This completes the proof. □

The description of the algorithm consists of a sequence of cases and subcases. To avoid a confusing nesting of if-then-else statements let us use the following convention: The first case which applies is used in the algorithm. Thus, inside a given case, the hypotheses of all previous cases are assumed to be false. The following algorithm computes for a given graph $G = (V, E)$ the maximum size of an induced matching.

1. **Maximum Degree-Two Rule.** All vertices are of degree at most two. In this case we solve the problem in polynomial time.
2. **Isolated-Vertex Rule.** There is an isolated vertex and let X be the set of all isolated vertices. Then $mim(G) = mim(G \setminus X)$.
3. **Isolated-Edge Rule.** There is an isolated edge and let X be the set of all isolated edges. Then $mim(G) = 2 \cdot |X| + mim(G \setminus V(X))$.
4. **Disconnected Rule.** If G is disconnected (let C be a component of G), solve the problem recursively in $G[C]$ and $G \setminus C$.
5. **False-Twin Rule.** There exist two vertices u and v that are false twins in G. Then $mim(G) = mim(G \setminus \{v\})$.
6. **Degree-One Edge Rule.** There is an edge (u, v) of degree one and let $X = N[\{u, v\}]$. Then $mim(G) = 2 + mim(G \setminus X)$.
7. **Degree-One Vertex Rule.** There is a degree-one vertex u, $v \in N(u)$, and $deg(v) \geq 3$. Then the algorithm either discards both u and v or selects both u and v. That is $mim(G) = \max\{mim(G \setminus \{u, v\}), 2 + mim(G \setminus N[v])\}$. The worst case happens when $deg(v) = 3$. The worst case branching vector is $(2, 4)$ and its branching number is 1.2721.
8. **True-Twin Rule.** There are two vertices v and z such that $N[z] = N[v]$, *i.e.*, v and z are true twins. Since Degree-One Edge Rule is not applicable, the degree of v and z are at least three. The algorithm either discards $\{v, z\}$ or selects $\{v, z\}$ and discards $N[v] \setminus \{v, z\}$. Hence $mim(G) = \max\{mim(G \setminus \{v, z\}), 2 + mim(G \setminus N[v])\}$. The worst branching vector is $(2, 4)$ and its branching number is 1.2721.
9. **Domination Rule.** Let v be a vertex of G and u be a neighbor of v satisfying that $N[u] \subset N[v]$. The degree of v is at least 3. The algorithm either discards v or selects $\{v, u\}$ and discards $N[v] \setminus \{v, u\}$. Hence $mim(G) = \max\{mim(G \setminus v), 2 + mim(G \setminus N[v])\}$. The worst branching vector is $(1, 4)$ and its branching number is 1.3803.
10. **Degree-Two Vertex Rule 1.** There is a degree-two vertex z satisfying the condition that the neighbor v with larger degree between the two neighbors of z is of degree at least three. Let x be the neighbor of z other than v. Since Domination Rule is not applicable, x and v are not adjacent and $|N[u] \setminus N[v]| \geq 1$ for any $u \in N(v)$. The degree of x is two. The algorithm either selects both x and z or selects v and one of v's neighbor p. That is $mim(G) = \max\{2 + mim(G \setminus (N[x] \cup N[z])), 2 + \max_{U \in \mathbb{U}}(mim(G \setminus (N[U])\}$ where $\mathbb{U} = \{\{v, p\} \mid p \in N(v)\}$. The worst case happens when degree of v is three. The branching vector is $(4, 5, 5, 5)$ and its branching number is 1.3413.
11. **Degree-Two Vertex Rule 2.** There is a degree-two vertex z satisfying the condition that the neighbor v with larger degree between the two neighbors of z is of degree at least three. Let x be the neighbor of z other than v. Since Domination Rule is not applicable, x and v are not adjacent and $|N[u] \setminus N[v]| \geq 1$ for any $u \in N(v)$. The degree of x is at least three. If we discard v then z becomes a degree-one vertex and hence there is an optimal solution that either discards x and z or selects $\{z, x\}$ and discards $N(x) \setminus \{z\}$. On the other hand, if v is selected, then there is an optimal solution that also selects one neighbor of v. Thus $mim(G) = \max\{mim(G \setminus N[z]), 2 + \max_{U \in \mathbb{U}}(mim(G \setminus (N[U])\}$ where $\mathbb{U} = \{\{x, z\}\} \cup \{\{v, p\} \mid p \in$

$N(v)\}$. The worst case happens when degree of v is three. The branching vector is $(3, 5, 5, 5, 5)$ and its branching number is 1.4336.

Note that for any $d \geq 3$ there is at most one d-regular graph assigned to a node of the search tree from the root to a leaf, since every instance generated by the algorithm is an induced subgraph of the input graph of the original problem. Hence any branching rule applied to d-regular graphs, for some fixed d, can only increase the number of leaves by a multiplicative constant. Hence we may neglect the branching rules needed for d-regular graphs in the time analysis. Therefore we neglect the case when G is 3-regular. If G is not 3-regular and the maximum degree is three, then one of the above rules applies. In the following rules, we may assume that the maximum degree of the input graph is at least four.

12. **Maximum-Degree Rule.** Let v be a vertex of maximum degree of G. In this case we apply **Standard Branching Rule**. With simple calculations we see that the worst case happens when the degree of v is four. The branching vector is $(1, 6, 6, 6, 6)$ and its branching number is 1.5099 in this case.

Theorem 1. *The* MAXIMUM INDUCED MATCHING *problem can be solved in time* $O^*(1.5099^n)$.

Proof. The worst case happens when the Maximum-Degree Rule is applied on graphs of maximum degree four and its branching number is 1.5099. Thus the running time of the branch-and-reduce algorithm is $O^*(1.5099^n)$. □

4 A More Efficient Algorithm

In this section, we give a more efficient algorithm for solving the MAXIMUM INDUCED MATCHING problem by adding a new rule called Sparse Graph Rule as the boundary condition of the algorithm. If the input graph does not satisfy the boundary condition, a rule in the branch-and-reduce algorithm given in Section 3 must be applied. We have the following algorithm.

1. Apply reduction rules: (i) Isolated-Vertex Rule, (ii) Isolated-Edge Rule, (iii) Disconnected Rule, (iv) False-Twin Rule, (v) Degree-One Edge Rule, if one of them is applicable
2. **Sparse Graph Rule.** $|E| \leq 2.044 \times |V|$. In this case, we construct $H = L^2(G)$ and compute $\alpha(H)$ where $\alpha(H)$ is the size of a maximum independent set in H. By Lemma 1, $mim(G) = 2 \times \alpha(H)$.
3. **Degree-One Vertex Rule.** The worst branching vector is $(2, 4)$ and its branching number is 1.2721.
4. **True-Twin Rule.** The worst branching vector is $(2, 4)$ and its branching number is 1.2721.
5. **Domination Rule.** The worst branching vector is $(1, 4)$ and its branching number is 1.3803.
6. **Degree-Two Vertex Rule 1.** The worst branching vector is $(4, 5, 5, 5)$ and its branching number is 1.3413.

7. **Degree-Two Vertex Rule 2.** The worst branching vector is $(3, 5, 5, 5, 5)$ and its branching number is 1.4336.
8. **Maximum-Degree Rule.** The worst case happens when the maximum degree is five. The branching vector is $(1, 7, 7, 7, 7, 7)$ and its branching number is 1.4786 in this case.

Theorem 2. *The* MAXIMUM INDUCED MATCHING *problem can be solved in time* $O^*(1.4786^n)$.

Proof. Due to the Maximum-Degree Rule, at some stage of the algorithm the input graph G satisfies the condition that $|E| \leq 2.044 \times |V|$. In this case we may apply the Sparse Graph Rule to construct $H = (E, E') = L^2(G)$ and compute the size of maximum independent sets in H. Suppose that the number of vertices in G is n. Since $|E| \leq 2.044n$, the number of vertices in H is at most $2.044n$. We apply Robson's algorithm [19] for finding maximum independent sets in H. The running time for graphs applying this rule is $O^*(2^{0.276 \times 2.044n}) = O^*(1.4786^n)$. For other branching rules, the worst case happens when the algorithm applies the Maximum-Degree Rule on graphs of maximum degree five and its branching vector and branching number are $(1, 7, 7, 7, 7, 7)$ and 1.4786. Thus the MAXIMUM INDUCED MATCHING problem can be solved in time $O^*(1.4786^n)$. □

Remark 1. The exact algorithm consists of two parts. When $|E| \leq 2.044n$, we reduce the MAXIMUM INDUCED MATCHING problem to the MAXIMUM INDEPENDENT SET problem. Otherwise, the problem is solved by a branch-and-reduced algorithm. This is an exact algorithm running in exponential space because Robson's algorithm for solving the MAXIMUM INDEPENDENT SET takes exponential space [19]. Notice that the branch-and-reduced part of the algorithm only takes polynomial space.

References

1. Binkele-Raible, D., Brankovic, L., Cygan, M., Fernau, H., Kneis, J., Kratsch, D., Langer, A., Liedloff, M., Pilipczuk, M., Rossmanith, P., Wojtaszczyk, J.O.: Breaking the 2^n–barrier for irredundance: two lines of attack. Journal of Discrete Algorithms 9, 214–230 (2011)
2. Cardoso, D.M., Kaminski, M., Lozin, V.: Maximum k-Regular Induced Subgraphs. Rutcor Research Report (RRR) 3 (2006)
3. Cameron, K.: Induced matching. Discrete Applied Mathematics 24, 97–102 (1989)
4. Cameron, K., Sritharan, R., Tang, Y.: Finding a maximum induced matching in weakly chordal graphs. Discrete Mathematics 266, 133–142 (2003)
5. Duckworth, W., Manlove, D.F., Zito, M.: On the approximability of the maximum induced matching problem. Journal of Discrete Algorithms 3, 79–91 (2005)
6. Erman, R., Kowalik, Ł., Krnc, M., Waleń, T.: Improved induced matching in sparse graphs. Discrete Applied Mathematics 158, 1994–2003 (2010)
7. Fomin, F.V., Kratsch, D.: Exact Exponential Algorithms. Springer (2010)
8. Fricke, G., Laskar, R.: String matching in trees. Congressum Numeratium 89, 239–243 (1992)
9. Gupta, S., Raman, V., Saurabh, S.: Fast Exponential Algorithms for Maximum *r*-Regular Induced Subgraph Problems. In: Arun-Kumar, S., Garg, N. (eds.) FSTTCS 2006. LNCS, vol. 4337, pp. 139–151. Springer, Heidelberg (2006)

10. Hosono, K.: Induced forests in trees and outerplanar graphs. Proceedings of the Faculty of Science of Tokai University 25, 27–29 (1990)
11. Kanj, I., Pelsmajer, M.J., Schaefer, M., Xia, G.: On the induced matching problem. Journal of Computer and System Sciences 77, 1058–1070 (2011)
12. Ko, C.W., Shepherd, F.B.: Bipartite domination and simultaneous matroid cover. SIAM Journal on Discrete Mathematics 16, 327–346 (2003)
13. Kober, D., Rotics, U.: Finding maximum induced matching in subclasses of claw-free and P_5-free graphs, and in graphs with matching and induced matching of equal maximum size. Algorithmica 37, 327–346 (2003)
14. Krishnamurthy, C.M., Sritharan, R.: Maximum induced matching problem on hhd-free graphs. Discrete Applied Mathematics 160, 224–230 (2012)
15. Lozin, V.V.: On maximum induced matchings in bipartite graphs. Information Processing Letters 81, 7–11 (2002)
16. Lozin, V.V., Mosca, R.: Maximum regular induced subgraphs in $2P_3$-free graphs. In: Theoretical Computer Science (2012), doi:10.1016/j.tcs.2012.06.014
17. Moser, H., Sikdar, S.: The parameterized complexity of the induced matching problem in planar graphs. Discrete Applied Mathematics 157, 715–727 (2009)
18. Moser, H., Thilikos, D.M.: Parameterized complexity of finding regular induced subgraphs. Journal of Discrete Algorithms 7, 181–190 (2009)
19. Robson, J.M.: Algorithms for maximum independent sets. Journal of Algorithms 7, 425–440 (1986)
20. Stockmeyer, L.J., Vazirani, V.V.: NP-completeness of some generalizations of the maximum matching problem. Information Processing Letters 15, 14–19 (1982)
21. Zito, M.: Linear time maximum induced matching algorithm for trees. Nordic Journal of Computing 7, 58–63 (2000)

2-Rainbow Domination and Its Practical Variation on Weighted Graphs

Chung-Kung Yen

Department of Information Management,
Shih Hsin University, Taipei, Taiwan
ckyen001@ms7.hinet.net

Abstract. Let $G(V, E)$ be a simple graph with n-vertex-set V and m-edge-set E. Two positive weights, $w_1(v)$ and $w_2(v)$, are assigned to each vertex v. For each $v \in V$, let $N(v) = \{u \mid u \in V$ and $(u, v) \in E\}$ and $N[v] = \{v\} \cup N(v)$. A 2-rainbow function of G is a function f mapping each vertex v to $f(v) = f_2(v) f_1(v)$, $f_2(v)$, $f_1(v) \in \{0, 1\}$. The weight of f is defined as $w(f) = \sum_{v\in V}[f_1(v)w_1(v) + f_2(v)w_2(v)]$. A 2-rainbow function f of G is called a 2-rainbow dominating function if $\Theta_{u\in N(v)} f(u) = 11$, for all vertices v with $f(v) = 00$, where $\Theta_{u\in N(v)} f(u)$ is the result of performing bit-wise Boolean OR on $f(u)$, for all $u \in N(v)$. Our problem is to obtain a 2-rainbow dominating function f of G such that $w(f)$ is minimized. This paper first proves that the problem is NP-hard on unweighted planar graphs. Then, an $O(n)$-time algorithm for the problem on trees is proposed using the dynamic programming strategy. Finally, a practical variation, called the weighted minimum tuple 2-rainbow domination problem, is proposed and the relationship between it and the weighted minimum domination problem is established.

Keywords: 2-rainbow domination functions, tuple 2-rainbow domination functions, planar graphs, trees, NP-hard.

1 Introduction

On algorithmic graph theory, domination and its variations have significant and important applications for various location problems in Operations Research and related fields. Thus, extensive research results have been done in literature [7–8]. The k-rainbow domination is one of important variations introduced in [2–3]. Let $G(V, E)$ be a simple graph which is connected and undirected, where V and E are the vertex-set and the edge-set of G, respectively. For each $v \in V$, let $N(v) = \{u \mid u \in V$ and $(u, v) \in E\}$ and $N[v] = \{v\} \cup N(v)$. Given a positive integer k, a *k-rainbow function* of G is a function h mapping each vertex v of G to $h(v) = h_k(v) \dots h_1(v)$, $h_i(v) \in \{0, 1\}$, $1 \le i \le k$. The *weight* of a k-rainbow function h of G is defined as $w(h) = \sum_{v\in V} \sum_{i=1}^{k} (h_i(v))$. In addition, the k-bit Boolean vectors with exactly k 1s' and k 0s' are denoted by $1^{(k)}$ and $0^{(k)}$, respectively. For any subset S of V, $\Theta_{u\in S} h(u)$ denotes the result of

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 59–68.
DOI: 10.1007/978-3-642-35452-6_8

performing bit-wise Boolean OR on $h(u)$, for all $u \in S$. A k-rainbow function h is called a *k-rainbow dominating function* of G if $\Theta_{u \in N(v)} h(u) = 1^{(k)}$, for all $v \in Z$, where $Z = \{v \mid v \in V$ and $h(v) = 0^{(k)}\}$. The following states the definition of the Minimum k-Rainbow Domination problem.

The Minimum *k*-Rainbow Domination Problem (The M*k*RD Problem). Given a graph $G(V, E)$ and a positive integer k, determine a k-rainbow dominating function h of G such that $w(h)$ is minimized, and $w(h)$ is also called the *k-rainbow domination number* of G in literature.

The M*k*RD problem was introduced in [2] and a linear-time algorithm on trees, for $k = 2$, was also given. For 2-rainbow domination on graphs, the authors in [1] gave some bounds for 2-ranibow domination numbers of 4-regular Harary graphs. The authors in [3] showed that it is NP-hard on bipartite graphs or chordal graphs, and they studied the exact values of 2-rainbow domination numbers of several graph classes, including generalized Petersen graphs. On generalized Petersen graphs, some further studies were made in [11]. For general k, the results in [4] showed that the M*k*RD problem is NP-hard on chordal graphs and bipartite graphs, respectively, and a linear-time algorithm on trees was also designed. The authors in [10] derived better Nordhaus-Gaddum bounds for k-rainbow domatic number of a graph.

This paper generalizes the 2-rainbow dominating functions to weighted graphs. To formally define our problem, let us introduce some terms and concepts. In the rest of this paper, each vertex v of all graphs $G(V, E)$ has two positive weights, $w_1(v)$ and $w_2(v)$. If $w_1(v) = 1$ and $w_2(v) = 1$, for all $v \in V$, then G is called an *unweighted graph*. Given a 2-rainbow function f of G, the *weight* of f is generalized to $w(f) = \sum_{v \in V}(f_1(v)w_1(v) + f_2(v)w_2(v))$. A vertex v of G is called a *00-vertex* if $f(v) = 00$. *01-vertices*, *10-vertices*, and *11-vertices* can be defined similarly. In addition, a vertex v of G is called *00-dominated* if $\Theta_{u \in N[v]} f(u) = 00$. *01-dominated vertices*, *10-dominated vertices*, and *11-dominated vertices* can be defined similarly. A *2-rainbow dominating function f* of G is a 2-rainbow function such that each vertex $v \in Z$ is 11-dominated, where $Z = \{v \mid v \in V$ and $f(v) = 00\}$. Based upon these descriptions, the problem studied in this paper can be defined as follows:

The Weighted Minimum 2-Rainbow Domination Problem (The WM2RD Problem). Let $G(V, E)$ be a graph with n-vertex-set V and m-edge-set E in which each vertex v is associated with two positive weights, $w_1(v)$ and $w_2(v)$. Find a 2-rainbow dominating function f of G such that $w(f)$ is minimized.

This paper will first prove that the problem is NP-hard on unweighted planar graphs. Then, an $O(n)$-time optimal algorithm on trees will be proposed based upon the dynamic programming strategy, which is a powerful strategy to solve optimization problems [9]. Finally, a practical variation, called the weighted minimum tuple 2-rainbow domination problem, will be proposed and the relationship between it and the weighted minimum domination problem will be established.

2 The WM2RD Problem on Unweighted Planar Graphs

A graph is a *planar graph* if this graph can be drawn into the plane without edge-crossings, except crossings occurring on end vertices of edges [5]. This section will show that the WM2RD problem is NP-hard on unweighted planar graphs. The task will be achieved by showing that the following NP-complete problem can be reduced to the MW2RD decision problem on unweighted planar graphs in polynomial time.

The Planar Three Satisfiability Problem (The P3SAT Problem). Suppose that a CNF Boolean formula consisting of $C = \{ c_1, \ldots, c_p \}$ Boolean clauses over the variable-set $X = \{ x_1, \ldots, x_t \}$ is given such that each clause consists of at most 3 literals, and the graph $G(V, E)$ with $V = C \cup X$ and $E = \{(c_i, x_j) \mid x_j$ or x'_j is a literal of $c_i \}$ is a planar graph. The output will be 'Yes' iff there exists an assignment S of X satisfying all clauses in C.

Theorem 2.1. [6] *The P3SAT problem is NP-complete.*

Theorem 2.2. *The WM2RD problem is NP-hard on unweighted planar graphs.*

Proof. Suppose that we are given $C = \{ c_1, \ldots, c_p \}$, $X = \{ x_1, \ldots, x_t \}$, and $G(V, E)$ of the P3SAT problem, where $G(V, E)$ is the planar graph defined in the definition of the P3SAT problem. The corresponding unweighted graph $G^*(V^*, E^*)$ of the WM2RD problem is constructed as, $V^* = C \cup X \cup X' \cup A \cup B \cup D$ and $E^* = \{(c_i, x_j) \mid x_j$ is a literal of $c_i\} \cup \{(c_i, x'_j) \mid x'_j$ is a literal of $c_i\} \cup \{(x_j, x'_j), (x_j, a_j), (a_j, x'_j), (x_j, b_j), (b_j, x'_j), (x_j, d_j), (d_j, x'_j) \mid 1 \le j \le t\}$, where $X' = \{ x'_1, \ldots, x'_t \}$, $A = \{ a_1, \ldots, a_t \}$, $B = \{ b_1, \ldots, b_t \}$, and $D = \{ d_1, \ldots, d_t \}$. Clearly, G^* can be constructed in polynomial-time with respect to p and t. The planarity of G^* can be easily verified because G is planar.

The following will claim that there exists an assignment S of X satisfying all clauses in C iff there exists a 2-rainbow domination function f of G^* such that $w(f) = 2t$. Suppose that there exists an assignment S of X satisfying all clauses in C. Let f be the 2-rainbow function of G^* obtained by the following rules.

1. For each x_j = TRUE in S, $f(x_j) = 11$ and $f(v) = 00$, for all $v \in (V^*_j - \{ x_j \})$.
2. For each x_j = FALSE in S, $f(x'_j) = 11$ and $f(v) = 00$, for all $v \in (V^*_j - \{ x'_j \})$.
3. $f(c_j) = 00$, for all clause vertices c_j.

It is clear that $w(f) = 2t$ and f is a 2-rainbow dominating function of G^*.

Next, assume that there exists a 2-rainbow dominating function f of G^* with $w(f) = 2t$. For any subset Q of V, let $w(f, Q) = \sum_{u \in Q}(f_1(u)w_1(u) + f_2(u)w_2(u))$. Examining all possible combinations of $f_2(x_j)f_1(x_j)$ and $f_2(x'_j)f_1(x'_j)$ for each j, we can ascertain that $w(f, V^*_j) \ge 2$. The assumption that $w(f) = 2t$ implies that

$w(f, V_j^*) = 2$, $1 \le j \le t$ This further implies that $f_2\ (c_j)\, f_1(c_j) = 00$, $1 \le j \le p$. Let S be the assignment in which x_j = TURE if $f_1(x_j) = 1$ and x_j = FALSE if $f_1(x_j) = 0$, $1 \le j \le t$. It is clear that S must satisfy all clauses in C. ■

3 The WM2RD Problem on Trees

Give a tree T, take any vertex r as the *root* and denote T as $T(r)$. For each vertex v of $T(r)$, par(v) means the *parent* of v and $T(v)$ means the subtree rooted at v. For each vertex v of $T(r)$, Children(v) = $\{x_1, \ldots, x_p\}$, $p \ge 0$, represents the set of the *child vertices* of u. Meanwhile, for each j, $1 \le j \le p$, the notation $BR(u, x_j \to x_p)$ is the subtree of $T(r)$ formed by $T(x_j) \cup \ldots \cup T(x_p) \cup \{(u, x_j), \ldots, (u, x_p)\}$. In this definition, $T(u) = BR(u, x_1 \to x_p)$, for all non-leaf vertices u of $T(r)$ and the root of each $BR(u, x_j \to x_p)$ is also the vertex u, $2 \le j \le p$.

Consider each non-leaf vertex u of $T(r)$. For all $c_1, c_2, d_1, d_2 \in \{0, 1\}$, define $\delta_{c_2c_1}^{d_2d_1}(T(u))$ = min$\{w(f) \mid f$ is a 2-rainbow function of $T(u)$ such that: (1) u is a $c_2\,c_1$-vertex, (2) u is $d_2\,d_1$-dominated, and (3) all other vertices x in $T(u) - \{u\}$ are 11-dominated if $f(x) = 00\}$. Note that the values of $\delta_{c_2c_1}^{d_2d_1}$ $(T(u))$ may not exist under this definition and it is reasonable to set those values to be ∞. For example, $\delta_{11}^{00}(T(u))$ means that u is assumed to be 11-vertex. This also means that u is 11-dominated by the definition of our problem, i.e., it is impossible that u is 00-dominated in this situation. Thus, we can set $\delta_{11}^{00}(T(u)) = \infty$. Similarly, we have $\delta_{11}^{01}(T(u)) = \delta_{11}^{10}(T(u)) = \delta_{01}^{10}(T(u)) = \infty$, etc.

Now, our objective can be viewed as to compute $\delta(T(r))$ = min$\{w(f) \mid f$ is a 2-rainbow domination function of $T(r)\}$. Consider the root r. r can be a 00-vertex, 01-vertex, 10-vertex, or 11-vertex. By the definition of 2-rainbow domination functions, we can have the following observation.

Observation 1. If r is a 00-vertex or an 11-vertex, then r must be 11-dominated. If r is a 01-vertex, then r can be 01-dominated or 11-dominated. If r is a 10-vertex, then r can be 10-dominated or 11-dominated.

Observation 1 derives that $\delta(T(r))$ = min$\{\delta_{00}^{11}(T(r))$, $\delta_{01}^{01}(T(r))$, $\delta_{01}^{11}(T(r))$, $\delta_{10}^{10}(T(r))$, $\delta_{10}^{11}(T(r))$, $\delta_{11}^{11}(T(r))\}$. The following will illustrate how to compute the sixteen values $\delta_{c_2c_1}^{d_2d_1}(T(v))$, for all $v \in T(r)$, in $O(n)$ time. For each non-leaf vertex u of $T(r)$, u can be a 00-vertex, 01-vertex, 10-vertex, or 11-vertex. The following will consider the cases that u is a 00-vertex and u is a 01-vertex, respectively. The other two cases can be resolved by similar way. Note that Children(u) = $\{x_1, \ldots, x_p\}$ is the set of children of u. It is trivial to handle the situations for $p = 1, 2$. The following assumes that $p \ge 3$.

Case 1. *u* is a 00-vertex

Consider the vertex x_1. First, we show how to compute $\delta_{00}^{00}(T(u))$. x_1 must be 11-dominated and must be a 00-vertex when we consider the subtree $T(x_1)$ recursively. Thus, the following equation can be obtained.

$$\delta_{00}^{00}(T(u)) = \delta_{00}^{11}(T(x_1)) + \delta_{00}^{00}(BR(u,\ x_2 \rightarrow x_p))$$

Second, we show how to compute $\delta_{00}^{01}(T(u))$. When we consider $T(x_1)$ recursively, x_1 can only be a 00-vertex or a 01-vertex. If x_1 is a 00-vertex in $T(x_1)$, then the vertex *u* must be 01-dominated when we consider $BR(u,\ x_2 \rightarrow x_p)$ recursively. If x_1 is a 01-vertex in $T(x_1)$, *u* can be 00-dominated or 01-dominated in $BR(u,\ x_2 \rightarrow x_p)$ and x_1 can be 01-dominated or 11-dominated in $T(x_1)$.

$$\begin{aligned}\delta_{00}^{01}(T(u)) = \min\{ & \delta_{00}^{11}(T(x_1)) + \delta_{00}^{01}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{01}(T(x_1)) + \delta_{00}^{00}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{01}(T(x_1)) + \delta_{00}^{01}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{11}(T(x_1)) + \delta_{00}^{00}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{11}(T(x_1)) + \delta_{00}^{01}(BR(u,\ x_2 \rightarrow x_p))\}\end{aligned}$$

By the similar reasoning, the following equations can be established.

$$\begin{aligned}\delta_{00}^{10}(T(u)) = \min\{ & \delta_{00}^{11}(T(x_1)) + \delta_{00}^{10}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{10}(T(x_1)) + \delta_{00}^{00}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{10}(T(x_1)) + \delta_{00}^{10}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{11}(T(x_1)) + \delta_{00}^{00}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{11}(T(x_1)) + \delta_{00}^{10}(BR(u,\ x_2 \rightarrow x_p))\}\end{aligned}$$

$$\begin{aligned}\delta_{00}^{11}(T(u)) = \min\{ & \delta_{00}^{11}(T(x_1)) + \delta_{00}^{11}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{01}(T(\mathrm{v})) + \delta_{00}^{10}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{01}(T(x_1)) + \delta_{00}^{11}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{11}(T(x_1)) + \delta_{00}^{10}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{01}^{11}(T(x_1)) + \delta_{00}^{11}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{10}(T(x_1)) + \delta_{00}^{01}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{10}(T(x_1)) + \delta_{00}^{11}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{11}(T(x_1)) + \delta_{00}^{01}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{10}^{11}(T(x_1)) + \delta_{00}^{11}(BR(u,\ x_2 \rightarrow x_p)),\\ & \delta_{11}^{11}(T(x_1)) + \delta_{00}^{00}(BR(u,\ x_2 \rightarrow x_p)),\end{aligned}$$

$$\begin{aligned}
&\delta_{11}^{11}(T(x_1)) + \delta_{00}^{01}(BR(u,\ x_2 \rightarrow x_p)),\\
&\delta_{11}^{11}(T(x_1)) + \delta_{00}^{10}(BR(u,\ x_2 \rightarrow x_p)),\\
&\delta_{11}^{11}(T(x_1)) + \delta_{00}^{11}(BR(u,\ x_2 \rightarrow x_p))\}
\end{aligned}$$

$$\begin{aligned}
&\delta_{00}^{00}(BR(u,\ x_j \rightarrow x_p)),\ 2 \le j \le (p-1)\\
&= \delta_{00}^{11}(T(x_j)) + \delta_{00}^{00}(BR(u,\ x_{j+1} \rightarrow x_p))
\end{aligned}$$

$$\begin{aligned}
&\delta_{00}^{01}(BR(u,\ x_j \rightarrow x_p)),\ 2 \le j \le (p-1)\\
&= \min\{\ \delta_{00}^{11}(T(x_j)) + \delta_{00}^{01}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{01}(T(x_j)) + \delta_{00}^{00}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{01}(T(x_j)) + \delta_{00}^{01}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{11}(T(x_j)) + \delta_{00}^{00}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{11}(T(x_j)) + \delta_{00}^{01}(BR(u,\ x_{j+1} \rightarrow x_p))\}
\end{aligned}$$

$$\begin{aligned}
&\delta_{00}^{10}(BR(u,\ x_j \rightarrow x_p)),\ 2 \le j \le (p-1)\\
&= \min\{\ \delta_{00}^{11}(T(x_j)) + \delta_{00}^{10}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{10}(T(x_j)) + \delta_{00}^{00}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{10}(T(x_j)) + \delta_{00}^{10}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{11}(T(x_j)) + \delta_{00}^{00}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{11}(T(x_j)) + \delta_{00}^{10}(BR(u,\ x_{j+1} \rightarrow x_p))\}
\end{aligned}$$

$$\begin{aligned}
&\delta_{00}^{11}(BR(u,\ x_j \rightarrow x_p)),\ 2 \le j \le (p-1)\\
&= \min\{\ \delta_{00}^{11}(T(x_j)) + \delta_{00}^{11}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{01}(T(x_j)) + \delta_{00}^{10}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{01}(T(x_j)) + \delta_{00}^{11}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{11}(T(x_j)) + \delta_{00}^{10}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{01}^{11}(T(x_j)) + \delta_{00}^{11}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{10}(T(x_j)) + \delta_{00}^{01}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{10}(T(x_j)) + \delta_{00}^{11}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{11}(T(x_j)) + \delta_{00}^{01}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{10}^{11}(T(x_j)) + \delta_{00}^{11}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{11}^{11}(T(x_j)) + \delta_{00}^{00}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{11}^{11}(T(x_j)) + \delta_{00}^{01}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{11}^{11}(T(x_j)) + \delta_{00}^{10}(BR(u,\ x_{j+1} \rightarrow x_p)),\\
&\qquad \delta_{11}^{11}(T(x_j)) + \delta_{00}^{11}(BR(u,\ x_{j+1} \rightarrow x_p))\}
\end{aligned}$$

$$\delta_{00}^{00}(BR(u,\ x_j\ \rightarrow\ x_p)) = \ \delta_{00}^{11}(T(x_p)), j = p$$
$$\delta_{00}^{01}(BR(u,\ x_j\ \rightarrow\ x_p)) = \min\{\ \delta_{01}^{01}(T(x_p)),\ \ \delta_{01}^{11}(T(x_p))\}, j = p$$
$$\delta_{00}^{10}(BR(u,\ x_j\ \rightarrow\ x_p)) = \min\{\ \delta_{10}^{10}(T(x_p)),\ \ \delta_{10}^{11}(T(x_p))\}, j = p$$
$$\delta_{00}^{11}(BR(u,\ x_j\ \rightarrow\ x_p)) = \ \delta_{11}^{11}(T(x_p)), j = p$$

Case 2. *u* is a 01-vertex

Employing the similar technique of Case 1, the following equations can be correctly established.

$$\delta_{01}^{00}(T(u)) = \infty$$
$$\begin{aligned}\delta_{01}^{01}(T(u)) = \ w_1(u) + \min\{\ &\delta_{00}^{10}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{00}^{11}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{01}^{01}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{01}^{11}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p))\}\end{aligned}$$
$$\delta_{01}^{10}(T(u)) = \infty$$
$$\begin{aligned}\delta_{01}^{11}(T(u)) = \ w_1(u) + \min\{\ &\delta_{00}^{10}(T(x_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{00}^{11}(T(X_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{01}^{01}(T(x_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{01}^{11}(T(x_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{10}^{10}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{10}^{10}(T(x_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{10}^{11}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{10}^{11}(T(x_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{11}^{11}(T(x_1)) + \ \delta_{01}^{01}(BR(u,\ x_2\ \rightarrow\ x_p)),\\ &\delta_{11}^{11}(T(x_1)) + \ \delta_{01}^{11}(BR(u,\ x_2\ \rightarrow\ x_p))\}\end{aligned}$$
$$\delta_{01}^{00}(BR(u,\ x_j\ \rightarrow\ x_p)) = \infty, 2 \le j \le (p-1)$$
$$\delta_{01}^{01}(BR(u,\ x_j\ \rightarrow\ x_p)), 2 \le j \le (p-1)$$
$$\begin{aligned}= \min\{\ &\delta_{00}^{10}(T(x_j)) + \ \delta_{01}^{01}(BR(u,\ x_{j+1}\ \rightarrow\ x_p)),\\ &\delta_{00}^{11}(T(x_j)) + \ \delta_{01}^{01}(BR(u,\ x_{j+1}\ \rightarrow\ x_p)),\\ &\delta_{01}^{01}(T(x_j)) + \ \delta_{01}^{01}(BR(u,\ x_{j+1}\ \rightarrow\ x_p)),\\ &\delta_{00}^{11}(T(x_j)) + \ \delta_{01}^{01}(BR(u,\ x_{j+1}\ \rightarrow\ x_p))\}\end{aligned}$$
$$\delta_{01}^{10}(BR(u,\ x_j\ \rightarrow\ x_p)) = \infty, 2 \le j \le (p-1)$$
$$\delta_{01}^{11}(BR(u,\ x_j\ \rightarrow\ x_p)), 2 \le j \le (p-1)$$
$$= \min\{\ \delta_{00}^{10}(T(x_j)) + \ \delta_{01}^{11}(BR(u,\ x_{j+1}\ \rightarrow\ x_p)),$$

$$\begin{aligned}
&\delta_{00}^{11}(T(x_j)) + \delta_{01}^{11}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{01}^{01}(T(x_j)) + \delta_{01}^{11}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{01}^{11}(T(x_j)) + \delta_{01}^{11}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{10}^{10}(T(x_j)) + \delta_{01}^{01}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{10}^{10}(T(x_j)) + \delta_{01}^{11}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{10}^{11}(T(x_j)) + \delta_{01}^{01}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{10}^{11}(T(x_j)) + \delta_{01}^{11}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{11}^{11}(T(x_j)) + \delta_{01}^{01}(BR(u, x_{j+1} \to x_p)),\\
&\delta_{11}^{11}(T(x_j)) + \delta_{01}^{11}(BR(u, x_{j+1} \to x_p))\}
\end{aligned}$$

$$\delta_{01}^{00}(BR(u, x_j \to x_p)) = \infty, j = p$$

$$\delta_{01}^{01}(BR(u, x_j \to x_p)), j = p$$

$$= \min\{\delta_{00}^{10}(T(x_p)), \delta_{00}^{11}(T(x_p)), \delta_{01}^{01}(T(x_p)), \delta_{01}^{11}(T(x_p))\}$$

$$\delta_{01}^{10}(BR(u, x_j \to x_p)) = \infty, j = p$$

$$\delta_{01}^{11}(BR(u, x_j \to x_p)) = \min\{\delta_{10}^{10}(T(x_p)), \delta_{10}^{11}(T(x_p)), \delta_{01}^{11}(T(x_p))\}, j = p$$

The boundary conditions are the following equations for all leaves y of $T(r)$.

$$\delta_{00}^{00}(T(y)) = 0,\ \delta_{00}^{01}(T(y)) = \infty,\ \delta_{00}^{10}(T(y)) = \infty, \text{ and } \delta_{00}^{11}(T(y)) = \infty$$

$$\delta_{01}^{00}(T(y)) = \infty,\ \delta_{01}^{01}(T(y)) = w_1(y),\ \delta_{01}^{10}(T(y)) = \infty, \text{ and } \delta_{01}^{11}(T(y)) = \infty$$

$$\delta_{10}^{00}(T(y)) = \infty,\ \delta_{10}^{01}(T(y)) = \infty,\ \delta_{10}^{10}(T(y)) = w_2(y), \text{ and } \delta_{10}^{11}(T(y)) = \infty$$

$$\delta_{11}^{00}(T(y)) = \infty,\ \delta_{11}^{01}(T(y)) = \infty,\ \delta_{11}^{10}(T(y)) = \infty, \text{ and } \delta_{11}^{11}(T(y)) = w_1(y) + w_2(y)$$

Theorem 3.1. *The WM2RD problem is O(n)-time solvable on trees.*

Proof. All reasoning so far can guarantee the correctness of above all equations, i.e., $\delta(T(r))$ can be computed correctly. For each non-leaf vertex u of $T(r)$ with children $x_1, \ldots, x_p$, Verifying that $\delta_{c_2c_1}^{d_2d_1}(T(u))$ and $\delta_{c_2c_1}^{d_2d_1}(BR(u, x_j \to x_p))$, $2 \le j \le p$, for all $c_1, c_2, d_1, d_2 \in \{0, 1\}$, can be computed in $O(p)$ time totally is a simple task. Thus, the total time complexity for computing $\delta(T(r))$ is $O((\sum_{\text{non-leaf vertex } u}|\text{Children}(u)|) + \gamma)$ $= O(n)$, where γ is the number of leaf vertices of $T(r)$. Finally, an optimal solution f of the WM2RD problem on $T(r)$ can be easily obtained via tracing the procedure for computing the value $\delta(T(r))$ by bottom-up manner. ■

4 A Practical Variation

Employing the notations defined in Section 1, a *tuple 2-rainbow dominating function f* of a graph G is a 2-rainbow function such that every vertex v of G must be 11-dominated. The new variation based upon this concept can be defined as follows:

The Weighted Minimum Tuple 2-Rainbow Domination Problem (The WMT2RD Problem). Let $G(V, E)$ be a graph with n-vertex-set V and m-edge-set E in which each vertex v is associated with two positive weights, $w_1(v)$ and $w_2(v)$. Identify a tuple 2-rainbow dominating function f of G such that $w(f)$ is minimized.

The WMT2RD problem was original proposed and studied in [12]. This section will show the relationship between the WMT2RD problem and the WMID problem, where the *WMID problem* is the problem of finding a minimum weight dominating set Q of a graph G in which each vertex v is associated a positive weight $w(v)$.

Theorem 4.1. *The WMT2RD problem is linearly equivalent to the WMID problem on any graph G.*

Proof. For a vertex-weighted graph $G(V, E)$, the corresponding instance $G^*(V^*, E^*)$ in which each vertex v is associated two positive weights $w_1^*(v)$ and $w_2^*(v)$ of the WMT2RD problem can be constructed as $V^* = V$, $E^* = E$, $w_1^*(v) = w(v)$, and $w_2^*(v) = 1$, for all $v \in V^*$, where $w(v)$ is the weight of v in G.

Suppose that f^* is an optimal solution of the WMT2RD problem on G^*. For $j = 1, 2$, let $D_j^* = \{v \mid v \in V^*$ and $f_j^*(v) = 1\}$, and $G_j(V_j, E_j)$ is the graph in which $V_j = V^*$ and $E_j = E^*$, and the weight of each vertex v is $w_j^*(v)$. By the definition of tuple 2-rainbow dominating functions, it is easy to see that $D_j = D_j^*$ forms a minimum weight dominating set of G_j, $j = 1, 2$. This implies that the WMID problem can be linearly reduced to the WMT2RD problem.

For another direction, let $G^*(V^*, E^*)$ be a graph in which each vertex v has two positive weights $w_1^*(v)$ and $w_2^*(v)$. By the notations $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ as above, if D_1 and D_2 are minimum weight dominating sets of G_1 and G_2, respectively, then it is easy to verify that the 2-rainbow function f obtained as follows must be a tuple 2-rainbow dominating function of G^* such that $w(f)$ is minimized. This implies that the WMT2RD problem can be linearly reduced to the WMID problem.

$f_1(v) = 1$, if $v \in D_1$, and $f_1(v) = 0$, otherwise.

$f_2(v) = 1$, if $v \in D_2$, and $f_2(v) = 0$, otherwise. ■

5 Concluding Remarks

This paper first generalized the 2-ranibow domination to weighted graphs. The problem is abbreviated as the WM2RD problem. Firstly, this paper showed that the problem is NP-hard on unweighted planar graphs. Then, an $O(n)$-time algorithm for the problem on trees was proposed using the dynamic programming strategy. In addition, a new variation, called the weighted minimum tuple 2-rainbow domination problem (the WMT2RD problem), was also discussed. We showed that the WMT2RD problem is linearly equivalent to the WMID problem on any graph G.

In the future, it deserves us to generalize the WM2RD/WMT2RD problems to WM*k*RD/WMT*k*RD problem, for $k \geq 3$. Extending the results of this paper to graph classes such as cactus graphs, block graphs, etc., is also a meaningful research issue.

References

1. Ali, M., Rahim, M.T., Zeb, M., Ali, G.: On 2-Rainbow Domination of Some Families of Graphs. International Journal of Mathematics and Soft Computing 1, 47–53 (2011)
2. Brešar, B., Henning, M.A., Rall, D.F.: Rainbow Domination in Graphs. Taiwanese Journal of Mathematics 12, 213–225 (2008)
3. Brešar, B., Šumenjak, T.K.: On the 2-Ranbow Domination in Graphs. Discrete Applied Mathematics 155, 2394–2400 (2007)
4. Chang, G.J., Wu, J., Zhu, X.: Rainbow Domination on Trees. Discrete Applied Mathematics 158, 8–12 (2010)
5. Chartrand, G., Zhang, P.: Introduction to Graph Theory. McGraw-Hill International Edition (2005)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Bell Laboratories, Freeman & Co., Murray Hill (1979)
7. Haynes, T.W., Hedetniemi, S.T., Slater, P.J.: Fundamentals of Domination in Graphs. Marcel Dekker, Inc., New York (1998)
8. Haynes, T.W., Hedetniemi, S.T., Slater, P.J.: Domination in Graphs. Advanced Topics. Marcel Dekker, Inc., New York (1998)
9. Lee, R.C.T., Tseng, S.S., Chang, R.C., Tsai, Y.T.: Introduction to the Design and Analysis of Algorithms. McGraw Hill Education, Asia (2005)
10. Meierling, D., Sheikholeslami, S.M., Volkmann, L.: Nordhaus-Gaddum Bounds on the *k*-Rainbow Domatic Number of a Graph. Applied Mathematics Letters 24, 1758–1761 (2011)
11. Tong, C., Lin, X., Yang, Y., Luo, M.: 2-Rainbow Domination of Generalized Petersen Graphs $P(n, 2)$. Discrete Applied Mathematics 157, 1932–1937 (2009)
12. Yen, W.C.-K., Liu, J.-J., Shih, C.-C.: The Weighted Minimum Tuple 2-Rainbow Domination on Graphs. World Academy of Science, Engineering and Technology 62, 1183–1186 (2012)

The Longest Path Problem on Distance-Hereditary Graphs

Yi-Lu Guo[1], Chin-Wen Ho[1], and Ming-Tat Ko[2]

[1] Department of Computer Science and Information Engineering,
National Central University, Chung-Li, Taiwan
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan

Abstract. The longest path problem is to find a path of maximum length in a graph. As a generalization of Hamiltonian path problem, it is NP-complete on general graphs. A graph is called distance-hereditary if the distances of each pair of vertices in every connected induced subgraph containing them are the same. In this paper, we present an $O(n^4)$ time algorithm to solve the longest path problem on a distance-hereditary graph of n vertices .

Keywords: Longest path problem, Distance-hereditary graphs, Polynomial-time algorithm.

1 Introduction

A graph is *distance-hereditary* [3, 23] if the distance of any two vertices are the same in every connected induced subgraph containing them, where the *distanc*e between two vertices is the length of shortest path connecting them. Distance-hereditary graphs form a subclass of perfect graphs [13, 22, 23] that are graphs in which the maximum clique size equals the chromatic number for every induced subgraph of G [4, 20]. Some graph classes, such as trees, cographs, block graphs, Ptolemaic graphs and bipartite distance-hereditary graphs belong to the class of distance-hereditary graphs [7, 11, 12, 24]. Properties of distance-hereditary graphs have been studied during the past three decades [3, 6, 9, 13, 17, 22, 23, 25-28, 30, 35, 38, 42- 45], which have resulted in sequential or parallel algorithms for solving several graph-theoretical problems on this special graph class.

The *longest path problem* is the problem of finding a path of maximum length in a graph. The problem generalizes the Hamiltonian path problem and thus it is NP-complete on general graphs. The Hamiltonian path problem is known to be NP-complete in general graphs [18, 19], and remains NP-complete even when restricted to some small classes of graphs such as split graphs [20], chordal bipartite graphs, split strongly chordal graphs [36], directed path graphs [37], circle graphs [14], planar graphs [19], and grid graphs [32]. On the other hand, there are several polynomial time algorithms of Hamiltonian path on several classes of graphs, such as proper interval graphs [5], interval graphs [1, 10, 15], circular-arc graphs [15], biconvex graphs [2], and cocomparability graphs [16].

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 69–77.
DOI: 10.1007/978-3-642-35452-6_9

There are few polynomial time solutions for the longest path problem. A linear time algorithm for finding a longest path in a tree was proposed by Dijkstra early in 1960, a formal proof is given by Bulterman *et al.* [8]. Through generalization of Dijkstra's algorithm for trees, Uehara and Uno [40] solved the longest path problem for weighted trees and block graphs in linear time and space, and for cactus graphs in $O(n^2)$ time and space, where n and m are number of vertices and edges of the input graph, respectively. Recently, polynomial algorithms have been proposed that solve the longest path problem on bipartite permutation graphs in $O(n)$ time and space [41], and on Ptolemaic graphs in $O(n^5)$ time and $O(n^2)$ space [39]. Furthermore, Uehara and Uno in [40] introduced a subclass of interval graphs, called interval biconvex graphs, which is a superclass of proper interval graphs and threshold graphs, and solved the longest path problem in $O(n^3(m + n \log n))$ time. As a corollary, they showed that a longest path of threshold graphs can be found in $O(n + m)$ time and space. More recently, Ioannidou *et al.* showed that the longest path problem has a polynomial solution on interval graphs in $O(n^4)$ time [33], and on cocomparability graphs in $O(n^7)$ time and space [34].

In this paper, we present an $O(n^4)$ time solution to solve the longest path problem on the class of distance-hereditary graphs. As to our best knowledge, it is the first polynomial-time algorithm. The rest of this paper is organized as follows. In Section 2, we review some properties of distance-hereditary graphs and give some basic definitions. In Section 3, we present a polynomial time algorithm to solve the longest path problem on distance-hereditary graphs.

2 Preliminaries

In this paper we consider finite, simple, and undirected graphs $G = (V, E)$, where V and E are the vertex and edge sets of G and let $n = |V|$ and $m = |E|$.

Bandelt and Mulder [3] showed that the house, holes, domino, and gem are neither induced subgraphs nor isometric subgraphs of a distance-hereditary graph. They also showed that every distance-hereditary graph G has a one-vertex-extension order which can construct G from a single vertex K_1 by following two operations: adding pendant vertices and splitting vertices [3]. Chang *et al.* [9] generalized the concept of one-vertex-extension ordering and define a new recursive construction of distance-hereditary graphs. In the recursive characterization, each distance-hereditary graph G is associated with a subset of vertices S called *twin set*, and a distance-hereditary graph can be constructed from two distance-hereditary graphs by three operations, which is described in following Theorem 1.

Theorem 1 (Chang et al. [9]**).** A graph is distance-hereditary if and only if it has the following recursive construction:

A graph consisting of a single vertex v is a distance-hereditary graph with twin set $\{v\}$ itself. Let G_1 and G_2 be distance-hereditary graphs with twin set S_1 and S_2, respectively. Then:

1. The graph G obtained from the union of G_1 and G_2 is a distance-hereditary graph with *twin set* $S_1 \cup S_2$. We call G is formed by *false twin* operation and denote $G = G_1 \odot G_2$.
2. The graph obtained from G_1 and G_2 by connecting every vertex of S_1 to all vertices of S_2 is a distance-hereditary graph with *twin set* $S_1 \cup S_2$. We called G is formed by *true twin* operation and denote by $G_1 \otimes G_2$.
3. The graph obtained from G_1 and G_2 by connecting every vertex of S_1 to all vertices of S_2 is a distance-hereditary graph with *twin set* S_1. We call G is formed by *attachment* operation and denote by $G_1 \oplus G_2$.

In the following, we use (G, S) to denote a distance-hereditary graph G associated with twin set S and also use G if its twin set is clear. Using the above definition, a distance-hereditary graph G can be represented by a binary tree form, called a *decomposition tree* T_G, which is defined as follows.

Definition 1 (Chang *et al.* [9]). The tree consisting of a single vertex v is a decomposition tree of a primitive distance-hereditary graph $G = (\{v\}, \{v\})$.

Let T_{G_1} and T_{G_2} be decomposition trees of distance-hereditary graphs (G_1, S_1) and (G_2, S_2) with root r_1 and r_2, respectively.

1. A tree consisting of T_{G_1}, T_{G_2} and a new vertex r as the root, marked by $\odot$, with r_1 and r_2 being the two children of r is a decomposition tree of $(G, S) = (G_1 \odot G_2, S_1 \cup S_2)$.
2. A tree consisting of T_{G_1}, T_{G_2} and a new vertex r as the root, marked by $\otimes$, with r_1 and r_2 being the two children of r is a decomposition tree of $(G, S) = (G_1 \otimes G_2, S_1 \cup S_2)$.
3. A tree consisting of T_{G_1}, T_{G_2} and a new vertex r as the root, marked by $\oplus$, with r_1 and r_2 being the left child and right child r respectively is a decomposition tree of $(G, S)= (G_1 \oplus G_2, S_1)$.

Lemma 1. (Chang *et al.* [9]). A decomposition tree of a distance-hereditary graph can be constructed in $O(n + m)$ time.

Let (G, S) be a distance-hereditary graph associated with $S \subseteq V$ as a twin set. A path of G is called *crucial* if it is a single vertex in S or its two end-vertices are in S. A path of G is called *semi-crucial* if it is not crucial path and exactly one of its end-vertices is in S. A set $\mathcal{P}$ of i disjoint crucial or semi-crucial paths in which there are exactly j semi-crucial paths is called a crucial path set of type (i, j), or an *(i, j)-crucial path set*. The number of vertices covered by $\mathcal{P}$ is denoted by $c(\mathcal{P})$. Note that there exists an (i, j)-crucial path set only for $i \le s$ and $j \le i$. In the following, by the nature of the problem, only the cases $j = 0, 1, 2$ are considered.

Let (G, S) be formed by two distance-hereditary graphs (G_1, S_1) and (G_2, S_2). An edge (a, b), $a \in S_1$ and $b \in S_2$, is called a *crossing edge*. A *crossing path* is a path having crossing edges. Consider a path P in (G, S), either P is a crossing path, or P is in G_1 or in G_2. In the former case, by removing the crossing edges, P is decomposed into several crucial paths and at most two semi-crucial paths in G_1 and G_2. For a crucial path set $\mathcal{P}$ of G, $\mathcal{P}$ can be decomposed into a crucial path set $\mathcal{P}_1$ of G_1 and a crucial path set $\mathcal{P}_2$ of G_2 by removing the crossing edges. $\mathcal{P}_1$ ($\mathcal{P}_2$ respectively) is called the *projection* of $\mathcal{P}$ on G_1 (G_2 respectively). Note that c($\mathcal{P}$) = c($\mathcal{P}_1$) + c($\mathcal{P}_2$). In the false twin case, $G = G_1 \odot G_2$, $\mathcal{P}_1$ ($\mathcal{P}_2$ respectively) is the set of paths of $\mathcal{P}$in G_1 (G_2 respectively). Therefore, if $\mathcal{P}$ is an (x, j)-crucial path set, $\mathcal{P}_1$ is an (x_1, j_1)-crucial path set and $\mathcal{P}_2$ is an (x_2, j_2)-crucial path set, then $x = x_1 + x_2$, and $j = j_1 + j_2$. On the other hand, an (x_1, j_1)-crucial path set of G_1 and an (x_2, j_2)-crucial path set of G_2 will then form an $(x_1 + x_2, j_1 + j_2)$-crucial path set of G. Following the similar argument of Lemma 3 and Lemma 4 in [29], we have the following two lemmas for the cases of true twin and attachment operations.

Lemma 2. Let $(G, S) = (G_1 \otimes G_2, S_1 \cup S_2)$. Given $\mathcal{P}_1$, an (x_1, j_1)-crucial path set of G_1, and $\mathcal{P}_2$, an (x_2, j_2)-crucial path set of G_2. There exists a (k, j)-crucial path set $\mathcal{P}$ of G such that $\mathcal{P}_1$ is the projection of $\mathcal{P}$ on G_1 and $\mathcal{P}_2$ is the projection of $\mathcal{P}$ on G_2 if and only if $j_1 + j_2 = j$, $x_1 - x_2 \leq k - j_2$, $x_2 - x_1 \leq k - j_1$, $k \leq x_1 + x_2$, $j_1 \leq x_1 \leq s_1$, $j_2 \leq x_2 \leq s_2$.

Proof. (only if) It is obvious that $j_1 + j_2 = j$, $j_1 \leq x_1 \leq s_1$, $j_2 \leq x_2 \leq s_2$. For a crucial path set $\mathcal{P}$, denote the subset of the paths in G_1 by $\mathcal{Q}_1$, the subset of the paths in G_2 by $\mathcal{Q}_2$ and the subset of crossing paths of both end vertices in G_1 (G_2 respectively) by $\mathcal{R}_1$ ($\mathcal{R}_2$ respectively) and the subset of crossing paths of one end vertex in G_1 and the other in G_2 by $\mathcal{R}$. The types of $\mathcal{Q}_1$, $\mathcal{Q}_2$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}$ are (y_1, i_1), (y_2, i_2), (t_1, r_1), (t_2, r_2), (t, r), respectively. Then, $j_1 \leq i_1 + r_1 + r$ and $j_2 \leq i_2 + r_2 + r$. Let P be a path. After removing all the crossing edges in P, it is decomposed into paths in G_1 and in G_2. Let the number of paths in G_1 and G_2 be p_1 and p_2. If P is in $\mathcal{R}_1$, there are even number of crossing edges in P, and $p_1 - p_2 = 1$. Similarly, if P is in $\mathcal{R}_2$, $p_1 - p_2 = -1$ and if P is in $\mathcal{R}$, $p_1 - p_2 = 0$. Thus, $x_1 - x_2 = y_1 - y_2 + t_1 - t_2 = k - 2y_2 - 2t_2 - t \leqq k - j_2$. Similarly, $x_2 - x_1 = y_2 - y_1 + t_2 - t_1 = k - 2y_1 - 2t_1 - t \leqq k - j_1$.

(if) For k and j satisfy the equations, we may construct a (k, j)-crucial path set of G following a similar construction in [29]. The only difference is semi-crucial path cannot join with other semi-crucial path. □

Lemma 3. Let $(G, S) = (G_1 \oplus G_2, S_1)$. Given $\mathcal{P}_1$, an (x_1, j_1)-crucial path set of G_1, and $\mathcal{P}_2$, an (x_2, j_2)-crucial path set of G_2. There exists a (k, j)-crucial path set $\mathcal{P}$ of G such that $\mathcal{P}_1$ is the projection of $\mathcal{P}$ on G_1 and $\mathcal{P}_2$ is the projection of $\mathcal{P}$ on G_2 if and only if $j_1 + j_2 \leq j$, $j_1 \leq x_1 \leq s_1$, $j_2 \leq x_2 \leq s_2$, $k = x_1 - x_2 + j - j_1 \leq s_1$.

Proof. (only if) It is obvious that $j_1 + j_2 \leq j$, $j_1 \leq x_1 \leq s_1$, $j_2 \leq x_2 \leq s_2$ and $k \leq s_1$. For a crucial path set $\mathcal{P}$ of G, denote the subset of the paths in G_1 by $\mathcal{Q}_1$, the subset of crossing paths with both end vertices in G_1 by $\mathcal{R}_1$ and the subset of crossing paths of one end vertex in G_1 and the other in G_2 by $\mathcal{R}$. Note that any path in crucial path set of G must have at least one end vertex in twin set $S = S_1$. The types of $\mathcal{Q}_1$, $\mathcal{R}_1$, and $\mathcal{R}$ are (y_1, i_1), (t_1, r_1) and (t, r) respectively. Since all paths in $\mathcal{R}$ are semi-crucial in G with an end vertex in G_2, its projection on G_1 is a set crucial paths in G_1 and $t = r$. Note that the projection of $\mathcal{R}_1$ on G2 is a set of crucial paths in G2. Thus, the semi-crucial paths of $\mathcal{P}_1$ are from $\mathcal{Q}_1$ and $\mathcal{R}_1$, i.e. $j_1 = i_1 + r_1$, and the semi-crucial paths of $\mathcal{P}_2$ are from $\mathcal{R}$, i.e. $j_2 \leq r$. We have $j = j_1 + j_2 = i_1 + r_1 + r = j_1 + r$. Let P be a path. After removing all the crossing edges in P, it is decomposed into paths in G_1 and in G_2. Let the number of paths in G_1 and G_2 be p_1 and p_2. If P is in $\mathbb{R}_1$, there are even number of crossing edges in P, and $p_1 - p_2 = 1$. Similarly, if P is in $\mathbb{R}$, $p_1 - p_2 = 0$. Thus, $x_1 - x_2 = y_1 + t_1$ and $k = y_1 + t_1 + t = x_1 - x_2 + r = x_1 - x_2 + j - j_1$.

(if) It can be proved by the similar construction in [29]. □

3 The Longest Path Problem

Let (G, S) be a distance-hereditary graph formed by two distance hereditary graphs (G_1, S_1) and (G_2, S_2). The size of twin sets S, S_1, and S_2 are s, s_1, and s_2, respectively. Let $L(G)$ denote the length of the longest path of G. Let $\mathcal{P}(G; i, j)$ denote the collection of all (i, j)-crucial path sets of G. Let $P(G; i, j)$ denote the maximum number of vertices covered by an (i, j)-crucial path set. In other words, $P(G; i, j) = \max\{c(\mathbb{P}) \mid \mathbb{P} \in \mathcal{P}(G; i, j)\}$, where $c(\mathbb{P})$ is the number of vertices covered by $\mathbb{P}$. The longest path of G is a path of G_1 or G_2, or a crossing path. In the latter case, the longest path can be decomposed into several *crucial* or *semi-crucial* paths in G_1 and G_2. In the following, assume we have a decomposition tree of G and we will compute $L(G)$, and $P(G; i, j)$ recursively from the leaves of the decomposition tree. First, the primitive distance-hereditary graph consisting a single vertex is of $L(G) = 1$, $P(G; 1, j) = 1$. For ease of computing, we set $P(G; 0, 0) = 0$.

Lemma 4. Let $G = G_1 \odot G_2$, then

$$\begin{aligned}
&P(G; k, 0)\\
&\quad = max\{P(G_1; x, 0) + P(G_2; k-x, 0) \mid max\{0, k-s_2\} \leq x \leq min\{k, s_1\}\},\\
&P(G; k, 1)\\
&\quad = max\{P(G_1; x, j) + P(G_2; k-x, 1-j) \mid max\{0, k-s_2\} \leq x \leq min\{k, s_1\}, j = 0, 1\},\\
&P(G; k, 2)\\
&\quad = max\{P(G_1; x, j) + P(G_2; k-x, 2-j) \mid max\{0, k-s_2\} \leq x \leq min\{k, s_1\}, j = 0, 1, 2\}.
\end{aligned}$$

Proof. Since $G = G_1 \odot G_2$, G_1 and G_2 are disconnected in G, a path of G is a path of G_1 or G_2. A (k, j)-crucial path set of G is consisted of an (x, y)-crucial path set in G_1, and a $(k - x, j - y)$-crucial path set of G_2, where $0 \leq x \leq s_1$, $0 \leq k - x \leq s_2$, $0 \leq y \leq j$, and $0 \leq\ j - y$. So, $max(0, k - s_2) \leq x \leq min(s_1, k)$. Thus, by the definition of $P(G; k, j)$,

$$\begin{aligned}
&P(G; k, j)\\
&\ = max\ \{c(\mathbb{P}) \mid \mathbb{P} \in \mathcal{P}(G; k, j)\}\\
&\ = max\{c(\mathbb{P}_1) + c(\mathbb{P}_2) \mid \mathbb{P}_1 \in \mathcal{P}(G_1; x, y), \mathbb{P}_2 \in \mathcal{P}(G_2; k - x, j - y),\\
&\qquad\qquad\qquad\qquad max\{0, k - s_2\} \leq x \leq min\{k, s_1\}, y = 0, \ldots, j\}\\
&= max\{P(G_1; x, y) + P(G_2; k - x, j - y) \mid max\{0, k - s_2\} \leq x \leq min\{k, s_1\}, 0, \ldots, j\}.
\end{aligned}$$

□

Lemma 5. Let $G = G_1 \otimes G_2$, then

$$P(G; k, 0) = max\ \{P(G_1; x, 0) + P(G_2; y, 0) \mid |x-y| \leq k \leq x+y,\ 0 \leq x \leq s_1,\ 0 \leq y \leq s_2\},$$

$$P(G;\, k, 1) = max\{P(G_1;\, x, 1\text{–}j) + P(G_2;\, y, j) \mid x\text{–}y \le k\text{–}j,\ y\text{–}x \le k\text{–}1+j,\ k \le x+y,$$
$$1\text{–}j \le x \le s_1,\ j \le y \le s_2,\ j = 0, 1\},\ \text{and}$$

$$P(G;\, k, 2) = max\{P(G_1;\, x, 2\text{–}j) + P(G_2;\, y, j) \mid x\text{–}y \le k\text{–}j,\ y\text{–}x \le k\text{–}2+j,\ k \le x+y,$$
$$2\text{–}j \le x \le s_1,\ j \le y \le s_2,\ j = 0, 1, 2\}.$$

Proof. By Lemma 2, a (k, l)-crucial path set of G can be made from an (x, l–j)-crucial path set of G_1 and a (y, j)-crucial path set of G_2, where $x\text{–}y \le k\text{–}j$, $y\text{–}x \le k\text{–}l+j$, $k \le x+y$, $l\text{–}j \le x \le s_1$, $j \le y \le s_2$, $j = 0, \ldots, l$. By the definition of $P(G;\, k, l)$,

$$\begin{aligned} P(G;\, k, l) &= max\ \{c(\mathcal{P}) \mid \mathcal{P} \in \mathcal{P}(G;\, k, l)\} \\ &= max\{c(\mathcal{P}_1)+c(\mathcal{P}_2) \mid \mathcal{P}_1 \in \mathcal{P}(G_1;\, x, l\text{-}j),\ \mathcal{P}_2 \in \mathcal{P}(G_2;\, y, j), \\ &\qquad x\text{–}y \le k\text{–}j,\ y\text{–}x \le k\text{–}l+j,\ k \le x+y,\ l\text{–}j \le x \le s_1,\ j \le y \le s_2,\quad j = 0, \ldots, l\} \\ &= max\{P(G_1;\, x, 1\text{–}j) + P(G_2;\, y, j) \mid x\text{–}y \le k\text{–}j,\ y\text{–}x \le k\text{–}l+j, \\ &\qquad k \le x+y,\ l\text{–}j \le x \le s_1,\ j \le y \le s_2,\ j = 0, \ldots, l\}. \quad \square \end{aligned}$$

Lemma 6. Let $G = G_1 \oplus G_2$, then

$$P(G;\, k, 0) = max\ \{P(G_1;\, x+k, 0) + P(G_2;\, x, 0) \mid 0 \le x \le min\{s_1\text{–}k, s_2\}\},$$
$$\begin{aligned} &P(G;\, k, 1) \\ &\quad = max\ \{\{P(G_1;\, x+k\text{–}j, 1\text{–}j) + P(G_2;\, x, j) \mid 0 \le x \le min\{s_1\text{–}k + j, s_2\}, j = 0, 1\} \cup \\ &\qquad\quad \{P(G_1;\, x+k\text{–}1, 0) + P(G_2;\, x, 0) \mid 0 \le x \le min\{s_1\text{–}k + 1, s_2\}\ \}, \\ &P(G;\, k, 2) \\ &\quad = max\ \{\{P(G_1;\, x+k\text{–}j, 2\text{–}j) + P(G_2;\, x, j) \mid 0 \le x \le min\{s_1\text{–}k + j, s_2\}, j = 0, 1, 2\} \cup \\ &\qquad\quad \{P(G_1;\, x+k\text{–}1\text{–}j, 1\text{–}j) + P(G_2;\, x, j) \mid 0 \le x \le min\{s_1\text{–}k + 1+j, s_2\}, j = 0, 1\} \cup \\ &\qquad\quad \{\ P(G_1;\, x+k\text{–}2, 0) + P(G_2;\, x, 0) \mid 0 \le x \le min\{s_1\text{–}k + 2, s_2\}\ \}. \end{aligned}$$

Proof. By Lemma 3, a (k, l)-crucial path set of G can be made from an (x+k–(l–i), i)-crucial path set of G_1 and an (x, j)-crucial path set of G_2, for some $0 \le x \le min\{s_1\text{–}k+j, s_2\}$, $i, j \ge 0$, $i+j \le l$. By the definition of $P(G;\, k, l)$,

$$\begin{aligned} &P(G;\, k, l) \\ &= max\ \{c(\mathcal{P}) \mid \mathcal{P} \in \mathcal{P}(G;\, k, l)\} \\ &= max\{c(\mathcal{P}_1)+c(\mathcal{P}_2) \mid \mathcal{P}_1 \in \mathcal{P}(G_1;\, x+k\text{–}(l\text{–}i), i),\ \mathcal{P}_2 \in \mathcal{P}(G_2;\, x, j), \\ &\qquad\qquad 0 \le x \le min\{s_1\text{–}k+(l\text{–}i), s_2\},\ i, j \ge 0,\ i+j \le l\ \} \\ &= max\{P(G_1;\, x+k\text{–}(l\text{–}i), i) + P(G_2;\, x, j) \mid 0 \le x \le min\{s_1\text{–}k+(l\text{–}i), s_2\},\ i, j \ge 0,\ i+j \le l\ \}. \quad \square \end{aligned}$$

In the above discussion, we compute the value $P(G;\, i, j)$ for each node of decomposition tree T. For false twin and attachment nodes, computing each $P(G;\, i, j)$ needs $O(n)$ time, but for true twin nodes, it takes $O(n^2)$ time.

Next, we show how to compute $L(G)$, the length of the longest path of a distance-hereditary graph G. If G is obtained by *false twin* operation, since G_1 and G_2 are disconnected, $L(G) = max\{\ L(G_1), L(G_2)\ \}$. For the case of true twin and attachment, consider a longest path Q of G. If Q is completely in G_1(or G_2), then Q is also a longest path of G_1(or G_2); in such a case, we have $L(G) = L(G_1)$ (or $L(G_2)$). If Q is a crossing path, by removing the crossing edges from Q, it will form a crucial path set $\mathcal{Q}_1$ on

G_1 and a crucial path set Q_2 on G_2. If two end vertices of Q are in G_1, then semi-crucial paths can only occur in Q_1 and Q_1 has one more path than Q_2. Thus, $L(G) = \max \{P(G_1; i+1, x) + P(G_2; i, 0)| \ 0 \le i \le \min(s_1-1, s_2), x = 0, 1, 2\}$. The case for two end vertices of Q are in G_2 can be argued similarly to yield $L(G) = max \{P(G_1; i, 0) + P(G_2; i+1, y) \mid 0 \le i \le \min(s_1, s_2-1), y = 0, 1, 2\}\}$. In the case that one end vertex of Q is in G_1 and the other is in G_2, Q_1 and Q_2 have the same number of paths and $L(G) = max \{P(G_1; i, x) + P(G_2; i, y) \mid 0 \le i \le \min(s_1, s_2), x, y = 0, 1\}$.

In summary, if G is constructed by *true twin* or *attachment* operation from G_1 and G_2,

$$\begin{aligned} L(G) = max \{ & L(G_1), \quad L(G_2), \\ & max\{P(G_1; i+1, x) + P(G_2; i, 0) \mid 0 \le i \le \min(s_1-1, s_2), x \in \{0, 1, 2\}\}, \\ & max \{P(G_1; i, 0) + P(G_2; i+1, y) \mid 0 \le i \le \min(s_1, s_2-1), y \in \{0, 1, 2\}\}, \\ & max \{P(G_1; i, x) + P(G_2; i, y) \mid 0 \le i \le \min(s_1, s_2), x \in \{0, 1\}, y \in \{0, 1\}\}\} \end{aligned}$$

Theorem 2. Given a distance-hereditary graph G and its decomposition tree T_G, $L(G)$ can be computed in $O(n^4)$ time.

Proof. First, for the given distance-hereditary graph G, we construct the decomposition tree T_G, which takes $O(n + m)$ times by Lemma 1. In each node v of T_G with twin set of size s_v, we compute $P(G_v; i, j)$, where $0 \le i \le s_v$, and $L(G)$. Thus for each node there are $O(n)$ terms to be computed, and for each term we need $O(n^2)$ time to determine the value of $P(G_v; i, j)$. Thus the time complexity to compute the longest path of G is $O(n^4)$. □

References

1. Arikati, S.R., PanduRangan, C.: Linear algorithm for optimal path cover problem on interval graphs. Inf. Process. Lett. 35, 149–153 (1990)
2. Asdre, K., Nikolopoulos, S.D.: The 1-fixed-endpoint path cover problem is polynomial on interval graphs. Algorithmica (2009), doi:10.1007/s00453-009-9292-5
3. Bandelt, H.J., Mulder, H.M.: Distance-hereditary graphs. Journal of Combinatorial Theory Series B 41(1), 182–208 (1989)
4. Berge, C.: Graphs and Hypergraphs. North-Holland, Amsterdam (1973)
5. Bertossi, A.A.: Finding Hamiltonian circuits in proper interval graphs. Inf. Process. Lett. 17, 97–101 (1983)
6. Brandstädt, A., Dragan, F.F.: A linear-time algorithm for connected-domination and Steiner tree on distance-hereditary graphs. Networks 31, 177–182 (1998)
7. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (1999)
8. Bulterman, R., van der Sommen, F., Zwaan, G., Verhoeff, T., van Gasteren, A., Feijen, W.: On computing a longest path in a tree. Inf. Process. Lett. 81, 93–96 (2002)
9. Chang, M.S., Hsieh, S.Y., Chen, G.H.: Dynamic Programming on Distance-Hereditary Graphs. In: Leong, H.-V., Jain, S., Imai, H. (eds.) ISAAC 1997. LNCS, vol. 1350, pp. 344–353. Springer, Heidelberg (1997)
10. Chang, M.S., Peng, S.L., Liaw, J.L.: Deferred-query: an efficient approach for some problems on interval graphs. Networks 34, 1–10 (1999)

11. Corneil, D.G., Lerchs, H., Stewart, L.K.: Complement reducible graphs. Discrete Appl. Math. 3, 163–174 (1981)
12. Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for cographs. SIAM J. Comput. 14, 926–934 (1985)
13. D'atri, A., Moscarini, M.: Distance-hereditary graphs, Steiner trees, and connected domination. SIAM J. Comput. 17(3), 521–538 (1988)
14. Damaschke, P.: The Hamiltonian circuit problem for circle graphs is NP-complete. Inf. Process. Lett. 32, 1–2 (1989)
15. Damaschke, P.: Paths in interval graphs and circular arc graphs. Discrete Math. 112, 49–64 (1993)
16. Damaschke, P., Deogun, J.S., Kratsch, D., Steiner, G.: Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm. Order 8, 383–391 (1992)
17. Dragan, F.F.: Dominating Cliques in Distance-Hereditary Graphs. In: Schmidt, E.M., Skyum, S. (eds.) SWAT 1994. LNCS, vol. 824, pp. 370–381. Springer, Heidelberg (1994)
18. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, New York (1979)
19. Garey, M.R., Johnson, D.S., Tarjan, R.E.: The planar Hamiltonian circuit problem is NP-complete. SIAM J. Comput. 5, 704–714 (1976)
20. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, NewYork (1980)
21. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Annals of Discrete Mathematics, vol. 57. North-Holland, Amsterdam (2004)
22. Hammer, P.L., Maffray, F.: Complete separable graphs. Discrete Appl. Math. 27(1), 85–99 (1990)
23. Howorka, E.: A characterization of distance-hereditary graphs. Quarterly Journal of Mathematics 28(2), 417–420 (1977)
24. Howorka, E.: A characterization of Ptolemaic graphs. J. Graph Theory 5, 323–331 (1981)
25. Hsieh, S.Y.: Parallel Decomposition of Distance-Hereditary Graphs. In: Zinterhof, P., Vajtersic, M., Uhl, A. (eds.) ACPC 1999 and ParNum 1999. LNCS, vol. 1557, pp. 417–426. Springer, Heidelberg (1999)
26. Hsieh, S.Y., Ho, C.W., Hsu, T.S., Ko, M.T., Chen, G.H.: Efficient parallel algorithms on distance-hereditary graphs. Parallel Process. Lett. 9(1), 43–52 (1999)
27. Hsieh, S.Y., Ho, C.W., Hsu, T.S., Ko, M.T., Chen, G.H.: A faster implementation of a parallel tree contraction scheme and its application on distance-hereditary graphs. J. Algorithms 35, 50–81 (2000)
28. Hsieh, S.Y., Ho, C.W., Hsu, T.-S., Ko, M.T., Chen, G.H.: Characterization of efficient parallel solvable problems on distance-hereditary graphs. SIAM J. Discrete Math. 15(4), 488–518 (2002)
29. Hsieh, S.Y., Ho, C.W., Hsu, T.S., Ko, M.T.: The Hamiltonian problem on distance-hereditary graphs. Discrete Applied Math. 154, 508–524 (2006)
30. Hung, R.W., Wu, S.C., Chang, M.S.: Hamiltonian cycle problem on distance-hereditary graphs. J. Inform. Sci. Engrg. 19, 827–838 (2003)
31. Hung, R.W., Chang, M.S.: Finding a minimum path cover of a distance-hereditary graph in polynomial time. Discrete Applied Math. 155, 2242–2256 (2007)
32. Itai, A., Papadimitriou, C.H., Szwarcfiter, J.L.: Hamiltonian paths in grid graphs. SIAM J. Comput. 11, 676–686 (1982)
33. Ioannidou, K., Mertzios, G.B., Nikolopoulos, S.D.: The Longest Path Problem has a Polynomial Solution on Interval Graphs. Algorithmica 61(2), 320–341 (2011)

34. Ioannidou, K., Nikolopoulos, S.D.: The Longest Path Problem is Polynomial on Cocomparability Graphs. In: Thilikos, D.M. (ed.) WG 2010. LNCS, vol. 6410, pp. 27–38. Springer, Heidelberg (2010)
35. Müller, H., Nicolai, F.: Polynomial time algorithms for Hamiltonian problems on bipartite distance-hereditary graphs. Inform. Process. Lett. 46, 225–230 (1993)
36. Müller, H.: Hamiltonian circuits in chordal bipartite graphs. Discrete Math. 156, 291–298 (1996)
37. Narasimhan, G.: A note on the Hamiltonian circuit problem on directed path graphs. Inf. Process. Lett. 32, 167–170 (1989)
38. Nicolai, F.: Hamiltonian problems on distance-hereditary graphs. Technique report, Gerhard-Mercator University, Germany (1994)
39. Takahara, Y., Teramoto, S., Uehara, R.: Longest path problems on Ptolemaic graphs. IEICE Trans. Inf. Syst. 91-D, 170–177 (2008)
40. Uehara, R., Uno, Y.: Efficient Algorithms for the Longest Path Problem. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 871–883. Springer, Heidelberg (2004)
41. Uehara, R., Valiente, G.: Linear structure of bipartite permutation graphs and the longest path problem. Inf. Process. Lett. 103, 71–77 (2007)
42. Yeh, H.G., Chang, G.J.: Weighted connected domination and Steiner trees in distance-hereditary graphs. Discrete Appl. Math. 87(1-3), 245–253 (1998)
43. Yeh, H.G., Chang, G.J.: Linear-time algorithms for bipartite distance-hereditary graphs (manuscript)
44. Yeh, H.G., Chang, G.J.: The path-partition problem in bipartite distance-hereditary graphs. Taiwanese J. Math. 2(3), 353–360 (1998)
45. Yeh, H.G., Chang, G.J.: Weighted connected k-domination and weighted k-dominating clique in distance-hereditary graphs. Theoret. Comput. Sci. 263, 3–8 (2001)

Balancing a Complete Signed Graph by Editing Edges and Deleting Nodes

Bang Ye Wu* and Jia-Fen Chen

National Chung Cheng University, ChiaYi, Taiwan 621, R.O.C.
bangye@ccu.edu.tw

Abstract. A signed graph is a simple undirected graph in which each edge is either positive or negative. A signed graph is balanced if every cycle has even numbers of negative edges. In this paper we study the problem of balancing a complete signed graph by minimum editing cost, in which the editing operations includes inserting edges, deleting edges, and deleting nodes. We design a branch-and-bound algorithm, as well as a heuristic algorithm. By experimental results we show that the branch-and-bound algorithm is much efficient than a trivial one and the heuristic algorithm performs well.

Keywords: algorithm, social network analysis, signed graph, balanced graph.

1 Introduction

A *signed graph* is a simple undirected graph $G = (V, E)$ in which each edge is labeled by a sign either +1 or -1, i.e., there is a function $s : E \mapsto \{1, -1\}$. A signed graph is *balanced* if every cycle has even numbers of negative edges. In social network analysis, it is believed that a social network usually evolves toward to a balanced system and a well-developed social network should be balanced [14]. Therefore computing the minimum changes to balance a network may be helpful for predicting how the system evolves and for detecting true/false relations. In this paper, we study the optimization problem how to balance a complete signed graph by minimum total cost of sign changes and node deletions.

Suppose that s is an edge-sign function on $V \times V$ and $B \subset V$. For $F \subset B \times B$, let $s' = s \oplus F$ be the modified sign function on $B \times B$ such that $s'(u, v) = s(u, v)$ if $(u, v) \notin F$ and $s'(u, v) = -s(u, v)$ if $(u, v) \in F$. We say that F is an (edge) editing set. When each sign change takes unit cost and deleting a node takes cost w, the problem to find a minimum total cost can be formulated as follows.

Problem: Balanced Graph Editing problem (BG-Editing)
Instance: A complete signed graph $G = (V, E = V \times V, s)$ in which s is an edge-sign function and $w > 0$.
Goal: Find a node subset B and an edge editing set F such that $G' = (B, E' = B \times B, s \oplus F)$ is balanced so as to minimize $|F| + w|V - B|$.

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 79–88.
DOI: 10.1007/978-3-642-35452-6_10

Basically we want to modify a complete signed graph into a balanced graph by deleting nodes and changing edge signs. It can be also thought of as a two-criteria optimization problem: minimizing the number of deleting nodes and minimizing the number of sign changes. There are different versions of the problems.

- To minimize sign changes subject to deleting at most k_1 nodes;
- To minimize the number of deleting nodes subject to changing at most k_2 edges;
- To find the minimum number of sign changes for each possible number of deleting nodes.

In the literature, BG-Editing also appeared in other forms. Modifying a given graph to satisfy some property is a wide interesting problem by its own reason in the area of graph theory. A *cluster graph* is an undirected graph consisting of disjoint maximal cliques and a cluster graph is a k-cluster graph if the number of clusters is at most k. Modifying a given graph into a cluster graph by minimum editing cost is an important graph modification problem since it relates to *graph clustering* and finds many applications. By Harary's theorem [10], a signed graph is balanced if and only if it can be partitioned into two clusters (node subsets) such that every positive edge is within the clusters and every negative edge crosses the two clusters. Therefore a complete signed graph G is balanced if and only if the (unsigned) graph formed by all positive edges of G is a 2-cluster graph. One way to realize the above result is as follows: A social network is balanced if all the actors can be partitioned into two parts. People within the same part are friends and any two people in different parts are enemies.

There are several previous results about cluster graph modification. Shamir et al. [13] studied the computational complexities of three edge modification problems. Cluster Editing asks for the minimum total number of edge insertions and deletions, while in Cluster Deletion (respectively, Cluster Completion), only edge deletions (respectively, insertions) are allowed. They showed that Cluster Editing is NP-hard, Cluster Deletion is Max SNP-hard, and Cluster Completion is polynomial-time solvable. For a constant integer $p \geq 2$, p-Cluster Deletion (similarly, p-Cluster Editing) is a variant of Cluster Deletion in which the desired solution must contain exactly p clusters. They also showed that p-Cluster Deletion is NP-hard for any $p > 2$ but polynomial-time solvable for $p = 2$, and p-Cluster Editing is NP-hard for any $p \geq 2$. In the literature, there are several results on the fixed-parameter time complexities for Cluster Editing and Cluster Deletion, for example [2–5, 7–9], and the most recent result can be referred to [1]. A variant with node (rather than edge) deletions was considered in [11], and another variant in which overlapping clusters are allowed was studied in [6]. A branch-and-bound algorithm for balancing a complete signed graph by minimum edge editing was proposed in [15].

Almost all the previous works focus on only edge modification or only node modification. In this paper, we study the problem with both edge editing and

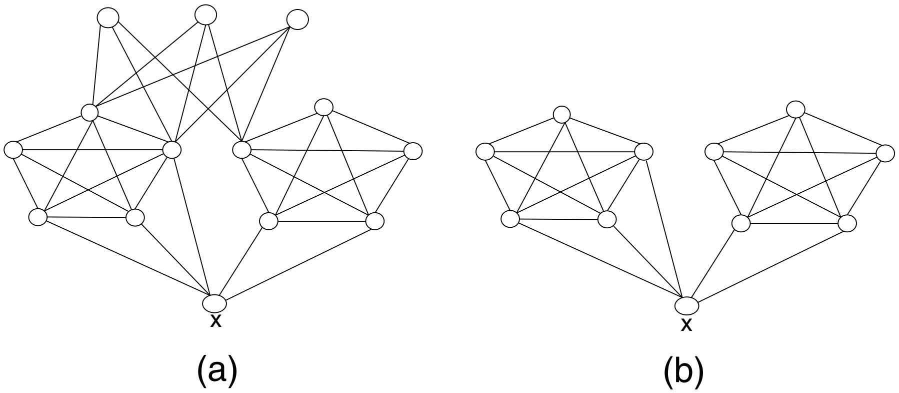


Fig. 1. A 2-clustering example

node deletion. This consideration may be very important in practical applications. For example, in a 2-clustering problem there may be some objects which should not be classified into any of the two clusters. This may be due to noises or experimental errors. In the balanced social network problem there may be someone who does not belong to either part. However, it may harm the resulting 2-clustering to enforce that every node must be clustered. To resolve this drawback, we propose the BG-Editing problem in which both edge editing and node deletion are considered and therefore not all the nodes need to be clustered. For example, consider the graph in Fig. 1.(a). There are two well-clustered groups (cliques) in the graph, each with five nodes. If the upper three nodes are not considered, as in Fig. 1.(b), the node x should be clustered with the left clique. Such a solution can be obtained if node deletions are allowed. However, if node deletions are not allowed as in the tradition clustering editing problem, the upper three nodes will be clustered with the left clique in the “optimal solution”, and the node x will be clustered with the right one.

When the weight w of node deletion is sufficiently large, BG-Editing is equivalent to Cluster Editing; and it degenerates to Cluster Vertex Deletion when w is sufficiently small. Therefore it is immediate that BG-Editing is NP-hard. In this paper we first give a branch-and-bound algorithm for BG-Editing. We show that with a good lower bound and a branching strategy it is much more efficient than a trivial brute force method. In addition, we also present a heuristic algorithm, and by experimental result we show that the heuristic algorithm performs well.

The paper is organized as follows. In Section 2, we give some notation and study some properties. We present the algorithms in Section 3, and the experiment results are given in Section 4. Finally, some concluding remarks are given in Section 5.

2 Preliminaries and Some Properties

For a graph G, $V(G)$ and $E(G)$ denote the node and edge sets, respectively. A clique is a complete subgraph. A k-clique is a clique of k nodes. A k-cycle is a cycle of k nodes. A 3-clique is also called a *triangle* which is also a 3-cycle. A cycle is positive if it contains an even number of negative edges. In the remaining paragraphs, the input graph is always a complete signed graph $G = (V, E, s)$ and we use $n = |V|$. Let E^+ be the set of positive edges, and E^- be the set of negative edges. For a node subset U and a node v, let $d^+(v, U)$ denote the number of positive edges between v and all nodes in U. Similarly $d^-(v, U) = |\{u \in U | s(u, v) = -1\}|$. Let $d^+(U) = (1/2) \sum_{v \in U} d^+(v, U)$ and $d^-(U) = (1/2) \sum_{v \in U} d^-(v, U)$ be the numbers of positive and negative edges in the subgraph induced by U, respectively.

A *perfect 2-clustering* of a singed graph $G = (V, E, s)$ is bipartition (V_1, V_2) of V such that $s(u, v) = -1$ if and only if u and v are in the different subsets. The next theorem is due to Harary.

Theorem 1. *A singed graph is balanced if and only if there is a perfect 2-clustering [10].*

For a bipartition (V_1, V_2), we mark each node in V_1 by +1 and each node in V_2 by -1. That is, we define a function $r \mapsto \{+1, -1\}$ such that $r(v) = +1$ for any $v \in V_1$ and $r(v) = -1$ for any $v \in V_2$. We say that an edge $e = (u, v)$ *agrees* with the bipartition if $r(u) \times r(v) = s(u, v)$ and e is an *agreement*, and otherwise e is a *disagreement*. We also abuse the term agreement to denote the total number of agreeing edges. By Theorem 1, any edge disagreeing with the 2-cluster must be edited.

At a glance, the problem BG-Editing seems to look for an edge subset to modify, as well as a set of nodes to delete. However, by Theorem 1, it is to find a 3-partition: a set of nodes to delete, and a 2-partition for the 2-clustering. Therefore we may have a brute force algorithm with $O(n^2 \cdot 3^n)$ time complexity. The next theorem appears in [14], which provides a simple method to check if a complete graph is balanced.

Theorem 2. *A complete signed graph is balanced if and only if all triangles are positive [14].*

3 Algorithms

3.1 Exact Algorithms

To find an exact solution, we use a standard branching algorithm as a basis and then improve the efficiency by designing a lower bound estimate and a branching strategy. In the standard branching algorithm, there are three choices for every node: discarding it or putting it into V_1 or V_2.

By a n-components vector $R =< r_1, r_2, r_3, \ldots, r_n >$, we denote a partial solution such that, for each $1 \leq i \leq n$, $r_i = 1$ means that v_i must be in V_1;

$r_i = 2$ for $v_i \in V_2$; $r_i = -1$ for v_i being discarded; and $r_i = 0$ if it is not determined yet. Let U be the set of undetermined nodes. For a partial solution, we can compute the cost for all determined nodes. Also we can estimate a lower bound for any undetermined node. Define

$$\begin{aligned} g(V_1, V_2, V_0) &= |E^- \cap \{(u,v) : u, v \in V_1\}| + |E^- \cap \{(u,v) : u, v \in V_2\}| \\ &\quad + |E^+ \cap \{(u,v) : u \in V_1, v \in V_2\}| + w|V_0| \\ &= d^-(V_1) + d^-(V_2) + d^+(V_1, V_2) + w|V_0| \end{aligned} \tag{1}$$

which is cost of partitioning $V_1 \cup V_2$ into V_1 and V_2 and discarding V_0. Define

$$h(V_1, V_2, V_0) = \sum_{v \in U} \min\{d^-(v, V_1) + d^+(v, V_2), d^-(v, V_2) + d^+(v, V_1), w\} \tag{2}$$

which is a lower bound of the cost to partition U. Therefore $f(V_1, V_2, V_0) = g(V_1, V_2, V_0) + h(V_1, V_2, V_0)$ is a lower bound function. If it is not less than the current best solution, we need not explore it any more. The two functions g and h can be computed from the previous values to reduce the computation cost. Precisely speaking, we have

$$g(V_1 + x, V_2, V_0) = g(V_1, V_2, V_0) + d^-(x, V_1) + d^+(x, V_2) \tag{3}$$

$$g(V_1, V_2 + x, V_0) = g(V_1, V_2, V_0) + d^-(x, V_2) + d^+(x, V_1) \tag{4}$$

and

$$g(V_1, V_2, V_0 + x) = g(V_1, V_2, V_0) + w \tag{5}$$

in which $V_1 + x$ denote $V_1 \cup \{x\}$ for short. For $v \notin V_1 \cup V_2 \cup V_0$, let $h_1(v, V_1, V_2) = d^-(v, V_1) + d^+(v, V_2)$ and $h_2(v, V_1, V_2) = d^-(v, V_2) + d^+(v, V_1)$. Then we have that $h(V_1, V_2) = \sum_{v \in U} \min\{h_1(v, V_1, V_2), h_2(v, V_1, V_2), w\}$. Furthermore, for $v, x \in U$ and $v \neq x$, we have

$$h_1(v, V_1 + x, V_2) = \begin{cases} h_1(v, V_1, V_2) + 1 & \text{if } (v, x) \in E^- \\ h_1(v, V_1, V_2) & \text{if } (v, x) \in E^+ \end{cases} \tag{6}$$

and

$$h_1(v, V_1, V_2 + x) = \begin{cases} h_1(v, V_1, V_2) & \text{if } (v, x) \in E^- \\ h_1(v, V_1, V_2) + 1 & \text{if } (v, x) \in E^+ \end{cases} \tag{7}$$

A similar formula for h_2 can be derived. By this way, we can update the lower bound in $O(n)$ time when a node is moved from U to V_1, V_2 or V_0. The algorithm is in Algorithm 1 and the efficiency will be shown by the experimental results in the next section.

3.2 A Heuristic Algorithm

By Theorem 1, the optimal balanced graph corresponds to a 2-clustering, and we can have a simple algorithm with agreement at least one half of $|E|$. Let

Algorithm 1. Branch-and-bound algorithm

```
Input: A complete graph $G = (V, E)$, in which $V = \{v_i | 1 \le i \le n\}$.
Output: The minimum cost to make $G$ balanced.
  initially a stack $T$; Best $\leftarrow \infty$;
  push $(\emptyset, \emptyset, \emptyset, 1)$ into $T$;
  **while** $T$ is not empty **do**
      pop $(V_1, V_2, V_0, i)$ from $T$;
      compute $g(V_1 + v_i, V_2, V_0)$, $g(V_1, V_2 + v_i, V_0)$ and $g(V_1, V_2, V_0 + v_i)$;
      **if** $i = n$ **then**
          Best $\leftarrow \min\{\text{Best}, g(V_1 + v_i, V_2, V_0), g(V_1, V_2 + v_i, V_0), g(V_1, V_2, V_0 + v_i)\}$;
      **end if**
      compute $h(V_1 + v_i, V_2, V_0)$, $h(V_1, V_2 + v_i, V_0)$ and $h(V_1, V_2, V_0 + v_i)$;
      **if** $f(V_1 + v_i, V_2, V_0) <$ Best **then**
          push $(V_1 + v_i, V_2, V_0, i + 1)$ into $T$;
      **end if**
      **if** $f(V_1, V_2 + v_i, V_0) <$ Best **then**
          push $(V_1, V_2 + v_i, V_0, i + 1)$ into $T$;
      **end if**
      **if** $f(V_1, V_2, V_0 + v_i) <$ Best **then**
          push $(V_1, V_2, V_0 + v_i, i + 1)$ into $T$;
      **end if**
  **end while**
  **return** Best;
```

$(v_1, v_2, \ldots, v_n)$ be an arbitrary ordering of the nodes. We first put v_1 into V_1 and then put each incoming node into V_1, V_2 or discard it greedily. That is, in each iteration, we put the incoming node into one cluster such that the number of agreement of its incident edges is maximized or discard the node if the cost is better. The algorithm is listed in Algorithm 2, and we show the performance in the next section.

4 Experimental Results

We did experiments on algorithms for the BG-Editing problem. In our experiments, we used the following two kinds of random graphs.

- First-type: A complete signed graph is constructed such that each edge is negative with an independent identical probability p.
- Second-type: We first randomly generated a balanced complete graph with specified two cluster sizes. To make the graph unbalanced, We randomly pick k edges and change their signs. We further add η additional nodes of which the edges are assigned positive or negative randomly. By this way we can control the optimal solutions.

By these data, we tested our algorithms and observed factors which affect the solution and the time complexity. By g and $g + h$, we denote the two versions of

Algorithm 2. Greedy algorithm

```
Input: A complete signed graph $G = (V, E)$.
Output: A 3-partition $(V_1, V_2, V_0)$ of $V$.
  $(V_1, V_2, V_0) \leftarrow (\emptyset, \emptyset, \emptyset)$;
  let $V = \{v_1, v_2, \ldots, v_n\}$ in which the nodes are arbitrarily labeled;
  for $i \leftarrow 1$ to $n$ do
      if $w < \min\{d^+(v_i, V_1) + d^-(v_i, V_2), d^-(v_i, V_1) + d^+(v_i, V_2)\}$ then
          put $v_i$ into $V_0$                                          ▷ discard
      else if $d^+(v_i, V_1) + d^-(v_i, V_2) \geq d^-(v_i, V_1) + d^+(v_i, V_2)$ then
          put $v_i$ into $V_1$;
      else
          put $v_i$ into $V_2$;
      end if
  end for
  output $(V_1, V_2, V_0)$.
```

the branch-and-bound algorithm which use g and $g + h$ as the lower bounds as described in the previous section. We estimate the efficiency by the number of pushes which is the number of recursive calls. We set a bound 15 millions of the number of pushes in the experiments. The program was aborted if no solution was found before exceeding the bound. The cost of edge editing is set to one while discarding a node takes cost w.

4.1 Lower Bound for Algorithm Efficiencies

The first experiment is used to test the effect of our lower bound function. We observe the following results. The running time increases as the discarding cost w increases. We set $w = n/3$ in this experiment.

- For the First-type data:
  - When p increases, i.e., the number of negative edges increases, the optimal cost increases and it also takes more time to find the optimal solution.
  - With only g as the lower bound, it can hardly find the optimal solutions for $n = 20$ within 15M pushes even for $p = 0.2$. Meanwhile, using $g + h$ as the lower bound, we can find optimal solutions for $n = 40$ within the same threshold. But when p increases to 0.4, the running time reaches the threshold when n is about 30.
- For the Second-type data:
  - When $k = \binom{n}{2}/10$ and $\eta = n/10$, using $g+h$ and using g can find optimal solutions for $n = 40$ and for $n = 20$, respectively.
  - When k increases to $\binom{n}{2}/5$, the problem size we can solved within the threshold is reduced to about $n = 20$ and $n = 30$, respectively.

4.2 Node Selection Method for Branching

In the branch-and-bound algorithm, we need to choose a node v in each iteration and make three branches according to putting v into V_1 or V_2, or discarding v. In this experiment, we would like to know if the method of selecting node can improve the time complexity. We tested three selection methods: with-order, largest-gap and smallest-gap.

- with-order: When choosing the next node to branch, we simply choose the next undetermined one.
- largest-gap: Choosing $v = \arg\max_v |h(V_1 + v, V_2, V_0) - h(V_1, V_2 + v, V_0)|$.
- smallest-gap: Choosing $v = \arg\min_v |h(V_1 + v, V_2, V_0) - h(V_1, V_2 + v, V_0)|$.

We used Second-type random graphs and count the numbers of pushes to compare the three methods. The result is shown in Table 1, and we can observe that using largest-gap reduces the running time to about one half.

Table 1. Average numbers of pushes for different selection methods ($w = n/3$, lower-bound: $g + h$)

n	η	k	with-order	largest-gap	smallest-gap
10	1	4	336	310	383
20	2	19	20K	11K	21K
30	3	43	568K	225K	380K
40	4	78	9075K	4202K	7446K

4.3 Performance of the Heuristic Algorithm

We compare the solutions found by the heuristic algorithm with the optimal solutions. For both the first-type and the second-type data, the performance ratio is below 1.26, in which the performance ratio is defined by the maximum ratio of the cost found by the heuristic algorithm to the optimal one.

4.4 Discussion

By the experiments, we make the following discussions.

- The optimal cost increases as the number of negative edges increases, and so does the running time for finding the optimal solutions.
- It takes more time to find the optimal solution when the discarding cost w increases. When w is large, the problem is reduced to the version without node deletion, and therefore can be solved easier.
- The better lower bound $g + h$ reduces the running time significantly.
- Using the largest-gap to select nodes for branching speeds up the algorithm at a factor about two.
- The heuristic algorithm finds solutions of costs within 1.26 times of the optimal solution. It can be thought of as a good method for finding solution for large graphs.

5 Conclusion

In this paper, we study the problem of balancing a complete signed graph by minimum editing cost of inserting edges, deleting edges, and deleting nodes. For this NP-hard problem we design a branch-and-bound algorithm and a heuristic algorithm. The experimental results show that the designed lower bound and selection method can significantly reduce the running time. Also we show that the heuristic algorithm performs well.

Although the heuristic algorithm usually finds good solution, no theoretical bound of the solutions has been derived. Interesting future works includes good approximation algorithms and exact algorithms with theoretical bounds on solution quality or time complexity.

Acknowledgement. This work was supported in part by NSC 100-2221-E-194-036-MY3 and NSC 101-2221-E-194-025-MY3 from the National Science Council, Taiwan, R.O.C.

References

1. Böcker, S., Damaschke, P.: Even faster parameterized cluster deletion and cluster editing. Information Processing Letters 111(14), 717–721 (2011)
2. Böcker, S., Briesemeister, S., Bui, Q.B.A., Truss, A.: Going weighted: Parameterized algorithm for cluster editing. Theor. Comput. Sci. 410(52), 5467–5480 (2009)
3. Chen, J., Meng, J.: A $2k$ Kernel for the Cluster Editing Problem. In: Thai, M.T., Sahni, S. (eds.) COCOON 2010. LNCS, vol. 6196, pp. 459–468. Springer, Heidelberg (2010)
4. Damaschke, P.: Bounded-Degree Techniques Accelerate Some Parameterized Graph Algorithms. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 98–109. Springer, Heidelberg (2009)
5. Damaschke, P.: Fixed-parameter enumerability of cluster editing and related problems. Theory Computing Syst. 46, 261–283 (2010)
6. Fellows, M.R., Guo, J., Komusiewicz, C., Niedermeier, R., Uhlmann, J.: Graph-based data clustering with overlaps. Discrete Optimization 8(1), 2–17 (2011)
7. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Graph-modeled data clustering: Fixedparameter algorithms for clique generation. Theory Computing Syst. 38, 373–392 (2005)
8. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Automated generation of search tree algorithms for hard graph modification problems. Algorithmica 39, 321–347 (2004)
9. Guo, J.: A more effective linear kernelization for cluster editing. Theor. Comput. Sci. 410, 718–726 (2009)
10. Harary, F.: On the notion of balance of a signed graph. Michigan Mathematical Journal, 143–146 (1953)
11. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. Theory of Computing Systems 47(1), 196–217 (2010)
12. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006)

13. Shamir, R., Sharan, R., Tsur, D.: Cluster Graph Modification Problems. In: Kučera, L. (ed.) WG 2002. LNCS, vol. 2573, pp. 379–390. Springer, Heidelberg (2002)
14. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press, Cambridge (1994)
15. Wei, P.-S., Wu, B.Y.: Balancing a complete signed graph by changing minimum number of edge signs. In: Proceedings of the 29th Workshop on Combinatorial Mathematics and Computation Theory, Taiwan, (2012)

Internally Disjoint Paths in a Variant of the Hypercube

Tsung-Han Tsai[1], Y-Chuang Chen[2,⋆], and Jimmy J.M. Tan[1]

[1] Department of Computer Science, National Chiao Tung University,
Hsinchu 300, Taiwan(R.O.C)
{tsaich,jmtan}@cs.nctu.edu.tw

[2] Department of Information Management, Minghsin University of Science
and Technology, Xinfeng Hsinchu 304, Taiwan(R.O.C)
cardy@must.edu.tw

Abstract. The hypercube is one of the most popular interconnection networks for parallel computer/communication system. The exchanged hypercube, which is a variant of the hypercube, maintains several desirable properties of the hypercube such as low diameter, bipancyclicity, and super connectivity. In this paper, we give internally disjoint paths for parallel routing in exchanged hypercubes and show the wide diameter of exchanged hypercubes.

Keywords: hypercube, exchanged hypercube, disjoint paths, diameter, wide diameter.

1 Introduction

An *interconnection network* is usually modeled as a *graph*, in which vertices represent processors or computers, and edges represent connections or communication links. Throughout this paper, a network is represented as a loopless undirected graph G. A graph G is a two-tuple (V, E), where V is a nonempty set, and E is a subset of $\{(u, v) \mid (u, v)$ is an unordered pair of $V\}$. We say that V is the vertex set and E is the edge set. Two vertices, u and v, of a graph G are *adjacent* if $(u, v) \in E(G)$. A *path* P of length k from vertex u to vertex v in a graph G is a sequence of distinct vertices written as $x_0 \to x_1 \to x_2 \to \cdots \to v_k$ where $x_0 = u$, $x_k = v$, and $(x_i, x_{i+1}) \in E(G)$ for every $0 \leq i \leq k-1$ if $k \geq 1$. We also write P as $u \to P \to v$ to emphasize its beginning and ending vertices. A *cycle* is a path with at least three vertices such that the last vertex is adjacent to the first one. For clarity, a cycle of length k is represented by $x_1 \to x_2 \to \ldots \to x_k \to x_1$. The length of a path P, denoted by $l(P)$, is the number of edges in P. The *distance* between two distinct vertices u and v in graph G, denoted by $d_G(u, v)$, is the length of the shortest path between u and v.

Due to many attractive properties of the *hypercube* such as *regularity*, *recursive structure*, *vertex and edge symmetry*, *maximum connectivity*, effective *routing* and

⋆ Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 89–96.
DOI: 10.1007/978-3-642-35452-6_11

broadcasting algorithm, the hypercube is one of the most popular interconnection networks for parallel computer/communication system [2,4]. The definition of hypercubes is presented as follows. A hypercube Q_n is a graph with 2^n vertices and each vertex u is denoted by an n-bit binary string $u = u_n u_{n-1} \dots u_1$. Two vertices are adjacent if and only if their strings differ exactly in one bit position. Let $u = u_n u_{n-1} \dots u_1$ and $v = v_n v_{n-1} \dots v_1$ be two n-bit binary strings. The Hamming distance $d_H(u, v)$ between two vertices u and v is the number of different bits in the corresponding strings of both vertices. Clearly, $d_H(u, v) = d_{Q_n}(u, v)$, where $d_{Q_n}(u, v)$ denotes the distance between two vertices u and v in Q_n, i.e., the length of a shortest uv-path in Q_n. In particular, Q_n is a vertex-transitive and edge-transitive bipartite graph, and has diameter n [10].

As a variant of the hypercube, the *exchanged hypercube* proposed by Loh et al. [5] is defined by removing some edges from the hypercube. However, it maintains several desirable properties of the hypercube such as low *diameter* [5], *bipancyclicity* [7] and *super connectivity* [8]. In this paper, we shall give *internally disjoint paths* in exchanged hypercubes and show the *wide diameter* of exchanged hypercubes.

In the next section, we give definition and some properties of exchanged hypercubes. Section 3 deals with internally disjoint paths in exchanged hypercubes and show the wide diameter of exchanged hypercubes.

2 A Variant of the Hypercube: The Exchanged Hypercube

The exchanged hypercube is defined as an undirected graph $EH(s, t) = G(V, E)$, where $s \geq 1, t \geq 1$. The set of vertices $V = \{a_{s-1} \dots a_0 b_{t-1} \dots b_0 c \mid a_i, b_j, c \in \{0, 1\}$ for $0 \leq i \leq s-1, 0 \leq j \leq t-1\}$, and the set of edges $E = E_1 \cup E_2 \cup E_3$ is as the following.

$$\begin{aligned}
E_1 &= \{(v_1, v_2) \in V \times V \mid v_1[s+t:1] = v_2[s+t:1], v_1[0] \neq v_2[0]\},\\
E_2 &= \{(v_1, v_2) \in V \times V \mid v_1[t:1] = v_2[t:1], H(v_1[s+t:t+1], v_2[s+t:t+1]) = 1,\\
&\qquad v_1[0] = v_2[0] = 0\},\\
E_3 &= \{(v_1, v_2) \in V \times V \mid v_1[s+t:t+1] = v_2[s+t:t+1], H(v_1[t:1], v_2[t:1]) = 1,\\
&\qquad v_1[0] = v_2[0] = 1\},
\end{aligned}$$

where $v[x : y]$ denotes the bit pattern of v from dimension y to dimension x, and $H(u, v)$ denotes the Hamming distance between u and v. By the definition of $EH(s, t)$, we know that the number of vertices is 2^{s+t+1} and the number of edges is $(s + t + 2)2^{s+t-1}$. Since $EH(s, t)$ is a subgraph of the $(s + t + 1)$-dimensional hypercube Q_{s+t+1}, it is also a bipartite graph. Loh et al. [5] state the following properties.

Property 1. The diameter of $EH(s, t)$ is $(s + t + 2)$.

Property 2. $EH(s, t)$ is isomorphic to $EH(t, s)$.

Property 3. $EH(s, t)$ can be decomposed into two copies of $EH(s - 1, t)$ or $EH(s, t - 1)$.

Property 4. The subgraphs induced by the vertices of the form $\overbrace{* \cdots *}^{s} b_{t-1} \cdots b_0 0$ and $a_{s-1} \cdots a_0 \overbrace{* \cdots *}^{t} 1$ in $EH(s,t)$ are isomorphic to Q_s and Q_t, respectively, where $* \in \{0,1\}$.

The subgraphs induced by the vertex sets $V(Q_s)$ and $V(Q_t)$ are denoted by S and T, respectively. Then $S \cong Q_s$ and $T \cong Q_t$. So there are 2^t distinct induced subgraphs Q_s^i in $EH(s,t)$ for $1 \leq i \leq 2^t$, and there are 2^s distinct induced subgraphs Q_t^i in $EH(s,t)$ for $1 \leq i \leq 2^s$. We denote that the hamming distance between u and v in the induced subgraphs Q_s of $EH(s,t)$, called $h_s(u,v)$ (simply abbreviated as h_s), is the number of bits in which labels of u and v differ from dimension $t+1$ to $s+t$. Similarly, we denote that the hamming distance between u and v in the induced subgraphs Q_t of $EH(s,t)$, called $h_t(u,v)$ (simply abbreviated as h_t) is the number of bits in which labels of u and v differ from dimension 1 to t.

3 Internally Disjoint Paths and Wide Diameter

A vertex set $F \subseteq V(G)$ is a *separating set* or a *vertex cut* if $G - F$ is disconnected. The *connectivity* of G, written as $\kappa(G)$, is the minimum size of a vertex cut. A graph G is *k-connected* if the connectivity $\kappa(G)$ is at least k. Moreover, a graph G has connectivity k if G is k-connected but not $(k+1)$-connected. Let $\delta(G)$ be the minimum degree of G. It follows from Menger's Theorem that the connectivity of a graph is at least k if and only if there exist k internally vertex-disjoint (abbreviated as disjoint) paths between any two vertices.

Let α and β be two positive integers such that $\alpha \leq \kappa$ and $\beta \leq \kappa - 1$. Given any two distinct vertices u and v of G, let $D(u,v)$ denote the set of all α disjoint paths between u and v. Each element of $D(u,v)$ consists of α disjoint paths. The number of elements in $D(u,v)$ denoted by $|D(u,v)|$. Let $l_i(u,v)$ denote the longest length among these α path of the ith element of $D(u,v)$. We define that $d_\alpha(u,v) = \min_{1 \leq i \leq |D(u,v)|} l_i(u,v)$. The *$\alpha$-wide diameter* of G, denoted by $D_\alpha(G)$, is defined as $D_\alpha(G) = \max_{u,v \in V} \{d_\alpha(u,v)\}$. In particular, we call $D_\kappa(G)$ to be the *wide diameter* of G. Note that $D_1(G)$ is simply the diameter $D(G)$ of G.

Obviously, $D(G) \leq D_\kappa(G)$. For the hypercube Q_n, Latifi [3] proved that $D_n(Q_n) = n + 1$ for $n \geq 2$. For the crossed cube CQ_n, Chang et al. [1] proved that $D_n(CQ_n) = \lceil \frac{n}{2} \rceil + 2$ for $n \geq 2$. In this section, we shall discuss the wide diameter of exchanged hypercubes, and prove that $D_{s+1}(EH(s,t)) = s + t + 3$ for $3 \leq s \leq t$.

Lemma 1. *[6] The connectivity of the exchanged hypercubes $EH(s,t)$ is $s+1$ for $1 \leq s \leq t$.*

From Menger's Theorem, there exist $s+1$ internal disjoint paths between any two distinct vertices in exchanged hypercube $EH(s,t)$.

Lemma 2. *[9] Let u, v be any two vertices of the n-dimensional hypercube Q_n and assume that $d_{Q_n}(u,v) = k$. Then there are n disjoint paths between u and v such that k of them are of length k, and the remaining $n-k$ paths are of length $k+2$.*

Lemma 3. *[7] The vertices in the vertex set $V_c = \{a_s \cdots a_1 b_t \cdots b_1 c \mid a_i, b_j \in \{0,1\}$ for $1 \le i \le s, 1 \le j \le t\}$ $(c \in \{0,1\})$ are vertex-transitive.*

The following theorem gives lower bound of the wide diameter of the exchanged hypercube. For convenience of the proof of the following theorem, we abbreviate some symbols. If there are s consecutive 0's, then we denote it by 0^s, that is, $0^s = \overbrace{00\cdots0}^{s}$. If there are t consecutive 1's, then we denote it by 1^t, that is, $1^t = \overbrace{11\cdots1}^{t}$.

Theorem 1. *$D_{s+1}(EH(s,t)) \ge s+t+3$ for $1 \le s \le t$.*

Proof. By Lemma 1, the connectivity of the exchanged hypercubes $EH(s,t)$ is $s+1$. From Menger's Theorem, there exist $s+1$ internal disjoint paths between any two distinct vertices, denoted by u and v, in $EH(s,t)$. We consider $u = \overbrace{00\cdots0}^{s}\overbrace{00\cdots0}^{t}0$ (simply abbreviated as 0^s0^t0) and $v = \overbrace{11\cdots1}^{s}\overbrace{11\cdots1}^{t}1$ (simply abbreviated as 1^s1^t1). Let $u' = 0^s0^t1$ be a neighbor of u and let $V' = N(u) - u'$. The shortest path P between u and v in $EH(s,t) - V'$ must pass u'. Loh et al. [5] showed that the length of the shortest path between u' and v is $H(u,v)+2 = s+t+2$. The $+2$ is because routing has to use dimension 0 twice: $1 \to 0$ and $0 \to 1$. For clarity, we write P as $< u, u', R, v >$, where R is the shortest path between u' and v in $EH(s,t) - V'$. Since $|R| = d(u',v) = s+t+2$, it follows that $d_{EH(s,t)-V'} = 1 + d(u',v) = s+t+3$. Therefore, $D_{s+1}(EH(s,t)) \ge s+t+3$ for $1 \le s \le t$. □

The following theorem gives upper bound of the wide diameter of the exchanged hypercube.

Theorem 2. *$D_{s+1}(EH(s,t)) \le s+t+3$ for $3 \le s \le t$.*

The proof of Theorem 2 can be derived from the following Lemmas 4 to 15.

Lemma 4. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \le s \le t$. If $u[0] = v[0] = 0$, and $h_t(u,v) = 0$, then there exist $s+1$ internally disjoint paths $P_i (1 \le i \le s+1)$ between u and v such that h_s of them are of length h_s, $s - h_s$ paths are of length $h_s + 2$, and one path is of length $h_s + 6$.*

Proof. By Lemma 3, without loss of generality, we consider $u = 0^s0^t0$ and $v = 0^{s-h_s}1^{h_s}0^t0$ are in the same induced subgraphs, denoted by Q_s^1. By Lemma 2, there exist s internally disjoint paths $H_i (1 \le i \le s)$ between u and v in the induced subgraph Q_s^1 such that h_s of them are of length h_s, and the other

$s - h_s$ paths are of length $h_s + 2$. Without loss of generality, we set $|H_i| = h_s$ for $1 \leq i \leq h_s$ and $|H_i| = h_s + 2$ for $h_s + 1 \leq i \leq s$.

The following sets of $s+1$ internally disjoint paths can be set between u and v.

For $1 \leq i \leq s$,

$$P_i : u = 0^s 0^t 0 \rightarrow H_i \rightarrow v = 0^{s-h_s} 1^{h_s} 0^t 0,$$

where $|P_i| = h_s$ for $1 \leq i \leq h_s$ and $|P_i| = h_s + 2$ for $h_s + 1 \leq i \leq s$.

$$\begin{aligned} P_{s+1} : u = 0^s 0^t 0 \rightarrow 0^s 0^t 1 \rightarrow 0^s 0^{t-1} 11 \rightarrow 0^s 0^{t-1} 10 \rightarrow L \rightarrow 0^{s-h_s} 1^{h_s} 0^{t-1} 10 \\ \rightarrow 0^{s-h_s} 1^{h_s} 0^{t-1} 11 \rightarrow 0^{s-h_s} 1^{h_s} 0^t 1 \rightarrow v = 0^{s-h_s} 1^{h_s} 0^t 0, \end{aligned}$$

where $|P_{s+1}| = h_s + 6$. Note that the path L is of length h_s in another induced subgraph, denoted by Q_s^2. □

Lemma 5. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] = v[0] = 0$ and $h_t(u,v) \neq 0$, then there exist $s+1$ internally disjoint paths $P_i (1 \leq i \leq s+1)$ between u and v such that $h_s + 1$ of them are of length $h_s + h_t + 2$, and $s - h_s$ paths are of length $h_s + h_t + 4$.*

Proof. By Lemma 3, without loss of generality, we may assume that $u = 0^s 0^t 0$ and $v = 0^{s-h_s} 1^{h_s} 0^{t-h_t} 1^{h_t} 0$. Depending on h_s, two cases are distinguished.

Case 1: $h_s = 0$. Consider $u = 0^s 0^t 0$ and $v = 0^s 0^{t-h_t} 1^{h_t} 0$ are in distinct induced subgraphs, denoted by Q_s^1 and Q_s^2, respectively. The following sets of $s+1$ internally disjoint paths can be set between u and v.

For $1 \leq i \leq s$,

$$\begin{aligned} P_i : u = 0^s 0^t 0 \rightarrow 0^{s-i} 1 0^{i-1} 0^t 0 \rightarrow 0^{s-i} 1 0^{i-1} 0^t 1 \rightarrow R_i \rightarrow 0^{s-i} 1 0^{i-1} 0^{t-h_t} 1^{h_t} 1 \\ \rightarrow 0^{s-i} 1 0^{i-1} 0^{t-h_t} 1^{h_t} 0 \rightarrow v = 0^s 0^{t-h_t} 1^{h_t} 0, \end{aligned}$$

where $|P_i| = h_t + 4$. Note that the path R_i is of length h_t in the induced subgraph, denoted by Q_t^i.

$$P_{s+1} : u = 0^s 0^t 0 \rightarrow 0^s 0^t 1 \rightarrow R_{s+1} \rightarrow 0^s 0^{t-h_t} 1^{h_t} 1 \rightarrow v = 0^s 0^{t-h_t} 1^{h_t} 0,$$

where $|P_i| = h_t + 2$. Note that the path R_{s+1} is of length h_t in the induced subgraph, denoted by Q_t^{s+1}.

Case 2: $1 \leq h_s \leq s$. Consider $u = 0^s 0^t 0$ and $v = 0^{s-h_s} 1^{h_s} 0^{t-h_t} 1^{h_t} 0$ are in distinct induced subgraphs, denoted by Q_s^1 and Q_s^2, respectively. By Lemma 2, there exist s internally disjoint paths $H_i (1 \leq i \leq s)$ between u and $v' = 0^{s-h_s} 1^{h_s} 0^t 0$ in the induced subgraph Q_s^1 such that h_s of them are of length h_s, and the other $s - h_s$ paths are of length $h_s + 2$. Without loss of generality, we set $|H_i| = h_s$ for $1 \leq i \leq h_s$ and $|H_i| = h_s + 2$ for $h_s + 1 \leq i \leq s$. We set $0^s 0^t 0 \rightarrow H_i' \rightarrow 0^{s-h_s} 1^{h_s - i} 0 1^{i-1} 0^t 0 \rightarrow 0^{s-h_s} 1^{h_s} 0^t 0$ as H_i for $1 \leq i \leq h_s$, and $0^s 0^t 0 \rightarrow H_i' \rightarrow 0^{s+h_s-i} 1 0^{i-h_s-1} 1^{h_s} 0^t 0 \rightarrow 0^{s-h_s} 1^{h_s} 0^t 0$ as H_i for $h_s + 1 \leq i \leq s$.

We denote that $N_i(v) = 0^{s-h_s}1^{h_s-i}01^{i-1}0^{t-h_t}1^{h_t}0$ is the neighbor of v in Q_s^1 for $1 \le i \le h_s$, and $N_i(v) = 0^{s+h_s-i}10^{i-h_s-1}1^{h_s}0^{t-h_t}1^{h_t}0$ is the neighbor of v in Q_s^1 for $h_s + 1 \le i \le s$.

The following sets of $s+1$ internally disjoint paths can be set between u and v.

$$\begin{aligned} P_1 : u = 0^s0^t0 &\to H_1 \to 0^{s-h_s}1^{h_s}0^t0 \to 0^{s-h_s}1^{h_s}0^t1 \to R_1 \\ &\to 0^{s-h_s}1^{h_s}0^{t-h_t}1^{h_t}1 \to v = 0^{s-h_s}1^{h_s}0^{t-h_t}1^{h_t}0, \end{aligned}$$

where $|P_1| = h_s + h_t + 2$. Note that the path R_1 is of length h_t in the induced subgraph, denoted by Q_t^1.

For $2 \le i \le h_s$,

$$\begin{aligned} P_i : u = 0^s0^t0 &\to H_i' \to 0^{s-h_s}1^{h_s-i}01^{i-1}0^t0 \to 0^{s-h_s}1^{h_s-i}01^{i-1}0^t1 \to R_i \\ &\to 0^{s-h_s}1^{h_s-i}01^{i-1}0^{t-h_t}1^{h_t}1 \to 0^{s-h_s}1^{h_s-i}01^{i-1}0^{t-h_t}1^{h_t}0 \\ &\to v = 0^{s-h_s}1^{h_s}0^{t-h_t}1^{h_t}0, \end{aligned}$$

where $|P_i| = h_s + h_t + 2$. Note that the path R_i is of length h_t in the induced subgraph, denoted by Q_t^i.

For $h_s + 1 \le i \le s$,

$$\begin{aligned} P_i : u = 0^s0^t0 &\to H_i' \to 0^{s+h_s-i}10^{i-h_s-1}1^{h_s}0^t0 \to 0^{s-h_s-i}10^{i-h_s-1}1^{h_s}0^t1 \to R_i \\ &\to 0^{s+h_s-i}10^{i-h_s-1}1^{h_s}0^{t-h_t}1^{h_t}1 \to 0^{s+h_s-i}10^{i-h_s-1}1^{h_s}0^{t-h_t}1^{h_t}0 \\ &\to v = 0^{s-h_s}1^{h_s}0^{t-h_t}1^{h_t}0, \end{aligned}$$

where $|P_i| = h_s + h_t + 4$. Note that the path R_i is of length h_t in the induced subgraph, denoted by Q_t^i.

$$\begin{aligned} P_{s+1} : u = 0^s0^t0 &\to 0^s0^t1 \to R_{s+1} \to 0^s0^{t-h_t}1^{h_t}1 \to 0^s0^{t-h_t}1^{h_t}0 \to L \\ &\to v = 0^{s-h_s}1^{h_s}0^{t-h_t}1^{h_t}0, \end{aligned}$$

where $|P_{s+1}| = h_s + h_t + 2$. Note that the path R_{s+1} is of length h_t in the induced subgraph, denoted by Q_t^{s+1}, and L is of length h_s in the induced subgraph Q_s^2. □

Since the proof of Lemmas 6 to 15 are long and similar to Lemmas 4 and 5, we omit the proof of Lemmas 6 to 15 here, and only state the lemmas.

Lemma 6. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \le s \le t$. If $u[0] = v[0] = 1$, and $h_s(u,v) = 0$, then there exist $t+1$ internally disjoint paths $P_i(1 \le i \le t+1)$ between u and v such that h_t of them are of length h_t, $t - h_t$ paths are of length $h_t + 2$, and one path is of length $h_t + 6$.*

Lemma 7. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \le s \le t$. If $u[0] = v[0] = 1$, and $h_s(u,v) \ne 0$, then there exist $t+1$ internally disjoint paths $P_i(1 \le i \le t+1)$ between u and v such that $h_t + 1$ of them are of length $h_s + h_t + 2$, and $t - h_t$ paths are of length $h_s + h_t + 4$.*

Lemma 8. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$ and $h_s(u,v) = h_t(u,v) = 0$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that s of them are of length 7, and one path is of length 1.*

Lemma 9. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $h_s(u,v) \neq 0$ and $h_t(u,v) = 0$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that h_s of them are of length $h_s + 5$, $s - h_s$ paths are of length $h_s + 7$, and one path is of length $h_s + 1$.*

Lemma 10. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $h_s(u,v) = 0$ and $h_t(u,v) \geq s$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that s of them are of length $h_t + 5$, and one path is of length $h_t + 1$.*

Lemma 11. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $h_s(u,v) = 0$ and $1 \leq h_t(u,v) \leq s-1$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that h_t of them are of length $h_t + 5$, $s - h_t$ of them are of length $h_t + 7$, and one path is of length $h_t + 1$.*

Lemma 12. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $h_s(u,v) = s$ and $h_t(u,v) \geq s$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that s of them are of length $s + h_t + 3$, and one path is of length $s + h_t + 1$.*

Lemma 13. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $h_s(u,v) = s$, $1 \leq h_t(u,v) \leq s-1$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that $h_t + 1$ of them are of length $s + h_t + 3$, $s - h_t - 1$ of them are of length $s + h_t + 5$, and one path is of length $s + h_t + 1$.*

Lemma 14. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $1 \leq h_s(u,v) \leq s-1$ and $h_t(u,v) \geq s$, then there exist $s+1$ internally disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that $h_s + 2$ of them are of length $h_s + h_t + 3$ and $s - h_s - 1$ paths are of length $h_s + h_t + 5$.*

Lemma 15. *Let u, v be two distinct vertices of $EH(s,t)$ for $3 \leq s \leq t$. If $u[0] \neq v[0]$, $1 \leq h_s(u,v) \leq s-1$ and $1 \leq h_t(u,v) \leq s-1$, then there exist $s+1$ internally vertex-disjoint paths $P_i(1 \leq i \leq s+1)$ between u and v such that the following two cases are distinguished.*

1. *If $h_s + h_t \geq t+1$, then $t + s - h_s - h_t - 1$ of them are of length $h_s + h_t + 5$ and $h_s + h_t - t + 2$ paths are of length $h_s + h_t + 3$.*
2. *If $s \leq h_s + h_t \leq t$, then $s - 1$ of them are of length $h_s + h_t + 5$ and two paths are of length $h_s + h_t + 3$.*
3. *If $h_s + h_t \leq s - 1$, then $h_s + h_t + 1$ of them are of length $h_s + h_t + 5$, $s - h_s - h_t - 1$ of them are of length $h_s + h_t + 7$ and one path is of length $h_s + h_t + 3$.*

By above Lemmas 4 to 15, we can construct internally disjoint paths in the exchanged hypercubes $EH(s,t)$ for $3 \leq s \leq t$. By Theorems 1 and 2, the following corollary can be obtained.

Corollary 1. *The wide diameter of the exchanged hypercube* $EH(s,t)$ *is* $s+t+3$ *for* $3 \leq s \leq t$.

Acknowledgements. This work was supported in part by the National Science Council of the Republic of China under Contract NSC 100-2221-E-159-014, NSC 99-2221-E-009-084-MY3, and in part by the Aiming for the Top University and Elite Research Center Development Plan.

References

1. Chang, C.P., Sung, T.Y., Hsu, L.H.: Edge congestion and topological properties of crossed cubes. IEEE Trans. on Parallel and Distributed Systems 11(1), 64–80 (2000)
2. Hsu, L.H., Lin, C.K.: Graph Theory and Interconnection Networks. CRC Press (2008)
3. Latifi, S.: Combinatorial analysis of fault-diameter of the n-cube. IEEE Trans. on Computers 42(1), 27–33 (1993)
4. Leighton, F.T.: Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes. Morgan Kaufmann, San Mateo (1992)
5. Loh, P.K.K., Hsu, W.J., Pan, Y.: The exchanged hypercube. IEEE Trans. on Parallel and Distributed Systems 16(9), 866–874 (2005)
6. Ma, M.: The connectivity of exchanged hypercubes. Discrete Mathematics Algorithms and Applications 2(2), 213–220 (2010)
7. Ma, M., Liu, B.: Cycles embedding in exchanged hypercubes. Information Processing Letters 110(2), 71–76 (2009)
8. Ma, M., Zhu, L.: The super connectivity of exchanged hypercubes. Information Processing Letters 111(8), 360–364 (2011)
9. Saad, Y., Schultz, M.H.: Topological properties of hypercubes. IEEE Trans. on Computers 37(7), 867–872 (1988)
10. Xu, J.M.: Topological Structure and Analysis of Interconnection Networks. Kluwer Academic Publishers, Dordrecht (2001)

Ranking and Unranking Algorithms for Loopless Generation of Non-regular Trees[⋆]

Ro-Yu Wu[1], Jou-Ming Chang[2], An-Hang Chen[2], and Ming-Tat Ko[3]

[1] Department of Industrial Management, Lunghwa University of Science and Technology, Taoyuan, Taiwan, ROC
[2] Institute of Information and Decision Sciences, National Taipei College of Business, Taipei, Taiwan, ROC
[3] Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC

Abstract. A non-regular tree T with a prescribed branching sequence $(s_1, s_2, \ldots, s_n)$ is an ordered tree whose internal nodes are numbered from 1 to n in preorder such that every node i in T has s_i children. Recently, Wu et al. (2010) introduced a concise representation called RD-sequences to represent all non-regular trees and proposed a loopless algorithm for generating all non-regular trees in a Gray-code order. In this paper, based on such a Gray-code order, we present efficient ranking and unranking algorithms of non-regular trees with n internal nodes. Moreover, we show that each of the algorithms can be run in $\mathcal{O}(n^2)$ time provided a preprocessing takes $\mathcal{O}(n^2 S_{n-1})$ time and space in advance, where $S_{n-1} = \sum_{i=1}^{n-1}(s_i - 1)$.

Keywords: loopless algorithms, ranking algorithm, unranking algorithm, lexicographic order, Gray-code order, non-regular trees.

1 Introduction

A family of combinatorial objects is usually encoded by a set of integer sequences. For convenience, we say objects to mean their integer sequences. To efficiently generate such a set of objects, a generated list is urgently demanded in a Gray-code order, and so that the generating algorithm can be run in a constant time for each generation (i.e., a *loopless algorithm*), where the measure of computation is the total amount of data structure change and not the time required to output the objects. Accordingly, a loopless algorithm is implemented non-recursively by using, after the initialization of the first object, only assignment statements and "if-then-else" statements. The reader is referred to [1, 9, 5–8, 15, 17] for loopless generation of combinatorial objects. In particular, see [12] for an excellent survey of generating combinatorial objects in Gray-code orders.

Trees are among the most fundamental of combinatorial objects. A *rooted tree* is a tree with a distinguished root node. A rooted tree is said to be *ordered*

[⋆] This work was partially supported by the National Science Council of Taiwan under contracts NSC101-2221-E-262-020 and NSC101-2115-M-141-001.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 97–106.
DOI: 10.1007/978-3-642-35452-6_12

if the relative order of the subtrees is fixed. An ordered rooted tree is *regular* (respectively, *non-regular*) if the number of subtrees for internal nodes is unvaried (respectively, varied). Many algorithms have been developed for generating regular trees including binary trees and k-ary trees. By contrast, the study of generating non-regular trees has received less attention except for [4, 15, 20]. Recently, Wu et al. [15] dealt with the problem of generating a family of non-regular trees whose internal nodes have a pre-specified degree sequence (called branching sequence) in preorder. Also, a new type of integer sequences called RD-sequences was introduced to represent such a set of non-regular trees. With the help of a coding tree structure, they presented a loopless algorithm to generate RD-sequences of non-regular trees with n internal nodes in a Gray-code order using $4n + \mathcal{O}(1)$ space.

Given a specific order on a family of combinatorial objects, a *ranking algorithm* is a function that determines the rank of a given object in the generated list, and an *unranking algorithm* is one that finds the object of a given rank. Many ranking algorithms [2, 1, 3, 10, 11, 13, 16, 19] and unranking algorithms [2, 1, 3, 11, 16] for diverse representations of regular trees have been proposed. Note that all proposed algorithms mentioned above dealt with the rank of objects in lexicographic order except for [1] in a Gray-order. In addition, Wu et al. [14] dealt with the ranking and unranking problems of non-regular trees encoded by RD-sequences in lexicographic order and showed that, given a prescribed branching sequence $(s_1, s_2, \ldots, s_n)$, both ranking and unranking problems can be solved in $\mathcal{O}(nS_{n-1})$ time, where $S_{n-1} = \sum_{i=1}^{n-1}(s_i - 1)$.

In this paper, based on the Gray-code order presented in [15], we propose efficient ranking and unranking algorithms for dealing with non-regular trees encoded by RD-sequences. We show that each of the algorithms can be run in $\mathcal{O}(n^2)$ time, after a preprocessing stage taking $\mathcal{O}(n^2 S_{n-1})$ time and space.

2 Non-regular Trees and Their Loopless Generation

Let $S = (s_1, s_2, \ldots, s_n)$ be an integer sequence where $s_i \geqslant 2$ for $i = 1, 2, \ldots, n$, and let T be a non-regular tree with n internal nodes numbered from 1 to n in preorder (i.e., visiting the root and then recursively the subtrees of T from left to right). S is called the *branching sequence* for T if node i in T has s_i children for each $1 \leqslant i \leqslant n$. For briefness, we say non-regular trees to mean the set of trees having a given branching sequence and let $\mathscr{T}_S$ denote the set consisting of all non-regular trees with S as their branching sequence. In [15], Wu et al. defined a concise representation, called *right-distance sequences* (abbreviated as RD-sequences), to describe non-regular trees. For each tree $T \in \mathscr{T}_S$, the *right-distance* of an internal node $i \in T$, denoted by d_i, is given as follows:

$$d_i = \begin{cases} 0 & \text{if } i = 1 \text{ (i.e., the root);} \\ d_{p(i)} + s_{p(i)} - k & \text{otherwise,} \end{cases}$$

where $p(i)$ is the parent of node i and k is the rank of node i among all sons of $p(i)$. Then, T can be represented by the sequence $rd(T) = (d_1, d_2, \ldots, d_n)$. Moreover, Wu et al. [15] gave the following characterization of RD-sequences: for a given branching sequence $S = (s_1, s_2, \ldots, s_n)$, an integer sequence $(d_1, d_2, \ldots, d_n)$ is the RD-sequence of a tree $T \in \mathscr{T}_S$ if and only if $d_1 = 0$ and $0 \leqslant d_i \leqslant d_{i-1} + s_{i-1} - 1$ for $i = 2, 3, \ldots, n$. Thus, the number of all possible non-regular trees with respect to a branching sequence S can be determined by the following lemma.

Lemma 1. [15] *Let $d_0 = 0$ and $s_0 = 1$. Given a branching sequence $S = (s_1, s_2, \ldots, s_n)$, the cardinality of $\mathscr{T}_S$ is as follows:*

$$|\mathscr{T}_S| = \sum_{d_1=0}^{d_0+s_0-1} \sum_{d_2=0}^{d_1+s_1-1} \cdots \sum_{d_n=0}^{d_{n-1}+s_{n-1}-1} 1 \qquad \textit{for } n \geqslant 1.$$

In particular, every sequence $(d_1, d_2, \ldots, d_n)$ in the above formula is an RD-sequence representing a tree in $\mathscr{T}_S$.

For example, $|\mathscr{T}_{(3,4,2)}| = \sum_{d_1=0}^{0+1-1} \sum_{d_2=0}^{d_1+3-1} \sum_{d_3=0}^{d_2+4-1} 1 = \sum_{d_2=0}^{2}(d_2 + 4) = 4 + 5 + 6 = 15$ and all non-regular trees with branching sequence $(3, 4, 2)$ are shown in Fig. 1.

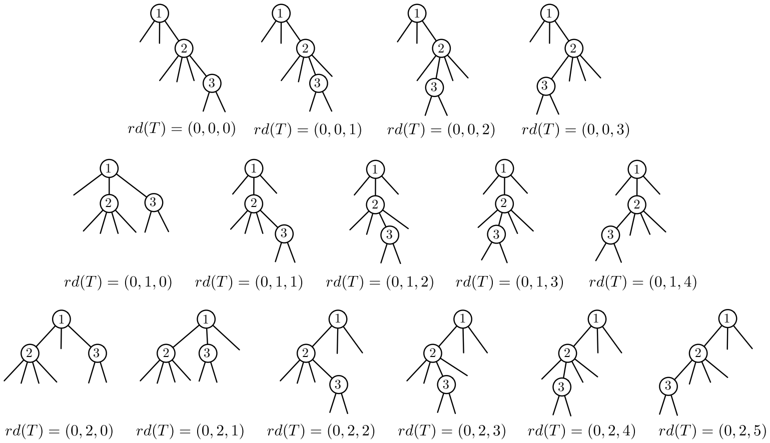


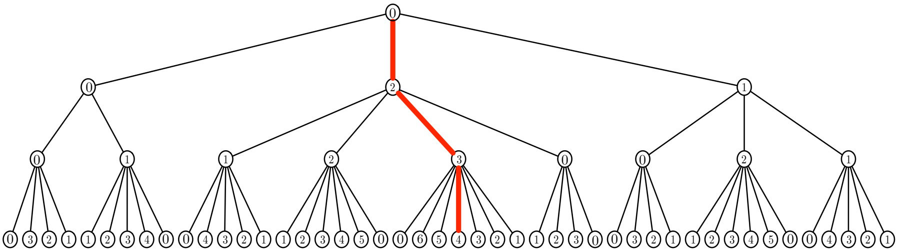
Fig. 1. Non-regular trees with branching sequence $(3, 4, 2)$

In what follows, we give the concept of the loopless Gray-code generation for non-regular trees proposed in [15]. To describe all non-regular trees in a systematic way, a coding tree called *flip-flap tree* was introduced. Given a branching sequence $S = (s_1, s_2, \ldots, s_n)$, a flip-flap tree $\mathbb{T}_S$ is a rooted labeled tree with n levels constructed by the follows rules:

(a) The first level contains the only one node (i.e., the root) with label 0.
(b) Every node with label d in the ith level, $i < n$, has $s_i + d$ sons labeled by $0, 1, \ldots, s_i + d - 1$ in the next level, where the sons are arranged in either $1, 2, \ldots, s_i + d - 1, 0$ (i.e., an up fragment) or $0, s_i + d - 1, \ldots, 2, 1$ (i.e., a down fragment).

(c) If the sons of a node x with label d are arranged in an up fragment, then the sons of its adjacency siblings (i.e., nodes with labels $d-1$ and $d+1$ in the same level as x), if exist, are arranged in a down fragment, and vice versa.

Accordingly, nodes on the boundary in every fragment are labeled by either 0 or 1. And, the full labels along a path from the root to a leaf in $\mathbb{T}_S$ indicate the RD-sequence of a non-regular tree. Moreover, any two adjacent RD-sequences in the flip-flap tree differ in exactly one digit. Therefore, this provides the base for designing a loopless Gray-code generation algorithm. For example, Fig. 2 shows a flip-flap tree with respect to the branching sequence $S = (3, 2, 4, 3)$, where the first fragment in each level is a down fragment. Thus, the first RD-sequence appeared in $\mathbb{T}_{(3,2,4,3)}$ is $(0, 0, 0, 0)$. Also, a path with bold lines from the root to a leaf in $\mathbb{T}_{(3,2,4,3)}$ represents the non-regular trees with RD-sequence $(0, 2, 3, 4)$. In [15], a procedure `NextRD()` written by `C++` was designed for generating the Gray-code order of RD-sequences.



0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 2 2 2 2 2 2 1 1 1 1 1

0 3 2 1 1 2 3 4 0 0 4 3 2 1 1 2 3 4 5 0 0 6 5 4 3 2 1 1 2 3 0 0 3 2 1 1 2 3 4 5 0 0 4 3 2 1

Fig. 2. A flip-flap tree $\mathbb{T}_{(3,2,4,3)}$

3 Ranking Algorithm

Let $S = (s_1, s_2, \ldots, s_n)$ be the prescribed branching sequence. For $1 \leqslant i \leqslant j \leqslant n$, let $S_{i,j} = \sum_{k=i}^{j}(s_k - 1)$, and simply write S_j instead of $S_{1,j}$. By convenience, we define $S_0 = 0$. We first observe that, for $i = 1, 2, \ldots, n$, the largest label for nodes in the ith level of $\mathbb{T}_S$ is exactly S_{i-1}. For instance, the largest labels are $S_0 = 0$, $S_1 = 3 - 1 = 2$, $S_2 = S_1 + (2 - 1) = 3$ and $S_3 = S_2 + (4 - 1) = 6$, respectively, in each level of the flip-flap tree shown in Figure 2. Next, we observe that if two nodes have the same label in the same level of $\mathbb{T}_S$, then they have the same degree (i.e., the number of children). This further implies that the numbers of leaves in the subtrees rooted at each of the two nodes are the same.

For $1 \leqslant \ell \leqslant n$, let $\mathbb{T}_S^{\ell}$ be the partial tree of $\mathbb{T}_S$ which is formed by the top ℓ levels. Thus, $\mathbb{T}_S^1$ consists of a rooted node and $\mathbb{T}_S^n = \mathbb{T}_S$. For $1 \leqslant i \leqslant \ell \leqslant n$, let $A_{i,k}^{\ell}$ denote the number of leaves in the subtree rooted at a node with label k

in the ith level of $\mathbb{T}_S^\ell$. Obviously, $A^\ell_{\ell,k} = 1$ for $1 \leqslant \ell \leqslant n$ and $0 \leqslant k \leqslant S_{\ell-1}$ and $A^\ell_{\ell-1,k} = s_{\ell-1} + k$ for $2 \leqslant \ell \leqslant n$ and $0 \leqslant k \leqslant S_{\ell-2}$. In general, we have

$$A^\ell_{i,k} = \sum_{j=0}^{s_i+k-1} A^\ell_{i+1,j} \tag{1}$$

where $1 \leqslant i < \ell \leqslant n$ and $0 \leqslant k \leqslant S_{i-1}$. For instance, if we consider the flip-flap tree $\mathbb{T}_{(3,2,4,3)}$, we can build the following tables for calculating $A^\ell_{i,k}$.

$A^4_{i,k}$		k						
		0	1	2	3	4	5	6
i	1	46						
	2	9	15	22				
	3	4	5	6	7			
	4	1	1	1	1	1	1	1

$A^3_{i,k}$		k			
		0	1	2	3
i	1	9			
	2	2	3	4	
	3	1	1	1	1

$A^2_{i,k}$		k		
		0	1	2
i	1	3		
	2	1	1	1

For the efficiency of computation, we define the following formula:

$$B^\ell_{i,k} = \sum_{j=0}^{k} A^\ell_{i,j} \tag{2}$$

where $1 \leqslant i \leqslant \ell \leqslant n$ and $0 \leqslant k \leqslant S_{i-1}$. That is, the term $B^\ell_{i,k}$ indicates the total number of leaves for those subtrees rooted at nodes with labels from 0 to k in the ith level of $\mathbb{T}^\ell_S$. Hence, a table built from Eq. (2) is named as the *accumulation table* with respect to $\mathbb{T}_S$. For instance, accumulation tables with respect to the partial trees of $\mathbb{T}_{(3,2,4,3)}$ are shown below.

$B^4_{i,k}$		k						
		0	1	2	3	4	5	6
i	1	46						
	2	9	24	46				
	3	4	9	15	22			
	4	1	2	3	4	5	6	7

$B^3_{i,k}$		k			
		0	1	2	3
i	1	9			
	2	2	5	9	
	3	1	2	3	4

$B^2_{i,k}$		k		
		0	1	2
i	1	3		
	2	1	2	3

Obviously, the time and space required for building accumulation tables are

$$\sum_{i=1}^{n}(S_{i-1}+1)i = \sum_{i=0}^{n-1}(S_i+1)(i+1) = \mathcal{O}(n^2 S_{n-1}).$$

We are now in a position to determine the rank of a non-regular tree in the Gray-code order of [15]. For a given non-regular tree $T \in \mathscr{T}_S$ associated with the RD-sequence $rd(T) = (d_1, d_2, \ldots, d_n)$, let x_i denote the node with label d_i in $\mathbb{T}_S$ such that the full labels along the path $x_1, x_2, \ldots, x_n$ represent the given RD-sequence. For each $i = 1, 2, \ldots, n$, let $R(i)$ be the rank of x_i in the ith level of $\mathbb{T}_S$.

Hereafter, the rank of a node in each level of $\mathbb{T}_S$ is ordered from left to right and a ranking always starts with 0. Note that $R(1) = 0$ and our ranking algorithm is to obtain $R(n)$. Since each level of $\mathbb{T}_S$ begins with a down fragment and the two types of fragments appear alternately, it is easy to check the following property.

Proposition 1. *For $1 \leqslant i < n$, if the rank of a node in the ith level of $\mathbb{T}_S$ is even (respectively, odd), then its sons are arranged in a down fragment (respectively, an up fragment).*

Initially, we set $R(i) = 0$ for $i = 1, \ldots, n$. Then, the computation of $R(i)$ requires $i - 1$ updates by means of Eqs. (3) and (4). For $i = 1, \ldots, n-1$, let $j = i + 1$. If x_j is not the first node in a fragment, let y_j be the node with rank $R(j) - 1$ in the j-th level of $\mathbb{T}_S$. Then, for each $h = j, j+1, \ldots, n$, we update $R(h)$ to represent the rank of the rightmost descendant of y_j in the h-th level of $\mathbb{T}_S$. By Proposition 1, there are two cases to compute $R(h)$. If $R(i) \equiv 0 \pmod 2$, the sons of x_i are arranged in a down fragment. In this case, we have

$$R(h) = R(h) + (B^h_{j,d_i+(s_i-1)} - B^h_{j,d_j} + B^h_{j,0}). \tag{3}$$

On the other hand, if $R(i) \equiv 1 \pmod 2$, the sons of x_i are arranged in an up fragment. In this case, we have

$$R(h) = R(h) + (B^h_{j,d_j-1} - B^h_{j,0}). \tag{4}$$

The last term of the right hand side in each of Eqs. (3) and (4) is called the *incremental change* of $R(h)$ in the update. The meanings of incremental changes are illustrated in Fig. 3 and Fig. 4, respectively. The details of the ranking algorithm is shown below.

Function Ranking$(d_1, d_2, \ldots, d_n)$

```
begin
    for $i = 1$ to $n$ do  $R(i) = 0$;
    for $i = 1$ to $n - 1$ do
        $j = i + 1$;
        if $R(i) \equiv 0 \pmod 2$ then  // a down fragment
            if $d_j \neq 0$ then
                for $h = j$ to $n$ do
                    $R(h) = R(h) + (B^h_{j,d_i+(s_i-1)} - B^h_{j,d_j} + B^h_{j,0})$;
        else  // an up fragment
            if $d_j \neq 1$ then
                for $h = j$ to $n$ do
                    $R(h) = R(h) + (B^h_{j,d_j-1} - B^h_{j,0})$;
    return $R(n)$;
```

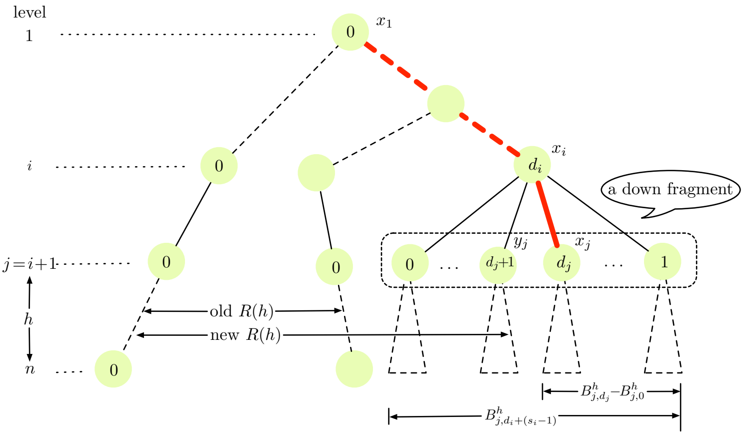


Fig. 3. Illustration of Eq. (3)

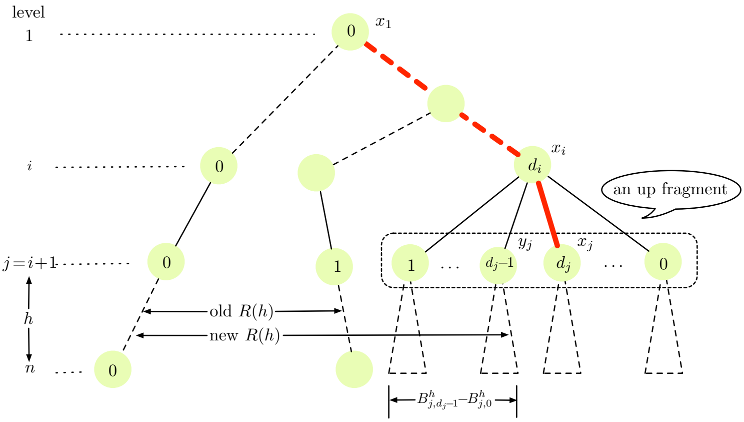


Fig. 4. Illustration of Eq. (4)

Example 1. We consider a non-regular tree $T \in \mathscr{T}_{(3,2,4,3)}$ with $rd(T) = (0, 2, 3, 4)$ and perform Ranking$(0, 2, 3, 4)$. Initially, $R(1) = R(2) = R(3) = R(4) = 0$. When $i = 1$, since $R(1) = 0$ is even, the sons of x_1 are arranged in a down fragment. Thus, we have

$R(2) = R(2) + B^2_{2,2} - B^2_{2,2} + B^2_{2,0} = 0 + 3 - 3 + 1 = 1,$
$R(3) = R(3) + B^3_{2,2} - B^3_{2,2} + B^3_{2,0} = 0 + 9 - 9 + 2 = 2,$
$R(4) = R(4) + B^4_{2,2} - B^4_{2,2} + B^4_{2,0} = 0 + 46 - 46 + 9 = 9.$

When $i = 2$, since $R(2) = 1$ is odd, the sons of x_2 are arranged in an up fragment. Then, we have the following updates:

$R(3) = R(3) + B^3_{3,2} - B^3_{3,0} = 2 + 3 - 1 = 4,$
$R(4) = R(4) + B^4_{3,2} - B^4_{3,0} = 9 + 15 - 4 = 20.$

When $i = 3$, since $R(3) = 6$ is even, the sons of x_3 are arranged in a down fragment. We obtain

$R(4)=R(4)+B^4_{4,6}-B^4_{4,4}+B^4_{4,0}=20+7-5+1=23.$
As a result, the algorithm outputs $R(4) = 23$. □

Since the time complexity and space requirement for building accumulation tables are both $\mathcal{O}(n^2S_{n-1})$ and the ranking algorithm can be run in $\mathcal{O}(n^2)$ time, we conclude the following.

Theorem 1. *Determining the rank of a non-regular tree T with a prescribed branching sequence $(s_1, s_2, \ldots, s_n)$ in a Gray-code order can be done in $\mathcal{O}(n^2S_{n-1})$ time and space, where $S_{n-1} = \sum_{i=1}^{n-1}(s_i - 1)$.*

4 Unranking Algorithm

In this section, we provide a reverse function called Unranking to convert a positive integer N to its corresponding RD-sequence of a non-regular tree with a branching sequence $(s_1, s_2, \ldots, s_n)$. The basic ideal of the function is to decompose N into a sequence incremental changes of $R(n)$. Initially, we set $R(i) = 0$ for all $i = 1, \ldots, n$. Then, we first determine the largest integer i on which $B^n_{i,0} > N$ and set $d_j = 0$ for all $1 \leqslant j \leqslant i$. For each round of the main loop, we let $j = i+1$. According to the parity of $R(i)$, we check that the arrangement of the sons of x_i is a down fragment or an up fragment. Then, we determine a relevant term k that can be used for calculating the current incremental change. For the case of a down fragment (respectively, an up fragment), if x_j is the leftmost child (respectively, the rightmost child) of x_i, then we set $d_j = 0$. Otherwise, we update N by subtracting the incremental change of $R(n)$ from N, and in the meanwhile the term $d_j = k$ (respectively, $d_j = k + 1$) is determined. The details of Unranking is shown below.

Example 2. Suppose we are given integers $N = 37$ and the branching sequence $S = (3, 2, 4, 3)$. Initially, $R(1) = R(2) = R(3) = R(4) = 0$. Since $B^4_{2,0} = 9 < N < B^4_{1,0} = 46$, we have $d_1 = 0$ and $i = 1$ before entering the main loop with condition $i \neq n$. There are three rounds to compute in the main loop. When $i = 1$ (i.e., $j = 2$), since $R(1) = 0$ is even, the sons of x_1 are arranged in a down fragment. We can check $N = 37 \not< B^4_{2,2} - B^4_{2,1} + B^4_{2,0} = 46 - 24 + 9 = 31$ and $k \neq d_1 + s_1 = 0 + 3$ (when $k = 1$). Thus, $d_2 = k = 1$ and $N = 37 - 31 = 6$. Moreover, we have
$R(2) = R(2) + B^2_{2,2} - B^2_{2,1} + B^2_{2,0} = 0 + 3 - 2 + 1 = 2,$
$R(3) = R(3) + B^3_{2,2} - B^3_{2,1} + B^3_{2,0} = 0 + 9 - 5 + 2 = 6,$
$R(4) = R(4) + B^4_{2,2} - B^4_{2,1} + B^4_{2,0} = 0 + 46 - 24 + 9 = 31.$
In this case, the incremental change of $R(4)$ is 31.

When $i = 2$ (i.e., $j = 3$), since $R(2) = 2$ is even, the sons of x_2 are arranged in a down fragment. We can check $N = 6 < B^4_{3,2} - B^4_{3,1} + B^4_{3,0} = 15 - 9 + 4 = 10$ and $k \neq d_2 + s_2 = 1 + 2$ (when $k = 1$). Also, $N = 6 \not< B^4_{3,2} - B^4_{3,2} + B^4_{3,0} = 15 - 15 + 4 = 4$ and $k \neq d_2 + s_2 = 1 + 2$ (when $k = 2$). Thus, $d_3 = k = 2$ and $N = 6 - 4 = 2$. Moreover, we have

$R(3) = R(3) + B^3_{3,2} - B^3_{3,2} + B^3_{3,0} = 6 + 3 - 3 + 1 = 7,$
$R(4) = R(4) + B^4_{3,2} - B^4_{3,2} + B^4_{3,0} = 31 + 15 - 15 + 4 = 35.$
In this case, the incremental change of $R(4)$ is 4.

When $i = 3$ (i.e., $j = 4$), since $R(3) = 7$ is odd, the sons of x_3 are arranged in an up fragment. We can check
$N = 2 < B^4_{4,5} - B^4_{4,0} = 6 - 1 = 5$ (when $k = d_3 + (s_3 - 1) = 2 + 3 = 5$);
$N = 2 < B^4_{4,4} - B^4_{4,0} = 5 - 1 = 4$ (when $k = 4$);
$N = 2 < B^4_{4,3} - B^4_{4,0} = 4 - 1 = 3$ (when $k = 3$);
$N = 2 \not< B^4_{4,2} - B^4_{4,0} = 3 - 1 = 2$ (when $k = 2$).
Thus, $N = 2 - 2 = 0$ and $d_4 = k + 1 = 3$. Moreover, we have
$R(4) = R(4) + B^4_{4,2} - B^4_{4,0} = 35 + 3 - 1 = 37.$
In this case, the incremental change of $R(4)$ is $3 - 1 = 2$. When $i = 4$, the main loop stops the running and the function outputs $0, 1, 2, 3$. □

```
Function Unranking($N$)
begin
    for $i = 1$ to $n$ do  $R(i) = d_i = 0$;
    $i = n$;
    while $B^n_{i,0} \leqslant N$ do  $i = i - 1$;
    while $i \neq n$ do
        $j = i + 1$;
        if $R(i) \equiv 0$ (mod) 2 then        // a down fragment
            $k = 1$;
            while $N < B^n_{j,d_i+(s_i-1)} - B^n_{j,k} + B^n_{j,0}$ and $k \neq d_i + s_i$ do  $k = k + 1$;
            if $k = d_i + s_i$ then        // $x_j$ is the leftmost child of $x_i$
                $d_j = 0$;
            else
                $d_j = k$;
                $N = N - (B^n_{j,d_i+(s_i-1)} - B^n_{j,k} + B^n_{j,0})$;
                for $h = j$ to $n$ do
                    $R(h) = R(h) + (B^h_{j,d_i+(s_i-1)} - B^h_{j,k} + B^h_{j,0})$;
        else                             // an up fragment
            $k = d_i + (s_i - 1)$;
            while $N < B^n_{j,k} - B^n_{j,0}$ do  $k = k - 1$;
            $N = N - (B^n_{j,k} - B^n_{j,0})$;
            if $k = d_i + (s_i - 1)$ then     // $x_j$ is the rightmost child of $x_i$
                $d_j = 0$;
            else
                $d_j = k + 1$;
            for $h = j$ to $n$ do
                $R(h) = R(h) + (B^h_{j,k} - B^h_{j,0})$;
        $i = i + 1$;
    return $d_1, d_2, \ldots, d_n$;
```

Theorem 2. *Let N be a positive integer and $S = (s_1, s_2, \ldots, s_n)$ a branching sequence. Determining the RD-sequence of a non-regular tree $T \in \mathscr{T}_S$ such that Rank$(T) = N$ in a Gray-code order can be done in $\mathcal{O}(n^2 S_{n-1})$ time and space, where $S_{n-1} = \sum_{i=1}^{n-1}(s_i - 1)$.*

References

1. Ahmadi-Adl, A., Nowzari-Dalini, A., Ahrabian, H.: Ranking and unranking algorithms for loopless generation of t-ary trees. Logic J. IGPL 19, 33–43 (2011)
2. Ahrabian, H., Nowzari-Dalini, A.: Generation of t-ary trees with ballot-sequences. Int. J. Comput. Math. 80, 1243–1249 (2003)
3. Er, M.C.: Lexicographic listing and ranking t-ary trees. Comput. J. 30, 569–572 (1987)
4. Er, M.C.: A simple algorithm for generating non-regular trees in lexicographic order. Comput. J. 31, 61–64 (1988)
5. Korsh, J.F.: Loopless generation of k-ary tree sequences. Inform. Process. Lett. 52, 243–247 (1994)
6. Korsh, J.F., LaFollette, P.: Loopless generation of Gray codes for k-ary trees. Inform. Process. Lett. 70, 7–11 (1999)
7. Korsh, J.F., LaFollette, P.: A loopless Gray code for rooted trees. ACM Trans. Algorithms 2, 135–152 (2006)
8. Korsh, J.F., Lipschutz, S.: Shift and loopless generation of k-ary trees. Inform. Process. Lett. 65, 235–240 (1998)
9. Roelants van Baronaigien, D.: A loopless Gray-code algorithm for listing k-ary trees. J. Algorithms 35, 100–107 (2000)
10. Roelants van Baronaigien, D., Ruskey, F.: Generating t-ary trees in A-order. Inform. Process. Lett. 27, 205–213 (1988)
11. Ruskey, F.: Generating t-ary trees lexicographically. SIAM J. Comput. 7, 424–439 (1978)
12. Savage, C.D.: A survey of combinatorial Gray codes. SIAM Review 39, 605–629 (1997)
13. Trojanowaki, A.E.: Ranking and listing algorithms for k-ary trees. SIAM J. Comput. 7, 492–509 (1978)
14. Wu, R.-Y., Chang, J.-M., Chang, C.-H.: Ranking and unranking of non-regular trees with a prescribed branching sequence. Math. Comput. Modelling 53, 1331–1335 (2011)
15. Wu, R.-Y., Chang, J.-M., Wang, Y.-L.: Loopless Generation of non-regular trees with a prescribed branching sequence. Comput. J. 53, 661–666 (2010)
16. Wu, R.-Y., Chang, J.-M., Wang, Y.-L.: Ranking and unranking of t-ary trees using RD-sequences. IEICE Trans. Inform. Sys. E94-D, 226–232 (2011)
17. Xiang, L., Ushijima, K., Tang, C.: Efficient loopless generation of Gray codes for k-ary trees. Inform. Process. Lett. 76, 169–174 (2000)
18. Zaks, S.: Lexicographic generation of ordered trees. Theore. Comput. Sci. 10, 63–82 (1980)
19. Zaks, S.: Generation and ranking of k-ary trees. Inform. Process. Lett. 14, 44–48 (1982)
20. Zaks, S., Richards, D.: Generating trees and other combinatorial objects lexicographically. SIAM J. Comput. 81, 73–81 (1979)

Completely Independent Spanning Trees on Complete Graphs, Complete Bipartite Graphs and Complete Tripartite Graphs*

Kung-Jui Pai[1], Shyue-Ming Tang[2], Jou-Ming Chang[3], and Jinn-Shyong Yang[3]

[1] Department of Industrial Engineering and Management,
Ming Chi University of Technology, New Taipei City, Taiwan, ROC
[2] Fu Hsing Kang School, National Defense University, Taipei, Taiwan, ROC
[3] Institute of Information and Decision Sciences, National Taipei College of Business,
Taipei, Taiwan, ROC

Abstract. Let T_1, T_2,..., T_k be spanning trees in a graph G. If for any two vertices x, y of G, the paths from x to y in T_1, T_2,..., T_k are vertex-disjoint except end vertices x and y, then T_1, T_2,..., T_k are called completely independent spanning trees in G. In 2001, Hasunuma gave a conjecture that there are k completely independent spanning trees in any $2k$-connected graph. Péterfalvi disproved the conjecture in 2012. In this paper, we shall prove that there are $\lfloor \frac{n}{2} \rfloor$ completely independent spanning trees in a complete graph with $n(\geqslant 4)$ vertices. Then, we prove that there are $\lfloor \frac{n}{2} \rfloor$ completely independent spanning trees in a complete bipartite graph $K_{m,n}$ where $m \geqslant n \geqslant 4$. Next, we also prove that there are $\lfloor \frac{n_1+n_2}{2} \rfloor$ completely independent spanning trees in a complete tripartite graph K_{n_3,n_2,n_1} where $n_3 \geqslant n_2 \geqslant n_1$ and $n_1 + n_2 \geqslant 4$. As a result, the Hasunuma's conjecture holds for complete graphs and complete m-partite graphs where $m \in \{2, 3\}$.

Keywords: completely independent spanning trees, edge-disjoint spanning trees, complete graphs, complete bipartite graphs, complete tripartite graphs.

1 Introduction

Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. Two paths P_1 and P_2 between vertices x and y are called *openly disjoint* if they are vertex-disjoint apart from their end vertices. Let T_1, T_2,..., T_k be spanning trees of G. If for any two vertices x, y of G, the paths from x to y in T_1, T_2,..., T_k are pairwise openly disjoint, then T_1, T_2,..., T_k are called *completely independent spanning trees* (abbreviated as CISTs) in G.

The CIST problem can be applied on the fault-tolerant broadcasting problem in interconnection networks. The CIST problem was first introduced by Hasunuma [1], and he also posed the following conjecture:

* This research was partially supported by National Science Council under the Grants NSC100-2628-E-141-001-MY2, NSC101-2221-E-131-039 and NSC101-2115-M-141-001.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 107–113.
DOI: 10.1007/978-3-642-35452-6_13

Conjecture 1. [2] There are k CISTs in any $2k$-connected graph.

Hasunuma [2] also showed that recognizing whether there exist two CISTs in an arbitrary graph G is NP-complete. Thus, related investigations tended to study validation of Conjecture 1 for certain families of graphs [1–3]. For any k-connected line graph $L(G)$, there are k CISTs in the underlying undirected graph of $L(G)$[1]. There are two CISTs in any 4-connected maximal plane graph[2]. There are two CISTs in the Cartesian product of any 2-connected graph[3]. In 2012, Péterfalvi gave a counterexample [4] to disprove Conjecture 1. He constructed a k-connected graph which does not contain two CISTs for each $k \geqslant 2$. This raises a new question that Conjecture 1 is affirmative for which families of graphs.

In this paper, we show that there are $\lfloor \frac{n}{2} \rfloor$ CISTs in a complete graph with $n(\geqslant 4)$ vertices. Then, we prove that there are $\lfloor \frac{n}{2} \rfloor$ CISTs in a complete bipartite graph $K_{m,n}$ where $m \geqslant n \geqslant 4$. Next, we also prove that there are $\lfloor \frac{n_1+n_2}{2} \rfloor$ CISTs in a complete tripartite graph K_{n_3,n_2,n_1} where $n_3 \geqslant n_2 \geqslant n_1$ and $n_1 + n_2 \geqslant 4$. As a result, we prove that Conjecture 1 holds for complete graphs and complete m-partite graphs where $m \in \{2, 3\}$.

2 Completely Independent Spanning Trees in Complete Graphs

A *tree* is a connected and acyclic graph. A *leaf* of a tree is a node with degree 1. A *complete graph* with n vertices, denoted by K_n, is a graph in which each pair of vertices is connected by an edge. Theorem 1 describes an important property in CISTs.

Theorem 1. [1] *$T_1, T_2, \ldots, T_k$ are CISTs in a graph G if and only if they are edge-disjoint spanning trees and for any $v \in V(G)$, there is at most one T_i such that v is not a leaf in T_i.*

Theorem 2. *There are $\lfloor \frac{n}{2} \rfloor$ CISTs in K_n for all $n \geqslant 4$.*

Proof. Let $k = \lfloor \frac{n}{2} \rfloor$. If n is even (respectively, odd), then $n = 2k$ (respectively, $n = 2k + 1$). Suppose that the vertices of K_n are labeled with $1, 2, \ldots, 2k$ for n even and are $0, 1, 2, \ldots, 2k$ for n odd. For the case that n is even, there are only two non-leaf nodes i and $k+i$ in T_i for $1 \leqslant i \leqslant k$. First, vertex i is adjacent to vertex $k+i$. Each leaf x is adjacent to vertex i for $i+1 \leqslant x \leqslant k+i-1$, and each leaf y is adjacent to vertex $k+i$ for $1 \leqslant y \leqslant i-1$ or $k+i+1 \leqslant y \leqslant 2k$. We note that edges $(1, i)$, $(2, i)$,..., $(i-1, i)$ (respectively, $(k+1, k+i)$, $(k+2, k+i)$,..., $(k+i-1, k+i)$) are used in $T_1, T_2, \ldots, T_{i-1}$, so vertices $1, 2, \ldots, i-1$(respectively, $k+1$, $k+2$,..., $k+i-1$) are adjacent to vertices $k+i$ (respectively, i) in T_i. For the case that n is odd, the trees are similar to the ones in the case that n is even. We let additional vertex 0 be adjacent to vertex $k+i$ in T_i for all $1 \leqslant i \leqslant k$.

Clearly, $T_1, T_2, \ldots, T_k$ are edge-disjoint spanning trees and, for $1 \leqslant i \leqslant k$, T_i contains two non-leaf nodes i and $k+i$. Thus, by Theorem 1, $T_1, T_2, \ldots, T_k$ are CISTs in K_n. These k CISTs are depicted in Figure 1, where we omit vertices 0 and its incident edges if n is even. □

3 Completely Independent Spanning Trees in Complete Bipartite Graphs

A complete bipartite graph, denoted by $K_{m,n}$, is a graph whose vertex set can be partitioned into two nonempty subsets V_1 and V_2 such that $|V_1| = m$, $|V_2| = n$, and two vertices a and b are adjacent if and only if $a \in V_1$ and $b \in V_2$.

Theorem 3. *There are $\lfloor \frac{n}{2} \rfloor$ CISTs in $K_{m,n}$ for all $m \geqslant n \geqslant 4$.*

Proof. Let $k = \lfloor \frac{n}{2} \rfloor$. If n is odd (respectively, even), then $n = 2k+1$ (respectively, $n = 2k$). Suppose that the m vertices of V_1 are labeled with $a_1, a_2, \ldots, a_m$ and the n vertices of V_2 are labeled with $b_1, b_2, \ldots, b_{2k}$ (respectively, $b_0, b_1, b_2, \ldots, b_{2k}$) for n even (respectively, odd). For the case that n is even, there are only four non-leaf nodes a_i, b_i, a_{k+i} and b_{k+i} in T_i for $1 \leqslant i \leqslant k$. First, edges (a_i, b_i), (a_i, b_{k+i}) and (a_{k+i}, b_i) are added in T_i. Each leaf a_x is adjacent to vertex b_i for $i+1 \leqslant x \leqslant k+i-1$, and each leaf a_y is adjacent to vertex b_{k+i} for $1 \leqslant y \leqslant i-1$ or $k+i+1 \leqslant y \leqslant m$. Then, each leaf b_x is adjacent to vertex a_i for $i+1 \leqslant x \leqslant k+i-1$, and each leaf b_y is adjacent to vertex a_{k+i} for $1 \leqslant y \leqslant i-1$ or $k+i+1 \leqslant y \leqslant 2k$. We note that edges (a_1, b_i), (a_2, b_i),..., (a_{i-1}, b_i) (respectively, (b_1, a_i), (b_2, a_i),..., (b_{i-1}, a_i)) are used in $T_1, T_2, \ldots, T_{i-1}$, so vertices $a_1, a_2, \ldots, a_{i-1}$(respectively, $b_1, b_2, \ldots, b_{i-1}$) are adjacent to vertices b_{k+i} (respectively, a_{k+i}) in T_i. For the case that n is odd, the trees are similar to the ones in the case that n is even. We let additional vertex b_0 be adjacent to vertex a_{k+i} in T_i for all $1 \leqslant i \leqslant k$. Clearly, $T_1, T_2, \ldots, T_k$ are edge-disjoint spanning trees and, for $1 \leqslant i \leqslant k$, T_i contains four non-leaf nodes a_i, b_i, a_{k+i} and b_{k+i}. Thus, by Theorem 1, $T_1, T_2, \ldots, T_k$ are CISTs. These k CISTs are depicted in Figure 2, where we omit vertices b_0 and their incident edges if n is even. □

4 Completely Independent Spanning Trees in Complete Tripartite Graphs

A complete tripartite graph, denoted by K_{n_3,n_2,n_1}, is a graph whose vertex set can be partitioned into three nonempty subsets V_1, V_2 and V_3 such that $|V_1| = n_1$, $|V_2| = n_2$, $|V_3| = n_3$. No two vertices in the same set are adjacent and every vertex of each set is adjacent to every vertex in the other two sets.

Theorem 4. *There are $\lfloor \frac{n_2+n_1}{2} \rfloor$ CISTs in K_{n_3,n_2,n_1} for all $n_3 \geqslant n_2 \geqslant n_1$ and $n_1 + n_2 \geqslant 4$.*

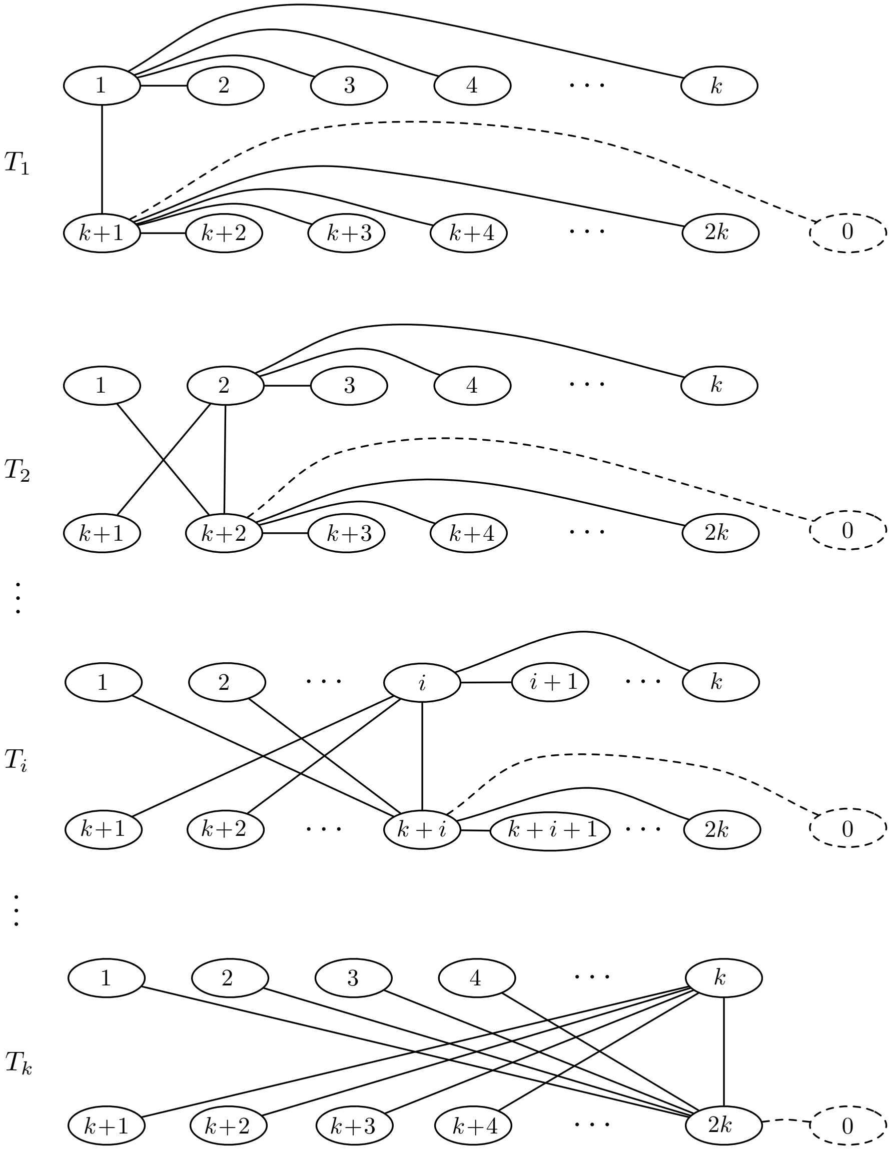


Fig. 1. CISTs of K_n. Dotted lines and dotted vertices are only for the case that n is odd.

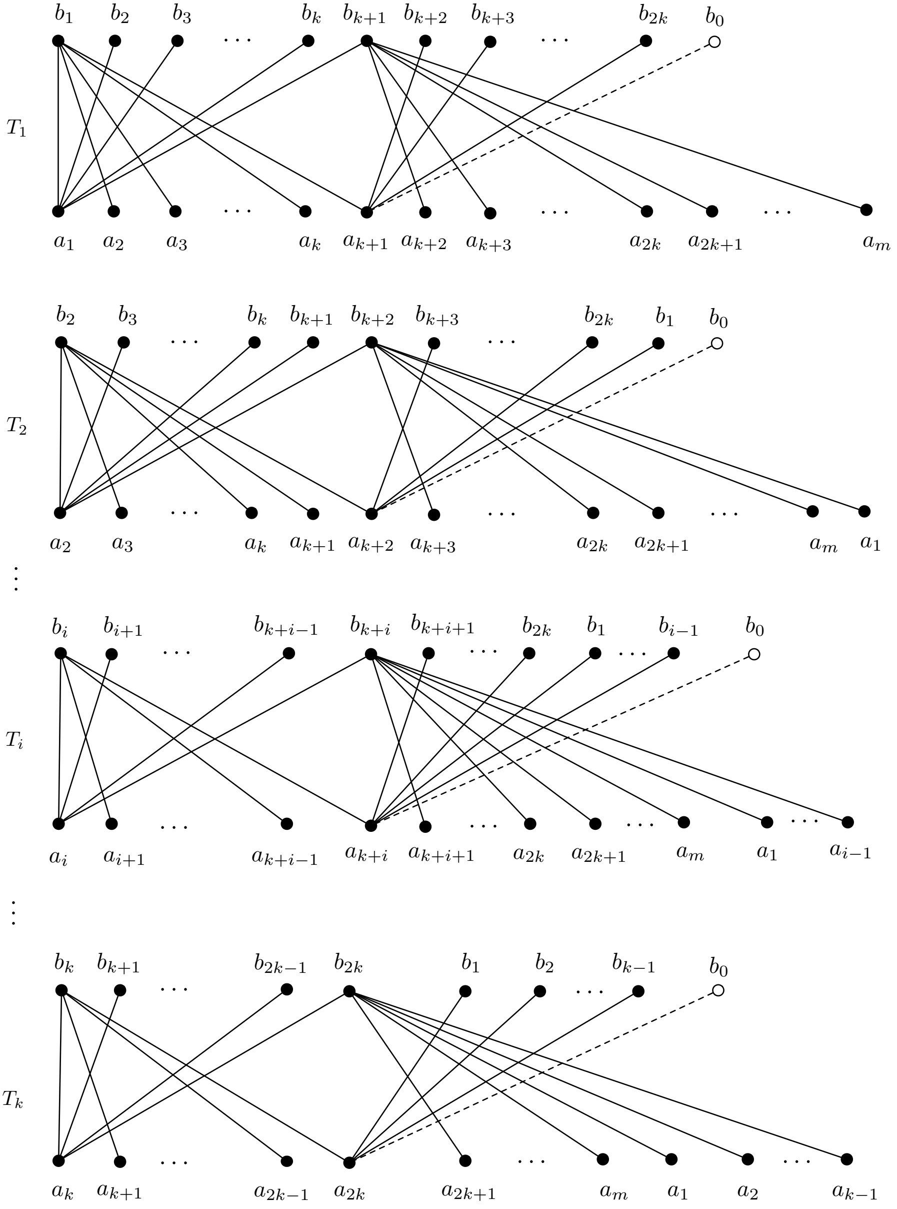

T_1
b_1 b_2 b_3 $\cdots$ b_k b_{k+1} b_{k+2} b_{k+3} $\cdots$ b_{2k} b_0
a_1 a_2 a_3 $\cdots$ a_k a_{k+1} a_{k+2} a_{k+3} $\cdots$ a_{2k} a_{2k+1} $\cdots$ a_m
T_2
b_2 b_3 $\cdots$ b_k b_{k+1} b_{k+2} b_{k+3} $\cdots$ b_{2k} b_1 b_0
a_2 a_3 $\cdots$ a_k a_{k+1} a_{k+2} a_{k+3} $\cdots$ a_{2k} a_{2k+1} $\cdots$ a_m a_1
$\vdots$
T_i
b_i b_{i+1} $\cdots$ b_{k+i-1} b_{k+i} b_{k+i+1} $\cdots$ b_{2k} b_1 $\cdots$ b_{i-1} b_0
a_i a_{i+1} $\cdots$ a_{k+i-1} a_{k+i} a_{k+i+1} $\cdots$ a_{2k} a_{2k+1} $\cdots$ a_m a_1 $\cdots$ a_{i-1}
$\vdots$
T_k
b_k b_{k+1} $\cdots$ b_{2k-1} b_{2k} b_1 b_2 $\cdots$ b_{k-1} b_0
a_k a_{k+1} $\cdots$ a_{2k-1} a_{2k} a_{2k+1} $\cdots$ a_m a_1 a_2 $\cdots$ a_{k-1}

Fig. 2. CISTs of $K_{m,n}$. Dotted lines and hollow vertices are only for the case that n is odd.

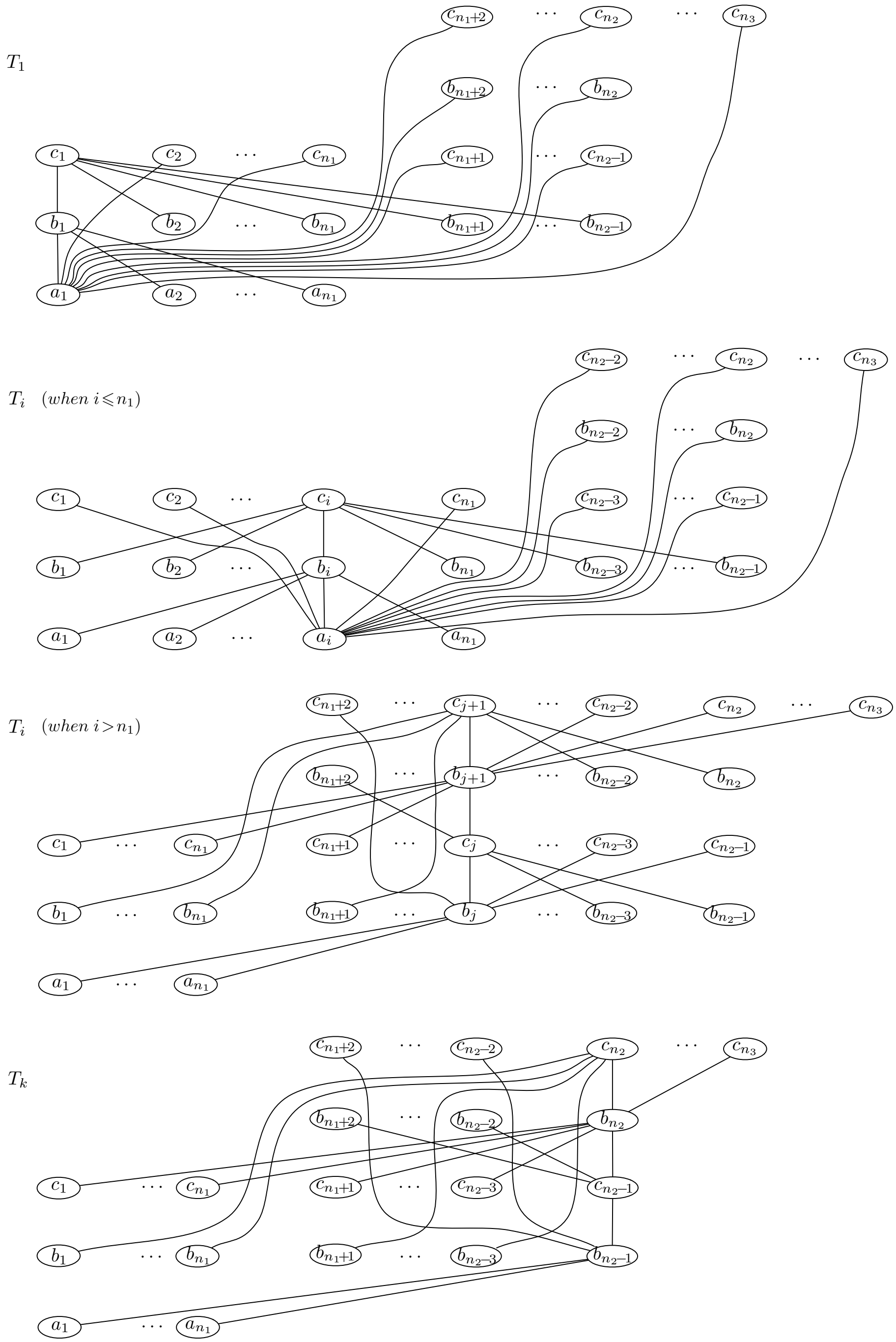

T_1
c_{n_1+2} ⋯ c_{n_2} ⋯ c_{n_3}
b_{n_1+2} ⋯ b_{n_2}
c_1 c_2 ⋯ c_{n_1} c_{n_1+1} ⋯ c_{n_2-1}
b_1 b_2 ⋯ b_{n_1} b_{n_1+1} ⋯ b_{n_2-1}
a_1 a_2 ⋯ a_{n_1}
T_i (when $i \leqslant n_1$)
c_{n_2-2} ⋯ c_{n_2} ⋯ c_{n_3}
b_{n_2-2} ⋯ b_{n_2}
c_1 c_2 ⋯ c_i c_{n_1} c_{n_2-3} ⋯ c_{n_2-1}
b_1 b_2 ⋯ b_i b_{n_1} b_{n_2-3} ⋯ b_{n_2-1}
a_1 a_2 ⋯ a_i a_{n_1}
T_i (when $i > n_1$)
c_{n_1+2} ⋯ c_{j+1} ⋯ c_{n_2-2} c_{n_2} ⋯ c_{n_3}
b_{n_1+2} ⋯ b_{j+1} ⋯ b_{n_2-2} b_{n_2}
c_1 ⋯ c_{n_1} c_{n_1+1} ⋯ c_j ⋯ c_{n_2-3} c_{n_2-1}
b_1 ⋯ b_{n_1} b_{n_1+1} ⋯ b_j ⋯ b_{n_2-3} b_{n_2-1}
a_1 ⋯ a_{n_1}
T_k
c_{n_1+2} ⋯ c_{n_2-2} c_{n_2} ⋯ c_{n_3}
b_{n_1+2} ⋯ b_{n_2-2} b_{n_2}
c_1 ⋯ c_{n_1} c_{n_1+1} ⋯ c_{n_2-3} c_{n_2-1}
b_1 ⋯ b_{n_1} b_{n_1+1} ⋯ b_{n_2-3} b_{n_2-1}
a_1 ⋯ a_{n_1}

Fig. 3. CISTs of K_{n_3,n_2,n_1}

Proof. Let $k = \lfloor \frac{n_2+n_1}{2} \rfloor$. Suppose that the vertices in V_1, V_2, and V_3 are labeled with $a_1, a_2, \ldots, a_{n_1}$, $b_1, b_2, \ldots, b_{n_2}$, and $c_1, c_2, \ldots, c_{n_3}$, respectively.

For the case that $i \leqslant n_1$, there are three non-leaf nodes a_i, b_i and c_i in T_i for $1 \leqslant i \leqslant n_1$. First, each leaf c_x is adjacent to a_i in T_i for all $1 \leqslant x \leqslant n_3$ and $x \neq i$. Then, edges (a_y, b_i) and (b_y, c_i) are added in T_i for all $1 \leqslant y \leqslant n_1$. Finally, each leaf b_{n_1+w} is adjacent to c_i (respectively, a_i) in T_i for all $1 \leqslant w \leqslant n_2 - n_1$ and w is odd (respectively, even).

For the case that $i > n_1$, there are four non-leaf nodes b_j, c_j, b_{j+1} and c_{j+1} in T_i for all $n_1 < j \leqslant n_2$ and $n_1 + j$ is odd. We note that $i = \lfloor \frac{j+n_1+1}{2} \rfloor$. First, edges (b_j, c_j), (c_j, b_{j+1}) and (b_{j+1}, c_{j+1}) are added in T_i. Then, edges (a_x, b_j), (b_x, c_{j+1}) and (c_x, b_{j+1}) are added in T_i for all $1 \leqslant x \leqslant n_1$. Next, edges (b_y, c_{j+1}), (c_y, b_{j+1}), (b_{y+1}, c_j) and (c_{y+1}, b_j) are added in T_i for all $n_1 < y < j$ and $n_1 + y$ is odd. Edges (b_w, c_j), (c_w, b_j), (b_{w+1}, c_{j+1}) and (c_{w+1}, b_{j+1}) are added in T_i for all $j < w \leqslant n_2$ and $n_1 + w$ is odd. Finally, each leaf c_v is adjacent to b_{j+1} in T_j for all $n_2 < v \leqslant n_3$.

Clearly, $T_1, T_2, \ldots, T_k$ are edge-disjoint spanning trees and, for $1 \leqslant i \leqslant k$, T_i contains three non-leaf nodes a_i, b_i and c_i (respectively, four non-leaf nodes $b_{2i-n_1-1}, c_{2i-n_1-1}, b_{2i-n_1}$ and c_{2i-n_1}) where $1 \leqslant i \leqslant n_1$ (respectively, $n_1 < i \leqslant n_2$). Thus, by Theorem 1, $T_1, T_2, \ldots, T_k$ are CISTs. These k CISTs are depicted in Figure 3. □

References

1. Hasunuma, T.: Completely independent spanning trees in the underlying graph of a line digraph. Discrete Math. 234, 149–157 (2001)
2. Hasunuma, T.: Completely Independent Spanning Trees in Maximal Planar Graphs. In: Kučera, L. (ed.) WG 2002. LNCS, vol. 2573, pp. 235–245. Springer, Heidelberg (2002)
3. Hasunuma, T., Morisaka, C.: Completely independent spanning trees in torus networks. Networks 60, 59–69 (2012)
4. Péterfalvi, F.: Two counterexamples on completely independent spanning trees. Discrete Math. 312, 808–810 (2012)

Diagnosable Evaluation of Enhanced Optical Transpose Interconnection System Networks[*]

Chang-Hsiung Tsai[**] and Jheng-Cheng Chen

Department of Computer Science and Information Engineering,
National Dong Hwa University, Hualien, Taiwan 970, R.O.C.
chtsai@mail.ndhu.edu.tw,p_p971@hotmail.com

Abstract. The process of identifying faulty processors is called the diagnosis of the system. Several models of diagnosis have been proposed and the most popular one is the PMC (Preparata, Metze and Chen) diagnostic model proposed by Preparata et al. in 1967. The precise strategy is correctly to identify all faulty processors and the pessimistic strategy is to isolate all faulty processors within a set containing one fault-free processor at most. For a multiprocessor system, its diagnosability is critical to measure the performance. The enhanced optical transpose interconnection system, enhanced OTIS, network has important applications in parallel processing. In this network architecture, n^2 processors are divided n groups where there are n processors, and processors in the same group are connected by electronic links, simultaneously, these groups are connected by optical links. In particular, an enhanced OTIS network is regular if its base graph G is regular. In this paper, we discuss the fault diagnosis issue of the enhanced OTIS network including precise strategy and pessimistic strategy under the PMC diagnostic model.

Keywords: Interconnection network, Reliability, System-level diagnosis, Pessimistic strategy, enhanced OTIS.

1 Introduction

Even though improvement in technology made it possible to realize very powerful multiprocessor systems, the bandwidth limitations imposed by electronic interconnects prove to be a major bottleneck. Optical interconnection networks would be the better candidate for the conventional interconnection networks of multiprocessor systems. A popular realization of optical communication is the Optical Transpose Interconnection System (OTIS) [10]. OTIS networks have a base graph G, on n vertices, and consist of n disjoint copies of G. These copies are labelled $G_1, G_2, \ldots, G_n$ and the vertices of any copy are $v_1, v_2, \ldots, v_n$. The edges involved in any one of these copies of G are intended to model electronic connections whereas additional edges, where there is an edge from vertex v_i of copy G_j to vertex v_j of copy G_i , for every $i, j \in \{1, 2, \ldots, n\}$, with $i \neq j$, are intended to model the optical connections. The resulting OTIS network

[*] This research was partially supported by the National Science Council of the Republic of China under contract NSC 101-2115-M-059-006-.

[**] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 115–122.
DOI: 10.1007/978-3-642-35452-6_14

is denoted by $OTIS(G)$. One slightly displeasing aspect of OTIS networks is that no matter what the base graph G is, the corresponding OTIS network $OTIS(G)$ cannot be regular. We note that there is no optical link from the node v_i of G_i for any copy of G_i in $OTIS(G)$. In order to improve the regularity of OTIS networks, enhanced OTIS networks were proposed by Das in 2007 [3]. Enhanced OTIS networks retains the corresponding OTIS as a subgraph and thus has almost all the desirable properties of the corresponding OTIS but contains some additional optical links. The advantage gained by adding those extra links, namely, uniform node degree, increasing diagnosability, and improved fault-tolerance. These advantages make enhanced OTIS a suitable architecture for multiprocessor interconnection network.

The process of identifying faulty processors is called the diagnosis of the system. The PMC diagnostic model proposed by Preparata, Metze and Chen [14] is the most popular among diagnostic models. In this model, every processor performs tests on its neighbors by communication links between them. When one processor tests another, the tester declares the tested processor to be fault-free or faulty depending on the test response; the result is always accurate if the tester is fault-free, but if the tester is faulty, the result is unreliable. According to the traditional *precise diagnosis* strategy, it is demanded that all processors are identified correctly. A system is *t-diagnosable* if all faulty processors can be identified under precise strategy provided that the number of faults presented does not exceed t. The maximum number of faulty processors that the system can guarantee to identify is called the diagnosability of the system. The *pessimistic diagnosis* strategy proposed by Kavianpour and Friedman [7] is a process to diagnose faults that allows all faulty processors can be isolated within a set having at most one fault-free processor. A system is t_1/t_1-diagnosable if, provided the number of faulty processors is bound by t_1, all faulty processors can be isolated within a set of size at most t_1 that a fault-free node may be mistaken as a faulty one. Numerous studies have been dedicated on the PMC model [2, 4, 6, 11–13, 15, 16].

In this paper, we show that an enhanced OTIS network with base graph G which has even number of nodes and degree of $t \geq 2$ is $(t+1)$-diagnosable, and $(t_1+2)/(t_1+2)$-diagnosable if G is a t_1/t_1-diagnosable regular graph. Our paper is organized as follow: In Section 2, we provide graph-theoretic terminology and fundamental properties. In Section 3, the precise diagnosability and pessimistic diagnosability of enhanced optical transpose interconnection system networks under the PMC model are evaluated. Finally, we give a conclusion in Section 4.

2 Preliminaries

The underlying topology of a multiprocessor system is usually represented by a graph $G = (V, E)$, where each of the vertices $u \in V$ denotes a processor and each edge $(u, v) \in E$ denotes an undirected communication link from processor u to processor v. Throughout this paper, the three terms - network, system, and graph - will be used interchangeably. A loop edge is an edge for which the two endpoints are the same vertex. Two edges are multiple edges if they have exactly the same two endpoints. A graph is simple if it does not contain loop edges and multiple edges. In this paper, we focus on simple graphs. For graph-theoretical terminology and notation not defined here, we follow [8].

Let $G = (V, E)$ be a graph. For a vertex $u \in V(G)$, we define $N_G(u)$ to be the set of vertices adjacent to u. The size of $N_G(u)$ is called the degree of u denoted by $deg_G(u)$. In addition, the degree of a graph G is defined as $\delta(G) = min_{u \in V} deg_G(u)$. For a vertex set $U \subseteq V(G)$, let $N_G(U) = \bigcup_{u \in U} N_G(u) \setminus U$. If $|N_G(u)| = k$ for any vertex u in G, then G is called a k-regular graph. A graph $H = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. The components of a graph G are its maximal connected subgraphs. A component is trivial if it has no edges; otherwise, it is nontrivial. In addition, For a vertex set $S \subseteq V$, the notation $G - S$ means the graph obtained by deleting all the vertices in S from G and all edges in G connected to S.

Optics, owing to its inherent parallelism, high spectral and spatial bandwidth, low latency and signal crosstalk, less power consumption and desirable topological properties [5], possesses the potential for a better solution to the communication problem in parallel and distributed computing. Optical interconnection networks would be the better candidate for the conventional interconnection networks of multiprocessor systems. A popular realization of optical communication is the Optical Transpose Interconnection System (OTIS) [10]. The OTIS(G) network has a base graph G with n vertices and consists of n copies of G, namely $G_1, G_2, \ldots, G_n$. Let the vertices of any copy are $v_1, v_2, \ldots, v_n$ and add an edge connecting v_i of copy G_j to v_j of G_i for $1 \le i, j \le n$ with $i \ne j$. The formal definition of $OTIS(G)$ is following.

Definition 1. *Let $G = (V, E)$ be an undirected graph. The $OTIS(G)$ graph is an undirected graph given by :*

$$\begin{aligned} V(EOTIS(G)) &= \{\langle v_g, v_p\rangle | v_g, v_p \in V(G)\} \textit{ and} \\ E(EOTIS(G)) &= \{(\langle v_g, v_r\rangle, \langle v_g, v_s\rangle) | v_g \in V(G), (v_r, v_s) \in E(G)\} \\ &\quad \cup \{(\langle v_g, v_p\rangle, \langle v_p, v_g\rangle) | v_g, v_p \in V(G),\ \textit{and}\, v_g \ne v_p\}. \end{aligned}$$

Obviously, the $OTIS(G)$ is not regular that no matter what the base graph G is. Therefore, in order to solve the problem of regularity, a variation of $OTIS(G)$ called *enhanced* $OTIS(G)$ was proposed in [3]. Given a base graph G with vertices set $\{v_1, v_2, \ldots, v_n\}$, the enhanced $OTIS(G)$ denoted by $EOTIS(G)$ is obtained from $OTIS(G)$ by adding an edge from node $\langle v_g, v_g\rangle$ to $\langle v_{\overline{g}}, v_{\overline{g}}\rangle$, where $1 \le g \le n$ and $\overline{g} = n - g + 1$. For example, graphs $OTIS(C_4)$ and $EOTIS(C_4)$ are illustrated in Fig. 1. Obviously, $EOTIS(G)$ includes self-loop edges if G has odd number of nodes. Hence we only consider the base graph G with even number of nodes in this paper.

Therefore, $EOTIS(G)$ is regular if its base graph G is regular, and the following lemma is obviously obtained.

Lemma 1. $\delta(EOTIS(G)) = \delta(G) + 1$.

There are many diagnostic models of system-level diagnosis have been proposed in the literatures [1, 9, 14]. Among these models, the most popular one is the PMC diagnostic model (or PMC model for short) proposed by Preparata, Metze, and Chen [14]. Under the PMC model, a self-diagnosable system G was often modelled by a digraph in which an arc direct from vertex u to vertex v means that u can test v. In this situation, u is a tester and v is tested vertex. The edge orientation is based on unit type (test or tested

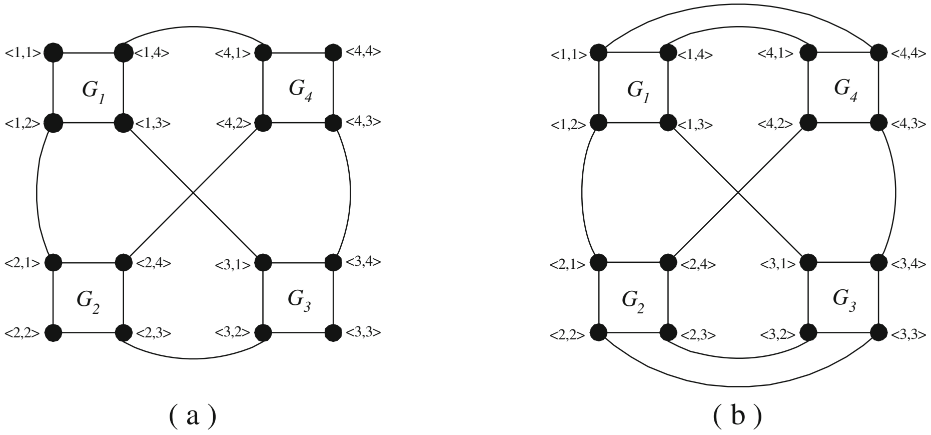


Fig. 1. (a)$OTIS(C_4)$ and (b)$EOTIS(C_4)$

unit). The directed edge (u, v), with its binary weight $\sigma(u, v)$, exists if and only if units u and v are interconnected and unit v (tested unit) is tested by unit u with binary outcome $\sigma(u, v)$. The test outcome $\sigma(u, v)$ is 0 (1) if u diagnoses v as fault-free (faulty). If u is faulty, then the outcome $\sigma(u, v)$ is unreliable. The collection of all test outcomes is called a $syndrome$.

The maximal number of faulty nodes that a system can guarantee to diagnose is called the degree of diagnosability of the system. We have definition of diagnosability under a precise strategy as follow.

Definition 2. *[14] A system is t-diagnosable if all faulty units can be identified without replacement, provided that the number of faults presented does not exceed t.*

Another representative diagnosis strategy based on the PMC diagnostic model was proposed by Friedman [7] and is called the pessimistic strategy, which defines the diagnosability of system as follow.

Definition 3. *[7] A graph G is pessimistically t_1/t_1-diagnosable if all the faulty vertices can be isolated to within a set of at most t_1 vertices, provided that the number of faulty vertices at any give time is bounded from above by t_1. The pessimistic diagnosability of G is the maximum number t_1 such that G is pessimistically t_1/t_1-diagnosable.*

3 Main Results

In this section, we study the precise diagnosability and pessimistic diagnosability of $EOTIS(G)$ where G has even number of nodes. It is verified that the $EOTIS(G)$ is $(t + 1)$-diagnosable, $t = \delta(G) \geq 2$, and $(t_1 + 2)/(t_1 + 2)$-diagnosable if G is a t_1/t_1-diagnosable regular graph with $\delta(G) \geq 2$. Hsu and Tan [6] proposed a special data structure, called *extending star* in here, $T_G(v;k)$ to evaluate the precise diagnosability of G. For example, Fig. 2 shows an extending star of order 3 at vertex $\langle 1,1\rangle$, $T_{EOTIS(C_4)}(\langle 1,1\rangle;3)$, of $EOTIS(C_4)$.

Definition 4. *[6] Letting $G(V,E)$ be a graph, $v \in V$ be a vertex, and k be an integer, $k \geq 1$, an extending star of order k at vertex v is defined to be a subgraph of G denoted by $T_G(v,k)$, where $V(T_G(v,k)) = \{v\} \cup \{v_i, x_i | 1 \leq i \leq k\}$ and $E(T_G(v,k)) = \{(v,v_i),(v_i,x_i)|1 \leq i \leq k\}$.*

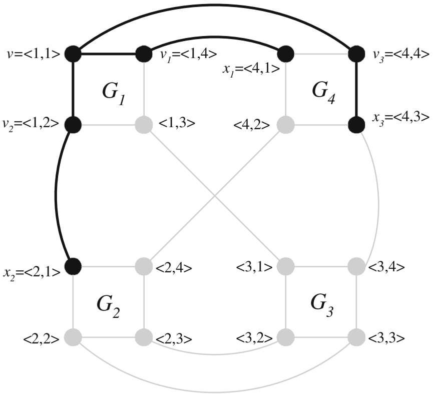


Fig. 2. An extending star $T_{EOTIS(C_4)}(\langle 1,1\rangle;3)$ order 3 at vertex $\langle 1,1\rangle$ illustrated with bold lines

Lemma 2. *[6] A graph G is t-diagnosable if and only if it contains an extending star of order t at each vertex v.*

By Lemma 2, the following theorem is verified by constructing an extending star structure at each vertex of $EOTIS(G)$.

Theorem 1. *Let G be a graph and $|V(G)| \geq 4$. Then $EOTIS(G)$ is $(t+1)$-diagnosable where $\delta(G) = t \geq 2$.*

Proof. Let $V(G) = n$ and $\delta(G) = t \geq 2$. Hence $EOTIS(G)$ has n copies of G. namely, $G_1, G_2, \ldots, G_n$. For sake of brevity, we reduce the node $\langle v_i, v_j\rangle$ in G_i to $\langle i,j\rangle$, where $1 \leq i \leq n$ and $1 \leq j \leq n$. Let $\overline{i} = n - i + 1$ for any $1 \leq i \leq n$.

Arbitrary pick a vertex $\langle i,j\rangle$ in G_i for some i. Suppose $i = j$. Clearly, the vertex $\langle \overline{i},\overline{i}\rangle$ is adjacent to $\langle i,i\rangle$ in $G_{\overline{i}}$. Since $\delta(G) = t \geq 2$, there are t different vertices in G_i joining $\langle i,i\rangle$, denoted by $\langle i,i_k\rangle$ for $1 \leq k \leq t$. Obviously, $\langle i_k,i\rangle$ is adjacent to $\langle i,i_k\rangle$ in G_{i_k} for $1 \leq k \leq t$. If $i_k \neq \overline{i}$ for $1 \leq k \leq t$, let $U = \{\langle i,i\rangle, \langle \overline{i},\overline{i}\rangle, \langle \overline{i},l\rangle\} \cup \{\langle i,i_k\rangle, \langle i_k,i\rangle | 1 \leq k \leq t\}$, where the vertex $\langle \overline{i},l\rangle$ is a neighbor of $\langle \overline{i},\overline{i}\rangle$ in G_j. Analogously, if $i_k = \overline{i}$ for some k, without loss of generality, we may assume $i_1 = \overline{i}$. Since $\delta(G_{\overline{i}}) \geq 2$, there is a vertex $\langle \overline{i},l\rangle$ adjacent to $\langle \overline{i},\overline{i}\rangle$ so that $l \neq i$. Let $U = \{\langle i,i\rangle, \langle \overline{i},\overline{i}\rangle, \langle \overline{i},l\rangle\} \cup \{\langle i,i_k\rangle, \langle i_k,i\rangle | 1 \leq k \leq t\}$, Clearly, the graph induced by U is an extending star of order $t+1$ at $\langle i,i\rangle$.

Similarly, if $i \neq j$, we easily prove that there is an extending star of order $t+1$ at vertex $\langle i, j\rangle$. Therefore, $EOTIS(G)$ is $(t+1)$-diagnosable by Lemma 2. □

Lemma 3. *[2] A graph $G = (V, E)$ is t_1/t_1-diagnosable if and only if for any integer p, $1 \leq p \leq t_1$, and each $V' \subset V(G)$, $|V'| = 2p$, $|N(V')| \geq t_1 - p + 1$.*

Lemma 4. *A graph G is a t_1/t_1-diagnosable if and only if for each vertex set $S \subset V(G)$ with $|S| = p$, $0 \leq p \leq t_1 - 1$, there exists at most one trivial component in $G - S$ and each nontrivial component C of $G - S$ satisfying $|V(C)| \geq 2(t_1 - p) + 1$.*

Proof. $(\Rightarrow)$ The proof is by contradiction. Suppose that there is a $S \subset V(G)$ with $|S| = p$, $0 \leq p \leq t_1 - 1$. The graph $G - S$ contains more than one trivial components and/or contains a nontrivial component C with $2(t_1 - p)$ vertices at most.

Let $V' = \{u, v\}$ where u and v are two trivial components in $G - S$. Obviously, $N_G(V') \subseteq S$. Hence, $|N_G(V')| \leq |S| = p \leq t_1 - 1$. By Lemma 3, it is contradiction that $|V'| = 2$, $|N_G(V')| \geq t_1$.

Suppose that there is a nontrivial component $C = (V(C), E(C))$ in $G - S$, where $|V(C)| = q \leq 2(t_1 - p)$. Let $k = \frac{q}{2}$ if q is even. Hence $k \leq t_1 - p$. Since $0 \leq p \leq t_1 - 1$, $1 \leq k \leq t_1$. Let $V' = V(C)$. Hence $|V'| = 2k$. Since $N_G(V') \subseteq S$, $|N_G(V')| \leq |S| \leq p$. This implies $|N_G(V')| \leq t_1 - k < t_1 - k + 1$. By Lemma 3, it is a contradiction.

Let $k = \frac{q-1}{2}$ if q is odd. Hence $k \leq t_1 - p - 1$. This implies $p + 1 \leq t_1 - k$. Since $0 \leq p \leq t_1 - 1$, $0 \leq k \leq t_1 - 1$. Let V' be any subset of $V(C)$ with $|V'| = q - 1$, i.e., $V' = V(C) - \{w\}$ for some $w \in V(C)$. Hence $N_G(V') \subseteq S \cup \{w\}$ and $w \neq S$. Hence $|V'| = 2k$. Since $|S| = p$, $|N_G(V')| \leq p + 1 \leq t_1 - k < t_1 - k + 1$. By Lemma 3, it is a contradiction.

$(\Leftarrow)$ Suppose, by contradiction, G is not t_1/t_1-diagnosable. By Lemma 3, there is an integer p, $1 \leq p \leq t_1$, and $V' \subset V$, $|V'| = 2p$ such that $|N_G(V')| < t_1 - p + 1$. Let $S = N_G(V')$. Hence $|S| = k \leq t_1 - p$. This implies $p \leq t_1 - k$ and $0 \leq k \leq t_1 - 1$. Since $|V'| = 2$ and the graph induced by V' contains at most one trivial component, there is a nontrivial component $C = (V(C), E(C))$ in the graph induced by V'. Hence $V_C \subseteq V'$. Obviously, $|V'| \geq |V(C)| \geq 2(t_1 - k) + 1 \geq 2p + 1$ which is contradiction. The proof is complete. □

Lemma 5. *Let G be a t_1/t_1-diagnosable regular graph. Then $|N_{EOTIS(G)}(\{u, v\})| \geq t_1 + 2$ for any $u, v \in V(EOTIS(G))$.*

Proof. Let u, v be any two vertices in $EOTIS(G)$. Also let $G_1, G_2, \ldots, G_n$ be n copies of G in $EOTIS(G)$ where $|V(G)| = n$. The proof is divided into two cases in the following.

Case 1. u and v are in the same copy G_i for some i. Hence $u = \langle i, j\rangle$ and $v = \langle i, k\rangle$ for some k, j where $k \neq j$. Since G is a t_1/t_1-diagnosable, $|N_{G_i}(\{u, v\})| \geq t_1$. It is obvious that $\langle j, i\rangle$ and $\langle k, i\rangle$ are adjacent to u and v in G_j and G_k, respectively. Hence $\langle j, i\rangle$ and $\langle k, i\rangle$ are not in $N_{G_i}(\{u, v\})$. Therefore, $N_{EOTIS(G)}(\{u, v\}) = N_{G_i}(\{u, v\}) \cup \{\langle j, i\rangle, \langle k, i\rangle\}$ and then $|N_{EOTIS(G)}(\{u, v\})| \geq t_1 + 2$.

Case 2. u and v are in different copies G_i and G_j, respectively. Since G is a regular and $|N_G(\{x, y\})| \geq t_1$ for any two different vertices x, y in G, $|N_G(x)| \geq$

$\frac{t_1}{2}+1$. By definition, $N_{EOTIS(G)}(\{u,v\}) = (N_{EOTIS(G)}(u) \cup N_{EOTIS(G)}(v)) \setminus \{u,v\}$. Clearly, $N_{G_i}(u) \subset N_{EOTIS(G)}(u)$ and $N_{G_j}(v) \subset N_{EOTIS(G)}(v)$. Therefore, $N_{G_i}(u) \cup N_{G_j}(v) \subseteq N_{EOTIS(G)}(\{u,v\})$. Since G_i and G_j are disjoint, $N_{G_i}(u) \cap N_{G_j}(v) = \emptyset$. This implies $|N_{EOTIS(G)}(\{u,v\})| \geq |N_{G_i}(u)| + |N_{G_j}(v)| \geq \frac{t_1}{2}+1+\frac{t_1}{2}+1 = t_1+2$. □

By above lemmas, we have

Theorem 2. *Let G be a regular graph with $\delta(G) \geq 2$. Then $EOTIS(G)$ is $(t_1+2)/(t_1+2)$-diagnosable if G is (t_1/t_1)-diagnosable.*

Proof. Let $|V(G)| = n$ and $G_1, G_2, \ldots, G_n$ be n copies of G in $EOTIS(G)$. Let $V = V(EOTIS(G))$. For simplicity, let $N(V')$ denote $N_{EOTIS(G)}(V')$ for any $V' \subseteq V$. The prove is by contradiction. Suppose $EOTIS(G)$ is not a $(t_1+2)/(t_1+2)$-diagnosable. By Lemma 4, there is a vertex set $S \subseteq V$, $|S| = p$, $0 \leq p \leq t_1+1$, so that $EOTIS(G) - S$ contains two trivial components at least and/or contains a nontrivial component C satisfying $2 \leq |V(C)| \leq 2(t_1+2-p)$.

Suppose u and v are two trivial conpoments in $EOTIS(G)$. Let $V' = \{u,v\}$. By Lemma 5, $|N(V')| \geq t_1+2$. Obviously, $N(V') \subseteq S$. Hence, $|S| \geq t_1+2$ which is contradiction.

Suppose there is a nontrivial component C in $EOTIS(G) - S$ satisfying $2 \leq |V(C)| \leq 2(t_1+2-p)$. For $1 \leq i \leq n$, let $S_{G_i} = S \cap V(G_i)$ and C_{G_i} be a graph induced by $V(C) \cap V(G_i)$, i.e., S_{G_i} is a vertex subset of S in G_i and C_{G_i} is a subgraph of C in G_i. The proof is divided into two parts: (1) $C = C_{G_i}$ for some i and (2) $C \neq C_{G_i}$ for all i.

Case 1. $C = C_{G_i}$ for some i. Without loss of generality, we may assume $C_{G_1} = C$. Clearly, $N(V(C)) \subseteq S$. Since $C_{G_1} = C$, $|N(V(C))| \geq |V(C)|$ and $|S_{G_1}| = |S| - |V(C)|$. Let $|S_{G_1}| = p_1$. Hence $p_1 = p - |V(C)|$. Since $|V(C)| \geq 2$, $p_1 \leq p-2$ and implies $p_1 \leq t_1 - 1$. Since G is t_1/t_1-diagnosable, $|V(C)| \geq 2(t_1-p_1)+1$. Therefore, $|V(C)| \geq 2(t_1-p+2)+1$ which is contradiction.

Case 2. $C \neq C_{G_i}$ for all i. In this case, the vertex set $V(C)$ is distributed to k copies of G. Without loss of generality, we may assume $V(C) = V(C_{G_1}) \cup V(C_{G_2}) \cup \ldots V(C_{G_k})$ where $k \geq 2$. Clearly, $\bigcup_{1 \leq i \leq k} S_{G_i} \subseteq S$. Since $k \geq 2$ and by pigeonhole principle, there is a S_{G_j}, $1 \leq j \leq k$, so that $|S_{G_j}| \leq t_1 - 1$. Without loss of generality, $|S_{G_1}| \leq t_1-1$ and $V(C_{G_2}) \neq \emptyset$. Since G_1 is t_1/t_1-diagnosable, $|V(C_{G_1})| \geq 2(t_1 - |S_{G_1}|)+1$. Thus, $|V(C)| \geq 2(t_1-|S_{G_1}|)+1+|V(C_{G_2})| \geq 2(t_1-p+1)+1+|V(C_{G_2})|$. Obviously, $|V(C)| \geq 2(t_1-p+2)+1$ if $|V(C_{G_2})| \geq 2$ which is contradiction.

Suppose $|V(C_{G_2})| = 1$. Since G_2 is a t_1/t_1-diagnosable regular graph, $\delta(G_2) \geq \frac{t_1}{2}+1$. This implies $|S_{G_2}| \geq \frac{t_1}{2}+1$ and $|S_{G_1}| \leq p - \frac{t_1}{2} - 1$. Hence $|V(C)| \geq 2(t_1-p+\frac{t_1}{2}+1)+1 \geq 2(t_1-p+2)+1$ due to $t_1 \geq \delta(G) \geq 2$. It is a contradiction. Therefore, $EOTIS(G)$ is $(t_1+2)/(t_1+2)$-diagnosable □

4 Conclusion

In this paper, we study the precise diagnosability and pessimistic diagnosability of an enhanced optical transpose interconnection system network ($EOTIS(G)$) with a base graph G containing even number of nodes under the PMC model. We get that the precise diagnosability of $EOTIS(G)$ is $(\delta(G) + 1)$ if $\delta(G) \geq 2$ and the pessimistic diagnosability of $EOTIS(G)$ is $t_1 + 2$ if G is a t_1/t_1-diagnosable regular graph with $\delta(G) \geq 2$.

References

1. Barsi, F., Grandoni, F., Maestrini, P.: A theory of diagnosability of digital systems. IEEE Transaction on Computers 25(6), 585–593 (1976)
2. Chwa, K.Y., Hakimi, S.L.: On fault identification in diagnosable systems. IEEE Transaction Computers C-30, 414–422 (1981)
3. Das, R.K.: Diameter and routing in enhanced OTIS cube. In: International Conference on Computing: Theory and Applications, pp. 188–192 (2007)
4. Fan, J., Yang, J., Zhou, G., Zhao, L., Zhang, W.: Diagnosable evaluation of DCC linear congruential graphs under the PMC diagnostic model. Information Sciences 179, 1785–1791 (2009)
5. Goodman, J.W., Leonberger, F.J., Kung, S.Y., Athale, R.A.: Optical interconnections for VLSI systems. Proceedings of the IEEE 72(7), 850–866 (1984)
6. Hsu, G.H., Tan, J.J.M.: A local diagnosability measure for multiprocessor systems. IEEE Transactions on Parallel and Distributed Systems 18(5), 598–607 (2007)
7. Kavianpour, A., Friedman, A.D.: Efficient design of easily diagnosable system. In: Proceedings of IEEE Computer Society's 3rd USA-Japan Computer Conference (1978)
8. Leighton, F.T.: Introduction to Parallel Algorithms and Architecture: Arrays, Trees, Hypercubes. Morgan Kaufmann, San Mateo (1992)
9. Maeng, J., Malek, M.: A comparison connection assignment for self-diagnosis of multiprocessors systems. In: Proc. 11th Intl Symp. Fault-Tolerant Computing, pp. 173–175 (1981)
10. Marsden, G.C., Marchand, P.J., Harvey, P., Esener, S.C.: Optical transpose interconnect system architectures. Optical Letters 18, 1083–1085 (1993)
11. Li, T.K., Tsai, C.H., Hsu, H.C.: A fast fault-identification algorithm for bijective connection graphs using the PMC model. Information Sciences 187, 291–297 (2012)
12. Liu, X., Yang, X., Xiang, M.: One-step t-fault diagnosis for hypermesh optical interconnection multiprocessor systems. Journal of Systems and Software 82, 1491–1496 (2009)
13. Peng, S.L., Lin, C.K., Tan, J.J.M., Hsu, L.H.: The g-good-neighbor conditional diagnosability of hypercube under PMC model. Applied Mathematics and Computation 218, 10406–10412 (2012)
14. Preparata, F.P., Metze, G., Chien, R.T.: On the connection assignment problem of diagnosis systems. IEEE Transaction Electronic Computers 16(12), 848–854 (1967)
15. Tsai, C.H.: A quick pessimistic diagnosis algorithm for hypercube-like multiprocessor systems under the PMC model. IEEE Transactions on Computers, `http://doi.ieeecomputersociety.org/10.1109/TC.2011.228`
16. Zhou, Q.Y., Liu, S.Y., Zhu, Q.: Local diagnosability of generic star-pyramid graph. Information Processing Letters 109, 695–699 (2009)

Three-Round Adaptive Diagnosis in Twisted Cubes[*]

Pao-Lien Lai[**], Zheng-da Liu, and Po-Chang Li

Department of Computer Science and Information Engineering,
National Dong Hwa University, Shoufeng, Hualien, Taiwan 97401, R.O.C.
baolein@mail.ndhu.edu.tw

Abstract. In this paper, we propose a scheme to solve the problem of adaptive fault diagnosis in n-dimensional twisted cubes in at most three test rounds, provided that the number of faulty vertices is at most n for $n \geq 9$. First, each vertex tests one specific neighbour and is tested by another specific neighbour to provide a basic syndrome in two rounds in our scheme. Then, we assign other necessary tests to diagnose the vertices that cannot be identified according to the previous syndrome in one more round. The scheme is optimal for at most three rounds since the adaptive diagnosis needs at least three rounds to complete.

Keywords: Interconnection networks, twisted cubes, hypercubes, adaptive diagnosis, Hamiltonian, distributed systems, multiprocessors.

1 Introduction

Interconnection networks play a major role in parallel computing systems. Since the increasing size of the multiprocessor system and the longer operation time in the system, the components of the multiprocessor system are easier failed than before. The concept of system-level fault-diagnosis has first been proposed by Preparata, Metze, and Chien [1]. Each processor performs tests of their neighbors in the system and output their test results. By analyzing the complete collection of the test results, the observer identifies which is $faulty$ and which is $fault-free$. A characterization of t-diagnosable systems is given by Hakimi and Amin [2].

Recently, Nakajima proposed an adaptive diagnosis model to improve the efficiency of the diagnosis process in [3] which test assignments are dynamically determined after checking the results of previous test results. S. Fujita and T. Araki proposed three round adaptive diagnosis in n-dimensional hypercube [4].

An n-dimensional twisted cube, TQ_n [5,6], is an important variation of hypercube Q_n and preserves many of its desirable properties. TQ_n has 2^n vertices and $n2^{n-1}$ links, same as hypercube Q_n. However, the diameter, wide diameter,

[*] This work was supported in part by the National Science Council of the Republic of China under Contract NSC 101-2115-M-259-007.

[**] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 123–132.
DOI: 10.1007/978-3-642-35452-6_15

and faulty diameter in twisted cubes are about half of those in comparable hypercubes [7]. In [8], P.L. Lai and C.H. Tasi studied the embedding of tori and grids into twisted cubes.

The paper is organized as follows. The preliminary knowledge and fundamental definitions for twisted cubes, the PMC model, t-diagnosable in the next section. Then, we describe an interesting scheme to transform a TQ_n into a hypercube-based hierarchical network in Section 3. A useful technique for detecting the status of each vertex (fault-free, faulty, or suspicious) from the basic syndrome is introduced in Section 4. In Section 5 we introduce our diagnosis scheme and some important concepts for assigning the furthermore necessary tests to identify all faults form the suspicious vertices. Conclusions are given in the final section.

2 Preliminary

For the graph definition and notation we follow [9]. G=(V, E) is a graph if V is a finite set and E is a subset of $\{(u, v)|(u, v)$ is an unordered pair of $V\}$. A *path* is a sequence of adjacent vertices written as $\langle v_0, v_1, \ldots, v_m\rangle$, in which all the vertices $v_0, v_1, \ldots, v_m$ are distinct except possibly $v_0 = v_m$. A *cycle* is a path with at least three vertices such that the first vertex is the same as the last one. An *isomorphism* from a simple graph G to a simple graph H is a one-to-one and onto function $f : V(G) \to V(H)$ such that $(u, v) \in E(G)$ if and only if $(f(u), f(v)) \in E(H)$. We say "G is *isomorphic* to H", if there is an isomorphism from G to H.

The twisted cube was first defined by Hilbers et al [5]. The n-dimensional twisted cube, denoted by TQ_n, is a variant of n-dimensional hypercube Q_n. Let u be a vertex of TQ_n and u will be written as $u_{n-1}u_{n-2}\cdots u_1u_0$, a binary string of length n, where u_{n-1} is the most significant bit and u_0 is the least significant bit. The i-bit of u is u_i for $0 \le i \le n-1$. The complement of u_i will be denote by $\overline{u}_i$ ($\overline{0} = 1$ and $\overline{1} = 0$). To define TQ_n, a parity function $f(x)$ is defined as $f_i(u) = u_i \oplus u_{i-1} \oplus \cdots \oplus u_1 \oplus u_0$, $0 \le i \le n-1$, where $\oplus$ is the exclusive-or operation.

Definition 1. *[10,5] The 1-dimensional twisted cube TQ_1 is a complete graph with two vertices labelled by 0 and 1, respectively. For an odd integer $n \ge 3$, the n-dimensional twisted cube TQ_n consists of four subcubes TQ^{00}_{n-2}, TQ^{01}_{n-2}, TQ^{10}_{n-2}, TQ^{11}_{n-2}, where TQ^{ij}_{n-2} is isomorphic to TQ_{n-2} and $V(TQ^{ij}_{n-2}) = \{ijx|x \in V(TQ_{n-2})\}$ and $E(TQ^{ij}_{n-2}) = \{(ijx, ijy)|(x, y) \in E(TQ_{n-2})\}$ for $i, j \in \{0, 1\}$; $V(TQ_n) = \bigcup_{i,j\in\{0,1\}} V(TQ^{ij}_{n-2})$, $E(TQ_n) = \bigcup_{i,j\in\{0,1\}} E(TQ^{ij}_{n-2}) \bigcup E'$, where for the nodes u, $v \in V(TQ_n)$, $(u, v) \in E$ if u and v satisfy one of the three conditions as follows:*

1. *$u = \overline{v}_{n-1}v_{n-2}v_{n-3}\cdots v_0$;*
2. *$u = \overline{v}_{n-1}\overline{v}_{n-2}v_{n-3}\cdots v_0$ and $f_{n-3}(v) = 0$;*
3. *$u = v_{n-1}\overline{v}_{n-2}v_{n-3}\cdots v_0$ and $f_{n-3}(v) = 1$.*

We say that v is the i-neighbor of u, denoted by $v = u^i$, and the edge (u, v) is an edge of dimension i if the following two conditions are satisfied: 1) either $u_{2x}u_{2x-1} = \overline{v}_{2x}v_{2x-1}$ for $i = 2x$; or, $u_{2x}u_{2x-1} = v_{2x}\overline{v}_{2x-1}$ with $f_{2x-2}(u) = 1$ or $u_{2x}u_{2x-1} = \overline{v}_{2x}\overline{v}_{2x-1}$ with $f_{2x-2}(u) = 0$ for $i = 2x - 1$; and 2) $u_j = v_j$ for $j \notin \{2x, 2x - 1\}$. We say that $V_o(u) = \{v \mid u_{2i+1} = v_{2i+1}$, for every $i = 0, 1, \cdots, \lfloor \frac{n}{2} \rfloor - 1\}$. Let $G[V_o(u)]$ be the subgraph of the TQ_n induced on $V_o(u)$ and let $dif(u, v)$ be the set of the positions of the different bits between u and v. For an instance, let u be 10001 and v be 00111, $dif(u, v)$ is $\{1, 2, 4\}$.

2.1 Diagnosis Model

Let u and v be a pair of adjacent nodes of G. The outcome of the test conducted by u and v is denoted by $\gamma(u, v)$. The test result $\gamma(u, v) = 0$ (resp. 1) if u evaluates v as *fault-free* (resp. *faulty*). A test with outcome 0 (resp., 1) is called *0-arrow* (resp., *1-arrow*). The collection of all test results for a test assignment is called *syndrome*. A system is *t-diagnosable* if there exists a test assignment such that all faulty nodes are identified, provided that the number of faulty nodes does not exceed t [1]. In non-adaptive diagnosis, all test assignment of a system must be scheduled in advance. In adaptive diagnosis, first proposed by Nakajima [3], the next tests are determined by previous test results. In this paper, we introduce a three-round adaptive diagnosis algorithm for twisted cubes.

3 Architectural Relation between Twisted Cubes and Hypercubes

The hypercube is one of the most popular interconnection networks because many parallel algorithms have been developed using this structure. It is well known that an n-dimensional crossed cube TQ_n is an important variation of hypercube Q_n. In this section, we furthermore show that a twisted cube can be transformed into a two-level hypercube-based hierarchical network.

Lemma 1. *For an odd integer $n \geq 3$ and any vertex $u \in V(TQ_n)$, the subgraph of the TQ_n induced on $V_o(u)$ is isomorphic to a $Q_{\lceil \frac{n}{2} \rceil}$.*

Proof: Clearly, there is $2^{\lceil \frac{n}{2} \rceil}$ vertices in the $V_o(u)$. Two vertices of $V_o(u)$ are adjacent if and only if their binary string differ exactly in one bit position, because one even bit position of TQ_n switch from 1 to 0 (or 0 to 1), do not affect other bits. And then, Two vertices of hypercube are adjacent if and only if their binary string differ exactly in one bit position. So, $V_o(u)$ is isomorphic to a $Q_{\lceil \frac{n}{2} \rceil}$. □

Lemma 2. *For an odd integer $n \geq 3$ and two vertices u and v of a TQ_n, $V_o(u) \cap V_o(v) = \emptyset$ if $u_{2i+1} \neq v_{2i+1}$ for some integer $0 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1$.*

Proof: Let $a \in V_o(u)$. Then $a_{2i+1} = u_{2i+1} \neq v_{2i+1}$. Clearly, $a \notin V_o(v)$. Then, $V_o(u) \cap V_o(v) = \emptyset$. □

To simplify the explanation, if no ambiguity arises, consider for an odd integer $n \geq 3$, an integer $m = \lceil \frac{n}{2} \rceil$ and a vertex u of a TQ_n, consider the $G[V_o(u)]$ as a super vertex, denoted as S_u.

For an integer $n \geq 3$, an even integer i and two vertices u and v of a CQ_n with $dif(u, v) = \{i\}$. There is an one-to-one and onto function $f : V_e(u) \rightarrow V_e(v)$ by one vertex of the $V_e(v)$ being an i-neighbor of the $V_e(u)$ in the S_u and the S_v.

Lemma 3. *For an odd integer $n \geq 3$, let u and v be two vertices of a TQ_n where the u's binary string and the v's differ exactly in one odd bit position i. There is an one-to-one and onto function $f : V_o(u) \rightarrow V_o(v)$ by one vertex of the $V_o(v)$ being an i-neighbor of the $V_o(u)$ in the S_u and the S_v.*

We say that S_u and S_v is adjacent if there is an one-to-one and onto function between vertices of them.

Theorem 1. *The structure of TQ_n consists of two levels of hierarchy. For an integer $m = \lceil \frac{n}{2} \rceil$, there is 2^{n-m} classes connected in a Hypercube fashion in the upper level of hierarchy, and each class is formed as a Q_m of 2^m vertices in the lower level.*

Proof: By Lemma 1 and 2, each class is formed as a Q_m. By lemma 3, each class connected in a hypercube fashion. □

Since a TQ_n can be transformed into a two-level hypercube-based hierarchical network, we introduce a lemma associated with the Q_n below which will be used in the following sections.

Lemma 4. *[4] For any subset $U \subseteq V(Q_n)$ of cardinality n, there is a set of vertices $W(\subset V(Q_n) - U)$ of cardinality n such that there is a matching of size n between U and W in Q_n.*

4 Cycle Partition

In the previous section, we show how to transform a TQ_n into 2^{n-m} m-cubes. Note, each m-cube contains a Hamiltonian cycle. Let C be the Hamiltonian cycle. Then we can get all the basic test results in C in two rounds. If we number the vertices in C, then the vertices of even number can test the vertices of odd number in the first round and the vertices of odd number can test the vertices of even number in the second round. In this section, we introduce an algorithm to partition a cycle into paths according to the test results obtained in the first two rounds in C.

Algorithm 1. *[1] CYCLE-PARTITION*

Step 1: Choose a 0-arrow a_0 followed by a 1-arrow.

Step 2: Let a be the arrow following a_0. If a is 0-arrow, then set $a_0 = a$ and proceed step 2; otherwise, go to step 3.

Step 3: Mark with an X the arrow following 1-arrow a. If it was not already marked, set a_0 the X-marked arrow and go to step 2; otherwise, the algorithm is terminated.

Now we can partition C into paths by removing the test result marked by X.

For example, as shown in Fig. 1, a cycle with sixteen vertices can be partitioned into three paths.

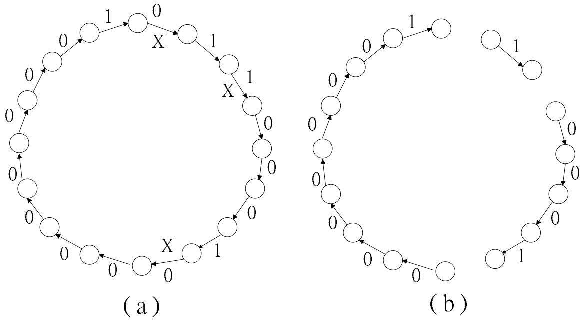


Fig. 1. (a) Before CYCLE-PARTITION (b) After CYCLE-PARTITION

Lemma 5. *[1] Each path contains at least one faulty vertex.*

5 Algorithm

In this section, we show that how to perform the diagnosis of the twisted cube in at most three rounds if $m \geq 5$ where $m = \lceil \frac{n}{2} \rceil$.

5.1 First Partition

At first, transforming the TQ_n into 2^{n-m} m-cubes, $H_1, H_2, \ldots, H_{2^{n-m}}$, and then executing the first two round tests in each m-cube's Hamiltonian cycle. Since $|V(Q_m)| \geq n + 1$, any m-cube with no 1-arrows must be fault-free. Any m-cube with at least one 1-arrow must contain at least one faulty vertex. Let x: the number of m-cubes which contain at least one 1-arrow and let S_1=$\{H_1, H_2, \ldots, H_x\}$ be the set of these m-cubes. In this paper, we suppose that there is an observer who can collect all the number of the paths in S_1 after CYCLE-PARTITION. Let s_i : the number of paths in H_i and let $s = \sum\limits_{i=0}^{x} s_i$. Next two subsections discuss how to identify faults according to the relation between x and $n - m$.

5.2 $x \leq n - m$

Note, a TQ_n can be treated as a Q_{n-m} of 2^{n-m} super vertices and S_1 can also be viewed as a set with x super vertices. Since $x \leq n - m$, by Lemma 4, there is a set $W \subset V(Q_{n-m}) - S_1$ such that there is a perfect matching of size $|S_1|$ between $|S_1|$ and W. Since each m-cube in W has been known as fault-free, each m-cube in S_1 can be tested by a fault-free m-cube, and the whole diagnosis will be done in three rounds.

5.3 $x > n - m$

Each m-cube in S_1 must contain at least one and at most m faulty vertices(In fact, if it contains $m + 1$ fault vertices, the total number of the fault vertices in the TQ_n will be at least $m + 1 + (x - 1) \geq n + 1$, a contradiction). According to Lemma 5 and CYCLE-PARTITION, we show an algorithm to distinguish the vertices of the m-cubes in S_1.

Algorithm 2. *[11] PATH-COLORING*

let $q = n - s + 1$. We give a color to each vertex in each path as follows.

1. If path has more than $q + 1$ vertices and has at least one 0-arrow, then we color the head of the path with black and color $q - 1$ vertices from the tail with grey, and all vertices that are not colored yet are colored white.

2. If path has at most $q + 1$ vertices, then we color all vertices in path with grey.

The white vertices must be fault-free, the black vertices must be faulty, and the grey vertices needs to be tested for identifying their status. Fig. 2 provides the results of performing PATH-COLORING on the three paths of Fig.1(b).

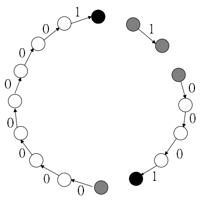


Fig. 2. An example of PATH-COLORING

Consider a Hamiltonian cycle constructed in the first two rounds, a (consecutive) part of the Hamiltonian cycle in a m-cube is called a *block*. A block is said to be "dirty" if it contains vertices to be colored with black or grey by PATH-COLORING and said to be "clean" if all the vertices in the block to be colored with white. Note that any vertex of the clean block is fault-free. Now we know that there are at least s faulty vertices in the TQ_n. By Lemma 5, each path contains at most $n-s+1$ faulty vertices. Now let $l = \lceil \log_2(n-s+1) \rceil$ and 2^l : the size of the block. Hence we can partition a m-cube into 2^{m-l} l-cubes, $B_1, B_2, \ldots, B_{2^{m-l}}$. Note that a path can produce at most two consecutive dirty blocks.

After counting, we have known the number of dirty blocks in each m-cube. Since $x > n-m$, it is possible that not all of the m-cubes in S_1 can be tested by the corresponding clean m-cubes. However, we hope some m-cubes can be diagnosed at first. In general, a m-cube contains more paths has less chances to be diagnosed by itself. So we can let m-cubes which owns more paths have higher priority to be tested by a clean m-cube. This can reduce the complexity of the whole diagnosis scheme. In other words, it means we can first make a maximum matching to let most of m-cubes be diagnosed with their respective clean neighbors in the three rounds. The remaining m-cubes cannot be matched with clean neighbors are discussed in the two cases below.

Case 1: If the number of dirty blocks in a m-cube doesn't exceed $m-l$, we can use the similar way to the case of $x \leq n-m$, the whole diagnosis will be done in three rounds.

Case 2: If the number of dirty blocks in some m-cube exceeds $m-l$, let S_2 be the set of such m-cubes. If a m-cube has only one or two dirty blocks, it's useful for the following discussion. We know that a m-cube with exactly one vertex has exactly one dirty block or two consecutive dirty blocks. So a m-cube with exactly one faulty vertex can have at least $2^{m-l}-4$ (there are at most two clean blocks are used to test the dirty block(s) in that m-cube)clean blocks to be uesd. We can use these m-cubes to test the m-cubes in S_2. Since the number of m-cubes which contains two or more faulty vertices doesn't exceed $m-1$ (if exceed $m-1$, the total number of the fault vertices in TQ_n will be at least $2m + (n-m+1-m) > n$, a contradiction), each m-cube in S_2 has at least $n-m-(m-2) = n-2m+2$ adjacent m-cubes which contains at most one faulty vertex. Note that $n-2m+2 \geq 2$. Hence we can assign two such m-cubes for each m-cube in S_2.

Let H_2 and H_3 be the two such m-cubes and H_1 be a m-cube in S_2 (The worst case is that two blocks in H_1 can't be tested by the blocks in H_2 and H_3, and such a situation occurs only when the dirty blocks in H_2 and H_3 are adjacent with those in H_1), we can use the clean blocks in H_2 and H_3 to test the dirty blocks in H_1. If the worst case occurs, let I be the set of those two dirty blocks in H_1. Since each m-cube contains at most m paths, we next discuss how to identify faults according to the relation between m and s_1 of H_1 in S_2 as the following subcases.

case 2.1 : $s_1 = m$

That is, H_1 has m faults. Since H_1 cannot be matched with a clean m-cube, all adjacent m-cubes of H_1 have faulty vertices. If there is a adjacent m-cube which has two or more faulty vertices, the total number of the faulty vertices in the TQ_n

will be $m + (n - m - 1) + 2 > n$, a contradiction. Hence each adjacent m-cube of H_1 has exactly one faulty vertex. Assigning a such m-cube to H_1 and almost all of the grey vertices in H_1 can be tested by the respective neighbors in the assigned m-cube. In the worst case, only one vertex can't be identified if its neighbor in the assigned m-cube is faulty. Since H_1 has m faults, each path of H_1 has exact one fault. That is, a grey vertex must belong to a path with only one edge. Note, such path has one 1-arrow and two end vertices of the 1-arrow are both grey. Clearly, one end vertex of the 1-arrow is faulty and another one is fault-free. Therefore, we only need to check the adjacent vertex of the unidentified vertex in the same 1-arrow. And the whole diagnosis will be done in three rounds.

Case 2.2 : $2 \leq s_1 \leq m - 3$

For all the adjacent m-cubes of H_1 have faulty vertices, $(n - m) + 2 \leq s \leq n$. If $s = m$, it's similar to the case of 2.1. If $(n - m) + 2 \leq s \leq n - 1$, $1 \leq l \leq \lceil \log_2(m - 1) \rceil$ and $m - l > s_1$ for $m \geq 5$. Each path can produce at most two consecutive dirty blocks. But we know that the adjacent blocks of the dirty block in I are not consecutive and each block in the m-cube has $m - l$ adjacent blocks, so each path contains at most one adjacent dirty block of the dirty block in I. Since $m - l > s_1$, it must exist at least one adjacent clean block of the dirty block in I. Hence the dirty blocks in I can be tested by the clean block(s). The whole diagnosis will be done in three rounds.

Case 2.3 : $s_1 = m - 2$

For all the adjacent m-cubes of H_1 have faulty vertices, $s \geq m - 2 + (n - m) = n - 2$. If $s = n - 1$, $l = 1$ and $m - l > s_1$, it's similar to the case 2.2. If $s = n$, it's similar to the case 2.1. If $s = n - 2$, then $l = 2$ and $m - l = s_1$ for $m \geq 5$. The dirty blocks in I can probably have adjacent clean blocks in m-cube, hence we first try to find if there is such clean blocks. If they exist, those dirty blocks can be tested by them and the whole diagnosis will be done in three rounds. If they don't exist, we have to discuss more deeply. Let B_i be such dirty block in I. Since each adjacent block of B_i is dirty, hence B_{i-1} and B_{i+1} must be dirty. Such three consecutive dirty blocks, B_{i-1}, B_i and B_{i+1}, must belong to at least two paths. And because $m - l = s_1$, B_i have exactly $m - l - 2$ adjacent blocks (not B_{i-1} and B_{i+1}) which belong to $s_1 - 2$ paths. Obviously, B_{i-1}, B_i and B_{i+1} must belong to exactly two paths. Now we can list all the case which can produce the situation described above and check if the whole diagnosis will be done in three rounds. By observing all the cases, they all will be done in three rounds. Although there are sixteen cases, we only show some in Fig. 3 for reducing paper length.

Case 2.4 : $s_1 = m - 1$

For all the adjacent m-cubes of H_1 have faulty vertices, $s \geq m - 1 + (n - m) = n - 1$. If $s = n - 1$, $l = 1$ and $m - l > s_1$, it's similar to the case 2.2. If $s = n$, it's similar to the case 2.1. If $s = n - 2$, then $l = 2$ and $m - l = s_1$ for $m \geq 5$. It's similar to case 2.3. By listing all the cases, the whole diagnosis will be done in three rounds. All the seven cases are shown in Fig.4.

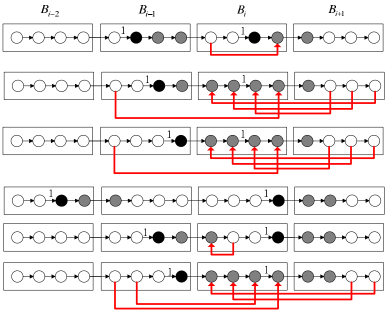


Fig. 3. Case 2.3

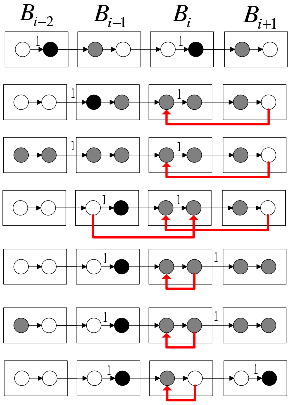


Fig. 4. Case 2.4

6 Conclusion

In this paper, we propose a scheme of adaptive diagnosis in the twisted cube with at most n faulty vertices for $n \geq 9$. And the whole diagnosis completes in at most three rounds through the scheme.

References

1. Preparata, F.P., Metze, G., Chien, R.T.: On the connection assignment problem of diagnosable systems. IEEE Transactions on Electronic Computers EC-16, 848–854 (1967)

2. Hakimi, S.L., Amin, A.T.: Characterization of connection assignment of diagnosable systems. IEEE Transactions on Computers 23, 86–88 (1974)
3. Nakajima, K.: A new approach to system diagnosis, pp. 697–706 (September 1981)
4. Fujita, S., Araki, T.: Three-Round Adaptive Diagnosis in Binary *n*-Cubes. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 442–451. Springer, Heidelberg (2004)
5. Hilbers, P.A.J., Koopman, M.R.J., van de Snepscheut, J.L.A.: The Twisted Cube. In: Treleaven, P.C., Nijman, A.J., de Bakker, J.W. (eds.) PARLE 1987. LNCS, vol. 258, pp. 152–159. Springer, Heidelberg (1987)
6. Abraham, S., Padmanabhan, K.: The twisted cube topology for multiprocessors: A study in network asymmetry. Journal of Parallel and Distributed Computing 13, 104–110 (1991)
7. Chang, C.P., Wang, J.N., Hsu, L.H.: Topological properties of twisted cube. Informatics and Computer Science: An International Journal 113, 147–167 (1999)
8. Lai, P.L., Tasi, C.: Embedding of tori and grids into twisted cubes. Theoretical Computer Science 411(40-42), 3763–3773 (2010)
9. West, D.B.: Introduction to Graph Theory. Prentice Hall (2001)
10. Fana, J., Linb, X., Panc, Y., Jiaa, X.: Optimal fault-tolerant embedding of paths in twisted cubes. Journal of Parallel and Distributed Computing 67, 205–214 (2007)
11. Araki, T.: Optimal adaptive fault diagnosis of cubic hamiltonian graphs, pp. 162–167 (May 2004)

On the Min-Max 2-Cluster Editing Problem

Li-Hsuan Chen[1], Maw-Shang Chang[2], Chun-Chieh Wang[1], and Bang Ye Wu[1,⋆]

[1] National Chung Cheng University, ChiaYi, Taiwan 621, R.O.C.
bangye@ccu.edu.tw
[2] Hungkung University, Taichung, Taiwan 433, R.O.C.

Abstract. In this paper, we study the problem Min-Max 2-Cluster Editing which asks for a modification of a given graph into two maximal cliques by inserting or deleting edges such that the maximum number k of the editing edges incident to any vertex is minimized. We show the NP-hardness of the problem and present a polynomial-time algorithm when $k < n/4$, in which n is number of vertices. In addition, we design a 2-approximation algorithm and a branching algorithm for finding an optimal solution. By experiments on random graphs, we show that the exact algorithm is much more efficient than a trivial one.

Keywords: algorithm, clustering editing, graph modification problem, NP-hard, approximation algorithm.

1 Introduction

A *cluster graph* is an undirected graph consisting of disjoint maximal cliques. The maximal cliques in a cluster graph are also called *clusters*. Given a graph, the problem Cluster Editing looks for the minimum number of edge insertions and deletions to modify the input graph to a cluster graph [21]. Cluster Editing is closely related to *graph clustering* and attracts many researchers. For an integer p, a cluster graph is a *p-cluster graph* if the number of clusters is at most p, and an important variant is p-Cluster Editing which modify the input graph to a p-cluster graph.

In this paper we study a natural variant, named Min-Max 2-Cluster Editing, of Cluster Editing. While Cluster Editing minimizes the total number of editing edges, for Min-Max Cluster Editing we hope that the number of inserting and deleting edges incident to any vertex is small. The decision version is called *max-k-editable* which determines if a graph can be modified into a 2-cluster graph in such a way that the number of editing edges incident to each vertex is at most k.

Shamir et al. [21] studied the computational complexities of three edge modification problems. Cluster Editing asks for the minimum total number of edge insertions and deletions, while in Cluster Deletion (respectively, Cluster Completion), only edge deletions (respectively, insertions) are allowed. They

[⋆] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 133–142.
DOI: 10.1007/978-3-642-35452-6_16

showed that Cluster Editing is NP-hard, Cluster Deletion is Max SNP-hard, and Cluster Completion is polynomial-time solvable. For a constant integer $p \geq 2$, p-Cluster Deletion (similarly, p-Cluster Editing) is a variant of Cluster Deletion in which the desired solution must contain exactly p clusters. They also showed that p-Cluster Deletion is NP-hard for any $p > 2$ but polynomial-time solvable for $p = 2$, and p-Cluster Editing is NP-hard for any $p \geq 2$. In the literature, there are several results on the fixed-parameter time complexities for Cluster Editing and Cluster Deletion, for example [5, 8–10, 15–17], and the most recent result can be referred to [4]. A variant with vertex (rather than edge) deletions was considered in [19], and another variant in which overlapping clusters are allowed was studied in [11].

Although Min-Max 2-Cluster Editing is a natural variant, there seems no study in the literature. In this paper, we show the following results.

- Min-Max 2-Cluster Editing is NP-hard.
- If the input graph is k-editable with $k < n/4$, Min-Max 2-Cluster Editing can be solved in $O(n^2)$ time.
- We design a heuristic algorithm which always reports a 2-approximation solution.
- We also present a branching algorithm for solving the problem exactly. By designing good reduction rules, the algorithm performs much better than the trivial brute force method. We show the performance by experiments.

The rest of the paper is organized as follows. In Section 2, we give some notation and definitions used in this paper. We show the NP-hardness in Section 3 and the polynomial-time solvable case is shown in Section 4. The approximation algorithm is presented in Section 5. The branching algorithm and experimental results are in Section 6. Finally concluding remarks are given in Section 7.

2 Preliminaries

The set-minus is denoted by "\", i.e., for two sets S and S', $S \setminus S' = \{e | e \in S \text{ and } e \notin S'\}$. For a graph G, $V(G)$ and $E(G)$ denote the vertex and the edge sets, respectively. In this paper, a graph is always undirected and simple. Two vertices u and v are *neighbors* of each other if $(u, v) \in E$. For a vertex subset U, the subgraph of G induced by U is denoted by $G[U]$. For an edge subset F, the subgraph induced by F is the graph whose edge set is F and its vertex set contains all the endpoints of edges in F. Since there will be no confusion from the context, we also use $G[F]$ to denote the edge-induced subgraph.

The degree of a vertex v in a graph G, denoted by $d_G(v)$, is the number of its neighbors in G. For a vertex subset U, let $d(v, U)$ denote the number of neighbors of v in U, i.e., $d(v, U) = |\{u \in U | (u, v) \in E\}|$. Let $\bar{d}(v, U) = |\{u \in U | (u, v) \notin E\}|$. For $F \subset V \times V$, we use $d_F(v)$ to denote $d_H(v)$ in which $H = (V, F)$.

A *clique* is a complete subgraph. A k-clique is a clique of k vertices. A clique is *maximal* if it is not properly contained in another clique. A *maximum clique* of a graph is a clique of maximum number of vertices.

For two sets S and T, let $S \oplus T = (S \setminus T) \cup (T \setminus S)$ denote the symmetric difference. For $G = (V, E)$, a set $F \subset V \times V$ is an *editing set* if $G' = (V, E \oplus F)$ is a 2-cluster graph.

Problem: Max-k Editable
Instance: A graph $G = (V, E)$ and a positive integer k.
Question: Is there an editing set F such that $d_F(v) \leq k$ for each $v \in V$?

We say that a graph is *max-k-editable*, or simply k-editable, if such an editing set exists. The optimization version will be called Min-Max 2-Cluster Editing, which finds the minimized k such that the given graph G is k-editable.

A *2-partition* $\pi = (V_1, V_2)$ of V is an unordered pair of vertex subsets V_1 and V_2 such that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. Immediately, $G' = (V, E \oplus F)$ is a 2-cluster graph if and only if there exists a 2-partition $\pi = (V_1, V_2)$ of V such that both $G'[V_1]$ and $G'[V_2]$ are cliques, as well as $(u, v) \notin E \oplus F$ for any $u \in V_1$ and $v \in V_2$. We say that π is the *feasible* 2-partition if G is k-editable. At a glance, it seems that we should try to find the best editing set for Max-k Editable. But it is sometimes more convenience to find the feasible 2-partition.

Two vertices u and v are co-clustered in π if they are in the same cluster. For a 2-partition π, define the *conflict* between two vertices as follows.

$$C_\pi(u, v) = \begin{cases} 1 & \text{if } (u, v) \notin E \text{ and they are co-clustered; or} \\ & (u, v) \in E \text{ and they are not co-clustered} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The relation between conflicts and editing set comes from the definitions.

Lemma 1. *Suppose that G is k-editable and F is the editing set. Let $G' = (V, E \oplus F)$ and π be the corresponding feasible 2-partition. Then, $C_\pi(u, v) = 1$ if and only if $(u, v) \in F$.*

For a vertex subset U, let $C_\pi(v, U) = \sum_{u \in U} C_\pi(v, u)$, and the *conflict number* of a vertex v is $C_\pi(v) = \sum_{u \in V} C_\pi(v, u)$. Max-$k$ Editable is equivalent to determining if there exists a 2-partition π such that $C_\pi(v) \leq k$ for each $v \in V$.

3 NP-Completeness

We shall show the NP-completeness of Max-k Editable by reducing from the 3-SAT problem. An instance of 3-SAT consists of n variables x_i, $1 \leq i \leq n$, and m clauses C_i, $1 \leq i \leq m$, in which each clause contains exactly three literals. We make the following two restrictions on the 3-SAT problem.

1. Assigning all variables false cannot satisfies all the clauses.
2. There is no true assignment making all literals of all clauses true.

It is not hard to see the restrictions do not affect the NP-completeness.

For an instance of the 3-SAT problem, we add additional n variables x_i, $n+1 \leq i \leq 2n$, and construct a graph $G = (V, E)$ as follows, in which V consists of pairwise disjoint subset M, U, and Q. Let $W > \max\{n, m\}$ be a sufficiently large integer and $J(i) = \{j | x_j \in D_i \vee \neg x_j \in D_i\}$ be the set of indexes for which variable x_j is involved in clause D_i.

- There is a vertex D_i for each clause. Let $M = \{D_i | 1 \leq i \leq m\}$ be a clique.
- Let U be the set of $2n$ disjoint cliques defined as follows. For $1 \leq i \leq 2n$, let X_i be a clique containing $2W$ vertices.
- Q is a clique of $4W$ vertices and there is no edge linking a vertex of Q to any vertex outside Q.
- For each clause D_i, if $x_j \in D_i$, then D_i is connected to all vertices of X_j. If $\neg x_j \in D_i$, then there is no edge between vertex D_i and any vertex in X_j. Otherwise, for $j \notin J(i)$, D_i is connected to W vertices in X_j. Note that D_i is connected to W vertices in X_j for any $n+1 \leq j \leq 2n$ since neither x_j nor $\neg x_j$ is in D_i.
- $k = (2n + 1)W$.

Definition 1. *For a 2-partition $\pi = (V_1, V_2)$, a vertex subset Y is* not split *by π if Y is entirely in V_1 or V_2. π is* regular *if it meets the following conditions: (1) $M \subset V_1$; (2) $Q \subset V_2$; and (3) X_j is not split by π for any j. For a 2-partition π not splitting any X_j, we define $T[\pi]$ as a truth assignment such that x_i is true if and only if $X_i \subset V_1$. Let $\bar{T}[\pi]$ be the reverse true assignment of $T[\pi]$.*

Since a 2-partition is an unordered pair of vertex subsets, $(V_1, V_2) = (V_2, V_1)$ and we shall assume $M \subset V_1$ and $Q \subset V_2$ when (V_1, V_2) is regular.

Proposition 1. *Suppose that $\pi = (V_1, V_2)$ is a regular 2-partition. The clause D_i is satisfied by $T[\pi]$ if and only if $C_\pi(D_i) \leq (2n + 1)W$.*

Proof. Since π is regular, $C_\pi(D_i, M) = C_\pi(D_i, Q) = 0$ and therefore $C_\pi(D_i) = C_\pi(D_i, U)$. For any literal $y_j = x_j$ or $\neg x_j$ in D_i, y_j is true if and only if $C_\pi(D_i, X_j) = 0$, and otherwise $C_\pi(D_i, X_j) = 2W$. For any variable x_j such that $j \notin J(i)$, $C_\pi(D_i, X_j) = W$ no matter which cluster X_j belongs to. There are exactly $2n - 3$ such variables since D_i contains three literals.

If D_i is satisfied by $T[\pi]$, at least one of its literals is true, and therefore $C_\pi(D_i, U) \leq 4W + (2n-3)W = (2n+1)W$. Conversely if $C_\pi(D_i, U) \leq (2n+1)W$, at least one of its literals is true.

Lemma 2. *If there is a truth assignment satisfying all the clauses, then G is k-editable.*

Proof. (Sketched) The corresponding regular 2-partition $\pi = (V_1, V_2)$ is as follows. For each $1 \leq j \leq n$, $X_j \subset V_1$ if and only if x_j is true. The other cliques X_j are distributed such that V_1 contains exactly $n + 1$ cliques of U. It is not hard to show that π is a feasible 2-partition. □

Next we show the other direction.

Lemma 3. *If G is k-editable, then there is a truth assignment satisfying all the clauses.*

Suppose that G is k-editable. By definition there exists a 2-partition $\pi = (V_1, V_2)$ such that $C_\pi(v) \leq k$ for each $v \in V$. The key point is to show that π must be regular, and then Lemma 3 is immediately followed from Proposition 1.

Claim. For $1 \leq i \leq 2n$, clique X_i is not split by π.

Proof. Let $U^+ = U \cup Q$. Suppose by contradiction that X_i is split. For $y_1 \in X_i \cap V_1$,

$$C_\pi(y_1, U^+) = |X_i \cap V_2| + |V_1 \cap (U^+ \setminus X_i)|$$

and for $y_2 \in X_i \cap V_2$,

$$C_\pi(y_2, U^+) = |X_i \cap V_1| + |V_2 \cap (U^+ \setminus X_i)|.$$

Since the sum of the right-hand sides of the above two equations is $|U^+| = (4n+4)W$, we have

$$\max_{y \in X_i} C_\pi(y) \geq (1/2)(C_\pi(y_1, U^+) + C_\pi(y_2, U^+)) = (2n+2)W > k,$$

which is a contradiction and therefore X_i is not split. □

The next claim can be shown similarly

Claim. Clique Q is not split by π.

By the above claim, we may assume $Q \in V_2$. Next we claim the vertices of U^+ are evenly distributed.

Claim. $|V_1 \cap U^+| = |V_2 \cap U^+| = (2n+2)W$.

Proof. By the above claims, cliques in U^+ are not split. If U^+ is not evenly divided, one of them contains at least $(2n+4)W$ vertices. For any vertex v in this set but not in Q, since at most $2W-1$ of them are its neighbors, $C_\pi(v) \geq (2n+2)W > k$, which is a contradiction. □

Claim. $M \subset V_1$.

Proof. Let $M_i = M \cap V_i$ for $i = 1, 2$. For $D_i \in M_2$,

$$C_\pi(D_i) \geq C_\pi(D_i, Q) + C_\pi(D_i, U) + |M_1| = 4W + (2n-3)W + |M_1|, \quad (2)$$

and the equality holds only when $C_\pi(D_i, X_j) = 0$ for any $j \in J(i)$. That is, all the three literals are set true by $\bar{T}[\pi]$.

If $M_1 \neq \emptyset$, we have $C_\pi(D_i) > k$. Therefore M_1 or M_2 is empty. If $M_1 = \emptyset$ and the equality of Eq. (2) holds for each D_i, then $\bar{T}[\pi]$ satisfies all literals of all clauses. It is a contradiction to our restriction 2 of the 3-SAT problem, and therefore we obtain $M_2 = \emptyset$. □

Lemma 3 is shown by the above claims and the next theorem follows from Lemmas 2 and 3.

Theorem 1. Min-Max 2-Cluster Editing *is NP-complete.*

4 Polynomial-Time Solvable Case

In this section we show how to solve the problem when $k < n/4$. Here and after we shall let $G = (V, E)$ be the input graph and $n = |V|$.

We define an operation "moving vertices" on 2-partitions as follows. Let $\pi = (V_1, V_2)$ be a 2-partition of V. Another 2-partition π' is obtained by moving a vertex u in π if $\pi' = (V_1 \oplus \{u\}, V_2 \oplus \{u\})$. Recall that $\oplus$ is the symmetric difference, and therefore

$$V_i \oplus \{u\} = \begin{cases} V_i \setminus \{u\} \text{ if } u \in V_i \\ V_i \cup \{u\} \text{ if } u \notin V_i \end{cases} \tag{3}$$

In other words, u is moved from its original part to the other. The operation of moving a set of vertices is defined similarly, and we denote by $\pi' = \Delta_U(\pi)$ that π' is obtained by moving a set U of vertices in π. The next relation follows from definitions directly.

$$C_{\pi'}(u, v) = \begin{cases} 1 - C_\pi(u, v) \text{ if exactly one of } u \text{ and } v \text{ is in } U \\ C_\pi(u, v) \qquad \text{otherwise} \end{cases} \tag{4}$$

Consequently, if $\pi' = \Delta_{\{u\}}(\pi)$, then $C_{\pi'}(u) = n - 1 - C_\pi(u)$.

For any vertex v, the *v-partition* is a 2-partition $\pi = (V_1, V - V_1)$, in which $V_1 = \{v\} \cup \{u | (u, v) \in E\}$.

Lemma 4. *If G is k-editable, for any $v \in V$ there exists $U \subset V$ with $|U| \leq k$ such that $\pi^* = \Delta_U(\pi)$ is feasible, in which π is the v-partition.*

Proof. By the assumption, there exists a vertex 2-partition π^* such that $C_{\pi^*}(u) \leq k$ for each $u \in V$. Therefore for any v, the size of the set U defined by $U = \{u | C_{\pi^*}(v, u) = 1\}$ is less than or equal to k. By definition, moving U in π^* results in the v-partition π, i.e., $\pi = \Delta_U(\pi^*)$. □

We shall call the set U in Lemma 4 a *feasible moving set.*

Lemma 5. *Suppose that π is the v-partition for arbitrary v and G is k-editable with $k < n/4$. For any vertex u and any feasible moving set U, if $C_\pi(u) \geq n/2$, then $u \in U$.*

Proof. By Lemma 4, there exists a feasible partition $\pi^* = \Delta_U(\pi)$ with $|U| \leq k$. Since $k < n/4$, if $C_\pi(u) \geq n/2$, we have $C_\pi(u) > 2k$. If $u \notin U$, moving U will reduce the conflict number of u by at most k and $C_{\pi^*}(u) > (n/2) - (n/4) > k$, which is a contradiction. Therefore $u \in U$. □

Lemma 6. *Suppose that π is the v-partition for arbitrary v and G is k-editable with $k < n/4$. For any vertex u and any feasible moving set U, if $C_\pi(u) < n/2$, $u \notin U$.*

Proof. By Lemma 4, there exists a feasible partition $\pi^* = \Delta_U(\pi)$ with $|U| \leq k$. By the assumption $k < n/4$, if $u \in U$ and $\pi' = \Delta_{\{u\}}(\pi)$, $C_{\pi'}(u) = n-1-C_\pi(u) \geq n/2$. Moving the other $|U|-1$ vertices will still have $C_{\pi^*}(u) \geq (n/2)-|U|+1 > k$, which is a contradiction. Therefore $u \notin U$. □

Theorem 2. *If G is k-editable and $k < n/4$, the problem* Min-Max 2-Cluster Editing *can be solved in $O(n^2)$ time.*

Proof. Starting from the v-partition π for an arbitrary vertex v, we compute $U = \{u | C_\pi(u) \geq n/2\}$. Then we move U to obtain $\pi' = \Delta_U(\pi)$, and compute $k' = \max_u C_\pi(u)$. It is trivial that all the steps can be done in $O(n^2)$ time.

If G is k-editable with $k < n/4$, by the above two lemmas, each vertex either must be moved or cannot be moved. Therefore π' is the only possible feasible 2-partition. If $k' < n/4$, we obtain the minimized k; and otherwise there is no feasible solution with $k < n/4$. □

5 An Approximation Algorithm

Consider the greedy algorithm beginning from an arbitrary partition π, repeatedly moves the vertex u with the maximum conflict number until all the vertices have conflict number at most $n/2$.

Lemma 7. *The greedy algorithm stops within n^2 iterations and takes $O(n^3)$ time.*

Proof. Consider the total conflict number over all vertices. The result follows from two facts. First the total conflict number strictly decreases after each iteration. Second, the initial total conflict number is at most $n(n-1)$. The time complexity follows from each iteration takes $O(n)$ time to update the conflict numbers and find the vertex to move. □

Corollary 1. *Any undirected simple graph is $n/2$-editable.*

By Theorem 2, we can check in $O(n^2)$ time if a graph is k-editable with $k < n/4$, as well as find the optimal k if the answer is yes. If the answer is no, we can find a feasible solution for $k \leq n/2$ by the greedy algorithm. Since in this case the optimal k is at least $n/4$, we find a 2-approximation. Therefore we have the next result.

Theorem 3. *A 2-approximation solution of* Min-Max 2-Cluster Editing *can be found in $O(n^3)$ time.*

6 A Branching Algorithm and Experiments

In this section we present an exact algorithm for Min-Max 2-Cluster Editing. In the following we focus on the decision version. That is, the algorithm determines if the input graph is k-editable. To find the minimized k, we first check if there is a solution with $k < n/4$. If not, we start from $k = n/4$ and iteratively increase k one by one to find the minimum k such that G is k-editable. Since Max-k Editable looks for a feasible 2-partition, a straightforward algorithm is to check the existence of a feasible 2-partition. But our algorithm focuses on the feasible moving set instead of the 2-partition. It takes the advantage that the size of a feasible moving set is at most k while there are 2^{n-1} possible 2-partitions. More importantly, we derive reduction rules to reduce the practical running time significantly. The efficiency will be shown by experimental results later.

In the branching algorithm, all vertices are divided into three subsets: M, $\bar{M}$, and U, which contain the vertices must in the moving set, must not in the moving set, and undetermined, respectively. Starting from a v-partition for an arbitrarily chosen vertex v and an initial moving set $M = \emptyset$, the algorithm chooses an undetermined vertex u and recursively searches the two branches: adding u into M or into $\bar{M}$. The worst case time complexity is exponential since the number of possible moving sets is $\binom{n}{k}$. But we may improve the practical performance by designing some reduction rules to avoid exhaustive search.

In section 4, we have derived two rules. When $k < n/4$, every vertex can be determined if it need to move to obtain a feasible partition. However, when $k \geq n/4$, not all the vertices can be determined. But we can generalize it to design reduction rules. For a partial solution (M, U) and an initial v-partition π, we can have the following two rules. Let $\pi' = \Delta_M(\pi)$.

Lemma 8. *For any $u \in U$, if $C_{\pi'}(u) > 2k - |M|$, then any feasible moving set $M' \supset M$ must contain u.*

Proof. By definition $|M'| \leq k$. Since $C_{\pi'}(u) > 2k - |M|$, if $u \notin M'$, moving $M' \backslash M$ from π' will reduce the conflict number of u by at most $k - |M|$ and result in a conflict number at least $C_{\pi'}(u) - (k - |M|) > k$, which is a contradiction. □

Lemma 9. *For any $u \in U$, if $C_{\pi'}(u) < n - 2k + |M|$, then any feasible moving set $M' \supset M$ cannot contain u.*

Proof. Since $C_{\pi'}(u) < n - 2k + |M|$, if $u \in M'$, then $C_{\pi''}(u) > n - 1 - (n - 2k + |M|) = 2k - 1 - |M|$, in which $\pi'' = \Delta_{\{u\}}(\pi')$. Then moving $M' \setminus (M \cup \{u\})$ in π'' will reduce the conflict number of u by at most $k - |M| - 1$ and result in a conflict number larger than $2k - 1 - |M| - (k - |M| - 1) = k$, which is a contradiction. □

In the remaining paragraphs of this section, we show the experimental results. Two types of random graphs are used in the experiments. In a random graph of the first type, an edge exists with an independent identical probability p. For the second type, a graph is generated from a 2-cluster graph and an editing set F such that $|F| = m$ and there are l vertices with degree k in F. Three kinds of experiments were performed: efficiency of the exact algorithm, practical performance of the approximation algorithm, and relation between optimal solution and the density of edges.

In Tables 1, we compare the execution times (in seconds) of the exact algorithm with and without reduction rules for Type-1 data. The numbers in the parentheses are running times without reduction rules. It is shown that using the reduction rules greatly reduces the execution time, especially on the graphs with higher density. In the density cases $p = 0.8$, the optimal solutions on graphs

Table 1. Running times for Type-1 random graphs

p	n=30	n=35	n=40	n=50	n=300
0.2	1.59(3.46)	13.1(48.6)	> 100	> 100	> 100
0.5	1.07(1.94)	10.7(21.2)	> 100	> 100	> 100
0.8	0.005(0.055)	0.0008(0.195)	0.0013(1.42)	$< 1(> 100)$	$< 1(> 100)$

with 300 vertices can be found within one second. We also performed tests on Type-2 data. The experimental results show that for $n = 100$ and $l \leq 10$, the algorithm with reduction runs very fast for $n/4 < k \approx n/3$.

In the second experiment, the approximation ratios are tested on both types of graphs with vertices less than 100 and $k \leq n/3$. In all the tests we performed, the approximation ratios are all within 1.3.

In the third experiment, the average solution in 20 Type-1 random graphs with 30 vertices and density 0.9, 0.8, and 0.7 are 6.4, 9.85, 12.9 respectively. The result shows that the maximum editing number rises very fast as the density decreases, and when the density is below 0.7, the editing number is very close to $n/2$.

7 Concluding Remarks

In addition to good approximation algorithms and exact algorithms for Min-Max 2-Cluster Editing, interesting future works also include how to generalize the results of this paper to the version that the number of cluster is more than two or the number of clusters is not specified.

Acknowledgment. L.-H. Chen and M.-S. Chang were supported in part by NSC 99-2221-E-241-015-MY3, and B.Y. Wu and C.-C. Wang were supported in part by NSC 100-2221-E-194-036-MY3 and NSC 101-2221-E-194-025-MY3 from the National Science Council, Taiwan, R.O.C.

References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: Ranking and clustering. J. ACM 55(5), 1–27 (2008)
2. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation: Combinatorial optimization problems and their approximability properties. Springer (1999)
3. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning, Special Issue on Clustering 56, 89–113 (2004)
4. Böcker, S., Damaschke, P.: Even faster parameterized cluster deletion and cluster editing. Information Processing Letters 111(14), 717–721 (2011)
5. Böcker, S., Briesemeister, S., Bui, Q.B.A., Truss, A.: Going weighted: Parameterized algorithm for cluster editing. Theor. Comput. Sci. 410(52), 5467–5480 (2009)

6. Bonizzoni, P., Della Vedova, G., Dondi, R.: A ptas for the minimum consensus clustering problem with a fixed number of clusters. In: Proc. Eleventh Italian Conference on Theoretical Computer Science (2009)
7. Bonizzoni, P., Della Vedova, G., Dondi, R., Jiang, T.: On the approximation of correlation clustering and consensus clustering. Journal of Computer and System Sciences 74(5), 671–696 (2008)
8. Chen, J., Meng, J.: A $2k$ Kernel for the Cluster Editing Problem. In: Thai, M.T., Sahni, S. (eds.) COCOON 2010. LNCS, vol. 6196, pp. 459–468. Springer, Heidelberg (2010)
9. Damaschke, P.: Bounded-Degree Techniques Accelerate Some Parameterized Graph Algorithms. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 98–109. Springer, Heidelberg (2009)
10. Damaschke, P.: Fixed-parameter enumerability of cluster editing and related problems. Theory Computing Syst. 46, 261–283 (2010)
11. Fellows, M.R., Guo, J., Komusiewicz, C., Niedermeier, R., Uhlmann, J.: Graph-based data clustering with overlaps. Discrete Optimization 8(1), 2–17 (2011)
12. Filkov, V., Skiena, S.: Integrating microarray data by consensus clustering. International Journal on Artificial Intelligence Tools 13(4), 863–880 (2004)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, NewYork (1979)
14. Giotis, I., Guruswami, V.: Correlation clustering with a fixed number of clusters. Theory Comput. 2, 249–266 (2006)
15. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Graph-modeled data clustering: Fixedparameter algorithms for clique generation. Theory Computing Syst. 38, 373–392 (2005)
16. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Automated generation of search tree algorithms for hard graph modification problems. Algorithmica 39, 321–347 (2004)
17. Guo, J.: A more effective linear kernelization for cluster editing. Theor. Comput. Sci. 410, 718–726 (2009)
18. Harary, F.: On the notion of balance of a signed graph. Michigan Mathematical Journal 2(2), 143–146 (1953)
19. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. Theory of Computing Systems 47(1), 196–217 (2010)
20. Micali, S., Vazirani, V.V.: An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In: FOCS, pp. 17–27 (1980)
21. Shamir, R., Sharan, R., Tsur, D.: Cluster graph modification problems. Discr. Appl. Math. 144(1-2), 173–182 (2004)
22. Talmaciu, M., Nechita, E.: Recognition algorithm for diamond-free graphs. Informatica 18(3), 457–462 (2007)
23. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press, Cambridge (1994)

A New Filtration Method and a Hybrid Strategy for Approximate String Matching

Chia Wei Lu, Chin Lung Lu, and R.C.T. Lee

Department of Computer Science, National Tsing Hua University, Hsinchu City, Taiwan
d9762807@oz.nthu.edu.tw, cllu@cs.nthu.edu.tw, rctlee@ncnu.edu.tw

Abstract. In this paper, we propose a new filtration algorithm, as well as a hybrid filtration strategy, to efficiently solve the approximate string matching problem (also called *k* differences problem), which aims to find all the positions *i*'s in a given text such that there exists a substring of the text ending at position *i* whose edit distance from a given pattern is less than or equal to a given error bound *k*. Our experimental results on simulated datasets of DNA sequences show that our filtration algorithm has better performance on the efficiency to filter out those positions of the text at which the pattern does not occur approximately. Moreover, our hybrid filtration strategy further improves the performance efficiency of our filtration algorithm greatly when the ratio of the error bound and the pattern size is about 0.2.

Keywords: approximate string matching, filtration, *q*-gram, hybrid.

1 Introduction

Before defining the approximate string matching problem, we first introduce the edit distance [21] to measure the similarity between two strings. The *edit distance* between two strings *A* and *B*, denoted by $ED(A,B)$, is the minimum number of insertion, deletion or substitution operations needed to transform *B* into *A*. Basically, the edit distance can be computed by using dynamic programming method [28]. In this paper, we are interested in the *approximate string matching problem* defined as follows: Given a text string $T = t_1t_2\dots t_n$, a pattern string $P = p_1p_2\dots p_m$ and an error bound *k*, the problem is to find all *i*'s such that $de_i \le k$, where $1 \le i \le n$ and $de_i = \min_{i-m-k+1\le j\le i} ED(P, t_j \dots t_i)$. The problem is also known as the *k differences problem* [16].

In general, there are two approaches to solve the *k* differences problem. The first approach, called *non-filtration approach*, is to check every position of *T* using the dynamic programming method. The first non-filtration algorithm [22] has a running time of O(*mn*). Some studies have improved the time complexity to O(*kn*) by using the properties of the dynamic programming matrix [12, 24]. Some other studies have used bit-parallelism in a computer word to reduce the number of edit operations so that the approximate string matching problem can be solved in $O(nk\lceil m/w \rceil)$ time [10, 13, 29].

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 143–155.
DOI: 10.1007/978-3-642-35452-6_17

Another approach to solve the k differences problem, named *filtration approach*, uses an efficient filtration algorithm to eliminate uninteresting regions of the text T, and then checks for possible solutions at the locations which are not eliminated by using the non-filtration algorithm. The filtration approach can be detailed in the following two steps.

- **Step 1:** Consider a window of T ending at t_i. Compute a filtration function F. If the computation of F indicates that $de_i > k$, then ignore the position i; otherwise, mark i.
- **Step 2:** For each i which is marked in Step 1, compute de_i by using a non-filtration algorithm and check if $de_i \le k$.

In general, computing de_i is much slower than computing F. The key of filtration approach is on the filtration function. Basically, the effectiveness of the filtration function F can be measured by counting the number of positions which can be eliminated. It has been shown that k differences problem can be efficiently solved by filtration algorithms if many locations can be eliminated [8, 16]. Therefore, it is meaningful to investigate the filtration approach because it will make the algorithms very efficient in average-case time complexity. We refer the readers to Section 8 in [16] for more details about the filtration algorithms. In this paper, we shall give an efficient method to filter out many locations of the text. In the following, we categorize the existing filtration algorithms into three different types.

1.1 Partial Exact Matching Approach

The idea of the partial exact matching approach is to search some substrings of P which exactly appear in T first. Suppose that we have a pattern P and we want to find a substring S in T such that $ED(P,S) \le k$. Then for reasonably small k, there must exist a substring in P which exactly appears in S. This approach therefore performs an exact string matching first. Different algorithms select different substrings in P to do the exact string matching, as described below.

Some algorithms [2, 3, 17, 18, 19, 29] used the fact that if $ED(P,S) \le k$, then after dividing P into (k+1) non-overlapping pieces, at least one of these (k+1) pieces must appear in the string S exactly, where S is a substring of T. Hence, the filtration step for these algorithms is first to divide P into (k+1) non-overlapping pieces that have approximately equal length. Then, for each window $W = T(i-m+1, i)$, it checks whether any of the divided pieces in P appears in W exactly. If none of the pieces appears exactly in W, then the checking step ignores this window W; otherwise, it checks W by using a non-filtration algorithm.

Other algorithms [4, 5, 6] found the maximal substrings of T that also appear in P. If a substring $S = t_i t_{i+1} \ldots t_j$ of T is also a substring of P while $t_i t_{i+1} \ldots t_j t_{j+1}$ is not, then we call the substring S is a *maximal substring* of T. The algorithm proposed by Chang and Lawler [4] works as follows: Traverse T from the beginning and consider it as the form $T = S_1 y_1 S_2 y_2 \ldots$, where S_i is a maximal substring that can be empty and y_i is a character. For each i, if the total size of (k+1) continuous (S_i, y_i) pairs is less than

$m-k$, i.e., $|S_i y_i S_{i+1} y_{i+1} \dots S_{i+k+1} y_{i+k+1}| < m-k$, then it can be proved that there is no solution starting from every position inside $S_i y_i$ and hence we can skip $S_i y_i$; otherwise, we need to check the region $S_i y_i S_{i+1} y_{i+1} \dots S_{i+k+1} y_{i+k+1}$. Note that $m-k$ is the minimum length of a possible solution. This algorithm, denoted by *LET* for simplicity, has a linear expected time. In [4], the authors also gave an algorithm, called *SET*, which runs in sublinear expected time. *SET* is similar to *LET* except that *T* is split into blocks with equal length $(m-k)/2$. It is obvious that any solution must contain at least one of these blocks. *SET* processes the text starting from the beginning of each block and finds first $k+1$ maximal substrings S_1, S_2, …, S_{k+1} in *T*, where S_i and S_{i+1} are separated by a character y_i in *T*. For each block, if the ending position of the $(k+1)$ th maximal substring, say *x*, is inside this block, then there is no solution containing this block and we can skip this block; otherwise, *SET* checks the substring of *T* starting from $(m+3k)/2$ characters to the left of the block and ending at *x*.

1.2 Bad *W*-Suffix Approach

Consider a window *W* of length $m-k$ in *T*. We define a *W-suffix* to be any suffix of the string *W*. If the distance between a *W*-suffix and every substring of *P* is larger than *k*, we call this *W*-suffix as a *bad W-suffix*. For example, if $P = acaac$, $W = agga$ and $k = 1$, then there is a bad *W*-suffix, $'gga'$, whose edit distance with every substring of *P* is larger than 1. It is not possible that a solution of the *k* difference problem contains a bad *W*-suffix. Thus, if there exists a bad *W*-suffix in *W*, then we can safely shift the window to the right such that it does not contain the whole bad *W*-suffix. More precisely, for a window $W = T(i-m+k+1, i)$, if there is a bad *W*-suffix equal to $T(j, i)$ for some *j*, where $i-m+k+1 \le j \le i$, we can conclude that there is no solution of the *k* difference problem starting from the positions in the region $T(i-m+k+1, j)$ and we can safely shift the window to the position $j+1$. This is due to the fact that any solution is a substring of *T* with length larger than or equal to $m-k$ and if the solution starts inside the region $T(i-m+k+1, j)$, it will definitely contain the bad *W*-suffix since the size of *W* is $m-k$. The algorithms using this rule first try to find whether there exists a bad *W*-suffix in the window $W = T(i-m+k+1, i)$. If there exists at least a bad *W*-suffix, then the algorithms prefer to use the shortest one and shift the window without any checking; otherwise, they check the substring $T(i-m-k+1, i)$ against *P* by applying a non-filtration algorithm.

There are several algorithms to find the bad *W*-suffix in a window. The algorithm proposed in [20] constructs a nondeterministic suffix automaton which recognizes every reverse prefix of *P* allowing at most *k* errors. Using this automaton, the algorithm reads the characters of *W* from right to left and stops reading if no pattern substring matches what was read with at most *k* errors. That is, the algorithm stops when it finds a bad *W*-suffix. Basically, this bad *W*-suffix is the shortest one. Another filtration algorithm using this approach was proposed in [23]. This algorithm also considers a window with length $m-k$. For each window, it examines each character of the window from right to left to see if the character appears in its corresponding characters in *P*. The corresponding characters are defined as follows. For a window $W = w_1 w_2 ... w_{m-k}$, the corresponding characters of w_i are p_{i-k}, p_{i-k+1}, …, p_{i+k}. If a character of *W* does not appear in its corresponding characters, this character must be an error and hence it is called a *bad character*. Thus, this algorithm examines each character of *W* from right to left and stops when it finds $k+1$ bad characters. If the algorithm finds $k+1$ bad characters, this means that it has found a bad *W*-suffix. The algorithm designed in [7] uses the *q*-gram technique to find the bad *W*-suffix. A *q-gram* is a string with length *q*. This algorithm first constructs a table *D*. For every possible *q*-gram, the edit distance of this *q*-gram with its most similar substring of *P* is recorded in *D*. For each *W*, the algorithm reads the non-overlapping *q*-grams of *W* from right to left and sum up their total values in *D*. If the total value exceeds *k*, it stops because a *bad W-suffix* is found. Note that this *bad W-suffix* may not be the shortest one.

1.3 *q*-Gram Counting Approach

The *q*-gram counting approach considers only the contents of the two strings by ignoring the character order in the contents. For example, let $x = abaaab$ and $y = bbaab$. We know that there are four $'a'$'s and two $'b'$'s in *x* and two $'a'$'s and three $'b'$'s in *y*. The difference between the number of $'a'$ in *x* and that in *y* is 2 and 1 for $'b'$. Thus, we must need at least two edit operations to make the numbers of $'a'$ and $'b'$ in *x* and *y* equal, for example, by deleting one $'a'$ in *x* and substituting another $'a'$ in *x* by $'b'$. Let Σ be a finite alphabet and Σ^q be the set of all possible strings of length *q* over Σ. Let $x = x_1 x_2 \ldots x_{|x|}$ be a string over the Σ. Clearly, *v* is a *q*-gram of *x* if $v = x_i x_{i+1} \ldots x_{i+q-1}$ for some *i*, where $1 \le i \le |x| - q + 1$. By following the notation used in [25], let $G(x)[v]$ denote the total number of the occurrences of *v* in *x*. Then the *q-gram distance* between two strings *x* and *y* is defined as $D_q(x,y) = \sum_{v \in \Sigma^q} |G(x)[v] - G(y)[v]|$. The relationship between the *q*-gram distance $D_q(P, t_{i-m+1} \ldots t_i)$ and the edit distance de_i is given in [25] as described as follows.

Theorem 1 [25] For $1 \le i \le n$, $D_q(P, t_{i-m+1} \ldots t_i) \le 2q \cdot de_i$.

This indicates that $D_q(P, t_{i-m+1} \ldots t_i)/(2q)$ is a lower bound of de_i and hence it can serve as a filter as follows. If $D_q(P, t_{i-m+1} \ldots t_i)/(2q) > k$ for some *i*, then $de_i > k$ and hence the position *i* can be ignored. In [25], Ukkonen also gave an efficient algorithm to compute $D_q(P, t_{i-m+1} \ldots t_i)$ for all *i*'s in linear time. This algorithm marks all *i*'s with $D_q(P, t_{i-m+1} \ldots t_i)/(2q) \le k$ and computes de_i only for those marked *i*'s. As a result, the *k* differences problem can be solved in O(min($n + rk^2, kn$)) time, where *r* is the number of indexes *i*'s with $D_q(P, t_{i-m+1} \ldots t_i)/(2q) \le k$. For example, suppose that

$P = abcd$, $T(3,6) = aaba$, $q = 1$ and $k = 1$. Then $D_1(P, t_3 \ldots t_6) = 4$ and clearly $D_1(P, t_3 \ldots t_6)/(2 \cdot 1) = 2 > k$, meaning that de_6 must be larger than 2 and hence the position 6 in *T* can be skipped.

On the other hand, let us consider another example where $P = abcdef$ and a window $T(8,13) = ceafbd$ in *T*. Let $q = 1$ and $k = 1$. We then have $D_1(P, t_8 \ldots t_{13}) = 0$ and $D_1(P, t_8 \ldots t_{13})/(2 \cdot 1) = 0 < 1$. In this case, the Ukkonen's algorithm in [25] needs to further check the position 13 in *T* by using a non-filtration algorithm. In the above example, however, if we consider the order of characters, it is quite obvious that the de_{13} would be quite large. Actually, it can be observed that if the value of *q* is too large, then the value of $D_q(P, t_{i-m+1} \ldots t_i)/(2q)$ would be very small, resulting in that many positions in *T* are needed to be checked.

The above idea was also used in [15]. In the next section, we propose a filtration method which combines the partial exact matching approach and the *q*-gram counting approach. Basically, our filtration uses a special property, called locality property, which was also used implicitly in [23].

2 A New Filtration Algorithm Based on the Locality Property

Our filtration algorithm is based upon the following ideas:

(1) We partition *P* into $(k+1)$ non-overlapping pieces $P^0, P^1, \ldots, P^k$. That is, for a pattern $P = p_1 p_2 \cdots p_m$, we have $P^j = p_{jl+1} p_{jl+2} \cdots p_{(j+1)l}$ for $0 \le j < k$, and $P^k = p_{kl+1} p_{kl+2} \cdots p_m$, where $l = \lfloor m/(k+1) \rfloor$. Let us assume that there is a substring *S* of *T* whose edit distance from *P* is less than or equal to *k*. Then there must be at least one of the divided pieces in *P* exactly appearing in *S* [29]. This is the partial exact matching approach introduced in Section1. In other words, for a window *W* with an appropriate size, if no P^i appears exactly in *W*, we ignore this window.

(2) To decide whether there exists one divided piece P^i in *P* that exactly appears in *W* is an exact string matching problem. There are many algorithms which can be used to solve this problem [1, 9, 11, 26, 27]. However, we shall define a new distance function $DL_q(A, B)$ between two strings *A* and *B* based on their *q*-grams below and use it to design a more efficient filtration algorithm.

$$DL_q(A,B) = \sum_{v \in \Sigma^q} \max(G(A)[v] - G(B)[v], 0) . \quad (1)$$

In fact, the value of $DL_q(A, B)$ is equal to the number of *q*-grams of *A* which are not in *B*. For the case when *A* appears in *B*, it is obvious that $G(A)[v] \le G(B)[v]$ for all $v \in \Sigma^q$ and as a result, $DL_q(A, B) = 0$. On the other hand, if $DL_q(A, B) \ne 0$, then *A* does not exactly appear in *B*. Thus we have the following lemma immediately.

Lemma 1. Let *A* and *B* be two strings and *q* be a positive integer less than or equal to *m*. If $DL_q(A, B) \ne 0$, then *A* does not exactly appear in *B*.

The above lemma suggests that we should compute only $DL_q(A,B)$ and check whether it is equal to 0 or not. If it is not 0, then we conclude that *A* does not exactly appear in *B*; otherwise, *A* may exactly appear in *B*. For example, if $A = aaccg$ and $B = aagacgcg$, then there is a 2-gram of *A*, namely $'cc'$, which is not in *B*. Hence, $DL_2(A,B) = 1 \neq 0$, meaning that *A* does not exactly appear in *B*. If $A = aaccg$ and $B = aaaccgcg$, then $DL_2(A,B) = 0$, indicating that *A* may exactly appear in *B*. After a further check, *A* indeed appears in *B* exactly in this example. In our filtration step, as indicated at the beginning of this section, we partition *P* into $k+1$ pieces P^0, P^1, …, P^k and determine whether any P^j exactly appears in the whole window $W = T(i-m-k+1, i)$, where $0 \leq j \leq k$. The window *W* can be ignored if every P^j does not exactly appears in *W*, which can be determined by checking whether $DL_q(P^j, W) \neq 0$ according to Lemma 1. However, we here do not use this approach to see if P^j does not appear in *W* exactly. Instead, we utilize the *locality property* described below to do this job in a more effective way. First of all, we define $W^j = w_{jl+1}w_{jl+2} \ldots w_{(j+1)l+2k}$ for $0 \leq j < k$ and $W^k = w_{kl+1}w_{kl+2} \ldots w_{m+k}$, where $l = \lfloor m/(k+1) \rfloor$. For example, if $W = aactgtccaa$, $m = 8$ and $k = 2$, then we have $l = \lfloor 8/(2+1) \rfloor = 2$ and hence $W^0 = aactgt$, $W^1 = ctgtcc$ and $W^2 = gtccaa$. Next, we check whether there is a P^j exactly appears in W^j by looking at the value of $DL_q(P^j, W^j)$. If $DL_q(P^j, W^j) \neq 0$, then P^j does not exactly appear in W^j according to Lemma 1. As will be explained later in Lemma 2, we can ignore the window *W* if no P^j appears in W^j exactly. This makes our filtration more efficient and more effective, because we compare P^j only with small piece W^j, illustrated in Figure 1, instead of whole *W* and the probability of P^j exactly appearing in W^j is much smaller than that of P^j exactly appearing in *W*.

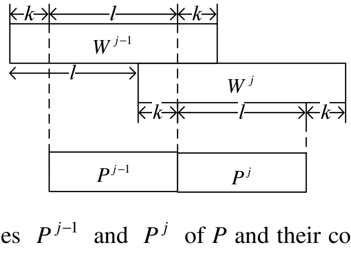


Fig. 1. Two consecutive pieces P^{j-1} and P^j of *P* and their corresponding substrings W^{j-1} and W^j in *W*

Lemma 2. Let $W = T(i-m-k+1, i)$, where $m+k \leq i \leq n$. If no P^j exactly appears in W^j, then $de_i > k$, where $0 \leq j \leq k$.

Proof: This lemma is equivalent to saying that if $de_i \leq k$, then there is at least one P^j exactly appears in W^j. Now, suppose that $de_i \leq k$ and let *S* be a suffix of *W* such that

$ED(P,S) \le k$. Then we divide *S* into $(k+1)$ pieces $S^0, S^1, \ldots, S^k$ such that P^j is aligned with S^j , as illustrated in Figure 2, for each $0 \le j \le k$. In other words, $ED(P,S)=ED(P^0,S^0)+\ldots+ED(P^k,S^k)$. Since $ED(P,S) \le k$, it can be verified that $ED(P^uP^{u+1}\ldots P^v, S^uS^{u+1}\ldots S^v) \le k$ for any $0 \le u \le v \le k$. Below, we first show that every S^j must be located inside W^j , denoted by $S^j \subseteq W^j$ for simplicity. We prove this property by induction on *j*. Again, for simplicity, we use $r(S^j)$ and $l(S^j)$ to denote the rightmost and leftmost positions of S^j in *W*, respectively.

(1) Suppose that $j=0$. Then $|P^1P^2\ldots P^k|-k \le |S^1S^2\ldots S^k| \le |P^1P^2\ldots P^k|+k$, since $P^1P^2\ldots P^k$ is aligned with $S^1S^2\ldots S^k$ and $ED(P^1P^2\ldots P^k, S^1S^2\ldots S^k) \le k$. Moreover, since $|P^1P^2\ldots P^k|=m-l$, we have $(m-l-k) \le |S^1S^2\ldots S^k| \le (m-l+k)$. This means that the smallest length of $S^1S^2\ldots S^k$ is $(m-l-k)$, suggesting that $r(S^0) \le m+k-(m-l-k)=l+2k$. It is clear that $l(S^0) \ge 1$. By definition, we have $W^0 = W(1, l+2k)$. Therefore, $S^0 \subseteq W^0$.

(2) Suppose that $S^0 \subseteq W^0, S^1 \subseteq W^1, \ldots, S^{j-1} \subseteq W^{j-1}$. Then we want to prove that $S^j \subseteq W^j$. It can be easily verified that $ED(P^jP^{j+1}\ldots P^k, S^jS^{j+1}\ldots S^k) \le k$ and $|P^jP^{j+1}\ldots P^k|=m-jl$, which implies that $m-jl-k \le |S^jS^{j+1}\ldots S^k| \le m-jl+k$. The result that the largest length of $S^jS^{j+1}\ldots S^k$ is $(m-jl+k)$ suggests that $l(S^j) \ge m+k-(m-jl+k)+1=jl+1$. On the other hand, it still can be verified that $ED(P^{j+1}P^{j+2}\ldots P^k, S^{j+1}S^{j+2}\ldots S^k) \le k$ and $|P^{j+1}P^{j+2}\ldots P^k|=m-(j+1)l$, which indicates that $m-(j+1)l-k \le |S^jS^{j+1}\ldots S^k| \le m-(j+1)l+k$. As a result, the smallest length of $S^{j+1}S^{j+2}\ldots S^k$ is $m-(j+1)l-k$, which suggests that $r(S^j) \le m+k-(m-(j+1)l-k)=(j+1)l+2k$. By definition, we have $W^j = W(jl+1,(j+1)l+2k)$. Therefore, $S^j \subseteq W^j$.

Based on the above discussion, we have proven that $S^j \subseteq W^j$ for all $0 \le j \le k$, where S^j is aligned with P^j . As mentioned before, there exists at least one P^j exactly appears in *W* according to the study in [29]. Therefore, we can conclude that there exists at least one P^j exactly appears in W^j . **Q.E.D.**

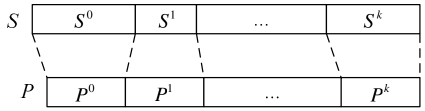


Fig. 2. Dividing *S* into (*k*+1) pieces $S^0, S^1, \ldots, S^k$ such that S^j is aligned with P^j in *P* for each $0 \le j \le k$

We shall call the condition in Lemma 2 as a *bad piece rule* for filtration of approximate string matching. To determine whether there is a P^j that exactly appears in W^j , a typical algorithm is to use the suffix tree approach [26, 27] by constructing a suffix tree of W^j . However, it is impractical for our problem because W^j changes all the time. Instead, we shall use the distance function defined in Equation (1) to do this job and show that the value of this distance function can be updated easily.

According to Lemma 1 and 2, the following theorem can be easily derived.

Theorem 2. Let $W = T(i-m-k+1, i)$, where $m+k \le i \le n$, and q be a positive integer less than or equal to *m*. If $DL_q(P^j, W^j) \ne 0$ for all $0 \le j \le k$, then $de_i > k$.

To use the condition in Theorem 2 as a filter, we need to compute the values $DL_q(P^j, W^j)$ for all $0 \le j \le k$. For example, let $q = 1$, $k = 1$, $P = abcdef$ and $W = T(8,15) = fceafbd$. We then divide P into $(k+1) = 2$ regions $P^0 = abc$ and $P^1 = def$, whose corresponding regions in *W* are $W^0 = fceaf$ and $W^1 = afbd$ respectively. It is not hard to verify that $DL_1(P^0, W^0) = 1 > 0$ and $DL_1(P^1, W^1) = 1 > 0$. By Theorem 2, we can conclude that $de_{15} > 1$. This means that there is no substring of *T* ending at position 15 whose edit distance with respect to *P* is less than or equal to 1 and we can therefore skip this position.

Below, we shall show that each $DL_q(P^j, W^j)$ can be updated efficiently when the window is shifted one step to the next position. For each W^j, where $0 \le j \le k$, we use array $G(W^j)[x]$ to store the number of the occurrences of x in W^j for all $x \in \Sigma^q$. For the first window $W = T(1, m+k)$, we compute $G(W^j)[x]$ for all $x \in \Sigma^q$ and $DL_q(P^j, W^j)$ directly. After that, we shift the window right by one step and update all $G(W^j)[x]$ and $DL_q(P^j, W^j)$. Actually, not all $G(W^j)[x]$ need to be updated. As illustrated in Figure 3, if *x* is the leftmost *q*-gram of W^j, denoted by Q_1, before the window is shifted, or the rightmost *q*-gram of W^j, denoted by Q_2, after the window is shifted, then $G(W^j)[x]$ is needed to be updated.

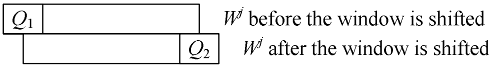

Q_1
W^j before the window is shifted
Q_2
W^j after the window is shifted

Fig. 3. The two *q*-grams need to be updated when shifting the window right by one step

Recall that $G(P^j)[x]$ denotes the number of *q*-gram *x* occurring in P^j, which can be computed in advance. Then we can use the following formula to update $DL_q(P^j, W^j)$ for all $0 \le j \le k$.

$$(1)\quad DL_q(P^j, W^j) = \begin{cases} DL_q(P^j, W^j) + 1, \text{if } G(P^j)[Q_1] \ge G(W^j)[Q_1]. \\ DL_q(P^j, W^j) - 1, \text{if } G(P^j)[Q_2] > G(W^j)[Q_2]. \end{cases}$$

Consider the case where the size *l* of P^j is small, say $l = 1$. It is very possible that every P^j exactly appears in its corresponding W^j, causing that the performance efficiency of our filtration method is decreased. For efficiently handling this case, we incorporate the Ukkonen's algorithm [25] into our filtration step which considers the whole pattern *P*, instead of just a piece P^j in *P*. Therefore, we compute $G(W_m)[x]$ for all $x \in \Sigma^q$ and $D_q(P, W_m)$, where $W_m = w_{k+1}w_{k+2}...w_{m+k}$. Based on the study in [25], $D_q(P, W_m)$ can be updated by the following two formulas. Note that the definitions of Q_1 and Q_2 with respect to W_m below are same as those of Q_1 and Q_2 we previously defined with respect to W^j.

$$(1)\ \ D_q(P,W_m)=\begin{cases} D_q(P,W_m)-1, \text{if } G(P)[Q_1]<G(W_m)[Q_1]. \\ D_q(P,W_m)+1, \text{otherwise.} \end{cases}$$

$$(2)\ \ D_q(P,W_m)=\begin{cases} D_q(P,W_m)-1, \text{if } G(P)[Q_2]>G(W_m)[Q_2]. \\ D_q(P,W_m)+1, \text{otherwise.} \end{cases}$$

After updating $D_q(P,W_m)$ and $DL_q(P^j,W^j)$, we also need to update $G(W^j)[Q_1]$ and $G(W_m)[Q_1]$ by decreasing them by one and update $G(W^j)[Q_2]$ and $G(W_m)[Q_2]$ by increasing them by one for all $0 \le j \le k$.

We have to pay attention to another point. Suppose that our filtration process decides that position *i* needs to be checked. We do not immediately perform a dynamic programming procedure on $T(i-m-k+1,i)$ and *P*. Instead, we look at position $i+1$. If position $i+1$ also needs to be checked, it will be more efficient to apply the dynamic programming procedure to $T(i-m-k+1,i+1)$ and *P*, which can tell us de_i and de_{i+1} at the same time. For this purpose, we use $Mark[i]$ to record whether position *i* in *T* needs to be checked or not, where $1 \le i \le n$. If position *i* needs to be checked, then $Mark[i]=1$; otherwise, $Mark[i]=0$. Note that our filtration algorithm starts from position $m+k$ in *T* and it is possible that there is a solution ending before position $m+k$. Therefore, we let $Mark[m+k-1]=1$ initially so that our algorithm can check $T(1,m+k-1)$ and *P* using the dynamic programming algorithm. In the following, we describe Algorithm 1 to solve the *k* differences problem.

Algorithm 1
Input: A text *T*, a pattern *P*, an error bound *k* and *q*.
Output: All the positions *i's* in *T* such that $de_i \le k$.

```
1. /*Initialization*/
   $l=\lfloor m/(k+1)\rfloor$ and $Mark[m+k-1]=1$.
2. /*Preprocessing Step*/
   Compute $G(P^j)[x]$ and $G(P)[x]$ for all $0 \le j \le k$ and $x \in \Sigma^q$.
3. Let $i=m+k$ and $W=T(1,i)$.
   Compute $G(W^j)[x]$ and $G(W_m)[x]$ for all $0 \le j \le k$ and $x \in \Sigma^q$.
   Compute $DL_q(P^j,W^j)$ and $D_q(P,W_m)$ for all $0 \le j \le k$.
4. if there exists j, where $0 \le j \le k$, such that $DL_q(P^j,W^j)=0$ then
       if $D_q(P,W_m)/2q \le k$ then
           $Mark[i]=1$.
       else $Mark[i]=0$.
   else $Mark[i]=0$.
5. $i=i+1$.
   if $G(P^j)[Q_1] \ge G(W^j)[Q_1]$ then $DL_q(P^j,W^j)=DL_q(P^j,W^j)+1$.
   if $G(P^j)[Q_2] > G(W^j)[Q_2]$ then $DL_q(P^j,W^j)=DL_q(P^j,W^j)-1$.
   if $G(P)[Q_1] < G(W_m)[Q_1]$ then $D_q(P,W_m)=D_q(P,W_m)-1$.
   else $D_q(P,W_m)=D_q(P,W_m)+1$.
   if $G(P)[Q_2] > G(W_m)[Q_2]$ then $D_a(P,W_m)=D_a(P,W_m)-1$.
```

```
    else $D_q(P, W_m) = D_q(P, W_m) + 1$.
      $G(W^j)[Q_1] = G(W^j)[Q_1] - 1$.
      $G(W_m)[Q_1] = G(W_m)[Q_1] - 1$.
      $G(W^j)[Q_2] = G(W^j)[Q_2] + 1$.
      $G(W_m)[Q_2] = G(W_m)[Q_2] + 1$.
    if $i < n$ then Go to Step 4.
6.  Obtain $\alpha$ non-overlapping substrings from T, say $T_1, \ldots, T_\alpha$, by deleting
    every position $v$ in T satisfying $Mark[v]=\ldots=Mark[v+m+k-1]=0$.
    Apply the algorithm proposed in [12] to $T_r$ and P for all $1 \le r \le \alpha$.
```

We analyze the time complexity of Algorithm 1 as follows. Since the size of P^j is less than or equal to l and there are $l-q+1$ q-grams in P^j, computing $G(P^j)[x]$ needs O(l) time. Since the size of P is m, the computation of $G(P)[x]$ needs O(m) time. Hence, the cost of Step 2 is O(kl+m)=O(m). Similarly, it can be verified that computing all $G(W^j)[x]$ and $G(W_m)[x]$ in Step 3 still requires O(m) time. As to all $DL_q(P^j, W^j)$ and $D_q(P, W_m)$ in Step 3, their computation costs O(Σ^q) time. Thus, the cost of Step 3 is O(m+Σ^q). Step 4 performs at most $(k+1)$ iterations, each of which needs just a constant time. Therefore, the cost of Step 4 is O(k). Step 5 takes O(k) time, since it updates only two q-grams for each W^j and W_m, where $0 \le j \le k$. Actually, both Steps 4 and 5 are performed at most O(n) times and their total cost is O(kn). As to Step 6, its worst-case scenario is that no position in the text is filtered out. In this case, the computation of Step 6 can be done in O(kn) time [12]. As a result, the total time complexity of Algorithm 1 is O(Σ^q +kn).

3 A Hybrid Filtration Strategy

In this section, we shall present a hybrid strategy to design a filtration algorithm for solving the k differences problem. Note that currently existing filtration algorithms have their strengths and weaknesses. For example, let $P = abcdef$, $k = 1$ and $W = T(9,15) = fceafbd$. Then our filtration method proposed in the previous section can skip this position 15 in T, since $P^0 = abc$, $P^1 = def$, $W^0 = fceaf$ and $W^1 = afbd$ and, as a result, $DL_1(P^0, W^0) = 1 > 0$ and $DL_1(P^1, W^1) = 1 > 0$. However, it can not be skipped using the Ukkonen's filtration algorithm [25]. Consider another example where $P = abbcba$, $k = 1$ and $W = T(2,7) = aaabbb$. If we use the filtration algorithm proposed in [20] based on the bad W-suffix approach, then position 7 in T can be skipped since $W(2,6) = aabbb$ is a bad W-suffix of W. However, neither Ukkonen's nor our filtration algorithm can filter out this position. In other words, we cannot say which flirtation method is superior to the others. But we may safely say that using some of them together will certainly be more effective.

We should point out that when we talk about filtration strategy, the error bound k should not be too small relative to the length of the pattern, because if k is small, any filtration method would easily filter out a large number of locations in the text. Of course, k should not be too large. Here, we focus on the cases where k is in a reasonable size. The basic idea of our hybrid filtration strategy is to use different filtration approaches in different stages. Particularly, for efficiency purpose, we use a faster filtration algorithm first to filter out most of the locations in the text and then apply another more powerful, but slower, filtration algorithm to the remaining locations. Note that if we use a powerful, but slow, filtration algorithm to start with, it will be very time-consuming. We note that the bad *W*-suffix approach [7, 14, 20] for flirtation is a very powerful one, but it is also time-consuming when *k* is large. If we use this approach at the very beginning, our algorithm will be quite slow, although it can filter out a large number of locations in the text. Hence, our hybrid strategy is to first use our filtration strategy which is fast and then the bad *W*-suffix approach. This makes our hybrid filtration quite efficient so that a large number of locations in the text are eliminated.

4 Experimental Results

In our experiments, we tested our filtration algorithms on some simulated datasets of DNA sequences and compared their results with those obtained by other filtration algorithms, including the linear expected time algorithm in [4] (CL90_LET for short), the sublinear expected time algorithm in [4] (CL90_SET), the algorithm proposed by Wu and Manber [29] (WM92), the algorithm designed by Ukkonen [25] (U92) and the algorithm proposed by Fredriksson and Navarro [7] (FN04). We have implemented all these algorithms using C and measured percentages of their unfiltered positions to compare their performance efficiencies. We also tested our hybrid filtration approach by combining our filtration algorithm with the bad *W*-suffix algorithm proposed in [7]. In our experiments, we randomly generated the text *T* with size $n = 10\text{M}$ and the pattern *P* with size $m \in \{10, 20, 50\}$ by using an alphabet $\Sigma = \{a, c, g, t\}$. We also used different error bounds *k* between 1 and 11 to test all the programs. Several algorithms [7, 25] have a parameter *q* for controlling the size of *q*-grams. Basically, different *q*-grams will have different efficiencies. Below, we only show the best performance of

Table 1. The percentage of positions needed to check

	m=10			*m*=20			*m*=50		
k	1	2	3	1	3	5	1	6	11
CL90_LET	48.70	98.97	100.00	0.01	47.44	100.00	0.00	0.02	99.99
CL90_SET	99.11	100.00	100.00	9.04	100.00	100.00	0.00	99.70	100.00
WM92	2.31	47.50	98.11	0.00	9.66	93.83	0.00	2.61	96.45
FN04	1.73	33.03	97.73	0.00	2.86	89.72	0.00	0.00	60.23
U92	1.50	49.39	92.96	0.00	6.25	95.38	0.00	0.00	98.64
LLL12	0.82	30.30	91.85	0.00	2.02	89.52	0.00	0.00	96.87
LLL12+FN04	0.26	15.78	87.26	0.00	0.24	75.61	0.00	0.00	52.22

each algorithm when it uses different q between 1 and 5. The results we obtained are shown in Table 1, each of which is the average over 10 repeated experiments, where LLL12 denotes our filtration algorithm and LLL12+FN04 denotes our hybrid filtration algorithm by using our filtration algorithm first and then the bad W-suffix algorithm FN04.

The experimental results show that our algorithm LLL12 indeed can filter out more positions in the text than most of the others. Without considering our hybrid filtration algorithm LLL12+FN04, our filtration algorithm LLL12 has the best performance in most of all the cases. However, the algorithm FN04 performs the best in some cases, supporting that the bad W-suffix approach is a powerful filtration as we noted before. On the other hand, our hybrid filtration algorithm LLL12+FN04 achieves the best performance in all cases. Particularly, it improves the performance efficiency greatly when the ratio of k/m is about 0.2. All the filtration algorithms have property that their performance efficiencies decrease when the error bound k increases.

5 Concluding Remarks and Future Research

We have proposed a filtering method, as well as a hybrid filtration strategy, to eliminate candidates in a text string for the k differences problem. The experimental results have also shown that our filtration method can filter more positions than most of the other algorithms, and our hybrid strategy can further improve the filtration efficiency in all the cases with different pattern size m and different error bound k. It will be interesting future works to analyze the average-case time complexity of our filtration algorithm for the k differences problem and apply the idea behind this algorithm to tackle the exact string matching problem.

References

1. Boyer, R.S., Moore, J.S.: A fast string searching algorithm. Communications of the ACM 20, 762–772 (1977)
2. Baeza-Yates, R., Navarro, G.: Faster approximate string matching. Algorithmica 23, 127–158 (1999)
3. Baeza-Yates, R., Perleberg, C.: Fast and practical approximate pattern matching. Information Processing Letters 59, 21–27 (1996)
4. Chang, W.I., Lawler, E.L.: Approximate string matching in sublinear expected time. In: Proceedings of the ACM-SIAM 31st Annual Symposium on Foundations of Computer Science, pp. 116–124 (1990)
5. Chang, W., Lawler, E.: Sublinear approximate string matching and biological applications. Algorithmica 12, 327–344 (1994)
6. Chang, W., Marr, T.: Approximate String Matching and Local Similarity. In: Crochemore, M., Gusfield, D. (eds.) CPM 1994. LNCS, vol. 807, pp. 259–273. Springer, Heidelberg (1994)
7. Fredriksson, K., Navarro, G.: Average-optimal single and multiple approximate string matching. ACM Journal of Experimental Algorithmics 9, 1–47 (2004)

8. Giegerich, R., Kurtz, S., Hischke, F., Ohlebusch, E.: A general technique to improve filter algorithms for approximate string matching. In: Proceedings of the 4th South American Workshop on String Processing (WSP 1997), pp. 38–52 (1997)
9. Horspool, R.N.: Practical fast searching in strings. Software - Practice & Experience 10, 501–506 (1980)
10. Hyyrö, H., Navarro, G.: Bit-parallel witnesses and their applications to approximate string matching. Algorithmica 41, 203–231 (2005)
11. Knuth, D.E., Morris, J.H., Pratt, V.R.: Fast pattern matching in strings. SIAM Journal on Computing 6, 323–350 (1977)
12. Landau, G., Vishkin, U.: Fast parallel and serial approximate string matching. Journal of Algorithms 10, 157–169 (1989)
13. Myers, G.: A Fast Bit-Vector Algorithm for Approximate Pattern Matching Based on Dynamic Programming. In: Farach-Colton, M. (ed.) CPM 1998. LNCS, vol. 1448, pp. 1–13. Springer, Heidelberg (1998)
14. Myers, G.: A fast bit-vector algorithm for approximate string matching based on dynamic programming. Journal of the ACM 46, 395–415 (1999)
15. Navarro, G.: Multiple approximate string matching by counting. In: Proceedings of the 4th South American Workshop on String Processing (WSP 1997), pp. 125–139. Carleton University Press (1997)
16. Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys 33, 31–88 (2001)
17. Navarro, G., Baeza-Yates, R.: Improving an algorithm for approximate pattern matching. Technical Report TR/DCC-98-5, Department of Computer Science, University of Chile (1998)
18. Navarro, G., Baeza-Yates, R.: Very fast and simple approximate string matching. Information Processing Letters 72, 65–70 (1999)
19. Navarro, G., Baeza-Yates, R.: A hybrid indexing method for approximate string matching. Journal of Discrete Algorithms 1, 205–239 (2000)
20. Navarro, G., Raffinot, M.: Fast and flexible string matching by combining bit-parallelism and suffix automata. ACM Journal of Experimental Algorithmics 5, 1–36 (2000)
21. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48, 443–453 (1970)
22. Sellers, P.H.: String matching with errors. Journal of Algorithms 20, 359–373 (1980)
23. Tarhio, J., Ukkonen, E.: Approximate Boyer-Moore string matching. SIAM Journal on Computing 22, 243–260 (1993)
24. Ukkonen, E.: Finding approximate patterns in strings. Journal of Algorithms 6, 132–137 (1985)
25. Ukkonen, E.: Approximate string matching with q-grams and maximal matches. Theoretical Computer Science 92, 191–211 (1992)
26. Ukkonen, E.: On-line construction of suffix trees. Algorithmica 14, 249–260 (1995)
27. Weiner, P.: Linear pattern matching algorithm. In: 14th Annual IEEE Symposium on Switching and Automata Theory, pp. 1–11 (1973)
28. Wagner, R.A., Fisher, M.J.: The string-to-string correction problem. Journal ACM 21, 168–173 (1974)
29. Wu, S., Manber, U.: Fast text searching: allowing errors. Communications of the ACM 35, 83–91 (1992)

A Novel Approximation Algorithm for Minimum Geometric Disk Cover Problem with Hexagon Tessellation[*]

Chi-Yu Chang, Chi-Chang Chen[**], and Cheng-Chun Liu

Department of Information Engineering, I-Shou University, Kaohsiung City 84001, Taiwan
pop449@hotmail.com, {ccchen,isu10103028m}@isu.edu.tw

Abstract. Given a set P of n points in the Euclidean plane, the minimum geometric disk cover (MGDC) problem is to identify a minimally sized set of congruent disks with prescribed radius r that cover all the points in P. It is known that the MGDC problem is NP-complete. Solutions to the MGDC problem can be used to solve the relay node placement problems of wireless sensor networks. In this study, we proposed an approximation algorithm for the MGDC problem that identifies covering disks via the regular hexagon tessellation of the plane. We show that the approximation ratio of the proposed algorithm is 5. Furthermore, we show that if the set of points in P is uniformly distributed, then there is 41.7% probability for the proposed algorithm to use less than or equal to 5 times the optimal number of disks, and 58.3% probability of using no more than 4 times the optimal number of disks.

Keywords: Minimum Geometric Disk Cover Problem, Hexagon Tessellation, Relay Node Placement Problems, Wireless Sensor Networks.

1 Introduction

Given a set P of n points in the Euclidean plane and a prescribed radius r, we say that a disk d covers a point $p \in P$ if the distance between the center of d and p is no greater than r. The minimum geometric disk cover (MGDC) problem is to find a set of disks with minimum cardinality covering all the points in P. If the radius r is equal to 1, the MGDC problem is called Minimum Geometric Unit Disk Cover (MGUDC) problem. In fact, MGDC problem is just a scale-up MGUDC problem. For simplicity of calculation, in this study we focus on solving the MGUDC problem. However, all the algorithms for the MGUDC problem can be easily extended to the MGDC problem by simply changing the radius from 1 to any fixed real number r. Solutions to the MGDC

[*] This work was partially supported by the I-Shou University under Grant ISU100-01-06 and the Ministry of Education under the Interdisciplinary Training Program for Talented College Students in Science, 100-B4-01.

[**] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 157–166.
DOI: 10.1007/978-3-642-35452-6_18

problem can be used to solve the relay node placement problems of wireless sensor networks.

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors that cooperatively monitor a deployed region for physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, and pollutants. The transmission power consumed by a wireless radio is proportional to the distance squared or even a higher order in the presence of obstacles. Thus, multi-hop routing is usually considered for sending collected data to the sink instead of direct communication.

One approach to prolong network lifetime of a WSN while preserving network connectivity is to deploy a small number of costly, but more powerful, relay nodes whose main task is to communicate with other sensors or relay nodes. The relay node placement problem of WSNs is to deploy the minimum amount of relay nodes such that each sensor can communicate with at least one relay node. Suppose the relay nodes in the WSN are homogeneous and their communication ranges are circles of radius r, then the relay node placement problem of WSNs can be transformed to the MGDC problem.

In this study, we proposed a simple yet effective approximation algorithm for the MGDC problem that identifies covering disks via the regular hexagon tessellation of the plane. A *tessellation* is the process of creating a two-dimensional plane using the repetition of a geometric shape with no overlaps and no gaps. A *regular tessellation* is a tessellation that covers a two-dimensional plane with regular polygons of the same shape and same size. It is known that there are only three regular tessellations (composed of the hexagon, square, and triangle) in the Euclidean plane [1].

A regular polygon enclosed by a unit disk (i.e., a circle with radius 1) is called a *unit polygon*. If the polygons of a regular tessellation are all unit polygons, we call the tessellation *unit polygon tessellation*. Therefore, there are three unit polygon tessellations: unit triangle tessellation, unit square tessellation, and unit hexagon tessellation (Fig. 1). It can be shown that to cover a sufficient large region in the plane, the unit hexagon tessellation requires less number of polygons than that of the other two unit polygon tessellations.

The rest of this paper is organized as follows: Section 2 reviews related research on MGDC problem and the relay node placement problem in WSNs. Section 3 presents the approximation algorithm and its analysis for the MGDC problem using hexagon tessellation. Section 4 provides a simulation of the proposed approximation algorithm and a comparison of its performance with the optimal solution. Finally, Section 5 offers a conclusion.

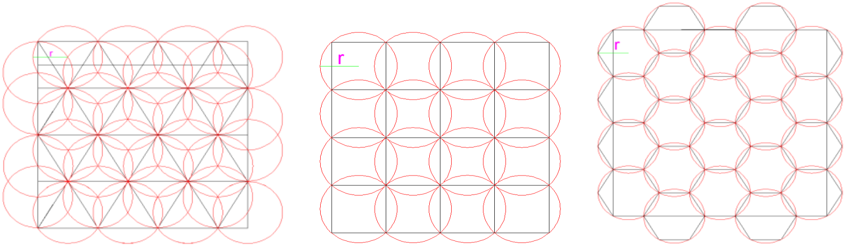


Fig. 1. Three unit polygon (triangle, square, hexagon) tessellations

2 Related Work

Given a set P of n points in the Euclidean plane, the minimum geometric disk cover (MGDC) problem is to identify a minimally sized set of congruent disks with prescribe radius r that cover all the points in P. It is known that the MGDC problem is NP-complete [2,3,4]. For the MGDC problem, there exists a polynomial time approximation scheme (PTAS) due to Hochbaum and Maass [2]. They used divide and conquer approach to solve the MGDC problem and proposed the famous Shifting Lemma to calculate the approximation ratio. By their approach, they obtain a family of algorithms with worst case approximation ratio of $(1 + \epsilon)$, with $0< \epsilon \leq 3$, and a running time that is polynomial when ϵ is fixed. However, The degree of the polynomial is very large as ϵ approaches 0, making their strategy not practical for even small number of input points to cover.

Other polynomial approximation algorithms which provide a suboptimal solution within a constant approximation factor to the optimal one have been proposed in [5,6,7]. These algorithms are based on the observation that the number of possible disks positions can be bounded by assuming (without loss of generality) that any disk that covers at least two points has two of these points on its border. An exhaustive search on all possible disc positions leads to an optimal solution, but runs in a time that is exponential in the number n of points to cover. By performing a search on a subset of the possible disks positions, the running time of the algorithm becomes polynomial in the number n of points to cover and the solution is still guaranteed to be within a constant factor from the optimal one.

Franceschetti, Cook, and Bruck [8] took a different approach to approximate the MGDC problem. They solved the MGDC problem by locating the covering disks at the centers of a set of tessellated squares. Consider an infinite square grid G. How many disks of given radius r, centered at the vertices of G, are required, in the worst case, to completely cover an arbitrary disk of radius r placed on the plane? They showed that this number is an integer in the set {3, 4, 5, 6} whose value depends on the ratio of r to the grid spacing. If the disks are congruent unit disks (i.e. $r = 1$) that circumscribe the squares of the grid G, then it needs 6 disks to cover an arbitrary disk of radius r in the worst case. This indicates that the number of covering disks used in their approach to solve the MGDC problem is no more than 6 times the optimal number of disks.

Lloyd and Xue [9] studied two versions of relay node placement problems. In the first version, they want to deploy the minimum number of relay nodes so that, between each pair of sensor nodes, there is a connecting path consisting of relay and/or sensor nodes. In the second version, they want to deploy the minimum number of relay nodes so that, between each pair of sensor nodes, there is a connecting path consisting solely of relay nodes. They present a polynomial time 7-approximation algorithm for the first problem and a polynomial time $(5+\epsilon)$-approximation algorithm for the second problem, where $\epsilon > 0$ can be any given constant.

Tang, Hao, and Sen [10] studied the relay node placement problem in large scale wireless sensor networks. Their objective is to place the fewest number of relay nodes in the playing field of a sensor network such that (1) each sensor node can communicate with at least one relay node and (2) the network of relay nodes is connected. They formulate the relay node placement in wireless sensor networks as a variant version of the MGDC problem. They present two polynomial time approximation algorithms to solve the problem, and prove that the ratio of the number of relay nodes needed by the approximation algorithm to the number of relay nodes needed by the optimal algorithm is bounded by 8 for the first algorithm and 4.5 for the second. However, both algorithms are with very high computational complexity [11].

3 Approximation Algorithm for the MGDC Problem

It is known that to cover a sufficient large area in the plane, unit hexagon tessellation uses less number of polygons than that of unit square tessellation. Using square tessellation to solve the MGDC problem has been studied in [8]. As far as we know, there are no research publications on using hexagon tessellation to solve the MGDC problem yet.

In Fig. 2, we provide a simple yet efficient approximation algorithm to the MGUDC problem with unit hexagon tessellation, and show that the algorithm needs no more than 5 times the optimal number of unit disks.

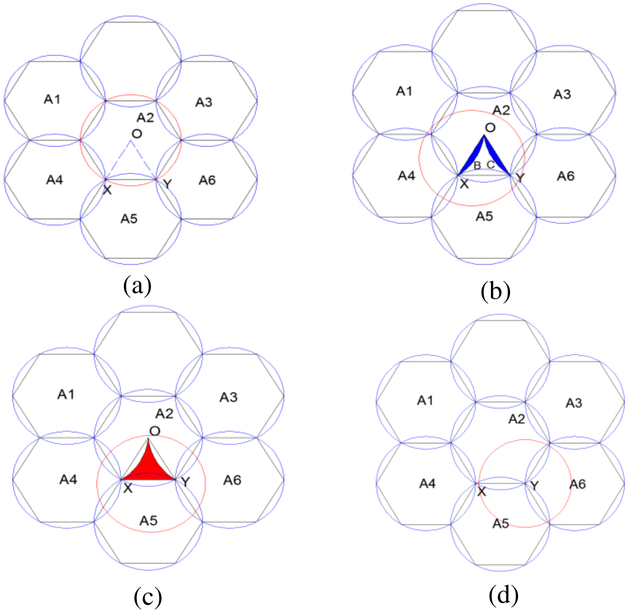
```
Algorithm
Input: A set $P$ of $n$ points, and a two-dimensional region R.
Output: A set of unit disks that covers $P$.
Step 1:
    Virtually tessellate the region R with unit hexagons. Calculate and store the
    coordinate of the center of each unit hexagon in the tessellation.
Step 2:
    For each unit hexagon $h$ (starting from left to right in a low-by-low manner)
        If the circumscribed circle of $h$ covers any uncovered points in $P$
        then choose the unit hexagon $h$ and mark all the uncovered points within
            the circumscribed circle.
        Otherwise, ignore the unit hexagon $h$.
    Next
Step 3:
    Output the circumscribed circles of the chosen unit hexagons.
```

Fig. 2. The proposed approximation algorithm for the MGDC problem

Time Complexity Analysis

We denoted the area of region R as |R|. Since the number of unit hexagons in the tessellation is depended on the area of the given region R, step 1 takes O(|R|)) time to calculate and store the centers of the unit hexagons. Since the number of unit hexagons is O(|R|) and there are n points in P, step 2 takes O(n×|R|) time. The complexity of step 3, again, depends on the number of unit hexagons, which is O(|R|). Hence, the overall complexity of the algorithm is O(|R|)+O(n×|R|)+O(|R|), which is O(n×(|R|). If the two-dimensional region R is a square with length m, then the complexity of the algorithm is O(n×m^2).

In the following, we will show that the number of output disks of the proposed algorithm is less than or equal to 5 times the optimal solution.



(a) (b) (c) (d)

Fig. 3. (a). The center of the unit disk is located at point O. (b). The center of the unit disk is located in the area of OBX (or OCY). (c). The center of the unit disk is located in the area of OXY (red region). (d). The center of the unit disk is located at point Y (or X).

Theorem 1

With the proposed algorithm, the number of output unit disks is no more than 5 times the optimal number of unit disks.

Proof

A circle circumscribes a unit hexagon is called a *hexagon disk*. We show that it is sufficient to cover any unit disk *D* on the plane with 5 hexagon disks.

Without loss of generality, we assume that the center of the unit disk *D* is located in a regular hexagon (e.g. A2 in Fig. 3) surrounding by its six neighbor regular hexagons. Since a regular hexagon can be divided into 6 congruent equilateral triangles, we analyze different cases using one of the triangles (e.g. ΔOXY in Fig. 3(a)). The analysis of the rest of triangles is the same.

Let *C* be the center of the unit disk *D*. We discuss the following cases according to the location of *C*.

Case 1: *C* is located at point O. (Figure 3(a))
The unit disk *D* can be completely covered by the circumscribed circle of hexagon A2.

Case 2: *C* is located in the area OBX (or OCY). (Figure 3(b))
The arc OBX is derived from the unit circle centered at the intersection point of A1, A2, and A4, and the arc OCY is derived from the unit circle centered at the intersection point of A2, A3, and A6. The unit disk *D* can be completely covered by 5

circumscribed circles of hexagons A1, A2, A4, A5, and A6 (or A2, A3, A4, A5, and A6 if the disk center is in OCY).

Case 3: *C* is located in the area OXY (the red region in Figure 3(c)).
The unit disk D can be completely covered by 4 circumscribed circles of hexagons A2, A4, A5, and A6.

Case 4: The center of the unit disk *D* is located at point Y (or X). (Figure 3(d))
The unit disk *D* can be complete covered by 3 circumscribed circles of hexagons A2, A4, and A5 (or A2, A5, and A6).

It is trivial to see that Case1 to Case 4 include all the area of the triangle OXY. Since the largest number of circumscribed circles (i.e., Case 2) needed to cover the unit disk *D* is 5, we conclude that it is sufficient to cover any unit disk on the plane with 5 hexagon disks. Therefore, suppose the optimal number of unit disks to cover *P* is *k*. Since each unit disk can be covered by at most 5 hexagon disks, we need no more than 5×*k* hexagon disks to cover *P*. Thus, with the proposed algorithm, the number of output unit disks is no more than 5 times the optimal number of unit disks. ■

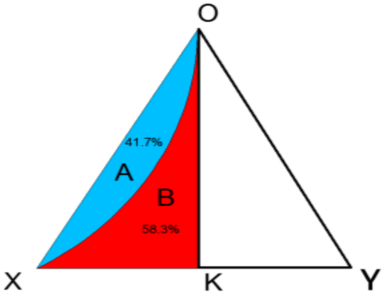


Fig. 4. Equilateral triangle OXY taken from the central unit hexagon of Fig. 3(a)

Corollary 1
If the set of points in P is uniformly distributed, then in the worst cases, there is 41.7% probability of using 5 times the optimal number of unit disks, and 58.3% probability of using 4 times the optimal number of unit disks

Proof
Because the triangle OXY can be divided into two symmetric right-angle triangles OXK and OYX (Fig. 4), we only consider triangle OXK in the proof. The proof can be done by calculating the ratios of areas A and B to the area of triangle OXK.

The area of triangle OXK is $\frac{1}{2} \times \left(\frac{1}{2} \times \frac{\sqrt{3}}{2}\right) = \frac{\sqrt{3}}{8}$

The area of A is $(2\pi - 3\sqrt{3})/12$, which is 41.7% of the area of triangle OXK.

The area of B is $(9\sqrt{3} - 4\pi)/24$, which is 58.3% of the area of triangle OXK.

According to Theorem 1, if the center of the unit disk is located in A, the unit disk can be covered by 5 hexagon disks, and if the center of the unit disk is located in B, the unit disk can be covered by 4 hexagon disks. Because the results of the cases in the proof of Theorem 1 are considered under the worst case, in fact, we may use less number of hexagon disks to cover all the points in the unit disk D. This indicates that in the worst cases, there is 41.7% probability of using 5 times the optimal number of unit disks, and 58.3% probability of using 4 times the optimal number of unit disks. ■

The proposed algorithm can be applied to the other two regular tessellations (i.e., square and triangle). Similar techniques can be used to analyze the cases of square and triangle. To condense the paper, we give the following theorems without proof.

Theorem 2
With the proposed algorithm using square tessellation, the number of output unit disks is no more than 6 times the optimal number of unit disks. In the worst cases, there is 57% probability of using 6 times the optimal number of unit disks, and 43% probability of using 4 times the optimal number of unit disks.

Theorem 3
With the proposed algorithm using triangle tessellation, the number of output unit disks is no more than 6 times the optimal number of unit disks. In the worst cases, there is 37.4% probability of using 6 times the optimal number of unit disks, 41.7% probability of using 5 times the optimal number of unit disks, and 20.9% probability of using 4 times the optimal number of unit disks.

The results are summarized in Table 1.

Table 1. The approximation ratios of the proposed algorithm for square, hexagon, and triangle tessellations

	Square tessellation	Hexagon tessellation	Triangle tessellation
Area needs 4 disks (percentage)	$\frac{4-\pi}{8}$ (43%)	$\frac{9\sqrt{3}-4\pi}{24}$ (41.7%)	$\frac{2\sqrt{3}-\pi}{4}$ (20.9%)
Area needs 5 disks (percentage)		$\frac{2\pi-3\sqrt{3}}{12}$ (58.3%)	$\frac{2\pi-3\sqrt{3}}{12}$ (41.7%)
Area needs 6 disks (percentage)	$\frac{\pi-2}{8}$ (57%)		$\frac{2\pi-3\sqrt{3}}{24}$ (37.4%)

Table 1 shows that in worst cases, both square tessellation and triangle tessellation need 6 times the optimal number of unit disks; however, hexagon tessellation only needs 5 times the optimal number of unit disks. If we consider $\sum(number\ of\ disks \times probability)$ for each polygon tessellation as the worst cases average number of disks needed to cover a unit disk, then square tessellation needs 5.14 disks, hexagon tessellation needs 4.583 disks, and triangle tessellation needs 5.165 disks. Again, hexagon tessellation needs less number of disks than that of the other two tessellations.

4 Simulation Results

In the simulation, we consider all the unit disks in MGDC problem are located at the integral coordinates. Thus, the optimal solution to the MGDC problem can be obtained by the binary integer programming. In this section, we first illustrate how to formulate the MGDC problem with binary integer programming, and solve it with Matlab. We then solve the MGDC problem by our proposed approximation algorithms with tessellations of hexagon, square, and triangle, respectively, and compare the simulation results with the optimal solutions.

Suppose the n points in P are uniformly distributed on an $m \times m$ square region, and the radius of the covering disk is r. The optimal solution to the MGDC problem can be computed through the following binary integer programming canonical form.

Minimize $I_1^T X$
Subject to:
$AX \geq I_2$
Where
I_1^T : the transpose of an $m^2 \times 1$ identity vector,
I_2 : an $n \times 1$ identity vector
X : an $m^2 \times 1$ binary vector, where each entry denotes whether the corresponding disk is chosen as a covering disk. 1 for yes, and 0 for no.
A : an $n \times m^2$ matrix where an element (i,j) is set to 1 if the distance between point i and the center of disk j is less than or equal to r; otherwise it is set to 0
We use Matlab binary ILP function bintprog() to compute the optimal results.

The simulation environment is set up as follows:

1. A square region of 200×200
2. Radius of covering disks is 10
3. Points are uniformly distributed
4. Number of points is ranged from 100 to 500
5. Each case runs 50 tests and take the average of their values

Fig. 5 shows the simulation results. As we expected, the performance of hexagon tessellation is better than that of square and triangle tessellations. More importantly, in our simulation, the number of disks generated by the proposed approximation algorithm (with hexagon tessellation) is less than 2 times the optimal solution for all the cases from 100 to 500 points. This indicates that our proposed approximation algorithm for the MGDC problem is practical for general cases.

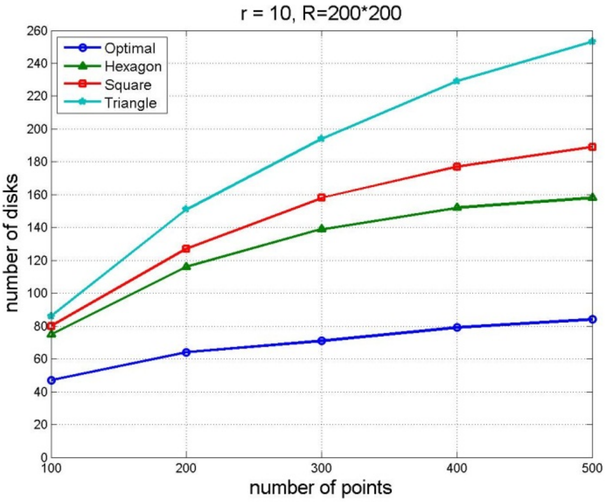


Fig. 5. Performance comparison of the optimal solutions to the solutions of the proposed algorithms with triangle tessellation, square tessellation, and hexagon tessellation

5 Conclusion

In this study, we propose an approximation algorithm for the MGDC problem, which has an application on the relay node placement problems of WSNs. The number of disks generated by the proposed algorithm is no more than 5 times the optimal solution. The time complexity of the algorithm is O(n×|R|), where n is the number of the points and |R| is the area of the deployed region.

We further show that if the set of points is uniformly distributed, then there is 41.7% probability of using less than or equal to 5 times the optimal number of disks, and 58.3% probability of using no more than 4 times the optimal number of disks.

According to our simulation experiments, the number of disks generated by our proposed algorithm is less than 2 times the optimal solution for the cases from 100 to 500 points in a square area of length 200 with $r = 10$. We believe that the worst cases of 5 times the optimal solution is very rare. In general, the number of disks generated by the proposed algorithm is usually less than 2 times the optimal solution, which makes the proposed algorithm practical under most of the conditions.

References

1. Ball, W.W.R.: Mathematical Recreations and Essays, 13th edn., pp. 105–107. Dover Publications, New York (2010)
2. Hochbaum, D.S., Maass, W.: Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI. J. ACM 32, 130–136 (1985)
3. Johnson, D.S.: NP-completeness columns: an ongoing guide. J Algorithms 3(2), 182–195 (1982)

4. Fowler, R., Paterson, M., Tanimoto, S.: Optimal packing and covering in the plane are NP-complete. Information Processing Letters 12, 133–137 (1981)
5. Franceschetti, M., Cook, M., Bruck, J.: A Geometric Theorem for Approximate Disc Covering Algorithms, Electronic Technical report ETR035 (2001), http://resolver.caltech.edu/CaltechPARADISE:2001.ETR035
6. Brönnimann, H., Goodrich, M.: Almost Optimal Set Covers in Finite VC-Dimension. Discrete and Computational Geometry 14, 463–479 (1995)
7. Gonzalez, T.: Covering a Set of Points in Multidimensional Space. Information Processing Letters 40(4), 181–188 (1991)
8. Franceschetti, M., Cook, M., Bruck, J.: A geometric theorem for network design. IEEE Transactions on Computers 53(4), 483–489 (2004)
9. Lloyd, E.L., Xue, G.: Relay Node Placement in Wireless Sensor Networks. IEEE Transactions on Computers 56(1), 134–138 (2007)
10. Tang, J., Hao, B.: ArunabhaSen Relay node placement in large scale wireless sensor networks. Computer Communications 29(4), 490–501 (2006)
11. Chen, Z.K., Chen, C.C.: Relay node placement in wireless sensor networks. In: Proceedings of the Fourth Workshop on Wireless Ad Hoc and Sensor Networks, Tainan, Taiwan, pp. 544–553 (2008)

Protein Structure Comparison and Visualization Tools on Cloud Platform

Yaw-Ling Lin[1], Chen-En Hsieh[1], Guan-Jie Hua[1], and Che-Lun Hung[2]

[1] Department of Computer Science and Information Engineering,
Providence University, Taichung, Taiwan
yllin@pu.edu.tw, {seanm7739,gt758215}@gmail.com
[2] Department of Computer Science and Communication Engineering,
Providence University, Taichung, Taiwan
clhung@pu.edu.tw

Abstract. The biological function of a protein molecule is decided by its 3D-shape, which eventually determines how the molecule interacts with other molecules in living cells. Identifying similar structures between proteins provide the opportunity to recognize homology that is undetectable by sequence comparison. Thus comparison and alignment of protein structures represents a powerful means of discovering functions, yielding direct insight into the molecular mechanisms.

This paper proposes approaches in providing visualization tools for pairwise 3D protein structure alignment; our web service takes advantage of the MapReduce paradigm as means of management and parallelizing tools under massive number of protein pairs examined under the experiment. It shows that our previously proposed sequential combinatorial algorithms are well parallelized under the map/reduce platform. These methods are tested on the real-world data obtained in from the RCSB PDB data set; the computation efficiency can be effectively improved proportional to the number of processors being used.

Keywords: protein structures comparisons, bioinformatics, visualization, VRML, MapReduce, Hadoop, cloud computing.

1 Introduction

Protein structures play critical roles in vital biological functions [1]. With more than 83,000 protein structures determined by the advances in X-ray crystallography and NMR spectroscopy to date, molecular biologists these days proceed in the direction of analyzing and classifying these protein structures in order to discover the structural relationships with protein functions [2].

Detection of proteins with a similar fold can suggest a common ancestor, and often a similar function [3]. Comparison of 3D structures makes it possible to establish distant relationships, even between protein families distinct in terms of sequence comparison alone. This is why structural alignment of proteins increases our understanding of more distant evolutionary relationships [4, 5]. The link between structural classification and sequence families enables us to study functions of various folds, or whole proteins [6].

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 167–176.
DOI: 10.1007/978-3-642-35452-6_19

There have been several methods proposed to compare protein structures and measure the degree of structural similarity based on alignment of secondary structure elements as well as alignment of intra and inter-molecular atomic distances. The basic ideas are rapid identification of pair alignments of secondary structure elements, clustering them into groups, and scoring the best substructure alignment. For examples, the VAST system [7] is based on continuous distribution of domains in the fold space. The FSSP/DALI system [8] provides two levels of description - a coarsegrained one and one with a fine-grained resolution. The method, CATH, provides the complete PDB fold classification by domains and links to other sources of information. The two methods, CE and LGscore2 [9] focus on the local geometry rather than global features such as orientation of secondary structures and overall topology (as in the case of VAST or DALI). VAST has been used to compare all known PDB domains to each other. The results of this computation are included in NCBI's Molecular Modelling Database at http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.html.

Note that there must be an atom-pairing scheme before one can do the structure alignment computation. The first atom of the first selection is compared to the first atom of the second selection, fifth to fifth, and so on. Incorporating with ideas of bipartite matching and 3-parameter isometric transformation, Lin et al. [10, 11] proposed methods of using parametric searching strategies with adaptive controls, and demonstrated that more accurate and similar protein structure pairings are possible comparing to previous known results like VAST [7] or CE [9].

One of the crucial steps of these algorithms is finding a good isometric transformation, which leads to the best atom-pairing alignment between two proteins. In this paper, we propose algorithms of for efficiently locating more suitable isometric transformations of one structure and aligning it to the other structure. Based upon the periodical property of the parametric settings, we propose parametric searching strategies by approximations with power series and trigonometric series. We show the eRectiveness of the proposed parametric searching strategies by a set of experiments, which leads to better alignments of structure pairing in general.

Hadoop [12] is a software framework intended to support data-intensive distributed applications. It is able to process petabytes of data with thousands of nodes. Hadoop supports MapReduce programming model [13] for writing applications that process large data set in parallel on Cloud Computing environment. The advantage of MapReduce is that it allows for distributed computing of the map and reduction operations. Each map operation is independent of the others and all maps can perform the tasks in parallel. In practice, the number of the map is limited by the data source and/or the number of CPUs near that data. Similarly, a set of reducers can perform the reduce operations. All outputs of the map operations which share the same key are presented to the same reducer, at the same time. In addition, one of the important benefits to use Hadoop to develop the applications is due to its high degree of fault tolerance. Even when running jobs on a large cluster where individual nodes or network components may experience high rates of failure, Hadoop can guide jobs toward a successful completion. Many applications of bioinformatics are often computationconsuming; sometimes it needs weeks or months to complete the jobs. The traditional parallel models, such as MPI, OpenMP and Multi-thread, are not suitable to such applications, where a fault occurred in some nodes leads the entire application into total failure. In these situations, the Hadoop platform are considered as a much better solution for these real-world applications. Recently, Hadoop has been applied in various domains in bioinformatics [14, 15].

2 Method and Materials

2.1 Protein Structure Analysis

The main idea of our local refinement algorithm for finding a suitable matching between two sets of points before utilizing the RMSD procedure to fine-tune the final result is by adjusting the suitable parameter sets by ways of searching the underlying parametric space.

Root Mean Squared Deviation

The smallest root mean squared deviation (rmsd) is a least-squares fitting method for two sequences of points [8]. The idea is to align atom vectors of the two given (molecular) structures, and use the common least averaged squared errors as a measurement of differences between these two (paired) sequences. Formally, let $P=\langle p_1,\ldots,p_n\rangle$ and $Q=\langle q_1,\ldots,q_n\rangle$ be two sequences of points. We assume that P is translated so that its centroid $(\frac{1}{n}\sum_{k=1}^{n} p_k)$ is at the origin. We also assume that Q is translated in the same way. For each point or vector x, let $(x)_i (i=1,2,3)$ denote the i-th(X,Y,Z) coordinate value of x ,and $\|x\|$denote the length of x. Let

$$\text{RMSD}(P,Q,R,\mathrm{a})=\sqrt{\frac{1}{n}\sum_{k=1}^{n}\|Rp_k+\mathrm{a}-q_k\|^2} \quad , \tag{1}$$

where " R " is a rotation matrix and " a is a translation vector. Then, the *rmsd* value $d(P,Q)$ between P and Q Is defined by $d(P,Q)=\min_{R,\mathrm{a}} d(P,Q,R,\mathrm{a})$. We refer to Martin's ProFit package (standing for protein fitting system) 16 and write a program to calculate the *rmsd* between C-α atoms of paired protein backbones with C language. Fitting was performed using the McLachlan algorithm [17]

Isometric Rotation Transformation

According to Euler's rotation theorem 18, any rotation about the origin point can be described by using three angular parameters. The rotation is determined by 3 consecutive rotations with 3 *Euler angles* (α,β,γ). The first rotation is done by the angle α around the *z*-axis, the second is done by the angle β around the *x*-axis, and the third rotation is done by the angle γ around the *z*-axis. Please refer to [19] for more detailed discussions about the transformation.

Analog to Euler's setting, we can also consider the point of north-pole $\mathbf{n} = (0, 0, 1)^T$ on the unit sphere. After the rotation, $\boldsymbol{R}$, say $\mathbf{n}$ is rotated to another point $\mathbf{p} = (x, y, z)^T$; i.e., $\mathbf{p} = \boldsymbol{R}\mathbf{n}$. Let α denote the angle $\angle \mathrm{n}O\mathrm{p}$. Note that α determines the *z*-coordinate of

p. To determine *x*-coordinate and *y*-coordinate of **p**, the point is rotated around the *z*-axis for the angle β on the unit sphere. Note that there are infinitely numbers of rotation that transform **n** to **p**. The particular rotation ***R*** can be decided by rotating all other points around the vector **p** by the angle γ. It is not hard to verified that, in such a way, any rigid rotation transformation can be parameterized by the three-tuple (α,β,γ). Thus, we call a vector $\mathbf{p} = (x, y, z)^T$ on the surface of the unit sphere a probe. Note that the ***movement*** of each probe is started from the northpole $(0, 0, 1)^T$ to other points in the sphere. The position of **p** is decided by the parameters (α,β), and exact rotation is fixed by the self-rotation angle γ.

As a result, we reduce the problem of finding the good rotation matrix to the new problem of finding a good 3-parameter setting. The rotation matrix is thus characterized by just adjusting the 3 uniformly distributed parameters.

Minimum Bipartite Matching

We use the minimum bipartite matching algorithm to find the best matching between two sets of points to decide the best matching for the rmsd procedure. We adopted the Munkres [20, 21] algorithm. In order to improve the efficiency of computation, we rewrite the Munkres algorithm, which is implemented on a public available perl module, by C language.

Parametric Adjustment with Trigonometric Series

In the trigonometric series estimation algorithm [22], the three parameters (rotational angles) are assumed to be independent. Three parameters are adjusted one by one, and the estimation function corresponding to the affected *rmsd* values is subsequently better estimated by increasing the curvature degree of the estimated function. The trigonometric series function is described as the following:

$$\begin{aligned} f(\theta) = C_1 &+ C_2 \cos \pi\theta + C_3 \sin \pi\theta \\ &+ C_4 \cos 2\pi\theta + C_5 \sin 2\pi\theta \\ &+ C_6 \cos 3\pi\theta + C_7 \sin 3\pi\theta + \ldots \\ &+ C_{2k} \cos k\pi\theta + C_{2k+1} \sin k\pi\theta. \end{aligned} \tag{2}$$

Note that $f(\theta)$ denote the corresponding rmsd value with respect of one of the three parameters, (α,β,γ). The *k* usually reflects the numbers of local maximal points in the approximated curve.

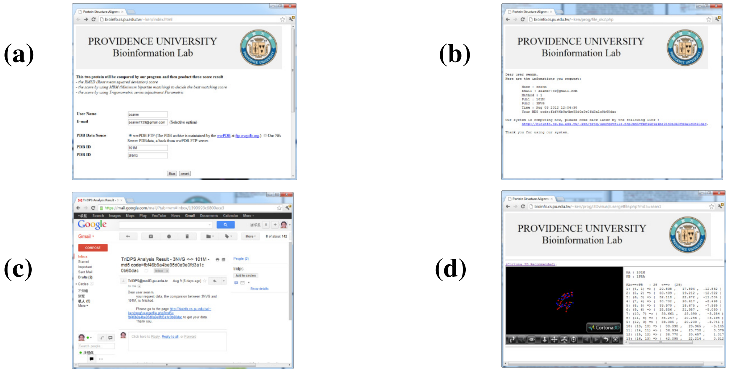
The algorithm [22] improves the ***rmsd*** value between a protein structure pair by finding better alignment list. A set of experiments is designed to test the parameters; these adjusted parameters are set to perform the experiments over a thousand of protein pairs, which are uniformly random sampled from the PDB database. As the results shown that our methods improve the alignment computed by the VAST by an averaged improvement ratios about 11 %. The results of Fig. 1 demonstrate that the method of using 3D Euclidean distance minimum bipartite matching with trigonometric series estimated parametric searching scheme indeed improves existed known system like the VAST.[23]

The initial method method	The average *rmsd* of initial method	The average *rmsd* after trigono. adj.
VAST	1.9572	1.7329 (11.46%)
semi-local-Alig	1.8378	1.7697 (9.58%)
cut1	2.1031	1.7947 (8.30%)
M1	1.9025	1.8047 (7.79%)
cut3	3.4563	1.8526 (5.34%)
M2	2.0612	1.9033 (2.75%)
Main Vector	4.2697	2.5907 (-32.37%)

Fig. 1. The rmsd values of trigonometric series adjustment after initial alignment of VAST, Seg-Alig(cut3), Seg-Alig(cut1), Semi-local-Alig, M1, M2 and Main Vector. Notice the improved ratio (in%) comparing to VAST's original rmsd.

PDB2VRML

Our visualization tools take advantage of the PDB2VRML tool [24] to convert the molecular structure information produced by the protein alignment algorithm into the VRML format (.wrl). We rewrite the PDB2VRML so that each acid pair is linked by a stick; the length of each stick measures the distance between two C-alpha, representing the corresponding amino acids pair. The comparison result can be viewed by molecular biologists in our web service via the Cortona3D Viewer. As an example, Fig. 2 shows the result of a sample PDB pair.



PROVIDENCE UNIVERSITY
Bioinformation Lab

This two protein will be compared by our program and then product three score result
- the RMSD (Root mean squared deviation) score
- the score by using MBM (Minimum bipartite matching) to decide the best matching score
- the score by using Trigonometric series adjustment Parametric

User Name seanm
E-mail seanm7739@gmail.com (Selective option)

PDB Data Souce ● wwPDB FTP (The PDB archive is maintained by the wwPDB at ftp.wwpdb.org.) ○ Our Nfs Server PDBdata, a back from wwPDB FTP server.
PDB ID 101M
PDB ID 3NVG

```
Dear user seanm.
Here are the infomations you request:

        Name : seanm
        Email : seanm7739@gmail.com
        Method : 1
        Pdb1 : 101M
        Pdb2 : 3NVG
        Time : Aug 09 2012 12:04:30
        Your MD5 code:fbf46b9a4be95d0a9e0fd3a1c0b60dac

Our system is computing now, please come back later by the following link :
        http://bioinfo.cs.pu.edu.tw/~ken/prog/usergetfile.php?md5=fbf46b9a4be95d0a9e0fd3a1c0b60dac.

Thank you for using our system.
```

TriDPS Analysis Result - 3NVG <-> 101M - md5 code=fbf46b9a4be95d0a9e0fd3a1c0b60dac

TriDPS@mail3.pu.edu.tw Aug 9 (6 days ago)
to me

Dear user seanm,
your request data, the comparsion between 3NVG and 101M, is finished.

Please go to the page http://bioinfo.cs.pu.edu.tw/~ken/prog/usergetfile.php?md5=fbf46b9a4be95d0a9e0fd3a1c0b60dac to get your data.
Thank you.

Click here to Reply, Reply to all, or Forward

(Cortona 3D Recommended).

```
PA : 101M
PB : 1PRA

PA<==>PB   : 29  <==>  (29)
1: (4, 1) => (  29.898 ,   17.894 ,  -12.882 )
2: (5, 2) => (  33.489 ,   19.212 ,  -12.922 )
3: (6, 3) => (  32.118 ,   22.472 ,  -11.504 )
4: (7, 4) => (  30.702 ,   20.617 ,   -8.498 )
5: (8, 5) => (  33.970 ,   18.675 ,   -7.985 )
6: (9, 6) => (  35.856 ,   21.987 ,   -8.080 )
7: (10, 7) => (  33.661 ,   23.390 ,   -5.284 )
8: (11, 8) => (  34.247 ,   20.256 ,   -3.195 )
9: (12, 9) => (  38.005 ,   20.200 ,   -3.741 )
10: (13, 10) => (  38.390 ,   23.965 ,   -3.145
11: (14, 11) => (  36.934 ,   23.758 ,    0.379
12: (15, 12) => (  38.770 ,   20.457 ,    1.017
13: (16, 13) => (  42.095 ,   22.214 ,    0.312
```

Fig. 2. Result of PDB comparison: (a) biologist enquiry a pair of proteins by entering two PDB ids. (b) The web service provides a resulting web link. (c) The completed protein structure alignment result. (d) Visualization of the final PDB alignment results in 3D view.

The computation needed to solve the structure comparison problem, like many scientific computation/simulation problem, is very time-consuming under cases of large structures and large number of paired structures. It is desirable to implement the system under massive parallel machines cluster, e.g., the grid-environment, to increase the throughput of the system.

2.2 Hadoop MapReduce Framework

Hadoop is a software framework for coordinating computing nodes to process distributed data in parallel. Hadoop adopts the map/reduce parallel programming model, to develop parallel computing applications. The standard map/reduce mechanism has been applied in many successful Cloud computing service providers, such as Yahoo, Amazon EC2, IBM, Google and so on. An application developed by Map/Reduce is composed of Map stage and Reduce stage (optionally). Fig. 3 illustrates the Map/Reduce framework. Input data will be split into smaller chunks corresponding to the number of Maps. Output of Map stage has the format of <*key, value*> pairs. Output from all Map nodes, <*key, value*> pairs, are classified by key before being distributed to Reduce stage. Reduce stage combines value by key. Output of Reduce stage are <*key, value*> pairs where each key is unique.

Hadoop cluster includes a single master and multiple slave nodes. The master node consists of a jobtracker, tasktracker, namenode, and datanode. A slave node, as computing node, consists of a datanode and tasktracker. The jobtracker is the service within Hadoop that farms out Map/Reduce tasks to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack. A tasktracker is a node in the cluster that accepts tasks; Map, Reduce and Shuffle operations from a jobtracker.

Hadoop Distributed File System (HDFS) is the primary file system used by Hadoop framework. Each input file is split into data blocks that are distributed on datanodes. Hadoop also creates multiple replicas of data blocks and distributes them on datanodes throughout a cluster to enable reliable, extremely rapid computations. The namenode serves as both a directory namespace manager and a node metadata manager for the HDFS. There is a single namenode running in the HDFS architecture.

2.3 Protein Structure Analysis on Map/Reduce Framework

Fig. 3 illustrates the MapReduce framework for the protein structure analysis. Assume that the number of compute node is *N* and the number of protein pairs lines is *P* in the protein pairs list file, the *P* lines will become *P* map tasks and send to hadoop by streaming program. Each computing node receives a map and then performs the analysis work. When a node completes the map task, the node passes the score to the Reduce and receives next map task to compute unless the total map are finished. Generally, each node will be assigned to deal with *P/N* maps.

Therefore, Reduce will have *P* RMSD scores. The RMSD score is described in the previous section 2.A. The reduce operation writes the protein structure pair analysis scores line by line to the output file on HDFS.

3 The Web Service (Cloud) Platform

Our protein structure comparison web service can be seen as a SAAS (Software as a service) under general cloud computing terminology [25]. Here we describe its platform structure as the following.

Platform Framework

Fig. 3 shows our web service platform consisting of a Receipting site and a Processing site. The Receipting site serves as a regular web page service responsible for processing general user requests; while the Processing site is the Hadoop map/reduce platform that is ideal for massive parallel processing for huge amount of processing data.

There are three programs within the receipting site: First, **get-req.php** is a PHP program receiving user request; the second program, **collect.pl,** is a Perl program responsible for collecting PDB data from NCBI PDB before preparing the needed ready list. The third program, **dispatch.pl,** is responsible for sending the ready-list to the Processing site, and collecting processed results from the Processing site before sending out the final web page to the user. Note that **dispatch.pl** regular checks for ready-list and processed results from time to time.

There is one program within the processing site: **process.pl** is a Perl program responsible for receiving the ready-list from the Receipting site, and invoking protein comparison in hadoop platform. Note that the **process.pl** regularly checks for the ready-list from time to time.

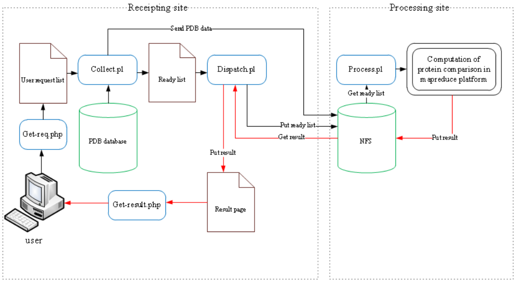


Fig. 3. Platform framework. The work flow between 2 sites.

4 The Experimental (Cloud) Platform

4.1 Experimental Environment and Data Source

The experiments are performed on one NFS server and four IBM blade server in the Providence University Cloud Computation Laboratory. Each server is equipped with two Quad-Core Intel Xeon 2.26GHz CPU, 24G RAM, and 296G disk under the Ubuntu version 10.4 with the virtualization platform KVM/QEMU. Under the current system environment, we create 8 virtual machines by KVM; each virtual machine is set to one core CPU, 2G RAM, and 30G disk running under the O.S. Ubuntu version 10.4 with Hadoop version 0.2 MapReduce platform. Each virtual machine is responsible for one map operation and one reduce operation. The total number of the map/reduce operations is up to 8 respectively.

The Protein structure data sources are gathered from the World Wide Protein Data Bank (http://www.wwpdb.org), which is a PDB format content with 4-letter entry name. The Protein Data Bank maintains this single archive of macromolecular structural data freely and publicly available to the global community. These protein structure data are downloaded from the wwPDB's ftp server (ftp://ftp.wwpdb.org/), where the number of protein structure data is 80,402. These data are treated as the testing data for our experiments.

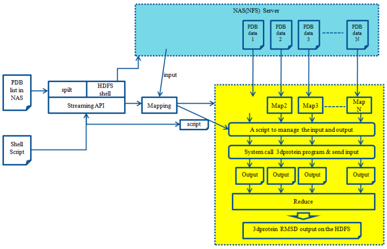


Fig. 4. Protein structure analysis use hadoop platform use Map/reduce framework

4.2 Experimental Result

To assess the performance of the proposed Hadoop MapReduce algorithm, we compare the computation time between various number of structure data and various number of map/reduce operations. Two factors, the number of protein structure data and the number of protein structure atoms, affect the performance of sequential algorithm and the proposed algorithm. The RMSD scores are calculated with protein ID pairs. Fig. 5 and Fig. 6 illustrate the comparisons between sequential algorithm and our proposed algorithm under the MapReduce framework. In Fig. 5 and Fig. 6, the maximum atom numbers of protein pairs are between 1-100 and 101-200, respectively.

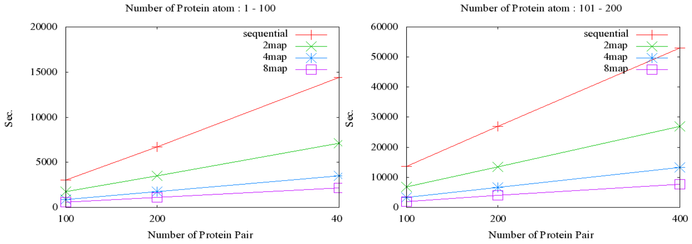
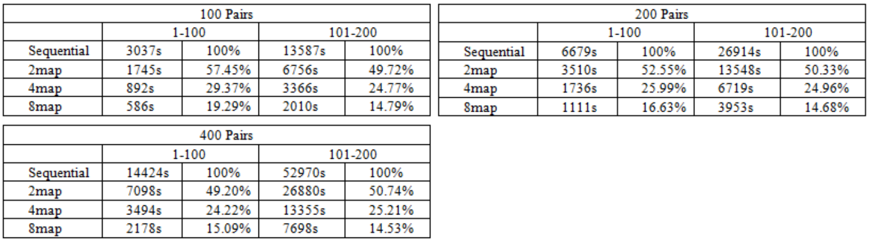


Fig. 5. Performance comparison between sequential compute RMSD of protein structure pair and MapReduce compute RMSD of protein structure pair using various number of map. The Number of Protein atom has been split up into two types, 1-100 and 101-200.



100 Pairs				
	1-100		101-200	
Sequential	3037s	100%	13587s	100%
2map	1745s	57.45%	6756s	49.72%
4map	892s	29.37%	3366s	24.77%
8map	586s	19.29%	2010s	14.79%

200 Pairs				
	1-100		101-200	
Sequential	6679s	100%	26914s	100%
2map	3510s	52.55%	13548s	50.33%
4map	1736s	25.99%	6719s	24.96%
8map	1111s	16.63%	3953s	14.68%

400 Pairs				
	1-100		101-200	
Sequential	14424s	100%	52970s	100%
2map	7098s	49.20%	26880s	50.74%
4map	3494s	24.22%	13355s	25.21%
8map	2178s	15.09%	7698s	14.53%

Fig. 6. Computing time and the percentage relationship of sequential algorithm and using MapReduce algorithm. The pairs is number of protein structure data pair and be separated form number of protein atom by 1-100 and 101-200.

The experimental result reveals that the computational time is effectively reduced when more map operations are deployed. Comparing to the original sequential algorithm, two, four and eight map operations improve the computation time by factor of two, four and eight times. The computation efficiency can be effectively improved proportional to the number of processors being used.

In Fig. 5 and Fig. 6, it is observed that computational time increases corresponding to number of protein pair and number of protein atom. The computation time with 1-100 atoms is less than that with 101-200 atoms. More number of atom lead to higher computational cost. These experimental results are corresponding to the algorithm analysis in previous section.

4.3 Web System

Our system is accessible through the web system at sites.google.com/site/pucspsas/ and bioinfo.cs.pu.edu.tw/~ken/3Dvisual.html for molecular biologists interested in three dimensional protein structures alignment/comparison problems. The user of our web service usually provides two protein PDB IDs, and the service will search protein structure data from wwPDB ftp and perform the desired protein structure comparison using the chosen set of algorithms.

To avoid the time delay, our web service also provides user access keys for user to check the result later on after they submit the desired query. User can come back and check the result after operation finished by our local server.

5 Concluding Remarks

In this paper, we have proposed visualization tools for pairwise protein structures alignment and compare the performances between the sequential version and the parallelized map/reduce version of our previously proposed protein structure alignment algorithms. 3D protein structures alignment is useful in analyzing and classifying these protein structures in order to discover the structural relationships with protein functions. Our methods provide the possibility to establish relevant relationships, even between protein families distinct in terms of sequence comparison alone.

In the future, we will investigate algorithmic methods for discovering bioinformatics functions, and try to parallelize these methods to provide more perspectives for biologists to improve the performance of these computation frameworks on for analyzing the increasingly huge bioinformatics data.

Acknowledgment. This research was partially supported by the National Science Council under the Grants NSC-99-2632-E-126-001-MY3.

References

1. Gerstein, M., Jansen, R., Johnson, T., Tsai, J., Krebs, W.: Motions in a data-base framework: from structure to sequence. In: Thorpe, M.F., Duxbury, P.M. (eds.) Rigidity Theory and Applications, pp. 401–442. Kluwer Academic/Plenum Publishers (1999)
2. Echols, N., Milburn, D., Gerstein, M.: Molmovdb: analysis and visualization of conformational change and structural flexibility. Nucleic Acids Res. 31, 478–482 (2003)

3. Dietmann, S., Holm, L.: Identification of homology in protein structure classification. Nature Struct. Biol. 8, 953–957 (2001)
4. Bujnicki, J.M.: Phylogeny of the restriction endonucleaselike superfamily inferred from comparison of protein structures. J. Mol. Evol. 50, 38–44 (2000)
5. Johnson, M.S., Sutcliffe, M.J., Blundell, T.L.: Molecular anatomy: Phyletic relationships derived from threedimensional structures of proteins. J. Mol. Evol. 30, 43–59 (1990)
6. Lin, Y.L., Lin, Y.H., Yu, P.S., Chang, H.C.: Randomized algorithms for three dimensional protein structures alignment. In: The 6th International Symposium on Computational Biology and Genome Informatics, pp. 122–125 (2005)
7. Gibrat, J.F., Madej, T., Bryant, S.H.: Surprising similarities in structure comparison. Curr. Opin. Struct. Biol. 6(3), 377–385 (1996)
8. Holm, L., Sander, C.: Touring protein fold space with DALI/FSSP. Nucleic Acids Res. 26, 316–319 (1998)
9. Shindyalov, I.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. 11, 739–747 (1998)
10. Lin, Y.L., Huang, S.P.: Tools and algorithms for refined comparison of protein structures. In: The 6th WSEAS International Conference on Microelectronics (MINO 2007), Istanbul, Turkey (2007)
11. Shin, H.S., Huang, S.P., Lin, Y.L.: Parametric searching algorithms with adaptive strategy for three dimensional protein structures alignments. In: National Computer Symposium (NCS 2007), Taichung, Taiwan, pp. 144–154 (2007)
12. Hadoop - Apache Software Foundation project home page, `http://hadoop.apache.org/`
13. Taylor, R.C.: An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC Bioinformatics 11, S1 (2010)
14. Dean, J., Ghemawat, S.: MapReduce: A Flexible Data Processing Tool. Communications of the ACM 53, 72–77 (2010)
15. Schatz, M.: Cloudburst: highly sensitive read mapping with MapReduce. Bioinformatics 25, 1363–1369 (2009)
16. Martin, A.C.R., `http://www.bioinf.org.uk/software/profit/`
17. McLachlan, A.D.: Rapid comparison of protein structures. Acta Cryst. A38, 871–8783 (1982)
18. Euler, L.: Formulae generales pro trandlatione quacunque corporum rigidorum. Novi Acad. Sci. Petrop. 20, 189–207 (1775)
19. Gray, A.: A treatise on gyrostatics and rotational motion. MacMillan, London (1918)
20. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5, 32–38 (1957)
21. Bourgeois, F., Lassalle, J.C.: An extension of the munkres algorithm for the assignment problem to rectangular matrices. Communications of the ACM 14, 802–804 (1971)
22. Shin, H.S., Lin, Y.L., Jiang, W.D.: Protein structures alignment algorithms by parametric searching with trigonometric series. In: Proceedings of the 25th Workshop on Combinatorial Mathematics and Computation Theory, Hsinchu, Taiwan, pp. 44–54 (2008)
23. Chen, H.S., Lin, Y.L.: Comparisons of Semi-local Alignment Algorithms for Protein Structures. In: Proceedings of the 27th Workshop on Combinatorial Mathematics and Computation Theory (CMCT 2010), Taichung, Taiwan, April 30-May 1, pp. V6-254–V6-259 (2010)
24. Copyright (c) 1998 by Horst Vollhardt. All rights reserved. This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself (1998)
25. Chou, T.: Introduction to Cloud Computing: Business & Technology

A New GPS Position Correction Method Based on Genetic Programming

Jung Yi Lin*, Ming Chih Tung, Chia Hui Chang, Chao Chung Liu, and Ju Fu Peng

Department of Computer Science and Information Engineering
Chien Hsin University of Science and Technology, Jhongli, Taiwan
jylin@ieee.org, {mctung,m10013001,m10013012}@uch.edu.tw,
by020827@gmail.com

Abstract. More and more mobile devices equipped with global positioning system (GPS) are helpful in tour navigation. However, such consumer-grade GPS receivers usually have low accuracy in positioning and require position correction algorithms. In this paper, we proposed an evolutionary computation based technique to correct a GPS receiver with another GPS receiver and a known reference point. The proposed technique can be implemented without changing any hardware. It generates a correction function from given NMEA (national Marine Electronics Association) information. Using such function to derive correct position information could be efficient. Experiments are conducted to demonstrate performance of the proposed technique. Positioning error could be reduced from in the order of 10 m to in the order of 1 m.

Keywords: Global Positioning System (GPS), Position Correction, Genetic programming, evolutionary computation.

1 Introduction

Global Positioning System, GPS, has been applied in various purposes such as ocean navigation, meteorology, military tasks, mapping, tour design, path tracking tools, and more [1]. Recently many mobile devices equip with GPS [2]. Those devices including tablet PC and smart phone provide city map to help users not lose their way or find a shortest path to their target.

GPS receiver receives satellite signal from some of constellation 24 GPS satellites. Those satellites are controlled by the United States Department of Defense [2]. The position of a GPS receiver, u, can be derived by the pseudorange ρ of a satellite. Let s be a satellite located at (x_s, y_s, z_s), and u is located at (x_u, y_u, z_u), ρ can be defined as

$$\begin{aligned}\rho &= \|s-u\| + ct_r + v_\rho \\ &= \sqrt{(x_s-x_u)^2+(y_s-y_u)^2+(z_s-z_u)^2} + ct_r + v_\rho\end{aligned} \tag{1}$$

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 177–185.
DOI: 10.1007/978-3-642-35452-6_20

where c is the speed of light; tr is the offset of receiver clock; and v_ρ stands for a random noise that is expected to be zero.

Theoretically, given four satellite coordinates without error, the exact position of u, (x_u, y_u, z_u), can be determined. However, GPS positional accuracy is affected by many factors [4-5], for example, radio signal corruption, ephemeris error, satellite and receiver clock offset, multipath error, receiver measurement noise, satellite geometry measures, tropospheric delay and ionospheric delay [6]. In general, GPS position accuracy is degraded to in the order of 10 m [7].

Many techniques are proposed to improve GPS position accuracy. A commonly used technique is to use relative positioning [8]. Relative positioning methods, including static, rapid static, pseudo-kinematic, kinematic, and real-time kinematic [8-10] have been proved capable of enhancing GPS accuracy. In [8], M. Berber et al. observed that pseudo-kinematic technique produces closest results, which could reduce the error to 2 centimeters.

Differential correction is an effective method to improve GPS positional accuracy. A GPS receiver with such technique is called dGPS. A typical differential correction requires a reference stationary receiver, or base station, at a known location [11-12], as shown in Fig1. The difference between the location obtained by the GPS receiver and the location of the reference stationary receiver is evaluated and is applied to a roving GPS receiver to correct its location under the assumption that relatively close receivers suffers similar errors. Ideally, dGPS is capable of reducing errors due to ionospheric and tropospheric delay, ephemeris error and satellite clock error. Unfortunately, those errors due to receiver noise, multipath error, or poor satellite measurement geometry cannot be eliminated.

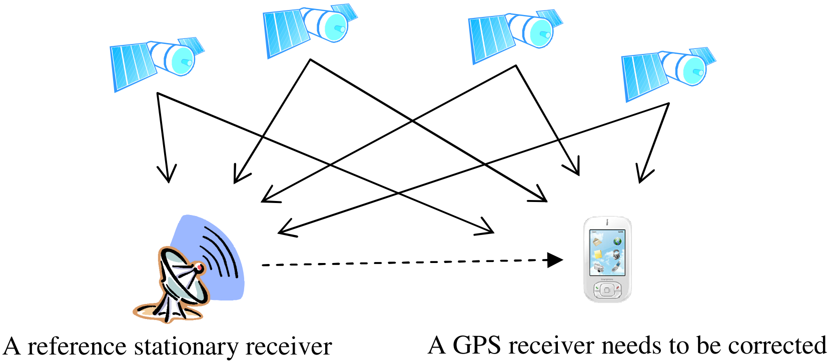

A reference stationary receiver
A GPS receiver needs to be corrected

Fig. 1. An illustration of dGPS technique. The precise location of the reference stationary receiver is known. A GPS receiver which requires correction accepts both signals from satellites and the stationary receiver.

The main drawback of dGPS technique is that reference stationary receivers are not common in many countries. However, many places have been precisely measured for their location and are open to access. A consumer-grade GPS receiver can easily be the role of a reference stationary receiver. Given two GPS receivers, G1 and G2, where G1 is placed on a known location, L. The location information obtained by G1 can be used to correct G2. Such scenario is shown Fig2. In this paper, we will use two consumer-grade GPS receivers to construct the scenario. Instead of using survey-grade GPS receivers, which have high accuracy and have been applied correction techniques, consumer-grade GPS receivers could be more common for most of users.

A navel position correction technique is proposed in this paper. This technique is based on differential correction and genetic programming (GP) [13]. GP will be used to generate a correction function obtained by NMEA information [15] derived from both a GPS receiver in a known location and a GPS receiver which needs to be corrected. The receiver which requires to be corrected will apply the function to have new location information.

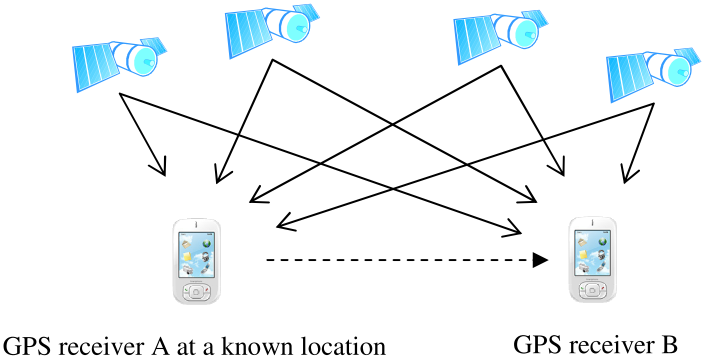


Fig. 2. A consumer-grade GPS receiver at a location which has exact known location

2 Genetic Programming

Genetic programming [13] is a research area of evolutionary computation. It has been proved that GP is capable of finding a solution efficiently. GP, like other techniques in evolutionary computation, generates possible solutions randomly for the given problem under given constrains, called *individuals*. In this paper, individuals are represented as functional expressions. Those individuals form a set named *population*. In order to produce new solutions, genetic operators such as crossover and mutation are applied on selected individuals, called *parents*, to create offspring and mutant. Comparing the fitness degree of those offspring and mutant with parents, which has higher fitness value will be kept as survived individuals. All survived individuals will replace the original population. Fitness degree of an individual is determined by a predefined fitness function, which is highly depending on the given problem. A *generation* is finished once the original population is fully replaced. After a number of generations, evolutionary process is over and the individual with highest fitness is the result of the population [14].

Training instances are constructed by raw information obtained from two GPS receivers and the known location. GPS receivers are capable of transferring different types of NMEA interpreted sentences [15]. In this work, we used GPGGA to represent position information, as shown in Table 1.The third, fifth, tenth, and twelfth field are symbols that can be harmlessly eliminated. The value of sixth field indicates GPS quality which is fixed. The thirteenth and fourteenth are available only when dGPS applied. The fifteenth is the checksum used to identify correctness of received data. In conclusion, 8 out of 15 fields can be removed. Two GPS receivers construct a 13-feature training instance after eliminating a redundant UTC time feature because two GPS should have identical UTC time information. Those features with longitude and latitude of the known location form a 15-feature training instance, as shown in Table 2. The target value is either known latitude or known longitude to which we intent to correct GPS receiver as close as possible.

Table 1. Fifteen fields of GPGGA sentence

Number	Meaning
1	UTC of position
2	Latitude
3*	N or S
4	Longitude
5*	E or W
6*	GPS quality indicator 1: invalid 2: GPS fix 3: dGPS fix
7	Number of satellites in use
8	Horizontal dilution of position
9	Antenna altitude above/below mean sea level (geoid)
10*	Meters (antenna height unit)
11	Geoidal separation
12*	Meters (units of geoidal separation)
13*	Age in seconds since last update from differential reference station
14*	Differential reference station ID
15*	Checksum

* removed features

An individual, idv, is defined as a functional expression composed of variables, operators, and constants:

$$idv = \left(S_v, S_{op}, C\right)$$

$$S_v = \{X_i \mid i = 1,2,...,17\}$$

$$S_{op} = \{+, -, \times, /, \sin, \cos, \log\}$$

$$C = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$$

An individual is a function mapping 17 real value features with constants into single real value, i.e., *idv*: ($\mathbf{R}^{|17|} \cup C$) → **R**.

A training instance, *t*, and the training set, *T*, are defined as follows:

$$t = (\text{target value}, f_1, f_2, ..., f_{15})$$

$$T = \{t_i \mid i = 1, 2, ..., |T|\}.$$

the target value is either known latitude or known longitude. The fitness of an individual is defined by

$$fitness = \sqrt{\sum_{i=1}^{|T|} \left(idv(t_i) - \text{target value}\right)^2} \qquad (2)$$

where idv(ti) stands for the calculated valued of ti by the individual. Over-fitting is a situation that a trained individual highly fits the training set but obtains relatively poor performance for the test set. To avoid such phenomenon happens, the validation process is applied. An individual having highest score is the output of the population.

$$\text{score} = \text{fitness} + \sqrt{\sum_{i=1}^{|V|} \left(idv(t_i) - \text{target value}\right)^2} \qquad (3)$$

where |V| is the number of instances in the validation set.

Table 2. Features of a training instance

Number	Meaning
Target value	Known latitude / known longitude
1-7	$GPGGA from GPS receiver 1 as shown in 0
8-13	$GPGGA from GPS receiver 2 as shown in 0without the UTC of position
14	Known latitude (where GPS receiver 1 locates)
15	Known longitude (where GPS receiver 1 locates)

3 Experiments

Two public reference positions are provided by Ching Yun University: CYU01 (24.94728, 121.22916) and CYU02 (24.94719, 121.22951). Satellite image extracted from Google Earth is shown in Fig.3. Two consumer-grade GPS, HOLUX GPSport 245 [16], precisely placed on CYU01 and CYU02 for 24 hours to collect position information. After eliminating noisy data, the dataset contains 59,209 instances. We used 19737, 19736, and 19736 instances as training set, validation set, and test set, respectively. To reduce the conversion error in calculating longitude/latitude format, CYU01 and CYU02 are transformed into NMEA format, (2456.8368, 12113.7496) and (2456.8314, 12113.7706), respectively. In this paper, we conduct experiments that use CYU02 to be the reference point, and use CYU01 as the target position. Five experiments with identical settings are executed to demonstrate average performance. Settings used for GP are shown in Table 3.

Average distance errors between position information obtained by GPS receivers and the two fixed positions are considerable. Average error is in order of 10 meter, as summarized in Table 4. We further show the image combining fixed positions and average positions in Fig 3. It is clear that the position determined by the GPS receiver is in high error.

The training phase of GP is time-consuming. Training time records of five experiments are summarized in Table 5. It requires more than one hour completing one experiment and the training time are vary. Long training time usually caused by long individuals because it takes more time to evaluate its fitness value. The content of an individual is generated randomly at initial or constructed by crossover or mutation. Since we do not know in prior what size is suitable for the optimum solution, it is unreasonable to cancel training phase by the reason that the sizes of individuals are long. It seems that the training time is not acceptable in real scenario. However, real scenario barely has 39,473 seconds for a position correction procedure. The training time is affected by the number of training instances. Fewer training instances would greatly cost less training time.

Table 3. Genetic programming parameters

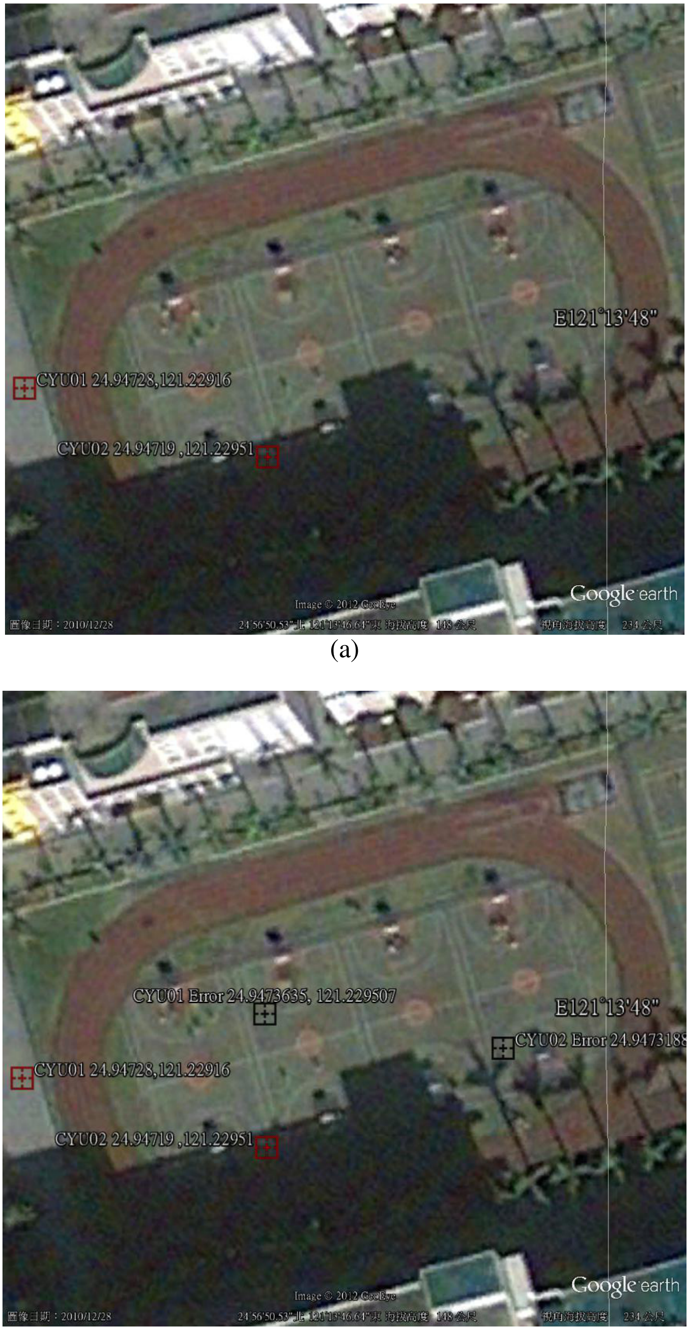
Parameter	Value
Population size	5000
Generations	200
Crossover rate	0.9
Mutation rate	0.1
Operators	+, −, ×, ⁄, sin, cos, natural log
Constants	0.1, 0.2, 0.3, …, 0.9, 1.0, π
Depth of an individual	8

Table 4. Average error between GPS receivers and reference positions

	Average error in latitude	Average error in longitude
GPS receiver at CYU01 (2456.8368, 12113.7496)	0.00501	0.02082
GPS receiver at CYU02 (2456.8314, 12113.7706)	0.00773	0.02033

Table 5. Training time (in second)

Experiments	Target value 2456.8368	Target value 12113.7496
Experiment 1	3825.03	8023.99
Experiemnt 2	1859.65	4252.42
Experiment 3	9644.12	3391.45
Experiment 4	2093.05	2885.01
Experiment 5	2919.44	2090.55
Average	4068.258	4128.684



(a)

(b)

Fig. 3. (a) Two fixed positions CYU01 and CYU02 provided by Ching Yun Unversity. (b) CYU01 and CYU02 and two position information collected by GPS receivers noted as CYU01 Error and CYU02 Error, respectively (image from Google).

Table 6. Average error of test set

Experiments	Target value 2456.8368	Target value 12113.7496
Experiment 1	0.000031	0.001009
Experiemnt 2	0.0	0.000133
Experiment 3	0.000077	0.0
Experiment 4	0.0	0.0
Experiment 5	0.0	0.0
Average	0.000022	0.000229

Experiment results of test set are shown in Table 6. It demonstrated that the proposed method achieved significant result in both latitude and longitude. Many experiments achieved perfect correction results. Experiment 4 and 5 achieved error-free test results. The average error of latitude reduces from 0.00501 to 0.000022 and that of longitude reduces from 0.02082 to 0.000229. The degree of correction is significant. The corrected position is very close to target position in less than 1 meter.

4 Conclusion

In this paper, we proposed a new GPS position correction technique based on genetic programming and differential GPS. Experiments have shown that even two GPS receivers have high error and noise, the proposed technique is capable of finding correction function to help one of the receivers to have accurate position. The proposed technique could be easily implemented on mobile devices because it does not need to modify or install any hardware component. A trained correction function can output latitude or longitude efficiently. Our future work will focus on training correction function with time limitation constraints. The training phase stops when given time limitation is reached. Such scenario would be closer to real world.

Acknowledgment. This work was supported by the National Science Council (grant number: NSC 100-2221-E-231 -021). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors only and do not necessarily reflect the viewpoints of the National Science Council.

References

1. Frenzel, L.E.: GPS take a global position in the portable market. Electronic Design 55(10), 47–56 (2007)
2. EI-Rabbany, A.: Introduct5ion to GPS: the global positioning system. Artech House, Boston (2002)
3. Lombardi, M.A., Nelson, L.M., Novick, A.N., Zhang, V.S.: Time and frequency measurements using the global positioning system. Cal Lab: International Journal of Metrology, 26–33 (2001)
4. Yeh, T.K., Wang, C.S., Lee, C.W., Liou, Y.A.: Construction and uncertainty evaluation of a calibration system for GPS receivers. Metrologia 43(5), 451–460 (2006)

5. Zhang, J., Zhang, K., Grenfell, R., Deakin, R.: GPS satellite velocity and acceleration determination using the broadcast ephemeris. Journal of Navigation 59, 293–306 (2006)
6. Klobuchar, J.A.: Ionospheric effects on GPS. In: Global Positioning System: Theory and Applications, vol. II, pp. 485–516. American Institute of Aeronautics and Astronautics, Inc., Washington, DC (1996)
7. Arnold, L.L., Zandbergen, P.A.: Positional accuracy of the wide area augmentation system in consumer-grade GPS units. Computers and Geosciences 37, 883–892 (2011)
8. Berber, M., Ustun, A., Yetkin, M.: Comparison of accuracy of GPS techniques. Measurement (in press, 2012), `http://dx.doi.org/10.1016/j.measurement.2012.04.010`
9. Ghilani, C.D., Wolf, P.R.: Elementary surveying – an introduction to geomatics, 12th edn. Peearson Prentice Hall, New Jersey (2007)
10. Van Sickle, J.: GPS for land surveyors, 3rd edn. CRC Press, New York (2008)
11. Chivers, M.: Differential GPS explained. ArcUser, 40–41 (January-March 2003)
12. Bolstad, P., Jenks, A., Berkin, J., Horne, K., Reading, W.H.: A comparison of autonomous, WAAS, real-time, and post-processed global positioning systems (GPS) accuracies in northern forests. Northern Journal of Applied Forestry 22(1), 5–11 (2005)
13. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
14. Lin, J.Y., Ke, H.R., Chien, B.C., Yang, W.P.: Designing a Classifier by a Layered Multi-Population Genetic Programming Approach. Pattern Recognition 40(8), 2211–2225 (2007)
15. National Marine Electronics Association (NMEA), `http://www.nmea.org`
16. HOLUX Technology Inc., GPSport 245, `http://www.holux.com/JCore/en/products/products_content.jsp?pno=349`

Maximum Likelihood DOA Estimation with Sensor Position Perturbation Using Particle Swarm Optimization

Jhih-Chung Chang[1,*] and Chia-Yi Chen[2]

[1] Department of Information Technology, Ling Tung University, Taichung, Taiwan, ROC
changjc@mail.ltu.edu.tw
[2] Institute of Applied Information Technology, Ling Tung University, Taichung, Taiwan, ROC
goccy0430@gmail.com

Abstract. In this paper, we consider the problem of direction-of-arrival (DOA) estimation for code-division multiple access (CDMA) signals in the presence of sensor position errors. For array calibration with sensor position errors, the estimation of the source directions and unknown array parameters is essential. The cost function is an extension of the maximum likelihood (ML) criteria that was originally developed for DOA estimation with a perfectly calibrated array. It is shown that the ML function is a complicated nonlinear multimodal function over a high-dimensional problem space. A modified particle swarm optimization (PSO) is proposed to compute the ML functions and find the global minimum cost function for array calibration. This proposed method has no requirement for calibration sources while the sensor position errors as well as the DOA of the incident signals can be estimated at the same time. Simulation results show that the proposed estimator has better performance over other popular methods.

Keywords: PSO, DOA, CDMA, ML.

1 Introduction

Adaptive array techniques have been developed for enhancing the performance of code-division multiple access (CDMA) systems. In the CDMA system, each user employs a unique pseudo-noise (PN) codeword to identify their data stream. Each user's transmission interferes with all the other users and causes the multiple access interference (MAI). The use of antenna arrays at the base station can provide accurate localization of user terminals [1]. In all these applications, estimation of direction of arrival (DOA) of the desired signals is required. With the advantage of code-matched filter inherent in the CDMA system, it has been proved that the multiple signal classification (MUSIC) estimator [2] can obtain an unbiased DOA estimation with low mean-square-error (MSE) [3]. It also contributes to solve the limitation that the number of array elements must be more than the number of impinging sources. But for the conventional

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 187–197.
DOI: 10.1007/978-3-642-35452-6_21

DOA estimation, the common presumption is that the array manifold is known. Nevertheless, in array signal processing, the sensors may be imperfect and the calibration measurement themselves are subjected to gain and phase errors. Due to changes in the surrounding environment, the response of the array and antenna position may be different from when it was calibrated before. Thus, the assumption of a known array response is seldom satisfied in practice, which makes the DOA estimation performance of these traditional methods degrade obviously [4, 5]. For array calibration with sensor position errors, the estimation of the source directions and unknown array parameters is essential. Many researchers have proposed various algorithms to tackle the problem. In [6], genetic approach (GA) is used to estimate the sensor positions as well as the DOAs, but it is also known as a slow optimization tool. On the other hand, a method for array shape calibration is proposed using sources in known locations in [7], however, it's difficult to get the sources in exactly known locations in practical applications. Subspace-based method was also evaluated for array shape calibration in [8]. However, the cost function with sensor position errors is an extension of the maximum likelihood (ML) criteria that was originally developed for DOA estimation with a perfectly calibrated array. The difficulty in optimizing the likelihood function prevents the ML technique from popular use.

Particle swarm optimization (PSO) is a stochastic optimization technique that simulates the social behavior of bird flocking to desired places [9]. Like GA, PSO starts by initializing a population of random solutions and searches for optima by updating generations. But, PSO does not use any evolution operators. In contrast to general heuristic methods, PSO is computationally efficient and has great capability of escaping local optima [10, 11]. In PSO, each agent is named particle and particles of the population are named swarm. Each particle is used to represent a potential solution. PSO has been applied to adaptive array signal processing like array calibration [12] and DOA estimation [13, 14]. For DOA estimation, direct maximization of the ML function over the large parameter space seems unrealistic in practice due to the huge computational burden. In general, it is difficult to completely separate the DOA angles from other unknown parameters. Therefore, to obtain the exact ML (EML) solutions, the DOAs must be jointly estimated along with other noise or signal parameters by optimizing a complicated nonlinear function. Although the computation complexity can be reduced via derivation of suboptimal approximate ML (AML) function using large sample assumption, nevertheless the AML estimator still require multi-dimensional search and the accuracy is lost to some extent. A PSO algorithm based on the ML methodology is also derived in [14], that was originally developed for angle estimation with some of the sensors are perfectly calibrated, while others are uncalibrated. Recently a modified PSO algorithm has also been successfully applied in MUSIC estimator with local scattered problem [15], that it is giving better performance. Here, the PSO with linear decreasing inertia weight is termed as hard-constraint PSO (HPSO). Unfortunately, the HPSO is easy to trap in local minima. And, it also suffers from overhead on computing the particle velocity limitation.

In this paper, we apply the PSO algorithm to ML criteria function for accurate DOA estimation of CDMA signals with sensor position errors. This paper presents an adaptive multiple inertia weight is proposed to rationally balance the global exploration and local exploitation abilities of the particle. The resulting estimator is called adaptive multiple inertia weight PSO (APSO-ML). A *u*-law based transform function is employed to dynamically calculate the values of inertia weight. At each iteration, during the run, every particle can choose appropriate inertia weight along every dimension of search space according to its own situation. By this fine strategy of dynamically adjusting inertia weight, the promising performance of APSO-ML estimator could be achieved.

2 Signal Model and Problem Formulation

2.1 Signal Model

Consider a DOA estimation scenario in a CDMA system with P users. The data bit $b_q(t) \in \{-1, 1\}$ of the qth user is spread by a pseudo-noise (PN) signature waveform $c_q(t)$. The signature waveform of the qth mobile is composed of a spreading sequence of L chips and $p_{T_c}(t)$ is the spreading pulse of duration T_c. Thus, the transmitted signal of the qth user during a bit interval T_b can be represented by $s_q(t) = \varepsilon_q b_q(t) c_q(t)$, where ε_q is the signal amplitude of the qth user. Let the processing gain of the spreading code be $L = T_b/T_c$. To simplify analysis, the PN code during a bit interval at a sample rate $1/T_c$ can be expressed as an L×1 vector of discrete sequence $\mathbf{c}_q = [c_{q1}, c_{q2}, \cdots, c_{qL}]^T$ for $q = 1, 2, \ldots, P$ where the superscript T is the matrix transpose operator. Employing the code-matched filter and DOA estimation algorithm, we propose a receiving base station of CDMA communication system with a uniform linear array (ULA). The antenna array consists of M identical omidirectional elements and receives signals from P users that arrive at the array from different angles θ_1, θ_2, …, θ_P with respect to the broadside of array in a cell. Assume that the array element space is d. The direction vector associated with the qth user is given by $\mathbf{a}(\theta_q) = [a_1(\theta_q), a_2(\theta_q), \cdots, a_M(\theta_q)]^T$, where $a_m(\theta_q) = \exp[-j2\pi d(m-1)\sin(\theta_q)/\beta]$ denotes the response of the mth sensor array to a signal with unit amplitude arriving from the direction angle θ_q, where β is the wavelength of the signals. Thus, the received baseband signal across the array can be written in vector form

$$\mathbf{x}(t) = \sum_{q=1}^{P} s_q(t)\mathbf{a}(\theta_q) + \mathbf{n}(t) \tag{1}$$

where $s_q(t) = \sum_{i=-\infty}^{\infty} \varepsilon_q b_q(i) c_q(t - iT_b - \tau_q)$, τ_q is the time delay, and the observation noise vector $\mathbf{n}(t)$ represents the spatially and temporally zero mean white Gaussian noise vector.

In order to pick out the pth user's signal, a code-matched filter containing the specified PN code vector $\mathbf{c}_p$ is applied to $\mathbf{x}(t)$. For simplicity, the bit sampling index is dropped to make the equations more readable in the following description. Therefore, the pth user's despread and sampled array vector signal $\mathbf{y}_p$ can be represented as

$$\mathbf{y}_p = \mathbf{x}\mathbf{c}_p = Ls_p\mathbf{a}(\theta_p) + \sum_{q=1,\, q\neq p}^{P} s_q\mathbf{a}(\theta_q) + \mathbf{n}_p \tag{2}$$

where $s_p = \varepsilon_p b_p$ and $s_q = \varepsilon_q b_q \kappa_{qp}$. Hence, $\pi_p = E\{|s_p|^2\}$ and $\pi_q = E\{|s_q|^2\}$ denote the average power of the desired pth user and each interferer user, respectively. The notation $E\{\bullet\}$ is used to denote the expectation operator. κ_{qp} is the inner product of two different PN code vectors $\mathbf{c}_q^T \mathbf{c}_p$, and the M×1 vector $\mathbf{n}_p$ with zero mean and variance π_n is the code-matched filter output of the noise vector **n**. However, the signals of the undesired users through the code-matched filter, i.e., MAI, can be viewed as a noise vector $\mathbf{n}_{MAI_p}$ with zero mean and variance π_{MAI_p} and $\mathbf{n}_{MAI_p} = \sum_{q=1,\, q \neq p}^{P} s_q \mathbf{a}(\theta_q)$. The correlation matrix of (2) in the observation interval is given by

$$\mathbf{R}_p = E\{\mathbf{y}_p \mathbf{y}_p^H\} = L^2 \pi_p \mathbf{a}(\theta_p) \mathbf{a}^H(\theta_p) + \pi_{np} \mathbf{I}_M \tag{3}$$

where the subscript H denotes conjugate transpose, $\pi_{np} = \pi_{MAI_p} + \pi_n$, and $\mathbf{I}_M$ is an $M \times M$ identity matrix.

2.2 Random Perturbations in Sensor Positions

In practical operating circumstances, there may exist the imprecise knowledge of sensor positions that can seriously degrade the performance of a direction finding system. For this case, assume an arbitrary array of identical unit-gain omnidirectional sensors in the x-y plane with randomly perturbed sensor positions. To show the circumstance of random perturbations in sensor positions, the position perturbations Δg_m of the mth sensor can be expressed as

$$\Delta g_m(\theta_q) = [\Delta x_m,\ \Delta y_m][\sin(\theta_q),\ \cos(\theta_q)]^T \tag{4}$$

where $(\Delta x_m,\ \Delta y_m)$ are the nominal coordinates position errors with the same variance σ_q^2 [5]. The position perturbation is assumed to remain constant during the adaptation period. The mth sensor response can be written as

$$\tilde{a}_m(\theta_q) = \exp(j \frac{2\pi}{\lambda} \Delta g_m(\theta_q)) a_m(\theta_q) \tag{5}$$

Then, from the above description of position perturbation, the received array data vector is given by

$$\mathbf{x} = \sum_{q=1}^{P} \mathbf{G}_q \mathbf{a}(\theta_q) s_q \ + \mathbf{n} \tag{6}$$

where $\mathbf{G}_q = diag\{\mathrm{e}^{(j\frac{2\pi}{\lambda}\Delta g_1(\theta_q))}, \mathrm{e}^{(j\frac{2\pi}{\lambda}\Delta g_2(\theta_q))}, \cdots, \mathrm{e}^{(j\frac{2\pi}{\lambda}\Delta g_M(\theta_q))}\}$.

Therefore, the pth user's despread and array vector signal $\hat{\mathbf{y}}_p$ can be represented as

$$\hat{\mathbf{y}}_p = \mathbf{x}\mathbf{c}_p = L s_p \mathbf{G}_p \mathbf{a}(\theta_p) + \sum_{q=1,\, q \neq p}^{P} s_q \mathbf{G}_q \mathbf{a}(\theta_q) + \mathbf{n}_p \tag{7}$$

The correlation matrix of (7) in the observation interval is given by

$$\hat{\mathbf{R}}_p = E\{\hat{\mathbf{y}}_p \hat{\mathbf{y}}_p^H\} = L^2 \pi_p (\mathbf{G}_p \mathbf{a}(\theta_p))(\mathbf{G}_p \mathbf{a}(\theta_p))^H + \pi_{np} \mathbf{I}_M \tag{8}$$

Comparing (1) and (8), we note that $\mathbf{G}_q$ represents the effect of sensor position errors, which depends on signal bearing and position perturbations. However, the correlation matrix formed by (8) loses the Toeplitz structure and its perturbation noise subspace also has components outside the true noise subspace. As a result, the performance of DOA estimation will be degraded in the presence of sensor position errors.

2.3 ML Estimators

We derive the ML estimate of DOAs and sensor positions is computed by [16] to maximizing the likelihood function with respect to all unknown real parameters. The ML criterion function can be given as

$$\begin{aligned} S_{ML\_p}(\theta) &= \arg\{\max_{\theta}\ L(\theta)\} \\ &= \arg\{\max_{\theta}\ tr\{\mathbf{P}_{\mathbf{A}(\theta)}\hat{\mathbf{R}}_p\},\ p = 1,\ 2,\ \cdots,\ P \end{aligned} \tag{9}$$

where $tr[\cdot]$ is the trace of the bracketed matrix, $\mathbf{P}_{\mathbf{A}(\theta)} = \mathbf{A}(\theta)(\mathbf{A}^H(\theta)\mathbf{A}(\theta))^{-1}\mathbf{A}^H(\theta)$ is the projection operator onto the space spanned by the columns of the matrix $\mathbf{A}(\theta) = [\mathbf{G}_1\mathbf{a}(\theta_1), \mathbf{G}_2\mathbf{a}(\theta_2), \cdots, \mathbf{G}_P\mathbf{a}(\theta_P)]$ which is the $M \times P$ source direction matrix. Apparently, the maximization of the log-likelihood is computationally expensive. If nominal sensor positions differ from the actual sensor positions, the estimates fail. In order to solve such a problem, we introduce the PSO algorithm as follows.

3 PSO Based Searching Algorithms for DOA Estimation

3.1 HPSO-ML Estimator

We use the PSO to achieve accurate angle problems of high computation cost in the presence of sensor position errors. The difficulty can be overcome by searching both the sensor positions and the DOA at the same time. For time-space array calibration, each particle can be treated as a point in the J-dimensional problem space and a swarm consisting of D random particles. The position of the ith particle is represented as $\boldsymbol{\theta}_i = [\Delta x_{i1}, \Delta x_{i2}, \cdots, \Delta x_{iM}, \Delta y_{i1}, \Delta y_{i2}, \cdots, \Delta y_{iM}, \theta_{i1}]$, where Δx_{im} and Δy_{im} are the unknown coordinates position errors of sensor m and $m = 1, 2, \cdots, M$, then the estimate of sensor position errors and DOAs with respect to $J = 2M+1$ unknown real parameters can be converted to the problem of optimization. The ith particle's velocity is updated according to $\mathbf{v}_i = [v_{i1},\ v_{i2},\ \cdots,\ v_{iJ}]$. The best previous position of the ith particle is recorded as $\mathbf{p}_i = [p_{i1},\ p_{i2},\ \cdots,\ p_{iJ}]$. And, the best particle among all the particles in the population is represented by $\mathbf{p}_g = [p_{g1},\ p_{g2},\ \cdots,\ p_{gJ}]$ and called global best location. In every iteration, the $\mathbf{p}_i$ and $\mathbf{p}_g$ are determined through evaluating the fitness values for the current population of particles. Two factors characterize a particle status on the search space: its position and velocity. The J-dimensional position for the ith particle in the kth iteration can be denoted as $\boldsymbol{\theta}_i(k)$. Similarly, the velocity is also a J-dimensional vector, for the ith particle in the kth iteration can be described as $\mathbf{v}_i(k)$. The velocity and the position of the ith particle at the $(k+1)$th iteration for $i = 1,\ 2,\ \cdots,\ D$ and $k = 1,\ 2,\ \cdots,\ k_{\max}$ are updated according to the following equations:

$$\begin{aligned}\mathbf{v}_i(k+1) = w(k)\mathbf{v}_i(k) + \phi_1\mathbf{r}_1(k) \otimes (\mathbf{p}_i(k) - \boldsymbol{\theta}_i(k)) \\ + \phi_2\mathbf{r}_2(k) \otimes (\mathbf{p}_g(k) - \boldsymbol{\theta}_i(k))\end{aligned} \tag{10}$$

$$\boldsymbol{\theta}_i(k+1) = \boldsymbol{\theta}_i(k) + \mathbf{v}_i(k+1) \tag{11}$$

where $\otimes$ denotes element-wise product. And, the positive acceleration constants ϕ_1 and ϕ_2 represent the weighting of the stochastic acceleration terms. $\mathbf{r}_1(k)$ and $\mathbf{r}_2(k)$ are J-dimensional vectors consisting of independent random numbers with uniform distributed between 0 and 1. The inertia weight $w(k)$ based on the linear decreasing strategy is considered critically for the convergence behavior of PSO [13], which is selected to decrease during the optimization process. Thus, PSO tends to have more global search ability at the beginning of the run while having more local search ability near the end of the optimization. Given a maximum value $w_{\max}$ and a minimum value $w_{\min}$, $w(k)$ is updated as follows:

$$w(k+1) = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} k \tag{12}$$

where $k_{\max}$ is the maximum number of iterations, $w_{\max}$ =0.9 and $w_{\min}$ =0.4 [13]. Because the PSO particle represents a series of priorities that search range from -90° to 90°. In this study, the search range can be normalized into -1 to 1, that the particle positions must be limited to $[\theta_{\min}, \theta_{\max}] = [-1, 1]$. Each parameter of the initialized or updated position that is beyond $[\theta_{\min}, \theta_{\max}]$ can be adjusted in the following form:

$$\begin{aligned}&\text{if } \theta_{ij}(k) > \theta_{\max} \text{ then } \theta_{ij}(k) = \theta_{\max} \\ &\text{elseif } \theta_{ij}(k) < \theta_{\min} \text{ then } \theta_{ij}(k) = \theta_{\min}\end{aligned} \tag{13}$$

where $j = 1, 2, \cdots, 2M+1$. The new particle position is calculated using (11). The velocity limit is applied to $\mathbf{v}_i$. A particle position vector is converted to a candidate solution vector in the problem space through a suitable mapping. For the DOA ML estimation problem which is to find the particle position or estimate angle $\theta_{ip}(k)$ of the pth user and maximize the following fitness function:

$$\text{fitness}_{ip}(k) = tr\{\mathbf{P}_{\mathbf{A}(\theta_{ip})}\hat{\mathbf{R}}_p\}\,,\ p = 1, 2, \cdots, P\,. \tag{14}$$

The final global best position $\mathbf{p}_g(k_{\max})$ is taken as the ML estimates of users. Unfortunately, the HPSO-ML is not so facile to implement for its overhead on computing the particle velocity limitation and particle position clipping.

3.2 The Proposed Adaptive Multiple Inertia Weight

The inertia weight is critical for the performance of PSO, which balances global exploration and local exploitation abilities of the swarm. A large inertia weight facilitates searching new area and global exploration. Conversely, a small inertia weight tends to facilitate local exploitation in the current search area. Suitable selection of the inertia weight requires less iteration on average to find the optimum. During the search every particle dynamically changes its position, so every particle locates in a complex environment and faces different situation. Therefore, each particle along every dimension may have different trade off between global and local search abilities. According to [17], it has been shown that the performance of the HPSO algorithm has the ability to quickly converge, and may lack global search ability at the end of a run. But to some extent, this can be overcome by employing a self-adapting strategy for adjusting the inertia weight. In this subsection, inertia weight is dynamically adapted for every particle along every dimension. The particle could decide to whether to increase or decrease the values of inertia weight by means of the transfer function. The fine strategy of dynamically adjusting inertia weight could lead to improvement in performance of PSO.

For velocity updating, the last two parts of (10) can be defined as $a_{ip}(k)=\phi_1\cdot\mathbf{r}_1(k)\cdot(p_{ip}(k)-\theta_{ip}(k))$ and $a_{gp}(k)=\phi_2\cdot\mathbf{r}_2(k)\cdot(p_{gp}(k)-\theta_{ip}(k))$. Here, we consider that the particle is moving with velocity of $v_{ip}(k)$ and acceleration of $a_{gp}(k)$. Suppose that the mass of the *i*th particle in the *p*th dimension is normalized to 1kg. According to the principle of mechanics, $f_{ip}(k)=a_{gp}(k)$, where $f_{ip}(k)$ is an outside force, which is put on the particle comes from the "pulling" of $p_{gp}(k)$. For ULA, the DOA is a one-dimensional searching problem. In order to make the particle fly towards optimal region quickly, $v_{ip}(k)$ should turn to the direction of $f_{ip}(k)$ as soon as possible. The $z_{ip}(k)=p_{gp}(k)-\theta_{ip}(k)$ is defined as the error between $p_{gp}(k)$ and $\theta_{ip}(k)$. Let $m_{ip}(k)=\ |z_{ip}(k)|/\theta_{\max}$ which is a number between 0 and 1. The velocity update problem of the *i*th particle on the *p*th dimension can be divided into two classes:

First, $v_{ip}(k)$ and $z_{ip}(k)$ are in the same direction, $z_{ip}(k)/v_{ip}(k)\geq 0$. If $m_{ip}(k)$ is relatively large, it means the particle is in the right direction, but the velocity is too small. Thus, the particle needs to speed up, and the inertia weight $w_{ip}(k)$ needs to be set larger. If $m_{ip}(k)$ is relatively small, it means the particle has come to the location, that is near the optimal region. So the velocity of this particle should slow down and the neighborhood of the state should be searched carefully, and $w_{ip}(k)$ should be set smaller. Next, consider $v_{ip}(k)$ and $z_{ip}(k)$ on different direction, $z_{ip}(k)/v_{ip}(k)<0$. If $m_{ip}(k)$ is relatively large, it means that the particle's state is far from the optimal region. So the particle needs to change its velocity as soon as possible, and the inertia weight $w_{ip}(k)$ needs to be set smaller. If $m_{ip}(k)$ is relatively small, it means that it's not urgent for the particle to change its direction on this dimension, and $w_{ip}(k)$ could be set a large value.

Based on the aforementioned analysis, an adaptive multiple inertia weight strategy is proposed. The individual normalized search ability of the *i*th particle along the *p*th dimension is defined as $m_{ip}(k)$. It is noted that the value of $w_{ip}(k)$ is a function of $m_{ip}(k)$. To get a balance of global search and local search ability, $w_{ip}(k)$ cannot be too large or too small, thus $w_{ip}(k)$ is limited in the range of $[w_{\min},w_{\max}]$. Therefore, we use *u*-law algorithm to achieve our strategy, normalization of the $m_{ip}(k)$ to $\overline{w}_{ip}(k)$. *u*-law algorithm is a compression law, which is widely used in digital telephone communication [18]. Increasing the value of u, the dynamic range capability of u-law can be improved and defined by

$$\bar{w}_{ip}(k)=\begin{cases}\dfrac{um_{ip}(k)}{1+\log u}, & 0\leq\ m_{ip}(k)\ \leq\dfrac{1}{u}\\ \dfrac{1+\log(um_{ip}(k))}{1+\log u}, & \dfrac{1}{u}\leq\ m_{ip}(k)\ \leq 1\end{cases} \tag{15}$$

where $m_{ip}(k)$ and $\bar{w}_{ip}(k)$ are normalized input and output values respectively, and the positive constant u is the compression parameter. For case 1, the value for small $m_{ip}(k)$ increases at the expense of the value for large $m_{ip}(k)$. To accommodate these conflicting requirements, a compromise parameter u=100 is made for u-law. The requirement of case 2 is opposite. Finally, based on our strategy and (15), we can define adaptive multiple inertia weight $w_{ip}(k)$ as:

$$w_{ip}(k)=\begin{cases}w_{\min}+\bar{w}_{ip}(k), & \dfrac{z_{ip}(k)}{v_{ip}(k)}\geq 0\\ w_{\min}+(1-\bar{w}_{ip}(k)), & \dfrac{z_{ip}(k)}{v_{ip}(k)}<0\end{cases} \tag{16}$$

In (16), $w_{\min}$ be added to avoid particles stop moving. Note that for every particle in population, $w_{ip}(k)$ is unique and can be computed individually. Therefore, the single inertia weight $w_i(k)$ can be replaced by an adaptive multiple inertia weight $w_{ip}(k)$. The proposed APSO-ML seems to be robust to control parameters due to the intrinsic advantages of the algorithm.

4 Computer Simulations

Several simulations examples are conducted for our method. Comparison results with other estimators, including the MUSIC algorithm, HPSO-ML. We use a 6-element ULA with half wavelength for simulation. Consider an asynchronous CDMA system with PN codes of length 31 and BPSK modulation. Eight users are impinging on the array with random impinging angles of uniform distribution in the interval $[-90^\circ,90^\circ]$. All the users are set to the equal power with signal-to-noise ratio (SNR) = 20 dB. The additive noise is assumed to be white Gaussian distribution with zero-mean and unit variance. For MUSIC, the search resolution is set to 0.1°. As an index of evaluation, the root-mean-squared error (RMSE) is defined as $\mathrm{RMSE}=[(1/FP)\sum_{f=1}^{F}\sum_{p=1}^{P}(\hat{\theta}_{fp}-\theta_p)^2]^{0.5}$, where F indicates the number of independent simulation runs, $\hat{\theta}_{fp}$ is the pth estimate DOA achieved in the fth run, and θ_p is the true DOA of the pth user. PSO parameters are set as $\phi_1=\phi_2=2$ and $D=k_{\max}=20$. All PSO algorithms start with random initializations and are terminated if the maximum iteration $k_{\max}$ is reached. Every simulation result is presented after 200 data bits were processed and it is averaged by 500 independent Monte Carlo runs.

Fig. 1 shows the RMSE of DOA estimation versus the variance of the position perturbations. All methods suffer the degradation of performance as the position error increases. It is seen that the proposed method results in better immunity against the position perturbations. Fig. 2 presents the RMSE of DOA estimation versus input SNR of desired user. For the low SNR case, all of methods may produce highly biased estimates. Clearly, with the compatible searching resolution, the APSO-ML can save the required number of searching grids and improve the RMSE performance, as compared with the other estimators. It has been shown that the performance of the PSO algorithm with linearly decreasing inertia weight has the ability to quickly converge, the PSO may lack global search ability at the end of a run and may fail to find the required optima in cases when the problem to be solved is too complicated and complex. But to some extent, this can be overcome by employing the proposed adaptive multiple strategy for adjusting the inertia weight.

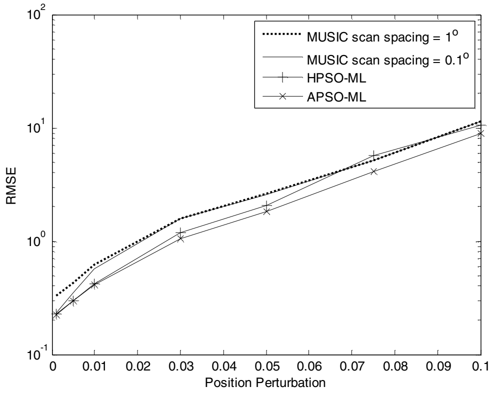


Fig. 1. DOA RMSE versus the position perturbation

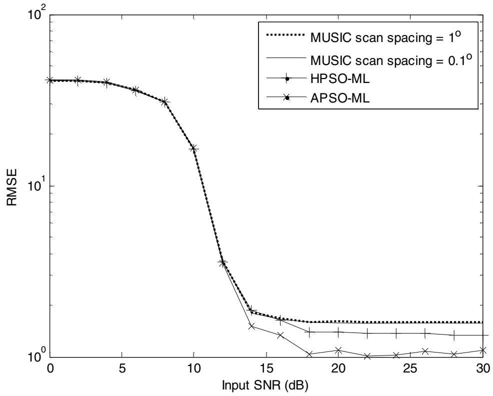


Fig. 2. DOA RMSE versus the input SNR of desired user

5 Conclusions

In this paper, a novelty PSO based searching DOA estimation method, named as APSO, is proposed to estimate the source DOA using ML function for CDMA signals. With the code-matched filter, the MAIs after code-decorrelation appear as noises for CDMA signals. However, by carefully selected evolution operators and fine-tuned parameters, the proposed techniques improve the estimation accuracy. With newly introduced multiple inertia weight has been incorporated in PSO. Computer simulations show that the proposed APSO-ML estimator attains better performance than other estimators like MUSIC and HPSO-ML.

Acknowledgment. This research was supported by the National Science Council under grant number NSC 101-2221-E-275-007.

References

1. Caffery, J.J., Stuber, G.L.: Overview of radiolocation in CDMA cellular networks. IEEE Communications Magazine 36(4), 38–45 (1998)
2. Schmidt, R.O.: Multiple emitter location and signal parameter estimation. IEEE Trans. Antennas and Propagation 34(3), 276–280 (1986)
3. Chiang, C.T., Chang, A.C.: DOA estimation in the asynchronous DS-CDMA system. IEEE Trans. Antennas and Propagation 51(1), 40–47 (2003)
4. Swindlehurst, A.L., Kailath, T.: A performance analysis of subspace-based methods in the presence of model errors-Part I: The MUSIC algorithm. IEEE Trans. Signal Process. 40(7), 1758–1774 (1992)
5. Ng, B.C., Samson See, C.M.: Sensor-array calibration using a maximum-likelihood approach. IEEE Trans. Antennas Propagation 44(6), 827–835 (1996)
6. Hong, J.S.: Genetic approach to bearing estimation with sensor location uncertainties. Electronic Letters 29, 2013–2014 (1993)
7. Zhang, M., Zhu, Z.D.: Array shape calibration using sources in known locations. In: Proc. of the IEEE National Aerospace and Electronics Conference, NAECO, pp. 67–69 (1993)
8. Weiss, A.J., Friedlander, B.: Array shape calibration using eigenstructure methods. Signal Processing 22(3), 251–258 (1991)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, Path, Australia, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
10. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, pp. 69–73 (1998)
11. Shi, Y., Eberhart, R.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proc. of the Congress on Evolutionary Computation, Piscataway, vol. 1, pp. 84–88 (2000)
12. Liao, B., Liao, G., Wen, J.: Array calibration with sensor position errors using particle swarm optimization algorithm. In: 2009 IET International Radar Conference, pp. 1–3 (2009)
13. Li, M., Lu, Y.: Maximum likelihood DOA estimation in unknown colored noise fields. IEEE Trans. Aerospace and Electronic Systems 44(3), 1079–1090 (2008)

14. Li, M., Lu, Y.: Source bearing and steering-vector estimation using partially calibrated arrays. IEEE Trans. Aerospace and Electronic Systems 45(4), 1361–1372 (2009)
15. Chang, J.C.: DOA estimation for local scattered CDMA signals by particle swarm optimization. Sensors 12, 3228–3242 (2012)
16. Ziskind, I., Wax, M.: Maximum likelihood localization of multiple sources by alternating projection. IEEE Trans. Acoustics, Speech and Signal Processing 36(10), 1553–1560 (1988)
17. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proc. of the Congress on Evolutionary Computation, vol. 3, pp. 1945–1950 (1999)
18. Haykin, S.: Communication Systems, 4th edn. John Wiley & Sons, New York (2001)

Optimistic Heuristics for MineSweeper

Olivier Buffet[3], Chang-Shing Lee[2], Woan-Tyng Lin[2], and Olivier Teytuad[1,2,4]

[1] TAO-INRIA, LRI, CNRS UMR 8623,
Université Paris-Sud, Orsay, France
[2] National University of Tainan, Taiwan
[3] MAIA team, LORIA, INRIA/Universit de Lorraine, France
[4] Montefiore Institute, Université de Liège, Belgium

Abstract. We present a combination of Upper Confidence Tree (UCT) and domain specific solvers, aimed at improving the behavior of UCT for long term aspects of a problem. Results improve the state of the art, combining top performance on small boards (where UCT is the state of the art) and on big boards (where variants of CSP rule).

Keywords: Constraint Satisfaction Problems, Upper Confidence Tree, Minesweeper.

1 Introduction

This sequential decision making research is based on the following requirements on our favorite application:

- anytime algorithm (behavior for unknown and possibly small time settings): we want our algorithm to be able to run in very limited time and provide an approximate solution. For example, Direct Policy Search (DPS) and Upper Confidence Tree (UCT) satisfy this requirement.
- asymptotic optimality (behavior for long time settings): many algorithms are fast, but have strong assumptions on the model, so that the real problem is modified in order to match the model requirement; as a consequence, the solution, even if it is optimal on the model, is not optimal on the real problem. For example, UCT satisfies this requirement (in discrete cases, and with some modifications also in continuous domains); many reinforcement learning tools also have this property. DPS, on the other hand, is usually limited by the policy structure.
- compatibility with long term effects: many Markov Decision Processes (MDP) have long term effects; you cannot make optimal decisions unless you take into account long term consequences of your actions. This requirement is harder to formalize: Direct Policy Search has no problem with this, because it is based on complete simulations of the problem; also, Stochastic Dynamic Programming takes into account long term effects. On the other hand, it is not the case for UCT. Theoretically, UCT will asymptotically find optimal solutions if the horizon is finite; but in real life, only the few initial time steps make sense, the remaining part being left to the pure Monte-Carlo part.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 199–207.
DOI: 10.1007/978-3-642-35452-6_22

- compatibility with high-dimension: many works are dedicated to improving the scalability of MDP-solving algorithms; a great success for this is UCT, with its ability to focus on the relevant parts of the search space.

In this paper, we propose the following principle:

- Using a UCT algorithm for handling the short term combinatorial explosion of the tree of possible futures.
- Using, in lieu of the standard pure Monte-Carlo simulator of UCT, a solver known for good performance for long-term behavior.

Section 2 presents some tools useful in the paper. Section 3 presents our algorithm, Optimistic Heuristics. Section 4 presents experimental results. Section 5 concludes.

2 Tools

In this section, we define the tools on which our algorithm is based: Upper Confidence Trees (Section 2.1), Constraint Satisfaction Problems (Section 2.2).

2.1 Upper Confidence Trees

Upper Confidence Trees [9] proceeds as explained in Alg. 1 for making a decision in a state s in time t.

Algorithm 1. The UCT algorithm in short.

```
Inputs: a state $s$, a time $t$.
Output: a decision.
while $t$ is not elapsed do
  $s' = s$  // initialization of a simulation
  while $s'$ is not a final state  // this loop is a simulation do
    if $s'$ has more than 5 simulations then
      Let $d$ be the decision in $s'$ which maximizes $Q_{ucb}(s', d)$
    else
      Let $d$ be a decision in $s'$ chosen by a heuristic (a.k.a Monte-Carlo part)
    end if
    Let $s''$ be a (possibly stochastic) next state obtained after decision $d$ in $s'$.
    $s' \leftarrow s''$
  end while
  Update $Q_{ucb}$ values and $\hat{Q}$ values
end while
Return the decision in $s'$ with maximum $\hat{Q}(s', d)$.
```

This algorithm uses the two following formulas:

$\hat{Q}(s,d) =$ mean reward of past simulations including decision d in state s

$$Q_{ucb}(s,d) = \hat{Q}(s,d) + \sqrt{\frac{\log(2+n(s))}{1+n(s,d)}}$$

where $n(s,d)$ is the number of simulations with move d in state s and $n(s) = \sum_d n(s,d)$.

It is widely reported that choosing a heuristic with high performance does not necessarily lead to high performance of the UCT built on top of it [5]. However, in the one player case, introducing strong heuristics is seemingly quite efficient [3, 13]. This is the principle of [13], using a strong heuristic, namely CSP, for improving UCT performance on MineSweeper on small boards.

2.2 Constraint Satisfaction Problems + Heuristics

In some problems in which rigorous optimality is unreachable, just optimizing the instantaneous reward might be a good idea. For example, in the MineSweeper problem, choosing the move with lowest probability of immediate loss is efficient. Nonetheless, it is not sufficient for rigorous optimality; as shown by [11], when two locations have the same probability of mine, choosing the one which is as close as possible to the frontier between covered locations and uncovered locations might be a good idea. For examples, if six locations *a*, *b*, *c*, *d*, *e*, *f* are uncovered, with probability of mine respectively 50%, 50%, 20%, 20%, 20%, 20%, and distance to the frontier respectively 1, 1, 2, 2, 2, 1, then move *f* should be chosen (at least according to this heuristic), because moves *c*, *d*, *e*, and *f* have lowest probability of mines, and among these locations, *f* has smallest distance to frontier. Formally, the frontier is defined as the set of locations which are uncovered and next to at least one mine. This is not formally proved as optimal (and Fig. 2 shows that it is not optimal), but it is on average quite efficient (see results in [11]) as a heuristic for breaking ties between equivalent moves. This heuristic "tie breaking" provides significant improvements. However, as shown by [3], there are cases where the optimal move is difficult to find among moves with minimum probability of mine. Therefore, CSP variants do not have the asymptotic optimality property discussed in the introduction.

3 Optimistic Heuristics

We define Optimistic Heuristics (OH), our UCT-based algorithm, as follows:

- the main structure of the program is UCT;
- we fix the first move at a corner of the board;
- we expand only children with minimum probability of mine;
- but the Monte-Carlo part is replaced by HCSP (Heuristic CSP), a variant of CSP with improved performance thanks to the heuristic of preferring locations close to the frontier between covered and uncovered locations.

Algorithm 2. Our Optimistic Heuristics algorithm. The two parts in bold involve specialized methods, namely CSP and HCSP.

```
Inputs: a state $s$, a time $t$.
Output: a decision.
while $t$ is not elapsed do
  $s' = s$    // initialization of a simulation
  while $s'$ is not a final state    // this loop is a simulation do
    CSP: Let $E$ be the set of moves with minimum probability of mine.
    if $s'$ has more than 5 simulations then
      Let $d$ be the decision in $E$ which maximizes $Q_{ucb}(s',d)$
    else
      Let $d$ be a decision in $E$ chosen by HCSP.
    end if
    Let $s''$ be a (possibly stochastic) next state obtained after decision $d$ in $s'$.
    $s'' \leftarrow s'$
  end while
    Update $Q_{ucb}$ values and $\hat{Q}$ values
end while
Return the decision in $s'$ with maximum $\hat{Q}(s',d)$.
```

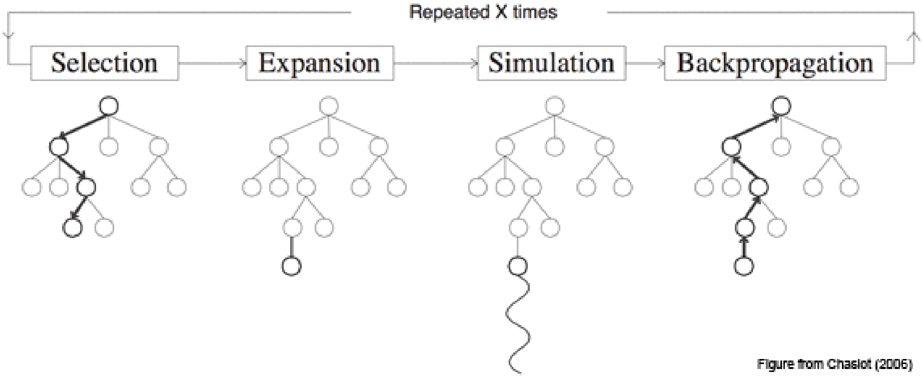


Fig. 1. A description of MCTS in one Figure, from [2]. Before making a decision, the algorithm performs many simulations. Each simulation is typically made of (i) a path in the current tree stored in memory (ii) a construction of one new node added to the tree (iii) a Monte-Carlo simulation from this new node.

The pseudo-code is given in Alg. 2. Describing UCT in details is beyond the scope of this paper; so we refer to http://www.mcts.ai/?q=mcts for a user-friendly introduction. Figure 1 provides a graphical overview.

4 Experiments

The well-known MineSweeper game is much more complex than expected at first sight [8, 1, 12], and a good tool for modeling in a clean way some real world problems [7]. The most classical approach for MineSweeper is based on Constraint Satisfaction Problems [14], as in Section 2.2: this provides a provably correct estimate of the belief state, and then classically after CSP one plays the covered location with least probability of being a mine. However, [3] has shown that this approach is suboptimal; there are situations, as in Fig. 2, where some moves with minimum probability of mines are nonetheless suboptimal moves. This fact was used in [11] for designing a version of CSP with better performance, by preferring the locations which are closest to the frontier in case of tie on the probability of mines; [11] got the best performance so far on MineSweeper on many board sizes, under time constraints, in particular big boards.

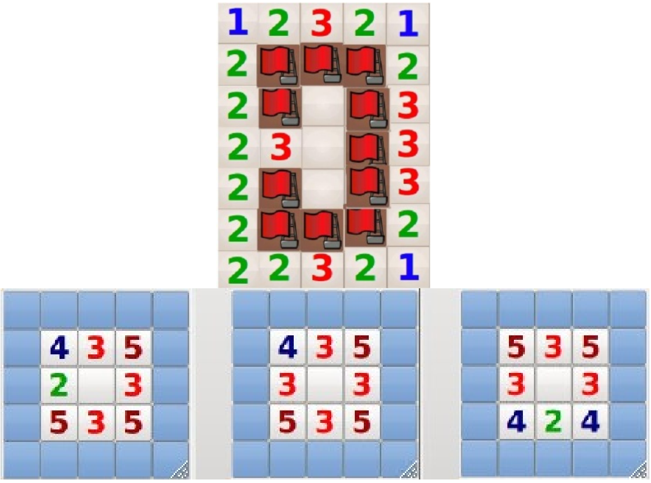


Fig. 2. Situations in which some moves with minimum probability of mine are suboptimal. We are not aware of situations where no move with minimum probability of mine is optimal; so in our program we expand only the parts of the tree corresponding to moves with lowest probability of mine (yet, we have no proof of optimality for this). Top: the top location and the bottom covered locations lead to a probability of winning $\frac{2}{3}$ whereas the middle location leads to a probability of winning $\frac{1}{3}$. Bottom: these three boards are the three possible cases when playing 5x5 GnoMine with 15 mines (up to rotation): they show that under GnoMine rule, playing in the center (for the first move) leads to a 100% winning rate. One can easily see that the success rate when playing any other move is < 100%. So, this case shows that even for the first move, choosing a move with minimal probability of mine is not enough for playing optimally; moreover, corners are not optimal for the GnoMine variant.

An alternate approach has been investigated in [3, 13]; using UCT, these approaches have asymptotic optimality, and so outperform CSP when given enough time. They got the best performance so far on small boards, outperforming CSP, but are too slow for big boards.

[13] and [11] outperformed the state of the art in two different settings; close to optimality on small boards for [13] (compliant with any variant of rule), and better than CSP on big boards for [11]; in this paper we combine both approaches, using UCT with the HCSP approach as a heuristic.

MineSweeper is parametrized by the number of mines, the number of rows, the number of columns. Importantly, there are different variants of MineSweeper:

- variants in which the mines can be anywhere on the board;
- variants in which the mines cannot be on the first move (as in most implementations);
- variants in which there are no mine in the neighborhood of the first move (e.g. the GnoMine implementation).

The optimal strategies are the same for the first two variants. It is widely believed that moves in the corners are optimal in the two first cases; but, as shown by the example in Fig. 2 (Bottom) this is not the case for the third family of variants. For the sake of sound comparisons with earlier results, we will present results in the second case, i.e. one cannot lose at the first move. Results are presented in Table 1.

5 Conclusions

We experimented a combination between UCT and CSP solvers. We got a reasonably fast solver, much faster than the one from [13], and benefiting from the heuristic from [11]. Results are at the state of the art on all board sizes. Further work consists in

- Pairing simulations: using the same sequence of possible children for all children of a given node.
- Using the additional information from HCSP for ordering legal decisions in given node; for example, we might:
  - consider moves in lexicographic orders on x-axis and y-axis;
  - consider moves in order of probability of mine, and in case of tie distance to the frontier (this is inspired by [11]);
  - consider random order (which avoids biases).
- Using progressive widening [2, 4, 10]; testing various coefficients, and maybe non-polynomial rules; or use rules depending on rewards obtained for the already sampled moves; see Fig. 3 for more on this.
- Simulate several times for evaluating a node, before starting new simulations from the root.
- Using rapid action value estimates [6] (either it brings an improvement, or it is an interesting counter-example; see Fig. 4 for a counter-example in the game of Go.).

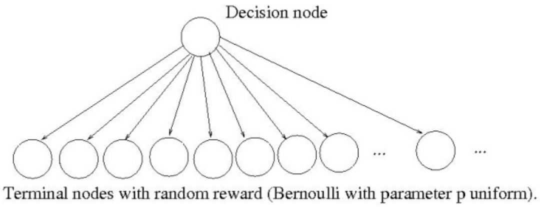

Decision node
...
...
Terminal nodes with random reward (Bernoulli with parameter p uniform).

Fig. 3. Example in which progressive widening is necessary; each decision leads to a reward which is a Bernoulli random variable with parameter p, uniformly drawn in [0, 1]. There are infinitely many possible decisions. Without progressive widening, all moves are tested just once, and the final arm is chosen randomly, whereas a progressive widening as in [15] is mathematically consistent.

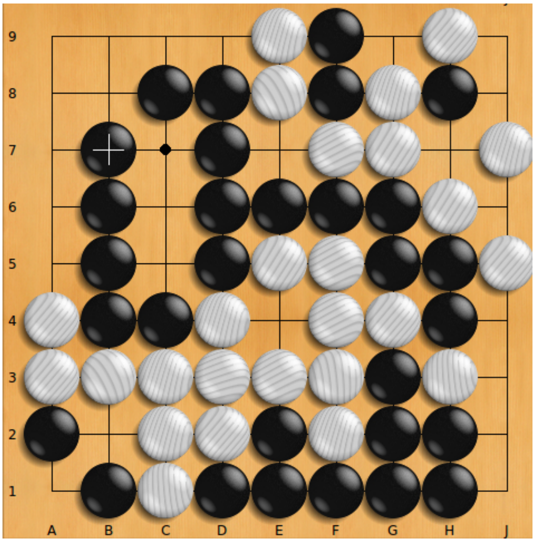


Fig. 4. Example provided by Martin Muller, in which RAVE values do not work well: WhiteB2 is the only good move (avoiding seki in bottom-left); but RAVE values are very bad for B2 because B2 only makes sense as a first move, and RAVE values propagate values from second time step, third time step, fourth time steps, . . . , to the current time step

More specifically for Minesweeper, we don't know if the assumption that the optimal move has minimum probability of mine is safe. We just break ties between various moves with minimum probability of mine, and we did not find any proof that doing otherwise (i.e. considering also moves with non-minimal probability of mine) can bring an improvement; but we have no proof. Also we have no proof that always playing in the corner for the first move is a good idea; it is known as a good heuristic, but maybe in some cases there are better moves (with the rule "first move is a 0", it is mathematically proved that the center is a good move in some cases; see [3]).

Table 1. Results of various implementations on the MineSweeper games. CSP-PGMS is the PGMS implementation of CSP (http://www.ccs.neu.edu/home/ramsdell/pgms/). HCSP is the implementation of CSP with heuristic breaking ties by playing as close as possible to the frontier between covered locations and uncovered locations [11]; results are averaged over 105 games except 16x30 which is averaged over 104 games. BSSUCT is the implementation of UCT from [13]. OH is our Optimistic Heuristics program. For OH, results are obtained with 10000 simulations per move, except expert mode and intermediate mode (99 mines on 16x30 and 40 mines on 16x16) which use 100 simulations per move.

Format	CSP-PGMS	HCSP	BSSUCT	OH
4 mines on 4x4	64.70%	67%	70% ± 0.9%	67% ± 0.5%
1 mine on 1x3	100%		100%	100%
3 mines on 2x5	22.60%	21%	25.4% ± 1%	23.4% ± 0.5%
10 mines on 5x5	8.20%	8.51%	9%(p-value: 0.14)	11.4% ± 0.4%
5 mines on 1x10	12.93%	12.70%	18.9% ± 0.2%	17% ± 0.4%
10 mines on 3x7	4.50%	4.76%	5.96% ± 0.16%	6.1% ± 0.2%
15 mines on 5x5	0.63%	0.63%	0.9% ± 0.1%	1.15% ± 0.1%
10 mines on 8x8		79.90%		80.2% ± 0.48%
10 mines on 9x9	80%	90.50%		89.9% ± 0.3%
40 mines on 16x16	45%	76.4% ± 0.4%		74.4% ± 0.5% (100 sims per move)
99 mines on 16x30	34%	38.1% ± 0.5%		38.7% ± 1.8% (100 sims per move)

Acknowledgements. The authors are grateful to NSC for funding NSC99-2923-E-024-003-MY3, NSC101-2815-C-024-021-E, NSC100-2811-E-024-001, and to ANR for funding COSINUS program (project EXPLO-RA ANR-08-COSI-004), to the authors of PGMS and PGMS-CSP for making their implementations freely available.

References

1. Ben-Ari, M.M.: Minesweeper as an np-complete problem. SIGCSE Bull. 37, 39–40 (2005)
2. Chaslot, G., Winands, M., Uiterwijk, J., van den Herik, H., Bouzy, B.: Progressive Strategies for Monte-Carlo Tree Search. In: Wang, P., et al. (eds.) Proceedings of the 10th Joint Conference on Information Sciences (JCIS 2007), pp. 655–661. World Scientific Publishing Co. Pte. Ltd. (2007)
3. Couetoux, A., Milone, M., Teytaud, O.: Consistent belief state estimation, with application to mines. In: Proceedings of the 2011 Conference on Technologies and Application of Artificial Intelligence (TAAI 2011), Taoyuan, Taiwan, November 11-13 (2011)
4. Coulom, R.: Computing elo ratings of move patterns in the game of Go. In: Computer Games Workshop 2007, Amsterdam, The Netherlands (2007)
5. Gelly, S.: Une contribution l'apprendissage par renforcement: application au Computer GO. PhD thesis, Université Paris-Sud (September 2007)
6. Gelly, S., Silver, D.: Combining online and offline knowledge in UCT. In. In: Proceedings of the 24th International Conference on Machine Learning (CML 2007), pp. 273–280. ACM Press, New York (2007)
7. Hein, K.B., Weiss, R.: Minesweeper for sensor networks–making event detection in sensor networks dependable. In: Proceedings of the 2009 International Conference on Computational Science and Engineering (CSE 2009), vol. 1, pp. 388–393. IEEE Computer Society, Washington, DC (2009)

8. Kaye, R.: Minesweeper is np-complete. Mathematical Intelligencer 22, 915 (2000)
9. Kocsis, L., Szepesvari, C.: Bandit based Monte-Carlo planning. In: 15th European Conference on Machine Learning (ECML), pp. 282–293 (2006)
10. Lee, C.S., Wang, M.H., Chaslot, G., Hoock, J.B., Rimmel, A., Teytaud, O., Tsai, S.R., Hsu, S.C., Hong, T.P.: The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments. IEEE Transactions on Computational Intelligence and AI in Games 1, 73–89 (2009)
11. Legendre, M., Hollard, K., Buffet, O., Dutech, A.: Minesweeper: Where to probe? Technical Report RR-8041, INRIA (2012)
12. Nakov, P., Wei, Z.: Minesweeper, #minesweeper. Technical report, EECS, Berkeley (2003)
13. Sebag, M., Teytaud, O.: Combining Myopic Optimization and Tree Search: Application to MineSweeper. In: Learning and Intelligent Optimization Conference, LION6, Paris, France, January 16-20 (2012)
14. Studholme, C.: Minesweeper as a constraint satisfaction problem. Unpublished project report (2000)
15. Wang, Y., Audibert, J.Y., Munos, R.: Algorithms for infinitely many-armed bandits. In: Advances in Neural Information Processing Systems, vol. 21 (2008)

Learning a Move-Generator for Upper Confidence Trees

Adrien Couetoux[1,2], Olivier Teytaud[1,2,3], and Hassen Doghmen[1]

[1] TAO-INRIA, LRI, CNRS UMR 8623,
Université Paris-Sud, Orsay, France
[2] OASE Lab, National University of Tainan, Taiwan
[3] Montefiore Institute, Université de Liège, Belgium

Abstract. We experiment the introduction of machine learning tools to improve Monte-Carlo Tree Search. More precisely, we propose the use of Direct Policy Search, a classical reinforcement learning paradigm, to learn the Monte-Carlo Move Generator. We experiment our algorithm on different forms of unit commitment problems, including experiments on a problem with both macrolevel and microlevel decisions.

1 Introduction

Monte-Carlo Tree Search (MCTS) [5] is a versatile algorithm for Markov Decision Processes (MDP) or games. It is elegant (can be described in a few lines), versatile (can be applied in many settings such as MDP, games, stochastic games), and can work with or without expert information in the playouts (hence a great success in general game playing[14]). It uses the framework of bandits[9] which does not depend on a particular application. It is moderately efficient when the number of time steps is big, but surprisingly stable on high-dimensional problems.

It has been greatly improved by including Progressive Widening and Double Progressive Widening[6,2], RAVE values[7], Blind Values[4], and handcrafted Monte-Carlo moves[17,10]. A crucial component is the Monte-Carlo move generator, also known as the playout generator.

In this paper, we focus on the addition of specialized Monte-Carlo moves, i.e. we modify default policy, to help dealing with stochastic planning problems. Finding a default policy that is optimal for all instances of a stochastic problem can be extremely difficult and time consuming. The solution we propose here is to apply a Direct Policy Search to the available default policy. This way, even an initially poor default policy can be improved to fit different instances of one stochastic planning problem.

In Section 2 we describe existing algorithms (Monte-Carlo Tree Search in Section 2.1, and existing algorithms for improving Monte-Carlo move generators in Section 2.2). We also introduce Direct Policy Search for improving a Monte-Carlo move generator in Section 2.3. In Section 3, we experiment our algorithms on three forms of a unit commitment problem (Section 3.1), and on an investment problems (Section 3.2).

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 209–218.
DOI: 10.1007/978-3-642-35452-6_23

2 Algorithms

In this section we will present the vanilla Monte-Carlo Tree Search algorithm (Section 2.1), Direct Policy Search, and Monte-Carlo move generators improvements (Sections 2.2 and 2.3).

2.1 Monte-Carlo Tree Search and Upper Confidence Trees

Monte-Carlo Tree Search (here presented in the framework of a MDP) consists in simulating plenty of series of decisions as long as we have time before choosing an action, and keeping statistics of these games. What follows, is the formal description of the state of the art continuous MCTS, i.e. MCTS with Double Progressive Widening (MCTS-DPW), as seen in [3]. As the reader can see, it mainly requires two things: (i) *a transition function*, capable of simulating what happens when an action a is taken in state s, and returns a new state s' and a reward r. (ii) *a default policy* φ, capable of returning an action a, given a state s. When nothing is specified, it is assumed that this function returns a random action, following a random distribution that covers the entire set of feasible actions in state s.

```
MCTS algorithm with DPW and default policy $\varphi$
Input: a state $S$.
Output: an action $a$.
Initialize: $\forall s, nbSims(s) = 0$
while Time not elapsed do
     // starting a simulation.
   $s = S$.
   while $s$ is not a terminal state  and $nbSims(s) > 0$ do
      Apply DPW in state $s$.
      Let $s'$ be the state given by DPW.
      $s = s'$
   end while
   while $s$ is not a terminal state // {happens when a non final and new state $s$ is
   visited} do
      Choose action $a$, according to $\varphi$
      Simulate action $a$; get a new state $s'$
      $s = s'$
   end while
   Get a reward $r = Reward(s)$  // $s$ is a final state, it has a reward.
   For all states $s$ in the simulation above, let $r_{nbVisits(s)}(s) = r$.
end while
Return the action which was simulated most often from $S$.
```

The algorithm therefore relies on a Monte-Carlo move generator, also called default policy, φ. The default policy can be a simple random uniform generator (when no expertise is available for making more reasonable simulations), but handcrafted functions can perform better.

Action Selection by Double Progressive Widening (DPW), applied in state s with constants $C > 0$, $\alpha \in]0,1[$, and $\beta \in]0,1[$.

```
Input: a state $s$.
Output: a state $s'$.
Let $nbVisits(s) \leftarrow nbVisits(s) + 1$
and let $t = nbVisits(s)$
Let $k = \lceil Ct^{\alpha} \rceil$.
Let $(o_i(s))_{i \geq 1}$ be the feasible actions in state $s$.
Choose an action $a_t(s) \in \{o_1(s), \ldots, o_k(s)\}$ maximizing $score_t(s,a)$ defined as follows:
  $totalReward_t(s,a) = \sum_{1 \leq l \leq t-1, a_l(s)=a} r_l(s)$
  $nb_t(s,a) = \sum_{1 \leq l \leq t-1, a_l(s)=a} 1$
  $score_t(s,a) = \frac{totalReward_t(s,a)}{nb_t(s,a)+1} + k_{ucb}\sqrt{\log(t)/(nb_t(s,a)+1)}$ $(+\infty$ if $nb_t(a) = 0)$
Let $k' = \lceil C nb_t(s, a_t(s))^{\beta} \rceil$
if $k' > \#Children_t(s, a_t(s))$ // {progressive widening on the random part} then
   Simulate action $a_t(s)$; get a new state $s'$
   if $s' \notin Children_t(s, a_t(s))$ then
      $Children_{t+1}(s, a_t(s)) = Children_t(s, a_t(s)) \cup \{s'\}$
   else
      $Children_{t+1}(s, a_t(s)) = Children_t(s, a_t(s))$
   end if
else
   $Children_{t+1}(s, a_t(s)) = Children_t(s, a_t(s))$
   Choose $s'$ in $Children_t(s, a_t(s))$ // $s'$ is chosen with probability $nb_t(s, a_t(s), s')/nb_t(s, a_t(s))$
end if
```

2.2 Heuristics and Monte-Carlo Move Generators

Whereas in the 2-player case, it is known that making a Monte-Carlo generator stronger (stronger in the sense: as a stand-alone policy), does not necessarily make the MCTS built on top of it stronger (see [17]), we conjecture that in the one-player case it is usually quite efficient.

The recent improvements in the world of Computer Go basically comes from improvements of the Monte-Carlo move generator, implemented so that the Monte-Carlo simulator does not contradict life&death known results; Zen, Crazy-Stone, Pachi, are examples of such strong programs, around 2 Dan for short time settings and 4 Dan for long time settings. Other tools have been proposed as generic solutions for learning Monte-Carlo move generators:

- Simulation balancing [15,8] has been proposed for automatically learning the Monte-Carlo move generator in 2-player games.
- PoolRave[13], in which the Monte-Carlo move is replaced, with a fixed probability $p \in (0,1)$, by a move uniformly drawn among the c moves with best RAVE score in the last node of the simulation with at least k simulations.

- Contextual Monte-Carlo[12] in which the Monte-Carlo move-generator is improved by online learning a tile-based value function.

These tools are efficient, but the main successes in Monte-Carlo Tree Search nonetheless come from handcrafted Monte-Carlo move generators. In this paper, we used a specialized Monte-Carlo move generator, chosen specifically on our main target problem, as well as a less specialized function. We improve them by Direct Policy Search (Section 2.3).

2.3 Direct Policy Search for Generating Monte-Carlo Move Generators

Direct Policy Search (DPS) is an approach very different from Upper Confidence Tree; it is based on selecting a policy among a parametric family of policies by optimization of its parameters. The pseudo-code is as follows:

```
Procedure $Simulate(s, MDP, p)$:
Inputs: a state $s$, a Markov Decision Process $MDP$, and a policy $p$.
Output: a reward.
Method: simulate $MDP$ from state $s$ with policy $p$ until a terminal state and return the obtained reward.
Procedure Direct Policy Search:
Inputs: (i) a parametric policy $\theta \mapsto \pi(\theta)$, where $\pi(\theta)$ is a mapping from states to actions.
(ii) a Markov Decision Process $MDP$. (iii) an initial state $s$.
Output: a parameter $\hat{\theta}$, leading to a policy $\pi(\hat{\theta})$.
Auxiliary method: a noisy optimization algorithm.
Apply the noisy optimization algorithm to the function $\theta \mapsto Simulate(s, MDP, \pi(\theta))$;
get $\hat{\theta}$ the approximate optimum.
Return $\hat{\theta}$.
```

Direct Policy Search is usually applied offline, i.e. a single $\hat{\theta}$ is obtained once and for all. However, optimizing Θ to maximize $\theta \mapsto Simulate(s, MDP, \pi(\theta))$ specifically for the current state s for which we look for a decision is possible. We apply DPS and use the obtained policy $\pi(\hat{\theta})$ as a Monte-Carlo move generator in our MCTS.

The paper in [1] proposes to apply DPS (the terminology in the paper is different, but it is essentially DPS) based on a heuristic function obtained by experts, by smoothing the heuristic and adding parameters in it (the smoothing is here for making the problem easier to optimize). This is our approach in the rest of this paper, except that we do not smooth policies as the randomized nature of our problems make the objective function smooth enough. We use self-adaptation[11] as a noisy optimization algorithm. As a summary, our algorithm is as follows for choosing a move in state s within time t:

```
Procedure $OptimisticHeuristics(s, \phi, t, MDP)$
Input: a state $s$, a time $t$, a parametric family of policies $\phi_\theta$.
Output: an action $a$.
Apply DPS with time budget $t/2$ for choosing $\hat{\theta}$ (use warm start if possible)
Apply MCTS with time budget $t/2$ for choosing action $a$.
```

3 Experiments

Here, we compare the performances of different sequential decision making algorithms. Namely, we implemented vanilla MCTS, MCTS with a fixed default policy, MCTS with a default policy improved online by DPS, and DPS alone.

We made experiments on three different forms of the unit commitment problem, and on a more general energy management problem called bilevel.

3.1 Unit Commitment Problem

We work on a stock management problem, from [16].

The main points in the problem are that: (1) Unit Commitment problems can not be solved efficiently by traditional methods; these problems are usually simplified so that classical methods, like Bellman's stochastic dynamic programming, can be applied. The motivation of our work on Unit Commitment by Monte-Carlo Tree Search methods is that we want to work without simplifying too much the model. (2) Unit Commitment problems exist at many time scales (from milliseconds, up to years for hydroelectric stocks or tenths of years if investments are included) and many dimensionalities (from a few stocks to thousands of state variables), depending on the scope under analysis. We here work on small scale problems for the sake of statistical significance (working on our full problems requires by far too much time for reproducing runs tenths of times).

In this paper, we will consider three variants of the unit commitment problem. The significant difference between these three variants is the way the stocks are connected. In the first one, they are lined up on a one dimension chain (we will call it the one river problem). In the second one, they are linked so that they form a binary tree, with the root being the last stock that the water goes through (we will call it the "binary rivers problem"). Finally, the third one is simply a random arrangement of the stocks, with one single constraint: no cycles are allowed.

Two Different Heuristics for the Unit Commitment Problem. The expert parametrized heuristic that we use has been designed using knowledge about the problem, to make it particularly efficient on the one river variant of the unit commitment problem. On the other hand, the naive heuristic uses almost no knowledge about the problem. Given a state s, it requires the current time to go t, and the average demand at the current time step D_{avg}. We provide below pseudo-codes of both heuristics.

Experimental Results on the Unit Commitment Problem. We present here the results obtained on all three variants of the unit commitment problem. Each time, we compared the following algorithms: (i) vanilla MCTS, as presented in Section 2.1, (ii) MCTS-naive, a MCTS using the naive heuristic as a default

Expert heuristic
Input: a state S, of dimension N.
Parameter: a vector θ of dimension 3. Default value is $[1, 0, 1]$.
Information required from the problem:

- $D(t)$: expected electricity demand during time step t.
- $D_{timeToGo}(t)$: expected total demand after time step t.
- $TSA(s, timeToGo)$: total stock available (this assume a 1 river structure).
- $TI(timeToGo, averageInflows)$: expected total usable water from inflows (this assume a 1 river structure).

Output: an action a.

```
1. initialize:
   – total water available = $TWA = (\theta_0 + \theta_1 \times timeToGo) \times (TSA + TI)$
   – $x = production\ by\ hydroelectricity = 0$
   – $increaseWater \leftarrow true$
   – $S_{available} = \sum_{0 \le i \le N-1} s_i$
2. while ($increaseWater$ and $x < S_{available}$) $s_i$ being the current level of stock $i$ do
     define the marginal cost $mc$ of increasing water, approximated as
       $mc = IC(x, s, t, D(t)) + \theta_2 \times LTC(x, TWA, t, D_{timeToGo}(t))$
     where:
     –  $IC(x, s, t, D(t))$ is negative; it is the marginal benefit associated to the reduction of thermal production.
     –  $LTC(\dots)$ is the sum of thermal production cost, if expected total demand $D_{timeToGo}$, decreased by the total production from the water stocks if equally distributed on the time steps to go, is produced thermally.
     if marginal cost $mc > 0$ then
        then $increaseWater \leftarrow false$
     else
        $x \leftarrow x + 1$.
     end if
3. end while
4. Compute $q$, the ratio $\min(0, \frac{x}{S_{available}})$
5. Return the action vector $a$ defined as follows: $\forall 0 \le i \le N-1, a_i = q.L_i$
```

Naive heuristic, polynomial with degree m
Input: a state S, of dimension N.
Parameter: a vector θ of dimension $m + 1$. Default values are $[1, 0, \dots, 0]$
Information required from the problem: t the remaining time steps, and D_{avg} the average demand after the current time step
Output: an action a.

```
1. Compute total amount of water to use $W_{use} = \max(0, D_{avg}.(\theta_0 + \theta_1 t + \cdots + \theta_{m+1} t^m))$
2. Given $S$ and the current level $L_i$ of each stock $i$, $W_{available} = \sum_{0 \le i \le N-1} L_i$
3. Compute $q$, the ratio $\min(0, \frac{W_{use}}{W_{available}})$
4. Return the action vector $a$ defined as follows: $\forall 0 \le i \le N-1, a_i = q.L_i$
```

policy,(iii) MCTS-expert, a MCTS using the expert heuristic as a default policy, (iv) MCTS-naive-DPS, a MCTS using the naive heuristic improved by DPS, (v) MCTS-expert-DPS, a MCTS using the expert heuristic improved by DPS and when relevant, (vi) the non tuned naive and expert heuristics.

The x axis shows the time budget allocated per decision made, in logarithmic scale, and went from 0.01 second to 2.56 second. The y axis shows the average reward. Each average reward was computed using 1000 runs. Error bars show the 95% confidence intervals. The higher the reward, the better the algorithm performed. It should be noted that the rewards cannot be compared between different variants of the unit commitment problem. Indeed, only the connections between the stocks change, and changing this changes the amount of water effectively available. Our results for the 1-river problem and for the binary rivers problem are shown on the left side and the right side of Fig. 1, respectively. We did not plot the results of the naive heuristic, that scored −150000 and −380000 respectively (far below other methods), for the sake of readability. In both experiment, MCTS-naive-DPS and MCTS-expert-DPS outperform by a factor of at least 100 the third placed algorithm, MCTS-expert. MCTS-naive and MCTS vanilla share the fourth and fifth places.

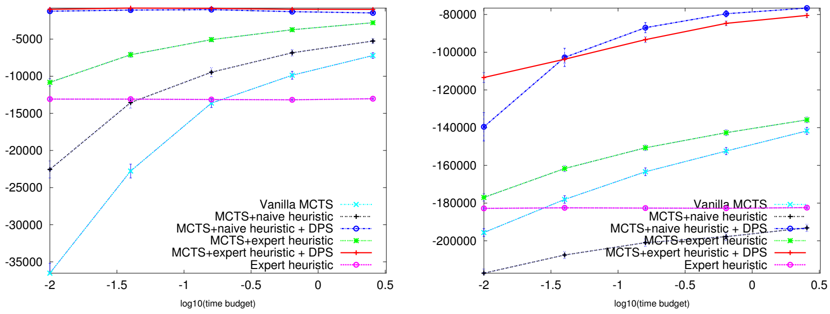


Fig. 1. Performances of different variants of MCTS on the 1 river unit commitment problem (left) and the binary rivers (right), with 7 stocks, 24 time steps. Y axis shows the reward (the higher the better).

Our results on the random rivers problem are shown in Fig. 2. In this experiment, the most significant difference in the results is that MCTS-naive-DPS is about 10 times faster than MCTS-expert-DPS.

Over all three versions of the unit commitment problem, the most efficient and robust version has been MCTS-naive-DPS. Even on the one river problem, that the expert heuristic was particularly well tuned for, we could not see huge benefits from using it as a parametric function for DPS.

3.2 Experiments on the Investment Problem

In this section we experiment our algorithm on a problem (simplified from [16]) as follows:

- At each time step, we decide investments; there is a limited amount of money to invest, and investments must be distributed over 7 different possible infrastructures. There are therefore 7 decision variables for each time step.

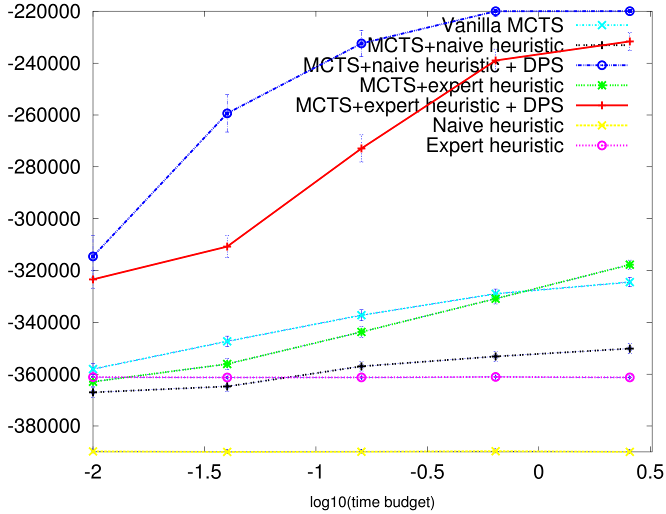


Fig. 2. Performances of different variants of MCTS on the randomly connected unit commitment problem (7 stocks, 24 time steps). Y axis shows the reward (the higher the better).

- At each time step, a lower level problem (the management of the energy production system) is built and solved, and its cost is the cost of the current transition of the investment problems.
- There are 10 time steps, the last one has a strong influence because it reflects the long-term.

Our results compare the following strategies:

- a heuristic which gives a constant ratio of the investment on each possible infrastructure (the parameters of the heuristic are this proportions); the default parametrization is the same ratio for all infrastructures;
- DPS on a “sum of Gaussians” policy (parameters: positions of the Gaussians, widths, associated decisions; see [16];
- DPS on a “neural network” policy (parameters: weights, theresholds; see [16]);
- DPS on a “sum of Gaussians” policy, added to the heuristic with default parametrization;
- DPS on a “neural network” policy, added to the heuristic with default parametrization;
- MCTS, on top of each of the above.

Results are presented in Fig. 3.

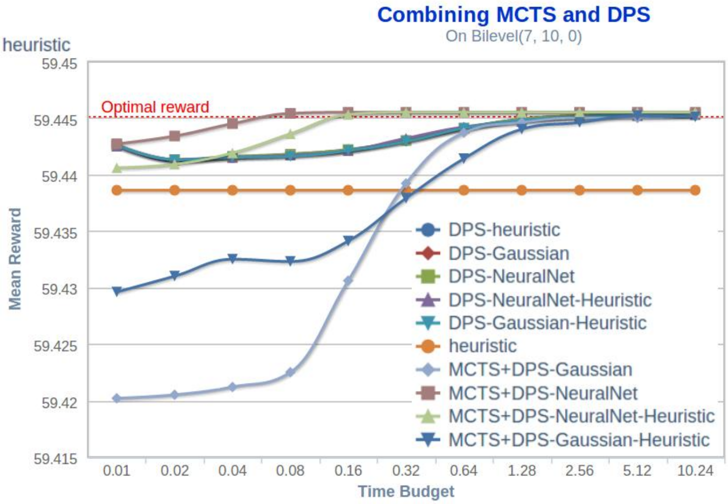


Fig. 3. Results on the energy investment problem. The five DPS curves (curves 1 to 5) are very close to each other; results are better than for the heuristic alone, and versions without the heuristic are almost the same as versions with the heuristic. The MCTS+DPS+neural network was the most efficient strategy, outperforming MCTS+DPS+neural network+heuristic. The sums of Gaussians require more time for learning, hence the poor results for moderate budgets.

4 Conclusion

We combine DPS and MCTS. The DPS provides the Monte-Carlo simulator of the MCTS. The resulting algorithm, has no free parameter and outperforms by far the vanilla MCTS. We use human expertise at two levels: (i) For partial observation handling, i.e. the belief state estimation was handcrafted, so that the problem is essentially a MDP rather than a partially observable MDP. The details of this are beyond the scope of this paper. (ii) In the Monte-Carlo move generator, because in spite of nice and interesting efforts in the literature, no generic algorithm, in the current state of the art, can define a Monte-Carlo move generator as efficiently as a human expert (in the case of Go, but also in the case of unit commitment problems). Nonetheless our DPS could strongly improve the heuristic by optimizing its parameters. We agree with the traditional statement that MCTS is surprisingly efficient when no human expertise is available, but we clearly see that human expertise was an easy key for a speed-up 100, as well as human expertise is the key of recent progress in MCTS for the classical challenge of the game of Go.

Importantly, the need for human expertise is considerably reduced by the use of DPS for optimizing the heuristics, so that our results are a step towards generic MCTS tools.

References

1. Bengio, Y.: Using a financial training criterion rather than a prediction criterion. CIRANO Working Papers 98s-21, CIRANO (1998)
2. Chaslot, G., Winands, M., Uiterwijk, J., van den Herik, H., Bouzy, B.: Progressive Strategies for Monte-Carlo Tree Search. In: Wang, P., et al. (eds.) Proceedings of the 10th Joint Conference on Information Sciences (JCIS 2007), pp. 655–661. World Scientific Publishing Co. Pte. Ltd. (2007)
3. Couëtoux, A., Hoock, J.-B., Sokolovska, N., Teytaud, O., Bonnard, N.: Continuous Upper Confidence Trees. In: Coello Coello, C.A. (ed.) LION 2011. LNCS, vol. 6683, pp. 433–445. Springer, Heidelberg (2011)
4. Couëtoux, A., Doghmen, H., Teytaud, O.: Improving the Exploration in Upper Confidence Trees. In: Hamadi, Y., Schoenauer, M. (eds.) LION 2012. LNCS, vol. 7219, pp. 366–371. Springer, Heidelberg (2012)
5. Coulom, R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In: van den Herik, H.J., Ciancarini, P., Donkers, H.H.L.M(J.) (eds.) CG 2006. LNCS, vol. 4630, pp. 72–83. Springer, Heidelberg (2007)
6. Coulom, R.: Computing elo ratings of move patterns in the game of go. In: Computer Games Workshop, Amsterdam, The Netherlands (2007)
7. Gelly, S., Silver, D.: Combining online and offline knowledge in UCT. In: ICML 2007: Proceedings of the 24th International Conference on Machine Learning, pp. 273–280. ACM Press, New York (2007)
8. Huang, S.-C., Coulom, R., Lin, S.-S.: Monte-Carlo Simulation Balancing in Practice. In: van den Herik, H.J., Iida, H., Plaat, A. (eds.) CG 2010. LNCS, vol. 6515, pp. 81–92. Springer, Heidelberg (2011)
9. Kocsis, L., Szepesvari, C.: Bandit based Monte-Carlo planning. In: 15th European Conference on Machine Learning (ECML), pp. 282–293 (2006)
10. Lee, C.-S., Wang, M.-H., Chaslot, G., Hoock, J.-B., Rimmel, A., Teytaud, O., Tsai, S.-R., Hsu, S.-C., Hong, T.-P.: The Computational Intelligence of MoGo Revealed in Taiwan's Computer Go Tournaments. IEEE Transactions on Computational Intelligence and AI in Games (2009)
11. Meyer-Nieberg, S., Beyer, H.-G.: Self-adaptation in evolutionary algorithms. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms. Springer, Berlin (2007)
12. Rimmel, A., Teytaud, F.: Multiple Overlapping Tiles for Contextual Monte Carlo Tree Search. In: Evostar, Istanbul, Turquie
13. Rimmel, A., Teytaud, F., Teytaud, O.: Biasing Monte-Carlo Simulations through RAVE Values. In: van den Herik, H.J., Iida, H., Plaat, A. (eds.) CG 2010. LNCS, vol. 6515, pp. 59–68. Springer, Heidelberg (2011)
14. Sharma, S., Kobti, Z., Goodwin, S.: Knowledge Generation for Improving Simulations in UCT for General Game Playing. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 49–55. Springer, Heidelberg (2008)
15. Silver, D., Tesauro, G.: Monte-carlo simulation balancing. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) *ICML*. ACM International Conference Proceeding Series, vol. 382, p. 119. ACM (2009)
16. Teytaud, O.: Including Ontologies in Monte-Carlo Tree Search and Applications - an Open Source Platform (2008)
17. Wang, Y., Gelly, S.: Modifications of UCT and sequence-like simulations for Monte-Carlo Go. In: IEEE Symposium on Computational Intelligence and Games, Honolulu, Hawaii, pp. 175–182 (2007)

Software Framework for Generic Game Development in CGDG

Hao-Yun Liu, I-Chen Wu, Ting-Fu Liao, Hao-Hua Kang, and Lung-Pin Chen

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
{hyliou,icwu,tfliao,kangbb,lpchen}@java.csie.nctu.edu.tw

Abstract. In this paper, we propose a software framework for easily facilitating computer game developments. Using the software framework, the developers can easily implement game record editors for computer game AI analysis. Additionally, using the framework, the developers can easily design job-level (JL) search algorithm to dispatch jobs to workers of a desktop grid, named CGDG. Based on the framework, we successfully developed several record editors for many games, such as Connect6, Go, Chinese Chess, Mahjong, and also developed some JL search algorithms, such as JL proof number search (JL-PNS), JL Monte-Carlo tree search (JL-MCTS), JL alpha-beta search (JL-ABS). Most importantly, once a JL algorithm such as JL-PNS is implemented, it can be applied to all games. This demonstrates the excellence of this framework.

Keywords: software framework, game record editor, job-level algorithm, CGDG.

1 Introduction

In the area of computer games, it is critical to have a good game development environment, in addition to the techniques of artificial intelligence (AI). With a good software development environment, a team can efficiently develop different computer games and different algorithms on them.

Upon developing a game such as Connect6 [26], a game record editor is needed for computer game AI analysis. An editor needs to include at least the following three functionalities. First, support an interactive interface for users to enter a game record. Second, support an easy way to create, maintain and delete branches of game positions. Third, transform a game record into a file.

In addition to game record editing, we also developed a job-level proof number search (JL-PNS) algorithm to help solve Connect6 openings [28-29]. Job-level (JL) is a kind of methodology used to parallelize game search algorithms [1-2], [10]. A JL search algorithm dispatches computer game jobs to idling workers. In [16], [25], we designed a desktop grid of workers, named CGDG, for performing the dispatched jobs.

For a team developing many computer games, it is critical to have a good game development framework in two aspects. First, this framework allows programmers to develop the GUI game record editing systems quickly via a plug-in mechanism for

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 219–229.
DOI: 10.1007/978-3-642-35452-6_24

different board games, such as Go-Moku, Connect6, Go, Chinese Chess, etc. Second, this framework allows designers to develop JL algorithms quickly, such as JL AI competitions (JL-AIC), JL-PNS, JL Monte-Carlo tree search (JL-MCTS), JL alpha-beta search (JL-ABS).

This paper proposes a software framework for easily facilitating computer game developments from the above two aspects. Based on the framework, we successfully developed several record editors for the above games and also the above JL search algorithms orthogonally. By “orthogonally”, we mean that a JL algorithm such as JL-PNS can be applied to all the other games once it is implemented, and that a game can use all the other JL algorithms once it is implemented.

In this paper, we design the software framework and describe the following three major modules: a generic module for developing game record editing for all games, a generic module for developing job-level search algorithms, and a library to communicate with CGDG.

This paper is organized as follows. Section 2 reviews the background of related work, and Section 3 explains the design of our framework. Two case studies are given in Section 4, and a conclusion is made in Section 5.

2 Background

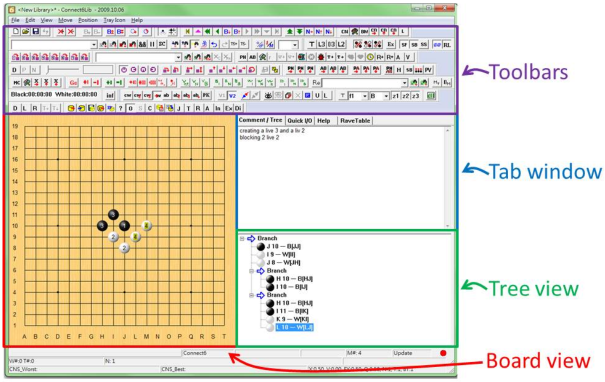
In the past, software engineering has gained prominence in academia, including agile programming [4], extreme programming [11-12], [15], [17], test-driven programming [3], [13], [19]. Other than the techniques of software engineering, this section reviews the past work for game record editors, CGDG, and job-level model in Subsections 2.1, 2.2 and 2.3, respectively.

2.1 Game Record Editor

As mentioned above, game record editors are important in the developments of computer games (especially with AI). Many game record editors were developed in the past. Among them, we chose an open-source editor for Renju, named Renlib [21], and then modified it to fit Connect6, called Connect6Lib [7] as shown in Fig. 1 (below).

In Fig. 1, the panel in the lower-left part is to show the board view which shows the current position is. The one in the lower-right part is to show the tree view which shows the game record tree. The one in the center-right part is to show a tab window which provides the users with some utilities, e.g., comments on a position, or a console output for debugging. Toolbars listed on the top are used to provide users with a variety of functionalities via buttons.

Connect6Lib was further modified to the editors for Go, Chinese Chess, etc. For these new games, we can simply modify part of the code, such as board view, some of toolbars, while sharing the same code for the rest, like most of tab window and tree view. Thus, it becomes important to develop a generic game editor to fit more games.



creating a live 3 and a liv 2
blocking 2 live 2

Fig. 1. Connect6Lib, our Connect6 editor

2.2 Job-Level Model

This subsection reviews the job-level (JL) model proposed by [28]. Owing to some requests, the editor Connect6Lib also supports a functionality in which a job can be dispatched to a worker for running. Jobs include finding the best move from a position, expand all moves of a position, or run a MCTS simulation from a position. The job result can be the best move, all the expanded moves, or the simulation result for updating the tree.

In our earliest version, it is the responsibility of users to select game position and then dispatch a job on it. However, one issue is the difficulty of dispatching jobs manually. Therefore, the authors in [28] proposed a JL algorithm, called JL-PNS, to dispatch jobs automatically. A generic job-level model was proposed in [29].

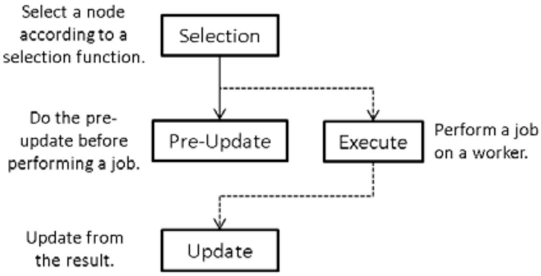

Select a node according to a selection function.
Selection
Do the pre-update before performing a job.
Pre-Update
Execute
Perform a job on a worker.
Update from the result.
Update

Fig. 2. Job-level model

The job-level model, shown in Fig. 2, includes four phases, selection, pre-update, execution, and update phases. First, in the selection phase, select a node according to a selection function based on some search technique. For example, PNS selects the most proving node [1-2]; and MCTS selects a node based on the so-called tree policy (defined in [6], [14]). Note that the search tree is supposed to be unchanged in this phase.

Second, in the pre-update phase, update the tree in advance to prevent from choosing the same node. In this phase, several policies can be used to update the search tree. For example, the flag policy is to set a flag on the selected node, so that the flagged nodes will not be selected again.

Third, in the execution phase, performs a job for a position on an idling worker as mentioned above. For example, find the best move from a node n, expand all moves of n, or run a simulation from n for MCTS.

Fourth, in the update phase, update the search tree according to the job result. For the above example, generate a node for the best move, generate nodes for all expanded moves, and update the status on the path to the root.

The job-level search model is described in greater detail in [29].

2.3 CGDG

For job-level computing, CGDG (Computer Game Desktop Grid), a desktop grid of workers developed in [25], is designed to allow many computer game developers to solve many computer game applications. It is also a kind of volunteer computing grid [5], [9], [22], [32]. CGDG consists of clients, broker, and workers. Broker is a central server to dispatch jobs, and worker is a remote computing resource. We have developed a worker program for Windows and Linux.

3 Framework

This section describes the design of our software framework for computer games. The design philosophy is to share common code for different games. Practically, we achieve this by extracting shared code segments into base classes. So, we can let new games simply extend the base classes and implement the differences.

The framework has three major modules, a generic module for developing game record editing for all games, a generic module for developing job-level search algorithms, and a library to communicate with CGDG. They are described in Subsections 3.1, 3.2 and 3.3, respectively.

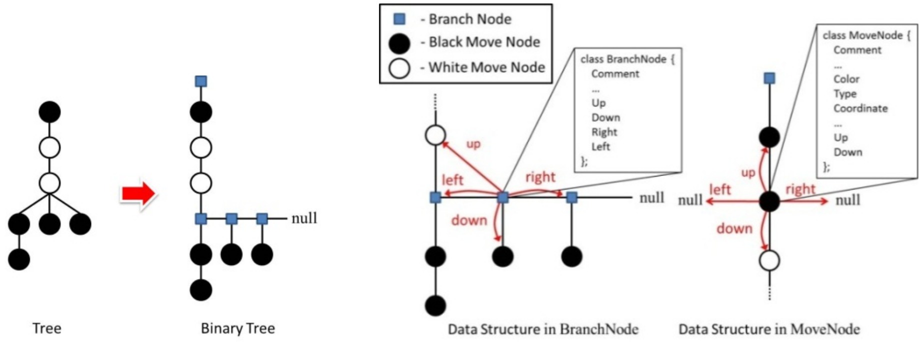
3.1 Module for Generic Game Editing

MVC [23-24], indicating the three components, model, view, and controller, is a software architecture commonly used for GUI program. Model refers to the domain logic, e.g. the game record tree in editor. View refers to the domain logic's presentation, e.g. the board view presenting the current position. Controller refers to the mapping of UI events to functionalities, e.g. the mapping of a "delete move" button to the move deletion function.

We use this architecture to design the module. The most important components in the game module are model and view. Because we use Microsoft Foundation Class (MFC) to develop our framework, the controller is distributed over model and view and not encapsulated into a class.

Model

The model component includes two main parts, the tree structure and the interpretation of moves, and some more minor functionalities.



(a) (b)

Fig. 3. (a) Tree structure and (b) node structure

A tree is a data structure used to store game records. We expect that it can be used in all game types, so it is designed as general as possible. For a game record tree, nodes represent game positions and normally contain multiple children, since users may choose many moves from a position. However, it is hard to implement and present in the tree view of editors. So, in practice, it is represented in a binary tree by inserting branching nodes, as shown in Fig. 3 (a).

As described above, a node represents a position (or a move) or a branch. For a position, it is also called a move node, since the data structure for it mainly contains information about a move such as coordinates, color of move, type of piece, links used to maintain the whole tree structure, comments for the move, etc. For a branch, the data structure mainly contains the links and some other information like comments. The data structures for nodes are shown in Fig. 3 (b).

Since it is hard to maintain links among the tree, we normally want to hide the details from the game developer, so that the maintenance can become easier. Since the links are hidden from developers, we provide the game developers with one mechanism to traverse tree. The mechanism for game developers is node pointer, or called node traverser. It uses iterator and visitor design pattern and offers developers an easy way to traverse or modify tree instead of modifying links in node structure directly.

Move interpretation is another important part in the model. Game records must be able to be saved into disk files or databases and loaded from disk files and databases. However, different games may need to define different formats of game records. Currently, we follow the Smart Game Formant (SGF) to store game records into files. It is a common game record format in Go and Connect6. However, the format becomes different for other games like Chinese Chess or even Mahjong. Practically, we provide base classes for serialization (the move interpretation), while game developers can extend the base classes for different game formats.

View

There are three major views, board view, tree view, and tab window. These three views are also abstracted into base classes, and game developers can extend them and redefine them.

Board view draws the board and some support display, e.g. showing available moves. It is more complex than tree view and tab window because the board changes for different games. For instance, Connect6 and Go use the Go board, and Chinese Chess has its own board. We leave a board drawing function for game developers and provide them with many utilities to draw symbols, lines, and pictures to easily draw their board.

Tree view displays the tree structure of game records, and is generally designed for the tree structure in the model as described above. Game developers usually do not need to redesign the tree view. If it is required to customize move strings in the tree view, they can customize a function to change the string shown on a move node. For example, the presentation for moves of Chinese Chess is much different from that for Go and Connect6.

Tab window offers an interface to conveniently add some sub-functionality. It includes a comment window, a console window, and a quick IO window by default. Game developers can change the default tabs by overriding the registration function.

3.2 Job-Level Module

The job-level (JL) module is used to automatically dispatch jobs to CGDG by applying different kinds of job-level search algorithm such as JL-PNS, JL-MCTS, and JL-AIC. To support these, the job-level module supports a generic job-level algorithm template.

Subsection 2.2 mentions that there are four common phases among different job-level search algorithms. From the above, the client only needs to handle three kinds of events: initialization, worker idling, and returning job result. Three event handlers for the three events are described respectively as follow.

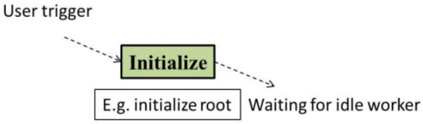


Fig. 4. Initialization event handler

Initialization event is triggered when a user demands to start. In the corresponding event handler, JL algorithm developers should initialize the tree if needed, as shown in Fig. 4.

Worker idling event is triggered when an idle worker is available (in CGDG). In the corresponding event handler, JL algorithm developers should select nodes, pre-update the tree, and dispatch jobs related to the selected nodes, as shown in Fig. 5.

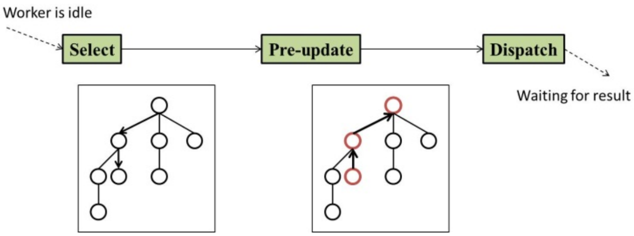


Fig. 5. Worker idling event handler

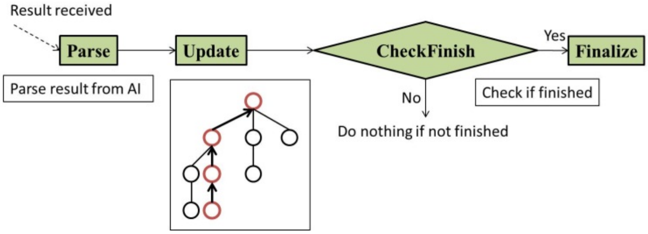


Fig. 6. Returning job result event handler

Returning job result event is triggered when job result is returned. In the corresponding event handler, JL algorithm developers should parse the result to update the tree and check if it is finished, and finalize algorithm if needed, as shown in Fig. 6.

In the above three event handlers, we abstract eight functions that different JL algorithm developers need to redefine or customize. So, we apply a design pattern named template method for JL algorithm developers to implement their algorithm quickly by redefining them.

Furthermore, for a JL algorithm, some portions of the eight functions should be provided by game developers. For example, for finding the best children for a position, JL algorithm developers should let game developers decide which children to be expanded; and similarly, let game developers decide position values.

In order to solve this problem, JL algorithm developers can provide game developers with an interface, `GamePolicy`, and let game developers implement it. It is very important to support orthogonality of JL algorithm developers and game developers based on this feature. Therefore, a JL algorithm is not limited to a certain games.

3.3 CGDG Library

In principle, the CGDG library hides the connection mechanism, XML analyzer, and event polling loop from JL algorithm and game developers. Developers can easily design their code by using the library. The library is composed of two kinds of modules, OS independent modules and OS dependent modules, to make it easy to combine with other frameworks such as MFC and X-window. Fig. 7 (below) shows the layer diagram for the application and the library.

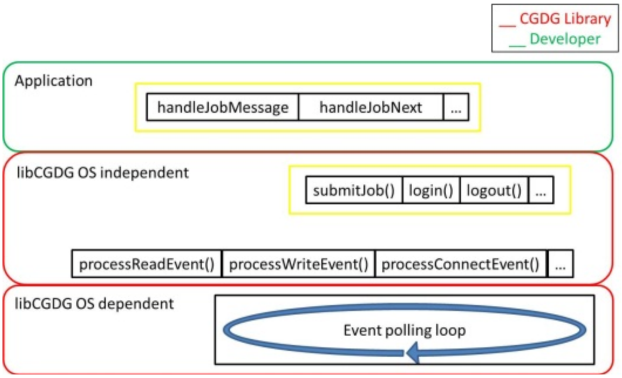


Fig. 7. CGDG library layer diagram

In the application part, developers only need to know the two sets of interfaces (depicted in yellow boxes in Fig. 7): event handlers for receiving commands from the broker, and operation requests for sending commands to the broker. In the OS dependent part, we implement the event polling loop and extract an interface for portability on different OSs. In the OS independent part, it handles all kinds of socket event handlers which notify the application.

4 Case Study

For illustration, we use Tic-Tac-Toe to demonstrate how to design a game record editor quickly, and use job-level PNS for Connect6 to demonstrate how to implement a job-level algorithm and apply this algorithm to any games rapidly. In this section, we show some results, while the details are described in [19].

To demonstrate how easy to develop based on our framework, we analyze the code on both game record editing module and JL module by comparing them with the earlier version of editor.

In game record editing module, the Tic-Tac-Toe editor has only 836 lines, and the base game module has 27854 lines. Additionally, with a rough approximation, the basic editor part of an earlier version has about 40000 lines. Apparently, it is much simpler to develop the game using our framework.

On the other hand, JL-PNS has 1178 lines, and the base JL module has 6658 lines excluding CGDG library which has 6904 lines. Also with a rough approximation, the basic job-level part of an earlier version has about 20000 lines. Apparently, again, it is much simpler to develop JL-PNS using our framework.

It seems that this software framework greatly reduces the overhead on developing game record editor for every game and every JL algorithm.

5 Conclusion

In this paper, we design a software framework with CGDG support which makes game development much easier. Our contributions are listed as follows.

- We define an API for rapid game record editor development for every game. Game developers can complete their own editor by extending the base classes and overriding the corresponding virtual functions.
- We define a template for JL search algorithm development. JL algorithm developers only need to override the eight functions in the template to implement a JL algorithm.
- We provide a library to access CGDG.
- Our case study also shows the benefit of using our framework to develop games and JL algorithms. For example, for Tic-Tac-Toe, the code size is reduced from about 35,000 lines to 836 lines; for JL-PNS the code size is reduced from about 15,000 lines to 1178 lines.

Table 1. Current status

Status	Pure Algorithm	Connect6	Go	Chinese Chess	Dark Chinese Chess	Mahjong	Tic-Tac-Toe
Pure Editor	✕	O	O	O	Δ	O	O
JL-PNS	O	O					
JL-MCTS	O	Δ	O				
JL-SSS*	Δ			Δ			
AI Competition	O	O		O			

The current status of our project is listed in Table 1. Circle denotes finished projects and triangle denotes ongoing projects. We believe that this software framework can make these support tools easy to maintain and easy to scale, and facilitate computer game development. In fact, the framework has already helped the researches in [28-29], and also help our game programs to win medals in many game tournaments, such as TAAI, TCGA tournaments [8],[18], [30-31], [33-34].

References

1. Allis, L.V.: Searching for solutions in games and artificial intelligence, Ph.D. Thesis, University of Limburg, Maastricht, The Netherlands (1994)
2. Allis, L.V., van der Meulen, M., van den Herik, H.J.: Proof-number search. Artificial Intelligence 66(1), 91–124 (1994)

3. Beck, K.: Test-Driven Development by Example. Addison Wesley - Vaseem (2003)
4. Beck, K., et al.: Manifesto for Agile Software Development. Agile Alliance (2001) (retrieved June 14, 2010)
5. BOINC website, http://boinc.berkeley.edu/
6. Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A Survey of Monte Carlo Tree Search Methods. IEEE Transactions on Computational Intelligence and AI in Games 4(1) (forthcoming 2012)
7. Chen, C.-P., Wu, I.-C., Chan, Y.-C.: ConnectLib – A Connect6 Editor (2009), http://www.connect6.org/Connect6Lib_Manual.html
8. Chou, C.-W., Yen, S.-J., Wu, I.-C.: TAAI 2011 Computer Go Tournaments. ICGA Journal 34(4), 251–252 (2011)
9. Condor website, http://www.cs.wisc.edu/condor/
10. Coulom, R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In: van den Herik, H.J., Ciancarini, P., Donkers, H.H.L.M(J.) (eds.) CG 2006. LNCS, vol. 4630, pp. 72–83. Springer, Heidelberg (2007)
11. Design Patterns and Refactoring, University of Pennsylvania (2003), lecture slide at http://www.cis.upenn.edu/~matuszek/cit591-2003/Lectures/49-design-patterns.ppt
12. Extreme Programming, USFCA.edu, lecture paper at http://www.cs.usfca.edu/~parrt/course/601/lectures/xp.html
13. Feathers, M.: Working Effectively with Legacy Code. Prentice Hall (2004)
14. Gelly, S., Silver, D.: Monte-Carlo Tree Search and Rapid Action Value Estimation in Computer Go. Artificial Intelligence 175, 1856–1875 (2011)
15. Human Centred Technology Workshop 2005 (2005), PDF webpage: ftp://ftp.informatics.sussex.ac.uk/pub/reports/csrp/csrp585.pdf
16. Chen, K.-Y.: The Development of the Multi-Broker Desktop Grid for Computer Games, National Chiao-Tung University, master thesis
17. Copeland, L.: Extreme Programming. Computerworld (December 2001) (retrieved January 11, 2011)
18. Lin, H.-H., Sun, D.-J., Wu, I.-C., Yen, S.-J.: The 2010 TAAI Computer-Game Tournaments. ICGA Journal (SCI) 34(1) (March 2011)
19. Liu, H.-Y., Wu, I.-C., Kang, H.-H., Liao, T.-F.: System Demonstration for Generic Game Development Framework. In: The 2012 Conference on Technologies and Applications of Artificial Intelligence (TAAI 2012), Tainan, Taiwan (November 2012)
20. Newkirk, J.W., Vorontsov, A.A.: Test-Driven Development in Microsoft .NET. Microsoft Press (2004)
21. Renlib, Renju – A Renju Editor, http://www.renju.se/renlib/
22. Smarr, L., Catlett, C.: Metacomputing. Communications of the ACM 35(6), 44–52 (1992)
23. Reenskaug, T.: Models-Views-Controllers (1979)
24. Reenskaug, T.: THING-MODEL-VIEW-EDITOR an Example from a planningsystem (1979)
25. Wu, I.-C., Chen, C.-P., Lin, P.-H., Huang, K.-C., Chen, L.-P., Sun, D.-J., Chan, Y.-C., Tsou, H.-Y.: A Volunteer-computing-based grid environment for Connect6 applications. In: IEEE Int. Conf. Comput. Sci. Eng., Vancouver, BC, Canada, August 29-31, pp. 110–117 (2009)
26. Wu, I.-C., Huang, D.-Y., Chang, H.-C.: Connect6. ICGA Journal 28(4), 234–242 (2006)

27. Wu, I.-C., Huang, D.-Y.: A New Family of *k*-in-a-Row Games. In: van den Herik, H.J., Hsu, S.-C., Hsu, T.-S., Donkers, H.H.L.M(J.) (eds.) ACG11. LNCS, vol. 4250, pp. 180–194. Springer, Heidelberg (2006)
28. Wu, I.-C., Lin, H.-H., Lin, P.-H., Sun, D.-J., Chan, Y.-C., Chen, B.-T.: Job-Level Proof-Number Search for Connect6. In: van den Herik, H.J., Iida, H., Plaat, A. (eds.) CG 2010. LNCS, vol. 6515, pp. 11–22. Springer, Heidelberg (2011)
29. Wu, I.-C., Lin, H.-H., Sun, D.-J., Kao, K.-Y., Lin, P.-H., Chan, Y.-C., Chen, B.-T.: Job-Level Proof-Number Search. Accepted by IEEE Transactions on Computational Intelligence and AI in Games (2012)
30. Wu, I.-C., Lin, P.-H.: NCTU6-Lite Wins Connect6 Tournament. ICGA Journal 31(4) (2008)
31. Wu, I.-C., Lin, Y.-S., Tsai, H.-T., Lin, P.-H.: The Man-Machine Connect6 Championship 2011. ICGA Journal (SCI) 34(2) (June 2011)
32. XtremWeb website, `http://www.xtremweb.net/`
33. Yang, J.-K., Su, T.C., Wu, I.-C.: TCGA 2012 Computer Game Tournament Report. Submitted to ICGA Journal (2012)
34. Yen, S.-J., Su, T.-C., Wu, I.-C.: The TCGA 2011 Computer-Games Tournament. ICGA Journal (SCI) 34(2) (June 2011)

The Art of the Chinese Dark Chess Program DIABLE

Shi-Jim Yen[1,*], Cheng-Wei Chou[1], Jr-Chang Chen[2], I-Chen Wu[3], and Kuo-Yuan Kao[4]

[1] Dept. of Computer Science and Information Engineering,
National Dong Hwa University, Taiwan
sjyen@mail.ndhu.edu.tw, d9721002@ems.ndhu.edu.tw
[2] Dept. of Applied Mathematics,
Chung Yuan Christian University, Taiwan
jcchen@cycu.edu.tw
[3] Dept. of Computer Science,
National Chiao Tung University, Hsinchu 30050, Taiwan
icwu@csie.nctu.edu.tw
[4] Dept. of Information Management,
National Penghu University, Taiwan
stone@npu.edu.tw

Abstract. DIABLE is a famous Chinese dark chess program, which won the Chinese dark chess tournaments in TAAI 2011, TCGA 2011, and TCGA2012 computer game tournaments. Chinese dark chess is an old and very popular game in Chinese culture sphere. This game is played with imperfect information. Most computer Chinese dark chess programs used alpha-beta search with chance nodes to deal with the imperfect information. DIABLE used a new nondeterministic Monte Carlo tree search model for Chinese dark chess. These tournament results show that the nondeterministic Monte Carlo tree search is promising for Chinese dark chess.

Keywords: Chinese dark chess, Monte Carlo tree search, Chance node, Partially observable games, Nondeterministic action.

1 Introduction

DIABLE is a famous Chinese dark chess program, which won many Chinese dark chess tournaments. 72223 Chinese dark chess, similar to Chinese chess, is a popular game played in oriental countries. Chinese dark chess is a two-player *partially observable* game. Most Chinese dark chess programs use the alpha-beta search with *chance nodes* 5, which usually explore only shallow levels because the chance nodes increase the branching factor massively. On the other hand, recent studies have shown that Monte Carlo Tree Search (MCTS) is an efficient algorithm not only for *complete information* games such as Go and Amazons 91113, but also for some *incomplete information* games such as Poker, Kriegspiel, Backgammon and Phantom Go 28151.

DIABLE extends MCTS, by incorporating the *chance node* concept, to a *nondeterministic* search model that deals with partially observable information, and

[*] Corresponding author.

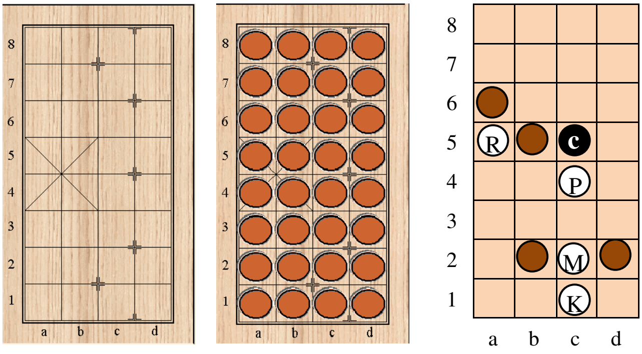
R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 231–242.
DOI: 10.1007/978-3-642-35452-6_25

apply the new model to Chinese dark chess. During the simulation, MCTS may suffer from repetition situations which are not forbidden in Chinese dark chess. We also propose two concepts – "*shorter simulation is better*" and "*prediction of game length*" to solve this problem.

The structure of this paper is organized as follows: Section 2 describes the rules of Chinese dark chess briefly. Section 3 gives an overview of the alpha-beta search with chance nodes in Chinese dark chess. Section 4 presents our methods. Section 5 demonstrates the experimental results and Section 6 makes conclusion.

2 The Rules of Chinese Dark Chess

The kinds and quantities of pieces used in Chinese dark chess are the same as those used in Chinese chess. One player, called *Red*, owns sixteen red pieces, and the other, called *Black*, owns sixteen black pieces. The sixteen pieces for each player include one King (K/k), two Guards (G/g), two Ministers (M/m), two Rooks (R/r), two Knights (N/n), two Cannons (C/c), and five Pawns (P/p), where, in the parentheses, the upper-cased letters are for Red and the lower-cased letters are for Black. Table 1 shows the pieces of Chinese dark chess. The board size of Chinese dark chess is 4×8 squares, as shown in Figure 1(a). All the thirty-two pieces are placed face-down at the beginning of a game, as shown in Figure 1(b), and thus their states are all hidden (or face-down).



(a) (b) (c)

Fig. 1. (a) The board. (b) The initial state. (c) An example of the Cannon's ability.

Two kinds of actions are allowed in the rule of Chinese dark chess: *flipping actions* and *moving actions*. Flipping actions are for *hidden* (*face-down*) pieces, whose states are hidden, e.g., b5 in Figure 1(c), and moving actions are for *observable* (*face-up*) pieces, whose states are observable, e.g., a5 in Figure 1(c). Each flipping action reveals a piece from face-down to face-up at the same location on the board, and the type of the piece is determined and known by both players. Thus, a flipping action is a *nondeterministic action*. By examining all pieces that has been flipped, we can find the *probability distribution* of the value of a piece in the hidden state. A player can only move pieces of his own color. Any piece *p* other than Cannon can be moved one empty square up, down, left or right as long as *p* stays within the board. Assume that there exists one opponent piece *p*' at the target square. The piece *p* can be moved to the targeted square to capture *p*' if *p* has equal or higher rank than *p*, and not otherwise. The ranks of piece types are shown in Table 1. One exception is that Pawn pieces can capture the opponent's King piece, but not vice versa. A moving action is a *deterministic action*.

Table 1. The pieces of Chinese dark chess

Type of Pieces	King	Guard	Minister	Rook	Knight	Cannon	Pawn
Number	1	2	2	2	2	2	5
Red type	***K***	***G***	***M***	***R***	***N***	***C***	***P***
Chinese icon	帥	仕	相	俥	傌	炮	兵
English icon	Ⓚ	Ⓖ	Ⓜ	Ⓡ	Ⓝ	Ⓒ	Ⓟ
Black type	***k***	***g***	***m***	***r***	***n***	***c***	***p***
Chinese icon	將	士	象	車	馬	砲	卒
English icon	ⓚ	ⓖ	ⓜ	ⓡ	ⓝ	ⓒ	ⓟ
Rank	1(Highest)	2	3	4	5	6	7(Lowest)

The legal actions for Cannon's are different from the ones for other pieces. A Cannon can be moved to an adjacent empty square in the same way as the actions for the other pieces, but cannot capture an adjacent opponent's piece. The only way for a Cannon to capture an opponent's piece is to move the Cannon jumping over a *carriage* and then stay on the square with the opponent's piece to be captured. The carriage can be a piece of any kind and any color, but subject to three constraints: (1) the Cannon, the carriage and the captured piece must stay at the same row (or column), (2) there are no pieces on the same row (or column) between the Cannon and the carriage, and (3) there are no pieces on the same row (or column) between the carriage and the piece been captured. This moving action is called to be *one-step-jump*. In Figure 2, the Black Cannon at c5 can move to up or right squares, but cannot capture the Red Pawn at c4. In contrast, the Black Cannon at c5 can capture either the Red Minister at c2 or the Red Rook at a5.

The color each player holds is decided by the piece first flipped by the first player. If the first flipped piece's color is red, the first player is Red and the second player is Black, and vice versa. A player wins when either the opponent has no legal move or the opponent's pieces are all captured. When neither player captures or flips a piece within forty plies, the game ends with a draw. The notation used to record moves (actions) is the same as the algebraic chess notation in 5. A moving action is recorded as *S*-*D*, where *S* stands for the source location and *D* stands for the destination location. A flipping action on a face-down piece is recorded as *P*(*T*), where *P* is the location of the face-down piece and *T* is the revealed type of the piece.

The state space complexity and the game tree complexity of Chinese dark chess were estimated to be 10^{37} and 10^{135} in 5, respectively. The game tree complexity is smaller than that of Chinese chess. However, since Chinese dark chess is a partially observable game, *chance nodes* should be added to the game tree. The complexity of a game tree with chance node for the flipping action is 10^{135}×32!×14^{32}, or roughly 10^{207}, which is bigger than that of Chinese chess and is close to that of Shogi. Furthermore, in the beginning stage of a Chinese dark chess game, the branching factor of chance nodes is huge. It is 32×14 (=448) for the first move played by the first player. This number is even bigger than that of Go. Therefore, the alpha-beta search with chance nodes cannot search deeply in Chinese dark chess programs.

3 Nondeterministic Monte Carlo Tree Search

In general, *Monte Carlo tree search* (MCTS) has four stages: *selection*, *expansion*, *simulation*, and *backpropagation* 4. This paper combines expansion into selection stage. Since Chinese dark chess is a partially observable game, we extend MCTS and Expectiminimax game tree concepts to *nondeterministic MCTS*. In contrast to all deterministic actions in Figure 2, we propose the *nondeterministic Monte Carlo tree search* model, since there are both deterministic actions (moving actions) and nondeterministic actions (flipping actions) in Chinese dark chess as illustrated in Figure 5. There are two kinds of nodes in the nondeterministic search tree: *deterministic state nodes* and *nondeterministic state nodes*. A nondeterministic state node is the big node which contains one or more than one *inner nodes*. The inner nodes represent the possible *temporary states* of the nondeterministic node. Each inner node has its own *visit count*, *win count*, and *sub tree*. Both visit and win counts of a nondeterministic node are the sums of visit counts and win counts of all its inner nodes, respectively. For the example in Figure 2, the leftmost nondeterministic state node in level 3 contains two inner nodes, represented by $\boldsymbol{k}^{\mathbf{1}}$ and $\boldsymbol{g}^{\mathbf{0}}$. It means that this nondeterministic node will be either Black King or Black Guard after it is flipped, and these inner nodes have been visited one and zero times, respectively. The nondeterministic state node can deal with the imperfect information of Chinese dark chess very well. The way to process deterministic state nodes in nondeterministic MCTS is the same as in normal MCTS. The following subsections will discuss the four stages of the nondeterministic MCTS.

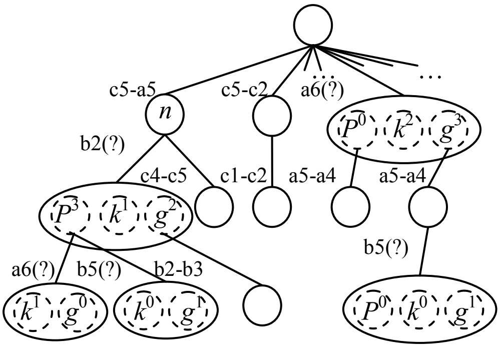


Fig. 2. The nondeterministic MCTS tree for the board in Figure 1(c). The possible types of the face-down pieces are ***P***, ***k*** and ***g***.

3.1 Selection and Expansion

In this stage, the strategy UCT 11 is used as formula (1), where v_i is the win rate of node i, N is the visited number of the parent node of node i, n_i is the visited number of node i, and *bias* is a constant, which is used to balance exploration and exploitation. We combine the expansion stage into the selection stage; a *leaf node* is either the first visited deterministic node or the first visited inner node. Initially, each created and non-visited node is assigned a large random UCT value. 11 The goal of selection is to find an action which maximizes formula (1) in every level of the game tree, and obtains the best sequence from the root to the leaf node.

$$v_i + bias \times \sqrt{\frac{\ln N}{n_i}} \ldots\ldots \qquad (1)$$

When the selected node is a nondeterministic node, we will choose a *temporary state node* from the inner nodes of the nondeterministic node by the *roulette wheel selection*, based on the probability distribution of the possible result states of the nondeterministic state node; the temporary state represents the current state of the selected nondeterministic node until the backpropagation stage is finished. If the temporary state node is a leaf node, the selection process stops, then continue to the simulation stage. Otherwise, it continues selecting the best child node of the temporary state node.

For the example in Figure 3(a), which is the early stage of the tree in Figure 2, node n is the leftmost node in the first level of the tree. Suppose the flipping action b2(?) is selected. Assume the possible types of the face-down pieces on the board of Figure 1(c) are ***P***, ***k***, and ***g***, the probabilities of each piece type at b2 are 1/4, 1/4 and 1/2 w.r.t. ***P***, ***k*** and ***g***. Assume that the "***P***" is chosen as the temporary state node, after the roulette wheel selection. Because the temporary state node $\boldsymbol{P}^{\mathbf{0}}$ is a leaf node, the selection process stops. Figure 4(b) shows the tree after the backpropagation stage. In Figure 4, assume that node n selects b2(?) action and then chooses k. Since the temporary state node k^0 is a leaf node, the selection process stops, and Figure 4(b) shows the tree after the backpropagation stage.

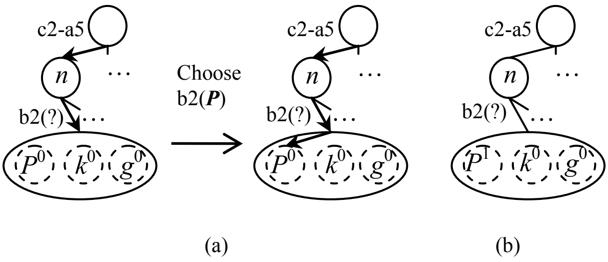


Fig. 3. (a) The early stage of the tree in Figure 2. (b) The tree after the backpropagation.

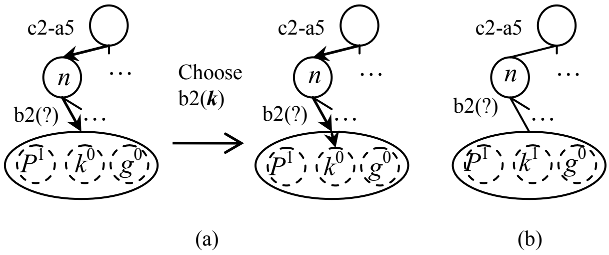


Fig. 4. (a) The tree after Figure 3. (b) The tree after the backpropagation.

In Figure 5, assume that node *n* selects b2(?) action and then chooses ***P*** again. Since $\boldsymbol{P}^{\mathbf{1}}$ is not a leaf node ($\boldsymbol{P}^{\mathbf{1}}$ has been visited once), it continues selecting the best child node of $\boldsymbol{P}^{\mathbf{1}}$. Assume the a6(?) is selected and ***k*** is chosen as the temporary state. Node $\boldsymbol{k}^{\mathbf{0}}$ is a leaf node and the selection process stops. The last tree in Figure 5 is the tree after the backpropagation.

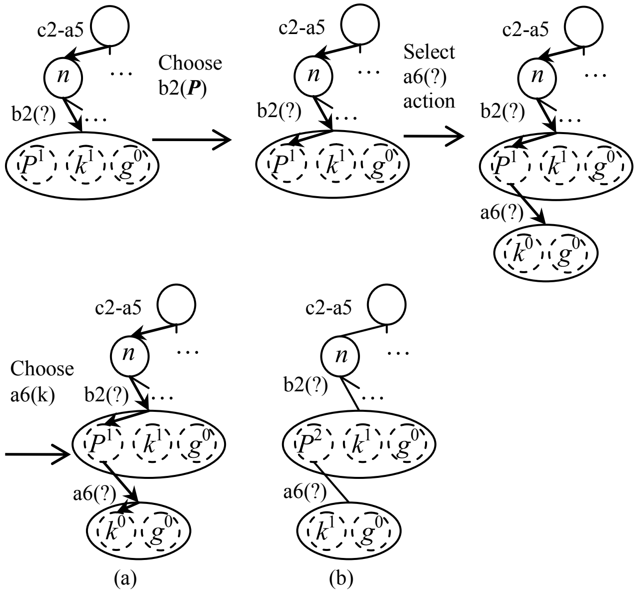


Fig. 5. (a) The trees after Figure 7. (b) The tree after the backpropagation.

3.2 Simulation

This stage will play the game from the position of the leaf node to the end, and obtain the simulation result, which is a win, draw or lose in Chinese dark chess. The result will be used in the backpropagation stage, as described in Section 4.3. The basic *default policy* of simulation only uses the "*capture-first*" heuristic (abbr. CF), that is

"*If there is some* ***capturing move*** *to capture the opponent's piece, choose the move. Otherwise, we randomly select a legal action*". The legal action includes moving action and flipping action. This paper proposes two other heuristics as follows.

The "*capture stronger piece first*" heuristic (abbr. CSPF) gives each capturing move a weight. If there is more than one capturing move, the heuristic will assign a probability to each of these capturing moves. The probability distribution is based on the scores of the pieces which are under the capturing threats. The scores of piece types are shown in Table 2. For example, if Black can only capture Red King and Guard, the probabilities of capturing moves, Red King and Guard, are (5500 / (5500 + 5000), 5000 / (5500 + 5000)), respectively. If there is no capturing move, we randomly select a legal action. Another heuristic is the "*capture and escape stronger piece first*" heuristic (abbr. CESPF), which considers both capturing and escaping moves. The probability distribution is also based on the score of the related pieces, as in Table 2.

Table 2. The scores of piece kinds

King	Guard	Minister	Rook	Knight	Cannon	Pawn
5500	5000	2500	1000	800	3000	800

3.3 Backpropagation

The result of a simulation will be used to update all nodes of the best sequence selected in the selection stage. If the result is not a draw, then the win counts of *winning nodes* in the best sequence are all incremented by one. If the result is a draw, then the win counts of all nodes in the best sequence are all incremented by 0.5. When the visit count and win count of a temporary inner node are updated, the counts of the nondeterministic node which contains the temporary inner node are also updated.

The game ends with a draw if neither player captures nor flips a piece within 40 plies in Chinese dark chess. Thus, the position with shorter simulation has higher probability to avoid a draw. For the example in Figure 6, the simulation of the left position may be shorter than that of the right position, and the left position has higher probability to win over draw.

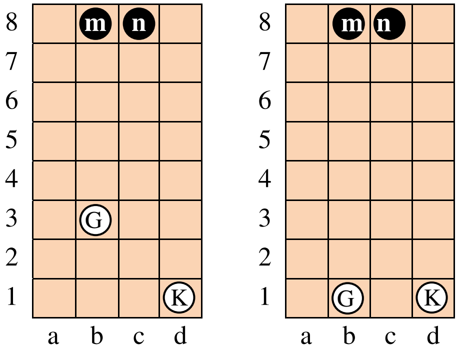


Fig. 6. Tow possible endgame positions

In computer Go, the winning position with shorter simulation means a simpler way to win. The Go program FUEGO modified the win/loss evaluation by small bonuses that favor shorter simulations and terminal positions with larger scores 9. In Chinese dark chess, choosing the position with shorter simulation can decrease the draw rate. Though the purpose is different, we modify the concept "*shorter simulation is better*" to avoid playing inefficient moves in winning situations. Thus, if the game situation is favoring our side, the rate draw decreases. Originally, when the best sequence was updated, the win count of the winning nodes will increment by one. With the shorter simulation heuristic, we give a winning and shorter simulation bonus as follows:

$$\boldsymbol{w} = \boldsymbol{w} + \mathbf{1} + \boldsymbol{d} \times(\boldsymbol{PGL} - \boldsymbol{S}) \ldots\ldots \qquad (2)$$

In the formula, w is the win count of the current node, S is the length of the simulation, and d is a small positive constant which is used to control the influence of this bonus. *Prediction of Game Length* (abbr. *PGL*) is a baseline of the game length, which is used to determine the length of this simulation. The value of PGL is obtained by simulating 500 times and averaging the lengths of these simulations before starting MCTS.

4 Experimental Results

In our experiments, we used as a *baseline player* program, developed by B.-N. Chen and T.-S. Hsu as mentioned in 5. It used the game tree search algorithm and the chance node with flipping heuristic. For the experiments, we used an Intel Core i7 920 CPU 2.4 GHz with 8GB memory, and with Windows 7. All of our experiments used only one core. Subsection 5.1 shows the experiment of changing the bias of formula 1 in Subsection 4.1. Both Subsections 5.2 and 5.3 demonstrate the concept of "*shorter simulation is better*" and the three simulation heuristics, respectively. Subsection 5.4 shows the scalability of our program. Section 5.5 is the competition results of our program in computer Chinese dark chess tournaments.

4.1 MCTS Parameter

The bias in the formula 1 of section 4.1 is to balance exploitation and exploration. In this part, our program played 400 games, 10000 simulations per move with the baseline player. Table 3 shows the results of changing the bias value with the balancing point 0.75.

Table 3. The results of changing the bias

Bias	Win	Loss	Draw
1.5	35.75%	18.50%	45.75%
1.25	38.50%	16%	45.50%
1	40.25%	15.25%	44.50%
0.75	**45.25%**	**13%**	**41.50%**
0.5	41.25%	15%	43.50%

4.2 Shorter Simulation is Better

In this part, the experiments are performed by changing the influence value *d* in formula (2) in Subsection 4.2. The times of games and simulations per move are the same as the setting in Subsection 5.1. The experimental results of changing the value *d* in Table 4 show this concept is useful in Chinese dark chess. First, the average length of games becomes shorter with the increase of *d*. This demonstrates that this concept can avoid inefficient moves. Second, the performance with $d = 0.01$ outperforms those with $d = 0$, in terms of the win and loss rates. This shows this concept actually improves the strength of the Chinese dark chess program. Finally, the optimal value of *d* is 0.01. When *d* is larger than 0.01, the program is forced to select the shorter game length and ignores the result of the game.

Table 4. The result of changing the *d* value

influence value *d*	Average length of games	Win	Loss	Draw
0	178.46	45.25%	13%	41.5%
0.001	166.63	47.5%	9.5%	43%
0.005	156.71	49.75%	9.75%	40.5%
0.01	**150.82**	**53.75%**	**7.75%**	**38.5%**
0.015	149.485	45.25%	10.25%	44.5%
0.03	145.38	34.5%	16.25%	49.25%

4.3 Simulation Heuristics

Table 5 lists the comparison of the three heuristics, called CF, CSPF and CESPF in Subsection 4.3. The number of simulations is 10,000, and “shorter simulation is better” heuristic is used. There are 400 games; our program played 200 games as being the first and the second players, respectively. The result of CESPF heuristic is the best. The loss rate of CESPF is 4.75% only. Table 5 also shows another interesting result: it favors the second player in Chinese dark chess. It is because the only choice for the first player in the beginning is a flipping action, which makes the second player have more information than the first player.

Table 5. Comparison of the three heuristics in simulation

	Turn	Win		Loss		Draw	
CF	First	108	54.00%	15	7.50%	77	38.50%
	Second	107	53.50%	16	8.00%	77	38.50%
	Total	215	53.75%	31	7.75%	154	38.50%
CSPF	First	105	52.50%	14	7.00%	81	40.50%
	Second	108	54.00%	15	7.50%	77	38.50%
	Total	213	53.25%	29	7.25%	158	39.50%
CESPF	First	120	60.00%	9	4.50%	71	35.50%
	Second	126	63.00%	10	5.00%	64	32.00%
	Total	**246**	**61.50%**	**19**	**4.75%**	**135**	**33.75%**

4.4 Scalability

Table 6 show the scalability of our program. As most MCTS programs, the win rate of the program grows as the number of simulations increases, but reaches a plateau after simulating 60,000 times. When the number of simulations is larger than 10000, the loss rate is below 5%. Then more simulations increase the winning rate and decrease the draw rate slightly. When the number of simulation is 240,000, it takes 30 seconds per move and the loss rate is only 1%. Our program with this setting is used to compare the three versions of the baseline program, named VAR1, VAR2 and VAR3 in 5, which are designed based on Alpha-Beta with three different flipping policies. Table 6 is the experimental results, and our program outperforms all the three versions. Compared to the results in Table 6, our version clearly decreases the draw rate and increases the win rate of the games.

Table 6. Scalability of our program

Simulations	Sec. / move	Win	Loss	Draw
3000	0.4	38%	16. 5%	45.5%
10000	1.25	61.5%	4.75%	32%
20000	2.5	73.25%	3%	27.5%
60000	7.5	81%	1.75%	17.25%
240000	**30**	**83.5%**	**1%**	**15. 5%**

Table 7. The experimental results of the three versions in 5

Version	Sec. / move	games	Win	Lose	Draw
VAR1	30	200	59.50%	1.00%	39.50%
VAR2	30	200	50.00%	0.50%	49.50%
VAR3	30	200	44.50%	0.50%	55.00%

4.5 Tournament Results

Table 8 shows the performance of our program in three computer Chinese dark chess tournaments, TCGA 2011, TAAI 2011 and TCGA 2012. 62222 Our program DIABLO, which used the methods proposed in Section 4, with 500,000 simulations per move, lost only two games in the three tournaments. Most of the opponent programs were state-of-the-art Chinese dark chess programs and designed by the methods described as in Section 3. These tournament results show that the methods proposed in this paper are promising for the development of Chinese dark chess.

Table 8. The results of Diablo in TCGA 2011, TAAI 2011 and TCGA 2012 computer Chinese dark chess tournaments

	Players	Win	Loss	Draw	Rank
TCGA 2011	7	7	0	5	1
TAAI 2011	4	4	0	2	1
TCGA 2012	6	7	2	1	1
Total	17	18(64%)	2(7%)	8(29%)	

5 Conclusion

DIABLE applies the nondeterministic MCTS for Chinese dark chess. This method can deal with the flipping/ nondeterministic actions very well. The proposed tree policy heuristic "shorter simulation is better" can decrease draw rate and increase the win rate. We also study three default policy heuristics and compare them. The experimental results show that the modified MCTS and proposed heuristics are promising for Chinese dark chess. Moreover, DIABLO won all the gold medals in computer Chinese dark chess tournaments in TAAI 2011, TCGA 2011 and TCGA 2012 computer game tournaments.

In the future, we plan to add more domain knowledge to improve the strength of our Chinese dark chess program. It is also worthy further studying to apply the nondeterministic MCTS to other partially observable games, such as Mahjong and card games.

References

1. Borsboom, J., Saito, J., Chaslot, G., Uiterwijk, J.: A comparison of Monte-Carlo methods for Phantom Go. In: Proc. 19th Belgian–Dutch Conference on Artificial Intelligence – BNAIC, Utrecht, The Netherlands (2007)
2. Van den Broeck, G., Driessens, K., Ramon, J.: Monte-Carlo Tree Search in Poker Using Expected Reward Distributions. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS (LNAI), vol. 5828, pp. 367–381. Springer, Heidelberg (2009)
3. Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A survey of Monte Carlo Tree Search. IEEE Transactions on Computational Intelligence and AI in Games 4(1) (March 2012)
4. Chaslot, G.M.J.-B., Winands, M.H.M., Uiterwijk, J.W.H.M., van den Herik, H.J., Bouzy, B.: Progressive strategies for Monte-Carlo Tree Search. New Mathematics and Natural Computation 4(3), 343–357 (2008)
5. Chen, B.N., Shen, B.-J., Hsu, T.S.: Chinese Dark Chess. ICGA Journal 33(2), 93–106 (2010)
6. Chen, J.-C., Lin, T.-Y., Hsu, S.-C., Hsu, T.-S.: Design and Implementation of Computer Chinese Dark Chess Endgame Database. In: Proceeding of TCGA Workshop 2012, Hualien, Taiwan, pp. 5–9 (2012)
7. Chou, C.-W., Yen, S.-J., Wu, I.-C.: TAAI 2011 Computer Go Tournaments. ICGA Journal 34(4), 251–252 (2011)
8. Ciancarini, P., Favini, G.P.: Monte Carlo tree search in Kriegspiel. Artificial Intelligence 174, 670–684 (2010)
9. Enzenberger, M., Müller, M., Arneson, B., Segal, R.: Fuego - An Open-Source Framework for Board Games and Go Engine Based on Monte Carlo Tree Search. IEEE Transactions on Computational Intelligence and AI in Games 2(4), 259–270 (2010)
10. Gelly, S., Wang, Y., Munos, R., Teytaud, O.: Modification of UCT with patterns in Monte-Carlo Go (Technical Report 6062). INRIA (2006)
11. Gelly, S., Silver, D.: Monte-Carlo tree search and rapid action value estimation in computer Go. Artificial Intelligence 175(11), 1856–1875 (2011)

12. Van den Broeck, G., Driessens, K., Ramon, J.: Monte-Carlo Tree Search in Poker Using Expected Reward Distributions. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS, vol. 5828, pp. 367–381. Springer, Heidelberg (2009)
13. Kloetzer, J., Iida, H., Bouzy, B.: The Monte-Carlo Approach in Amazons. In: Proceedings of the Computer Games Workshop 2007 (CGW 2007), Amsterdam, The Netherlands, pp. 185–192 (2007)
14. Kocsis, L., Szepesvari, C.: Bandit based Monte-Carlo planning. In: 15th European Conference on Machine Learning, pp. 282–293 (2006)
15. Lai, S.-C.: Research and Implementation of Computer Dark Chess Program with Heuristics, National Dong Hwa University, Matser thesis (2008) (in Chinese)
16. Van Lishout, F., Chaslot, G., Uiterwijk, J.: Monte-Carlo Tree Search in Backgammon. In: Proceedings of the Computer Games Workshop 2007 (CGW 2007), Amsterdam, The Netherlands, pp. 175–184 (2007)
17. Lou, W.-C.: Artificial Intelligence Improvement of Chinese Dark Chess, National Taiwan Normal University, Master thesis (2011) (in Chinese)
18. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall (2003)
19. Shieh, Y.-A.: The Design and Implementation of Computer Dark Chess, National Taiwan Normal University, Master thesis (2008) (in Chinese)
20. Shih, H.-C., Lin, S.-S.: The Design and Implementation of Computer Dark Chess. In: Proceeding of TCGA Workshop 2012, Hualien, Taiwan, pp. 29–36 (2012) (in Chinese)
21. Veness, J., Ng, K.S., Hutter, M., Uther, W., Silver, D.: A Monte-Carlo AIXI Approximation. J. Artif. Intell. Res. 40, 95–142 (2011)
22. Yang, J.-K., Su, T.-C., Wu, I.-C.: TCGA 2012 Computer Game Tournament Report. Submitted to ICGA Journal (2012)
23. Yen, S.-J., Su, T.-C., Wu, I.-C.: The TCGA 2011 Computer-Games Tournament. ICGA Journal 34(2), 108–110 (2011)
24. Yen, S.-J., Chiu, S.-Y., Wu, I.-C.: Modark Wins Chinese Dark Chess Tournament. ICGA Journal 33(4), 230–231 (2010)

Mining High Utility Itemsets Based on the Pre-large Concept

Chun-Wei Lin[1], Tzung-Pei Hong[2,3,*], Guo-Cheng Lan[4], Jia-Wei Wong[3], and Wen-Yang Lin[2]

[1] Innovative Information Industry Research Center (IIIRC)
School of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus Shenzhen University Town, Xili, Shenzhen 518055 P.R. China
jerrylin@ieee.org
[2] Department of Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung, 811, Taiwan, R.O.C.
{tphong,wylin}@nuk.edu.tw
[3] Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, 804, Taiwan, R.O.C.
jwwong.alex@gmail.com
[4] Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, 701, Taiwan, R.O.C.
rrfoheiay@gmail.com

Abstract. In this paper, an incremental mining algorithm is proposed for efficiently maintaining the discovered high utility itemsets based on the pre-large concept. It first partitions itemsets into nine cases according to whether they are large (high), pre-large or small transaction-weighted utilization in the original database and in the inserted transactions. Each part is then performed by its own procedure. Experimental results also show that the designed incremental high utility mining algorithm has better performance than the bach one for handling inserted transactions.

Keywords: Utility mining, pre-large itemset, high utility itemset, incremental mining, two-phase approach.

1 Introduction

In conventional data mining techniques, mining association rules [1, 2, 4] is the most popular approach among data mining techniques. In association rule mining, each item is treated as a binary variable for discovering interesting relationships between

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 243–250.
DOI: 10.1007/978-3-642-35452-6_26

itemsets. The frequency of an itemset, however, is insufficient for identifying highly profitable itemsets.

Utility mining [3, 13] was thus proposed to improve the above limitations of frequent itemsets. It may be thought of as frequent-itemset mining with sold quantities and item profits considered as 1. In practice, the utility value of an itemset can be labeled as cost, profit, or some other user-defined term. Liu et al. designed a two-phase algorithm for efficiently extracting high utility itemsets based on the downward closure property [11].

In the above approaches, the database is assumed to be static and the mining processes are performed in batch mode. In real-world applications, transactions may be inserted into the database. Hong et al. proposed pre-large itemsets to solve these problems [6-7]. Upper and lower thresholds are used to define large and pre-large itemsets and induced a safety number of new transactions to reduce the number required database rescans. In this paper, an incremental mining algorithm based on pre-large concepts to update the discovered high utility itemsets is thus proposed. Based on the proposed algorithm, only a little amount of itemsets is required to rescan the original database for maintaining the high utility itemsets.

2 Related Work

Traditional data mining is used to extract useful itemsets or rules from binary database [1, 2, 4, 9]. In real-world applications, transaction database usually grow over time and time and the procedure for mining association rules is performed in a batch way. Cheung *et al.* thus proposed the Fast UPdated algorithm (abbreviated as FUP) [5] to effectively handle transaction insertion for maintaining the frequent itemsets. Hong *et al.* proposed the pre-large concepts for efficiently maintaining the discovered rules in incremental data mining [6-7]. The pre-large algorithm is unnecessary to rescan the original database until a number of new transactions have been inserted. Since rescanning database spent much computational time, the maintenance cost could thus be reduced.

Utility mining [3, 13] is an extension of frequent itemset mining based on the measurement of local transaction utility and external utility. Yao *et al.* proposed an algorithm for efficiently mining high utility itemsets [12]. Liu *et al.* proposed a two-phase algorithm for efficiently discovering high utility itemsets [11] based on downward closure property. The two-phase algorithm consists of two phases to level-wisely generate-and-test high utility itemsets. In the first phase, the transaction utility was used as an effective upper bound of each candidate itemset in the transactions according to the downward closure property of transaction-weighted utilization. In the second phase, an additional database scan was then performed to find the real utility values of the remaining candidate itemsets for discovering high utility itemsets.

3 The Proposed Incremental High Utility Mining Algorithm

When new transactions are inserted into the original database, the proposed incremental algorithm is performed to update the discovered high utility itemsets. The candidate 1-itemsets in the new transactions are firstly derived with their transaction-weighted utilizations [11] and their actual utility values. Based on the two utility thresholds [6-7], the generated candidate 1-itemsets in the new transactions can be divided into three cases according to whether they have large, pre-large, or small transaction-weighted utilization in the original database. Individual procedures are then applied to each part to maintain and update the candidate 1-itemsets for generating high transaction-weighted utilization 1-itemsets in the updated database. The candidate 2-itemsets are then formed from the remaining large (high) and pre-large transaction-weighted utilization 1-itemsets. The same procedure is performed to find the candidate 3-itemsets. This procedure is repeated level by level until all high utility itemsets have been maintained and updated. The detail of the proposed algorithm is shown below.

***The proposed algorithm*:**

INPUT: An original database D, the total utility TU^D of D, the large (high) transaction-weighted utilization itemsets $HTWU^D$ and the pre-large transaction-weighted utilization itemsets $PTWU^D$ with their transaction-weighted utilization values and actual utility values discovered from D, the safety transaction utility buffer *buf* for preserving the total utility value of the last processed transactions, a profit table P, an upper utility threshold S_u (the same as the minimum high utility threshold), a lower utility threshold S_l, and a set of new transactions d.

OUTPUT: A set of high utility itemsets (HU^U) for the updated database U ($D \cup d$).

STEP 1: Calculate the safety transaction utility bound f as:

$$f = \frac{S_u - S_l}{1 - S_u} \times TU^D.$$

STEP 2: Calculate the item utility value u_{kj} of each item i_j in each new transaction t_k as:

$$u_{kj} = q_{kj} \times p_j,$$

where q_{kj} is the quantity of i_j in t_k and p_j is the profit of i_j. Sum up the utility values of all the items in each transaction t_k as the transaction utility tu_k. Add all transaction utilities for all new transactions as the total utility of the new transactions, which is denoted as TU^d. They are then calculated as:

$$tu_k = \sum_{j=1}^{m} u_{kj} \text{ and } TU^d = \sum_{k=1}^{n} tu_k .$$

STEP 3: Calculate the total utility TU^U for the updated database as:

$$TU^U = TU^D + TU^d,$$

where TU^D is the total utility in the original database, and TU^d is the total utility in the new transactions.

STEP 4: Generate the candidate *1*-itemsets C_1 in the new transactions d.

STEP 5: Set $k = 1$, where k records the size of the itemsets currently being processed.

STEP 6: For each candidate k-itemset X in C_k, calculate the transaction-weighted utilization $twu^d(X)$ and the actual utility $au^d(X)$ respectively as:

$$twu^d(X) = \sum_{t_k \in d} \sum_{X \subseteq t_k} tu_k \text{ and } au^d(X) = \sum_{t_k \in d} \sum_{X \subseteq t_k} \sum_{i_j \in X} u_{kj},$$

where $twu^d(X)$ is the sum of the transaction utilities containing the itemset X in the new transactions d, and $au^d(X)$ is the sum of the item utilities containing itemset X in the new transactions d.

STEP 7: For each high (large) transaction-weighted utilization itemset in $HTWU_k^D$ in the original database, do the following substeps (**Cases 1**, **2**, and **3**):

Substep 7-1: Set the updated transaction weighted-utilization $twu^U(X)$ of itemset X in the entire updated database as:

$$twu^U(X) = twu^D(X) + twu^d(X),$$

where $twu^D(X)$ was kept in the set of $HTWU^D$ in the original database, and $twu^d(X)$ was calculated in STEP 6 in the new transactions.

Substep 7-2: Set the updated actual utility $au^U(X)$ of itemset X in the entire updated database as:

$$au^U(X) = au^D(X) + au^d(X),$$

where $au^D(X)$ was kept in the set of $HTWU^D$ in the original database, and $au^d(X)$ was calculated in STEP 6 in the new transactions.

Substep 7-3: If $\frac{twu^U(X)}{TU^U} \geq S_u$, put itemset X in the set of $HTWU_k^U$ as the large(high) transaction-weighted utilization k-itemset in the updated database;

Otherwise, if $S_l \leq \frac{twu^U(X)}{TU^U} < S_u$, put itemset X in the set of $PTWU_k^U$ as the pre-large transaction-weighted utilization k-itemset in the updated database; Otherwise, itemset X is small after the database is updated, neglect itemset X.

STEP 8: For each pre-large transaction-weighted utilization itemset in $PTWU_k^D$ in the original database, do the following substeps (**Cases 4**, **5**, and **6**):

Substep 8-1: Set the updated transaction weighted-utilization $twu^U(X)$ of itemset X in the entire updated database as:

$$twu^U(X) = twu^D(X) + twu^d(X),$$

where $twu^D(X)$ was kept in the set of $PTWU^D$ in the original database, and $twu^d(X)$ was calculated in STEP 6 in the new transactions.

Substep 8-2: Set the updated actual utility $au^U(X)$ of itemset X in the entire updated database as:

$$au^U(X) = au^D(X) + au^d(X),$$

where $au^D(X)$ was kept in the set of $PTWU^D$ in the original database, and $au^d(X)$ was calculated in STEP 6 in the new transactions.

Substep 8-3: If $\frac{twu^U(X)}{TU^U} \geq S_u$, put itemset X in the set of $HTWU_k^U$ as the large(high) transaction-weighted utilization k-itemset in the updated database;

Otherwise, if $S_l \leq \frac{twu^U(X)}{TU^U} < S_u$, put itemset X in the set of $PTWU_k^U$ as the pre-large transaction-weighted utilization k-itemset in the updated database; Otherwise, itemset X is small after the database is updated, neglect itemset X.

STEP 9: For each candidate k-itemset X in C_k from STEP 6, which is not appearing both in the set of $HTWU_k^D$ and $PTWU_k^D$, but its $\frac{twu^d(X)}{TU^d} \geq S_l$, put candidate k-itemset X in the set of *Rescan_Items* which is used when rescanning original database in STEP 10 is required.

STEP 10: If $(buf + TU^d) \leq f$ or the set of *Rescan_Items* is ***null***, nothing is done in this STEP;

Otherwise, do the following substeps for each candidate k-itemset X in the set of *Rescan_Items*:

Substep10-1: Rescan the original database to calculate the transaction-weighted utilization $twu^D(X)$ and the actual utility $au^D(X)$ respectively as:

$$twu^D(X) = \sum_{t_k \in D} \sum_{X \subseteq t_k} tu_k \text{ and } au^D(X) = \sum_{t_k \in D} \sum_{X \subseteq t_k} \sum_{i_j \in X} u_{kj},$$

where $twu^D(X)$ is the sum of the transaction utilities containing the itemset X in the original database D, and $au^D(X)$ is the sum of the item utilities containing itemset X in the original database D.

Substep 10-2: Set the updated transaction weighted-utilization $twu^U(X)$ of itemset X in the entire updated database as:

$$twu^U(X) = twu^D(X) + twu^d(X).$$

Substep 10-3: Set the updated actual utility $au^U(X)$ of itemset X in the entire updated database as:

$$au^U(X) = au^D(X) + au^d(X).$$

Substep 10-4: If $\frac{twu^{U}(X)}{TU^{U}} \geq S_u$, put itemset *X* in the set of $HTWU_k^U$ as the large(high) transaction-weighted utilization *k*-itemset in the updated database; Otherwise, if $S_l \leq \frac{twu^{U}(X)}{TU^{U}} < S_u$, put itemset *X* in the set of $PTWU_k^U$ as the pre-large transaction-weighted utilization *k*-itemset in the updated database;
Otherwise, itemset *X* is small after the database is updated, neglect itemset *X*.

STEP 11: Form candidate (*k*+1)-itemsets C_{k+1} according to the large transaction-weighted utilization *k*-itemsets $HTWU_k^U$ and pre-large transaction-weighted utilization *k*-itemsets $PTWU_k^U$ ($HTWU_k^U \cup PTWU_k^U$) found in STEP 10.

STEP 12: Set $k = k + 1$.

STEP 13: Repeat Step 6 to 12 until no updated large (high) or pre-large transaction-weighted utilization itemsets are found.

STEP 14: Process each itemset *X* in the set of $HTWU^U$ and set of $PTWU^U$, if $\frac{au^{U}(X)}{TU^{U}} \geq S_u$, *X* is concerned as a high utility itemset. Put itemset *X* into the set of HU^U;
Otherwise, if $S_l \leq \frac{au^{U}(X)}{TU^{U}} < S_u$, *X* is concerned as a pre-large utility itemset. Put itemset *X* into the set of $PU^{U.}$.

STEP 15: If (buf + TU^d) > f, set TU^D = TU^U and $buf = 0$; Otherwise, set $buf = buf + TU^d$.

STEP 16: Set $HTWU^D$ = HU^U and $PTWU^D$ = PU^U as set of the updated large(high) transaction-weighted utilization and the set of pre-large transaction-weighted utilization, respectively for the last updated database.

After STEP 16, the final high utility itemsets are found from the set of HU^U in STEP 14, and the proposed algorithm is then terminated.

4 Experimental Results

Experiments were conducted to evaluate the performance of the two-phase (TP-HUI) algorithm in a batch way [11], the incremental algorithm for record insertion based on FUP concepts (FUP-HUI) [10], and the proposed approach for transaction insertion based on pre-large concepts (PRE-HUI) in a simulation database. The IBM data generator [8] was used to generate a simulation database called T10I4N4KD200K. The 200,000 transactions were used to initially mine the large (high) and pre-large transaction-weighted utilization itemsets with their actual utility values. Each 2,000 transactions were then sequentially extracted from the original database bottom up, thus forming the new transactions for transaction insertion at each time. The minimum high utility threshold (upper utility threshold) was set at 0.2% to evaluate the performance of TP-HUI, FUP-HUI and as the upper utility threshold of PRE-HUI algorithms. The lower utility threshold was set at 0.17% for PRE-HUI algorithm. Figure 1 shows the execution times for the TP-HUI, FUP-HUI, and PRE-HUI algorithms.

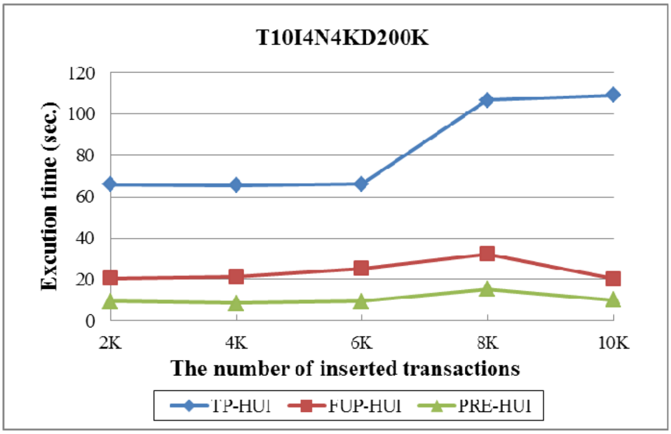


Fig. 1. The comparisons of the execution times for transaction insertion

It is obvious to see from Figure 1 that the proposed PRE-HUI needed less execution time than TP-HUI and FUP-HUI algorithms for handling transaction insertion at various numbers of inserted transactions.

5 Conclusions

In this paper, an incremental algorithm for efficiently mining high utility itemsets is proposed for transaction insertion based on pre-large concept. Experimental results are also shown that the proposed incremental high utility mining algorithm has a better performance rather than the batched two-phase approach and the high utility mining algorithm based on FUP concept.

References

[1] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: The International Conference on Very Large Data Bases, pp. 487–499 (1994)

[2] Berzal, F., Cubero, J.C., Marín, N., Serrano, J.M.: TBAR: An efficient method for association rule mining in relational databases. Data and Knowledge Engineering 37, 47–64 (2001)

[3] Chan, R., Yang, Q., Shen, Y.D.: Mining high utility itemsets. In: IEEE International Conference on Data Mining, pp. 19–26 (2003)

[4] Chen, M.S., Han, J., Yu, P.S.: Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering 8, 866–883 (1996)

[5] Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of discovered association rules in large databases: an incremental updating technique. In: The International Conference on Data Engineering, pp. 106–114 (1996)

[6] Hong, T.P., Wang, C.Y., Tao, Y.H.: A new incremental data mining algorithm using pre-large itemsets. Intelligent Data Analysis 5, 111–129 (2001)

[7] Hong, T.P., Wang, C.Y.: Maintenance of association rules using pre-large itemsets. In: Intelligent Databases: Technologies and Applications, pp. 44–60 (2006)

[8] IBM quest data mining project, Quest synthetic data generation code, `http://www.almaden.ibm.com/cs/quest/syndata.html`

[9] Lin, C.W., Hong, T.P., Lu, W.H.: The Pre-FUFP algorithm for incremental mining. Expert Systems with Applications 36, 9498–9505 (2009)

[10] Lin, C.W., Lan, G.C., Hong, T.P.: An incremental mining algorithm for high utility itemsets. Expert Systems with Applications 39, 7173–7180 (2012)

[11] Liu, Y., Liao, W.-K., Choudhary, A.K.: A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 689–695. Springer, Heidelberg (2005)

[12] Yao, H., Hamilton, H.J.: Mining itemset utilities from transaction databases. Data and Knowledge Engineering 59, 603–626 (2006)

[13] Yao, H., Hamilton, H.J., Butz, C.J.: A foundational approach to mining itemset utilities from databases. In: The SIAM International Conference on Data Mining, pp. 211–225 (2004)

Spatial Query Processing on Distributed Databases

Jiun-Wen Bai, Jun-Zhe Wang, and Jiun-Long Huang

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, ROC
{jwbai,jzwang,jlhuang}@cs.nctu.edu.tw

Abstract. In recent years, with the ubiquity of location-aware mobile devices and widespread deployment of wireless networks, location-based services (LBSs) have become popular rapidly. Spatial queries, one useful LBS, enable users to query about the interested data objects near them. Due to the rapid growth in spatial data, it is challenging to index data objects and answer spatial queries. In this paper, we propose novel index structures and companion algorithms to efficiently solve representative spatial queries, namely *k*NN and window queries, in the case of vast amounts of data objects. The proposed index structures are built on the top of the distributed database HBase and are separately designed according to the characteristics of *k*NN and window queries. With the index structures, we devise efficient *k*NN and window query processing algorithms to achieve fast query search. The experimental results show that the proposed algorithms and the index structures are effective and efficient in solving *k*NN and window queries. Moreover, the results also demonstrate the scalability of the proposed algorithms and index structures.

1 Introduction

In recent years, due to the proliferation of location-aware mobile devices and widespread deployment of WiFi and 3G networks, location-based services (LBSs) have been gaining popularity rapidly. Spatial queries, one useful LBS, allow users to query about nearby interested data objects with respect to their locations. The two most common spatial queries are *k* nearest neighbor (*k*NN) queries and window queries. A *k*NN query retrieves the *k* nearest objects with respect to the location of the query user, while a window query displays all the objects within the requested window centered at the query location. A real-world example of *k*NN queries is that "tell me the 3 nearest restaurants with respect to my current location"; a window query example is that "show all the coffee shops within the navigation frame centered at my location."

To answer *k*NN and window queries, the R-tree [1] and its variants (e.g., R*-tree [2]) are the most commonly used index structures for storing data objects due to the efficiency. However, the rapid growth of spatial data leads to a enormous amount of data objects that cannot be stored using one single storage disk. Thus, the R-tree and its variants cannot be employed as the index structure for such a significant number of data objects. In addition to the storage issue, spatial query processing on such a vast amount of data objects is time-consuming. In view of these two issues, it is necessary to maintain data objects in a distributed database. However, current distributed databases such as HBase [3] and Cassandra [4] are not designed for spatial query processing. As

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 251–260.
DOI: 10.1007/978-3-642-35452-6_27

such, it would suffer from poor performance when applying traditional spatial querying approaches to current distributed databases directly.

In this paper, we propose new index structures and companion algorithms to efficiently resolve *k*NN queries and window queries in the case of enormous amounts of data objects. The proposed index structures are built on the top of HBase, a representative distributed database. HBase is an open-source scalable database based on Google BigTable [5]. All operations on HBase are transformed into MapReduce [6] operations run on Hadoop [7]. Due to this feature, it is natural to achieve parallel processing with HBase. For *k*NN queries, we design a novel index structure, which mainly uses the one-dimensional sequence generated by applying Hilbert curve to two-dimensional data objects. With the index structure, we propose an efficient *k*NN query processing algorithm to retrieve the answer of *k*NN queries. On the other hand, for window queries, we introduce another index structure because of the distinct features of *k*NN and window queries. The index structure is constructed according to the distribution of the data objects. A companion algorithm is presented to address window queries with less computational cost. For performance evaluation, we conduct several experiments using the real dataset. The experimental results show that the proposed index structures and algorithms are efficient for *k*NN and window query processing. Moreover, the experimental results indicate that the proposed algorithms are scalable due to the high efficiency in large datasets.

The rest of this paper is organized as follows. Section 2 presents the preliminaries. In Section 3, we elaborate the spatial query processing. Experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

2 Preliminaries

2.1 HBase

HBase [3] is the open-source version of Google BigTable. It is mainly used to store enormous sparse data in a fault-tolerant way. HBase is the database component in Hadoop ecosystem and it uses Hadoop distributed file system (HDFS) as the storage back-end. Its operations would be transformed into Hadoop job to execute. To access the HBase, in addition to using the Java API [8], it is feasible to invoke Stargate[9], Avro[10], and Thrift[11] gateway APIs.

Different from traditional relational database systems, The data model of Base is a sparse, non-volatile, multi-dimensional, sorted hash table. The index structure is composed of three parts: row key, column, and timestamp. Given the values of the three entries, a unique value will be retrieved. Byte array is the only data type supported by HBase. When storing data into the table of HBase, the row key is determined and is associated with an arbitrary number of columns. Note that the columns corresponding to the same row key may be different. To retrieve the rows in the table of HBase, there are three ways listed below.

- Visit the row via row key directly.
- Visit the rows via the range of row keys.
- Scan the full table from the beginning of the table.

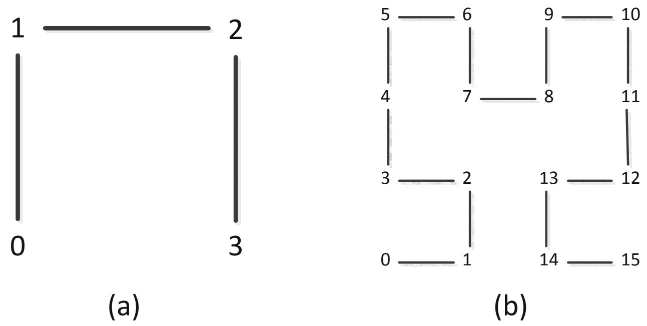


Fig. 1. An example of Hilbert curve

2.2 Hilbert Curve

As mentioned above, the data with the same row key are stored together and are sorted in a lexicographical order. However, the coordinates of spatial data objects are two-dimensional and thus cannot be directly stored at HBase. To address this issue, we exploit Hilbert curve to transform the two-dimensional data objects into one-dimensional data. The reason of employing Hilbert curve is that the transformed one-dimensional data are able to preserve the spatial locality of data objects [12]. The spatial locality means that data objects close to each other in the two-dimensional space remain close to each other after transforming into the one-dimensional space. The transformation is based on the a basic curve (depicted in Fig. 1(a)) and recursively using the same basic curve until all the data spaces are processed (as shown in Fig. 1(b)). In the following, we will utilize Hilbert curve to design the index structures for *k*NN queries.

3 Spatial Query Processing

In this section, we elaborate the proposed index structures and companion algorithms for *k*NN and window queries search. Since the proposed *k*NN query processing algorithm takes advantage of the window query processing algorithm, we first introduce window query processing , followed by the description of the *k*NN query processing.

3.1 Window Query Processing

Index Structure for Window Queries. To enable the efficient retrieval of the answer objects of a window query, we design an index structure on the top of HBase for window queries. To solve window queries, it is required to identify answer objects and filter irrelevant objects from the perspectives of x-axis and y-axis. Based on this observation, we use one of the coordinate components (x-coordinate and y-coordinate) as the row key for the index structure. With one coordinate component as the row key, the another components of the data objects with the same row key are stored into the corresponding column. Table 1 shows an example of using the x-coordinate as the row

Table 1. The index structure for window queries with x-coordinate as row key

Row Key	Timestamp	Column ”Y:”	
121.589341	t5	”Y:0”	”37.740023”
	t8	”Y:1”	”37.606252”
121.589346	t6	”Y:0”	”37.765364”
121.589348	t3	”Y:0”	”37.609897”
	t9	”Y:1”	”37.664402”

key for the data objects. From Table 1, we can see that the data objects with coordinates $(121.589341, 37.740023)$ and $(121.589341, 37.606252)$ are stored in the same column since they have the same x-coordinate. With the index design and the range scan of HBase, the window query resolving can be achieved in an efficient manner. Note that using the x-coordinate or y-coordinate as the row key depends on the distribution of the data objects. Specifically, if the data objects are spatially close in terms of y-coordinate, it would be more efficient to employ y-coordinate as the row key since the range scan on y-coordinate is able to filter more irrelevant data objects. On the other hand, adopting x-coordinate as the row key is more effective when the data objects are spatially close regarding x-coordinate.

Window Query Processing Algorithm. With the proposed index structure for window queries, we detail the window query processing algorithm here. When receiving a window query with query length l, query width w and query location (x,y), the algorithm first identifies the four vertices of the query window, which are $v_1(x-\frac{l}{2}, y-\frac{w}{2})$, $v_2(x-\frac{l}{2}, y+\frac{w}{2})$, $v_3(x+\frac{l}{2}, y+\frac{w}{2})$, $v_4(x+\frac{l}{2}, y-\frac{w}{2})$. Then, according to the distribution of data objects, the algorithm uses the x-coordinate range or y-coordinate range of the query window to perform range scan on the index structure. For all the row keys in the coordinate range, the data objects in the corresponding columns are retrieved and put into the candidate set *CandSet*. For each object in *CandSet*, the algorithm examines whether this object is inside the query window. If so, the object is an answer object and is added to the *Win* (the set of answer objects). Due to the effective index structure, the window query processing algorithm needs to evaluate only one coordinate of the retrieved data objects, thereby achieving efficient window query answering. The window query processing algorithm is given in Algorithm 1.

3.2 *k*NN Query Processing

Index Structure for *k*NN Queries. Due to the distinct characteristics of *k*NN and window queries, we design a new index structure for *k*NN query search. The idea is to take advantage of the aforementioned Hilbert curve to transform two-dimensional data objects into one-dimensional sequence. The transformed sequence number of an object serves as the row key and the x-coordinate and y-coordinate of the object are stored at the corresponding columns of the row key, as shown in Table 2. With such design, the data objects are able to be easily retrieved for *k*NN query solving. To realize the index structure, we partition the entire data space into disjoint $n \times n$ cells where n is a

Algorithm 1. Window query processing algorithm: $WindowQuery(x,y,l,w)$

```
Input:
    $x$ : x-coordinate of the query location
    $y$ : y-coordinate of the query location
    $l$ : window length
    $w$ : window width
Output:
    Answer objects $Win$
$Win \leftarrow \emptyset$
$CandSet \leftarrow$ all objects with x-coordinate between $x - \frac{l}{2}$ and $x + \frac{l}{2}$
for all $p \in CandSet$ do
    if $p_y$ between $y - \frac{w}{2}$ and $y + \frac{w}{2}$ then
        add $p$ to $Win$
    end if
end for
return $Win$
```

Table 2. The index structure based on Hilbert curve

Row Key	Column "X:"	Column "Y:"
0394080828547	"122.293898"	"37.89772"
0394081175471	"122.293728"	"37.897715"
0394081238797	"122.293505"	"37.897709"
0394082017744	"122.29372"	"37.897876"
0394082142912	"122.293696"	"37.898391"

power of 2. Note that n is set to an extremely large value in practice so that no more than one data object is contained in one cell. The cells are represented by two integers (c_x, c_y). The cell at the bottommost and leftmost is denoted by $(0,0)$ while the cell at the topmost and rightmost is denoted by $(n-1, n-1)$. After transform, cell $(0,0)$ will be represented by 0 and serves as the starting point of the Hilbert curve whereas cell $(n-1, n-1)$ will be denoted by $n^2 - 1$ and is the ending point. With the index structure, we introduce the proposed *k*NN query process algorithm next.

***k*NN Query Processing Algorithm.** Since the index structure is created based on Hilbert curve, the data objects with close row keys are likely to be spatially close. Thus, to answer a *k*NN query with query location (x,y), the proposed algorithm first identifies the corresponding cell and the transformed sequence number of (x,y). With the sequence number, the proposed algorithm then retrieves the data objects whose row keys are close to the sequence number. To ensure the correctness of the answer to the *k*NN query, however, the proposed algorithm uses four Hilbert curves to fetch the nearby data objects instead of one single Hilbert curve. The reason is that searching nearby data objects employing one Hilbert curve guarantees only that at least one fourth of the data objects are spatially close. Thus, using one Hilbert curve takes longer time to retrieve sufficient data objects for answering the *k*NN query.

To overcome this problem, the proposed algorithm constructs four Hilbert curses as follows. The first curve is same as the one for index structure creation (starting cell

Algorithm 2. *k*NN query processing algorithm: $k\mathrm{NN}(x,y,k)$

```
Input:
    $x$ : x-coordinate of the query location
    $y$ : y-coordinate of the query location
    $k$ : the requested number of objects
Output:
    $knn$ : $k$ nearest objects with respect to $(x,y)$
$d1$, $d2$, $d3$, $d4$ is the 1D value of $(x,y)$ transformed by the Hilbert Curve on four directions.
$C \leftarrow \emptyset$
$range \leftarrow$ the initial search range specified by data
$multi \leftarrow 1$
while the number of objects in $C < k$ do
    for $d \in \{d1,d2,d3,d4\}$ do
        $D \leftarrow$ all objects whose 1D value transformed by Hilbert Curve between $d - range \times multi$ and $d + range \times multi$
        $C \leftarrow C \cap D$
    end for
    $multi \leftarrow multi + 1$
end while
$r \leftarrow$ the distance between $(x,y)$ and the $k$th nearest point in $C$
$W \leftarrow WindowQuery(x,y,2r,2r)$ // Algorithm 1
$knn \leftarrow$ the top $k$ nearest points in $W$
return $knn$
```

$(0,0)$ and ending cell $(n-1,0)$) and the second one is generated by rotating the first curve 90° counterclockwise (starting cell $(n-1,0)$ and ending cell $(n-1,n-1)$). Similarly, the third one is determined with starting $(n-1,n-1)$ and ending cells $(0,n-1)$ while the fourth one has starting and ending cells $(0,n-1)$ and $(0,0)$, respectively. Although the four Hilbert curves allow to retrieve the correct answer to a *k*NN query, the algorithm may need to access $\frac{2}{3}(n^2-1)$ data objects when the query location (x,y) is located in the vicinity of cell $(0,0)$ or cell $(n-1,n-1)$. To increase the search efficiency, we observe that the answer objects of a *k*NN query must be within the circle centered at (x,y) with radius r where r is the distance between q and the kth object found by the used Hilbert curves. As such, when fetching the kth object found by the used Hilbert curves, the proposed algorithm runs a window query $(x,y,2r,2r)$ and identifies the answer objects from the result of the window query. The algorithmic form of the proposed *k*NN query processing algorithm is described in Algorithm 2.

4 Performance Evaluation

4.1 Environmental Setup

To evaluate the performance of the proposed index structures and companion algorithms, we deploy Hadoop version 1.0.0 and HBase version 0.92.1 on a cluster with 8 servers. The specification of the servers is shown in Table 3. Because the configuration of Hadoop and HBase could have a substantial impact on the experimental results,

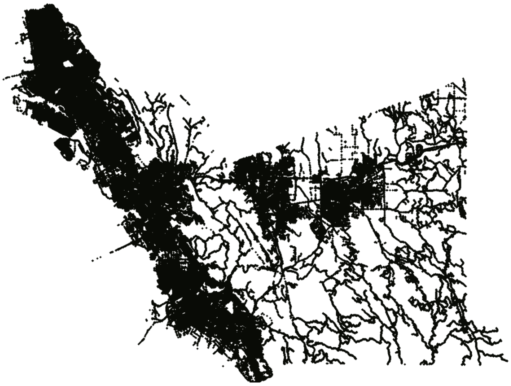


Fig. 2. Real dataset for evaluation

Table 3. The specification of the deployed servers

CPU	Intel(R) Xeon(R) E5520 @ 2.27GHz (8 cores)
Memory	12GB
OS	FreeBSD 8.0-RELEASE
Storage	SATA II 500GB

we use the same configuration on the 8 servers. We conduct the experiments using the real dataset (shown in Fig. 2) that consists of 488,760 points in the roads of California [13]. The query locations are randomly generated and each experiment is run 100 times. We adopt query time, the most commonly used performance metric, to measure the performance.

4.2 Experiments on *k*NN Queries

To measure the performance of the proposed index structure and process algorithm for *k*NN queries, we first conduct the experiment varying the values of *k* from 1 to 20. Besides, we use 1 and 4 Hilbert curves for data object search in this experiment to validate the efficiency of 4 Hilbert curves. From Fig. 3, the proposed index structure and algorithm are efficient for *k*NN query solving and are slightly affected by the increase in the value of *k*. The reason is that the distributed nature of HBase and the design of the proposed index structure and algorithm are beneficial for effective parallel computation of query search. In addition, we can see from Fig. 3 that 4 Hilbert curves substantially increase the search efficiency due to the parallel search and the smaller radii of executed window queries. Next, we change the number of data objects from 400,000 to 2,000,000 to investigate the scalability of the proposed index structure and algorithm. Fig. 4 shows that the query time increases less than 40% when the datasize is 5 times larger. This results show that the proposed index structure and algorithm are very scalable since the search can be executed in a parallel manner.

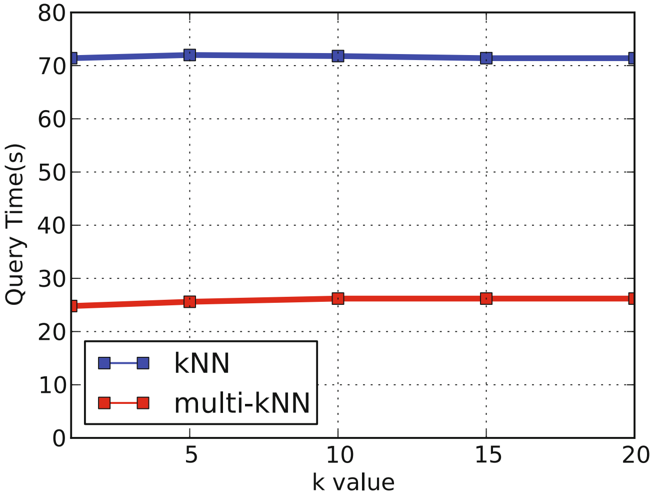


Fig. 3. The value of *k* vs. query time for *k*NN queries

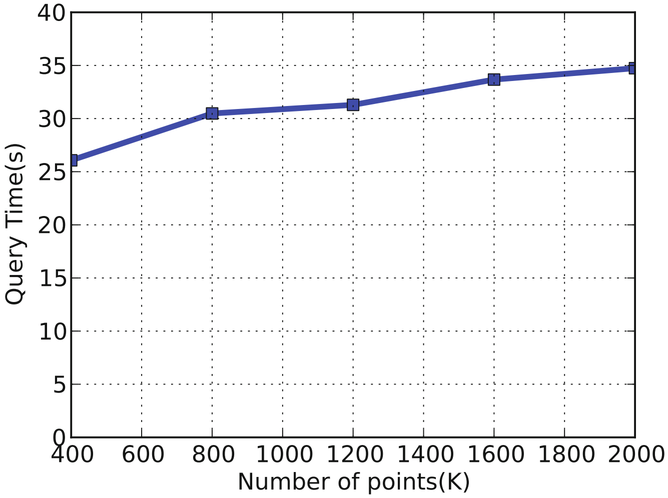


Fig. 4. Data size vs. query time for *k*NN queries

4.3 Experiments on Window Queries

In this subsection, we evaluate the performance of the proposed index structure and algorithm on window queries. First, we study the effect of different window sizes by increasing the window length and width from 1% to 20% of the data space length. Fig. 5 depicts that the window size has a slight effect on the performance due to effectiveness of the proposed index structure and algorithm. Besides, we also study the scalability of

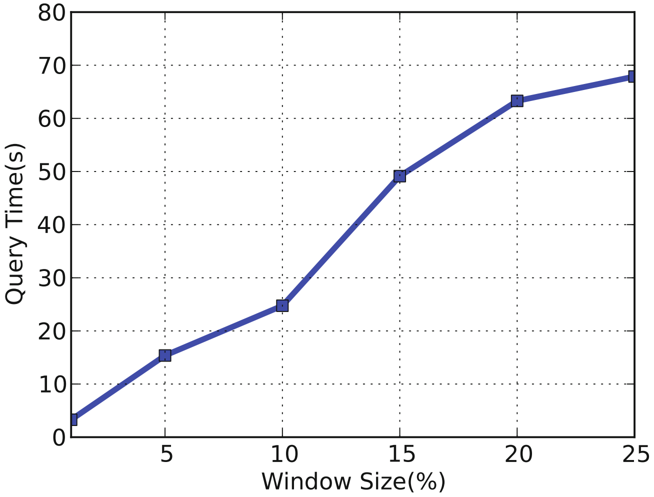


Fig. 5. The size of window queries vs. query time for window queries

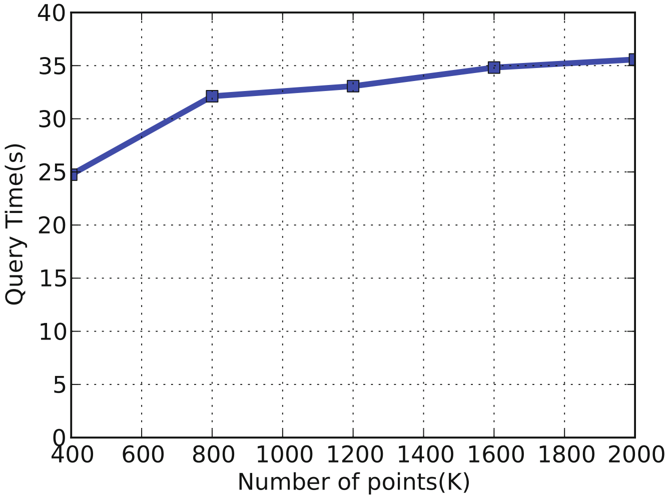


Fig. 6. Data size vs. query time for window queries

the proposed index structure and algorithm on window queries by increasing the number of data objects from 400,000 to 2,000,000. Similar to the *k*NN query case, the proposed index structure and algorithm are fairly scalable because the significant increase in the number of data objects increases the query time slightly, as shown in Fig. 6.

5 Conclusion

In this paper, we propose novel index structures and companion algorithms to achieve efficient spatial query processing in the case of enormous amounts of data objects. The proposed index structures are built on the top of the distributed database HBase and are separately designed according to the characteristics of *k*NN and window queries. With the index structures, we devise efficient *k*NN and window query processing algorithms to achieve fast query search. The experimental results on the real dataset show that the proposed algorithms and the index structures are efficient. Moreover, the results reveal that the larger dataset size slightly increases the query time of the proposed algorithms and thus the proposed algorithms are scalable.

Acknowledgments. This work of Dr. Huang described in this paper was partially supported by ITRI under Grant No. B301AA3241.

References

1. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data (1984)
2. Beckmann, N., Kbegel, H.P., Schneider, R., Seeger, B.: The R*-tree: an efficient and robust access method for points and rectangles. In: Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data (1990)
3. Apache HBase, `http://hbase.apache.org/`
4. Apache Cassandra, `http://cassandra.apache.org/`
5. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: BigTable: A distributed storage system for structured data. In: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (2006)
6. MapReduce, `http://hadoop.apache.org/mapreduce/`
7. Apache Hadoop Project, `http://hadoop.apache.org/`
8. HBase Java API, `http://hbase.apache.org/apidocs/overview-summary.html`
9. HBase Stargate API, `http://archive.apache.org/dist/hadoop/hbase/hbase-0.20.1/docs/api/org/apache/hadoop/hbase/stargate/package-summary.html`
10. Apache Avro, `http://avro.apache.org/`
11. Apache Thrift, `http://thrift.apache.org/`
12. Gotsman, C., Lindenbaum, M.: On the metric properties of discrete space-filling curves. IEEE Transactions on Image Processing 5(5) (1996)
13. U.S. Census Beruea Tiger, `http://www.census.gov/geo/www/tiger/`

Path Tree: Mining Sequential Patterns Efficiently in Data Streams Environments

Guanling Lee, Kuo-Che Hung, and Yi-Chun Chen

Department of Computer Science and Information Engineering
National Dong Hwa University, Hualien, Taiwan, R.O.C
guanling@mail.ndhu.edu.tw

Abstract. Although issues of data streams have been widely studied and utilized, it is nevertheless challenging to deal with sequential mining of data streams. In this paper, we assume that the transaction of a user is partially coming and that there is no auxiliary for buffering and integrating. We adopt the *Path Tree* for mining frequent sequential patterns over data streams and integrate the user's sequences efficiently. Algorithms with regards to accuracy (*PAlgorithm*) and space (*PSAlgorithm*) are proposed to meet the different aspects of users. Many pruning properties are used to further reduce the space usage and improve the accuracy of our algorithms. We also prove that PAlgorithm mine frequent sequential patterns with the approximate support of error guarantee. Through experiments, synthetic dataset is utilized to verify the feasibility of our algorithms.

Keywords: Data Mining, Sequential Patterns, Frequent Patterns.

1 Introduction

In this paper, we focus on the problem of frequent sequence mining in data streams. The key characteristics of existing algorithms in sequences database are the multiple scans and secondary memory accesses involved. The characteristics are diametrically opposed to the limitations of the data stream environment: one pass scheme, short response time and limited system resources. Consequently, on-line fashion algorithms are needed to tie in the data stream environment. Li et al. introduced an on-line algorithm, *StreamPath*, to mine frequent sequence patterns [4]. In recent years, top-k sequences mining have been discussed in [3][5]. The issues of mining closed sequential patterns in a data stream environment are introduced in [1][2][6].

In this paper, the problem of finding frequent sequential patterns across multiple streams in an on-line fashion is discussed. In [4] and [5], they aimed to mine frequent sequences, called maximal forward references from data streams. They adapt FP-Tree based structures to achieve mining a frequent sequence in one pass; however, they do not take the problem of incremental transaction into account. In this paper, we propose algorithms for mining sequential patterns without any auxiliary buffer. An approximation for counting the algorithms with a guaranteed error boundary was proposed. Furthermore, we also proposed an algorithm with compressed space.

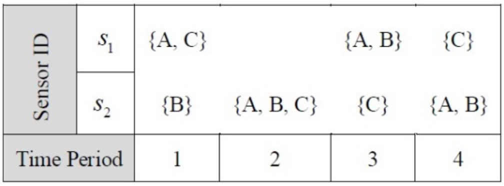
R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 261–268.
DOI: 10.1007/978-3-642-35452-6_28

The rest of the paper is organized as follows. The problem statement is provided in section 2. In section 3, the structure and algorithms are explained, followed by the performance result with thoughtful experiments in section 4. Conclusions are discussed in section 5.

2 Preliminaries

2.1 Stream Environment

Let $i = \{i_1, i_2, ..., i_n\}$ be the monitored items and *n* be the total number of items that can be a variable or fixed value depending on applications. An itemset $e = \{i_{j_1}, ..i_{j_k}\}$ is a subset of *I*. The monitoring sensors set is $S = \{s_1, s_2, ..., s_m\}$ where *m* is the total number of monitoring sensors. The data stream generated by s_j is a sequence with consecutive and boundless data items and has the form $\{e_{s_j t_1}, e_{s_j t_2}, ..., e_{s_j t_k}\}$ where $e_{s_i t_k}$ is the set of items detected by s_j during the time period t_k and named as an appearance. An example is shown in figure 1. Referring to Fig. 1, $e_{s_1 t_1}$ ={A, C} means item A and C appear at monitoring sensor s_1 in time period t_1. In this paper, we assume that an item can not only appear once during a time period, but multiple times in different periods.



Sensor ID					
	s_1	{A, C}		{A, B}	{C}
	s_2	{B}	{A, B, C}	{C}	{A, B}
Time Period		1	2	3	4

Fig. 1. Data of sensor gathering in each time period

A *path* of an item i_l, $path_{i_l} =< s_{t_1}, s_{t_2}, ..., s_{t_k} >$ where $s_{t_j} = \{s_\beta,$ if $i_l \in e_{s_\beta t_j}\}$, is a time ordered sequence, and, s_{t_1} is the *head* of the $path_{i_l}$. Moreover, *k* increases as time advances. A $path_{i_l} =< \alpha_1, \alpha_2, ..., \alpha_j >$ is a *sub-path* of a $path_{i_p} =< \beta_1, \beta_2, ..., \beta_k >$, denoted by $\text{path}_{i_l} \subseteq \text{path}_{i_p}$, if there exist integers $1 \le l_1 \le l_2 \le ... \le l_j \le k$ such that $\alpha_1 = \beta_{l_1}$, $\alpha_2 = \beta_{l_2}$,..., $\alpha_j = \beta_{l_j}$. The support count of a path *x* is the number of items whose path contains *x*. And the support of *x*, denoted by *support*(*x*), is defined by the ratio of the support count of *x* to the total number of items. Taking Fig. 1 as an example, the support count of path <$s_1 s_2$> is three.

2.2 Problem Definition

Since people have more interest in recent data, we can define a *stream window*, $W_j = \{w_{j-D+1}, \ldots w_{j-2}, w_{j-1}, w_j\}$, $j - D + 1 \geq 0$, to delimit which data we are interested in, where *j* is the current time period and *D* is the user defined window size. The timestamp of a path formed by item *i* is denoted by timestamp(*i*, $path_i$), that is the time that *i* appeared in the head. A path formed by item *i* is expired whenever its timestamp is out of a current time window.

Our goal is to mine the paths with support that exceed a *minimum support* within the current window. Minimum support is a user-defined parameter *min_sup*∈ (0,1). Given a user-defined error threshold $\varepsilon \in (0,1)$ and ε<<min_sup, the answer of our algorithm is required to have the following guarantees:

1. All paths whose support exceeds min_sup are output. There are no *false negatives.*
2. No paths whose support has a value of less than (min_sup- ε) is output.

3 Algorithms

3.1 Path Tree

Conceptually, the complete search space of path mining forms a *path tree* (*PTree*), which can be constructed in the following way: The root node $node^0_\varnothing$ of the PTree is at level 0 and labeled $\varnothing$. From root node $\text{node}^0_\varnothing$ but excluding $\text{node}^0_\varnothing$, each node $\text{node}^l_{s_j}$ PTree forms a path path($\text{node}^l_{s_j}$) = < s_x, …, s_j>, where $\text{node}^1_{s_x}$ is in the path from $\text{node}^0_\varnothing$ to $\text{node}^l_{s_j}$ and s_x is the label of the node following $\text{node}^0_\varnothing$. A set of items denoted by $item\_set(node^l_{s_j})$ maintained in node $\text{node}^l_{s_j}$ is used to store the relationship between paths and items, and assist the recognition of the item's path whenever the new arrival tuple is inserted into the PTree. Moreover, each item *i* maintained in node $\text{node}^l_{s_j}$ is associated with a timestamp $\text{timestamp}(i, \text{node}^i_{s_j})$ which is set to $\text{timestamp}(i, \text{path}(\text{node}^l_{s_j}))$.

In the PTree, each node represents a path from root to the node and contains a set of items. It is obvious that PTree stores all the information of items path and sub-paths. The PTree construction process is illustrated by the following example: in Fig.s 1 and 2, $w_2 = \{e_{s_2 t_2 = \{\text{A,B,C}\}}\}$ is incoming, item A will be inserted into $\text{node}^1_{s_2}$ with timestamp 2, since $\text{node}^1_{s_2}$ already exists. In addition, because A has visited $\text{node}^1_{s_1}$ before and $\text{node}^1_{s_1}$ has no child with label s2, a new child $\text{node}^2_{s_2}$ is

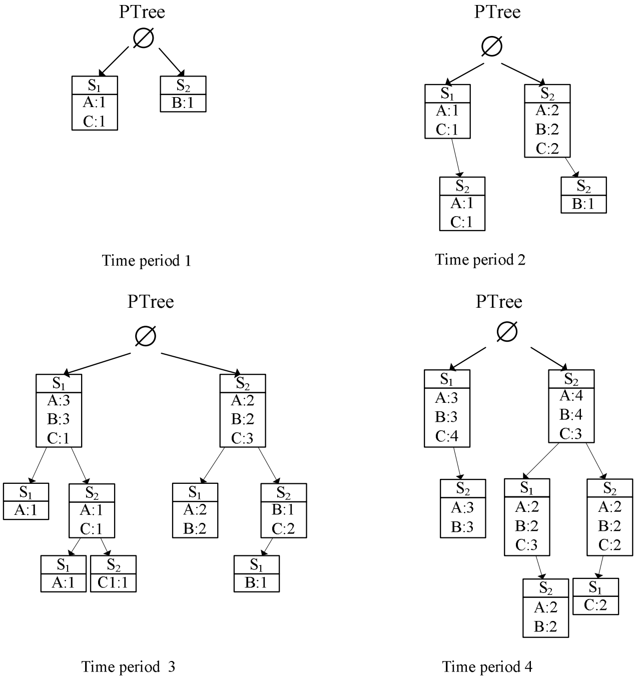


Fig. 2. PTree constructed from Fig. 1

inserted into $\text{node}_{s_1}^{1}$ to maintain the path $< s_1, s_2 >$ formed by A. Moreover, A is inserted into item_set($\text{node}_{s_2}^{2}$) and timestamp(A, $\text{node}_{s_2}^{2}$) is set to timestamp(A, $\text{node}_{s_1}^{1}$) since timestamp(A, $\text{node}_{s_1}^{1}$) is the timestamp of path$< s_1, s_2 >$ formed by A. Recursively, we can extend a node $\text{node}_{s_j}^{l}$ at level *l* with label s_j in PTree by adding sensor id s_m as its child node $\text{node}_{s_m}^{l+1}$ at the next level *l*+1. This is called the *path extension* and the extended path is denoted by $path(node_{s_j}^{l})\cdot s_m$ (the corresponding node is denoted by $node_{s_j}^{l}\cdot s_m$). An item does not contain $\text{path}(\text{node}_{s_j}^{l})$ in the time period of t_k whenever the timestamp(i, $\text{node}_{s_j}^{l}$)+*D* is no greater than t_k . Obviously, the support count of a path formed by node $\text{node}_{s_j}^{l}$ (we will use the node to represent the path formed by itself for short), *sup_count*($\text{node}_{s_j}^{l}$), is the number of items it maintains. By traversing the PTree and identifying the node whose support count exceeds $\min\_\sup\times n$, we can enumerate all frequent paths.

3.2 Algorithms

In our approach, a user-specified error parameter ε is used as the candidate generation threshold, *ε-generation.* After an update iteration, nodes with support count exceeding ε are identified. These identified nodes can be taken into account to generate new candidates, the candidates of which are then counted in later iterations.

Because $\text{path}(\text{node}_{s_j}^{l})\cdot s_m$ would not be generated until the support counts of $\text{node}_{s_j}^{l}$ and its sibling $\text{node}_{s_m}^{l}$ exceed $\varepsilon \times n$, we could not determine the items whose path contains path($\text{node}_{s_j}^{l}$) and path($\text{node}_{s_m}^{l}$) forms $path(node_{s_j}^{l})\cdot s_m$ or $path(node_{s_m}^{l})\cdot s_j$ whenever the path extension is performed. Therefore, we set the items maintained by extended nodes $\text{node}_{s_j}^{l}\cdot s_m$ and $\text{node}_{s_m}^{l} s_j$ to the possible maximum set of items. The possible items whose path contain $path(node_{s_j}^{l})\cdot s_m$ is the intersection of the items maintained by $\text{node}_{s_j}^{l}$ and its sibling $\text{node}_{s_m}^{l}$. Similarly, the sub-tree rooted at node $\text{node}_{s_j}^{l}$ can be removed whenever the sup_count($\text{node}_{s_j}^{l}$) is less than $\varepsilon \times n$.

Lemma 1: All nodes whose true support exceeds min_sup can be recognized in the PTree constructed by ε-generation.

Proof: Assume *t`* is the support count of a node *γ* accumulated so far and *t* is the true support of *γ*, we have $t' \geq t \geq t' - \varepsilon$, since sup_count(*γ*) is always set to ε whenever it was generated. If *γ*'s true support exceeds min_sup, we have $t' \geq t \geq \text{min\_sup}$. Thus, nodes with true support count that exceed the min_sup will be recognized. □

Lemma 2: Each node *γ* with estimated support t' no smaller than the min_sup in PTree constructed by using ε-generation satisfies: $t' - t \leq \varepsilon$.

Proof: Because the over estimate of *γ* is $t' - t$, where *t* is the true support of *γ*, and we know $t \geq t' - \varepsilon$. Therefore, $t' - t \leq t' - (t' - \varepsilon)$. And we get $t' - t \leq \varepsilon$. □

From lemmas 1 and 2, we can examine that the PTree constructed by using ε-generation is satisfying those guarantees outlined in the previous section.

In the following, we introduce two algorithms to enumerate the complete set of frequent paths with regards to the accuracy and space. The first algorithm, PAlgorithm, keeps the full information of the timestamp in nodes to meet the requirements of the guarantee discussed in section 2. The second algorithm, PSAlgorithm, keeps less information of the timestamp in nodes. It calculates an item's timestamp through an estimation function.

Through PAlgorithm, PTree persistently maintains a root and a set of child nodes { $node^{l}_{s_m} \mid s_m \in S$ }, only the monitored nodes are counted in each update. Generally, PAlgorithm traverses PTree in a DFS manner. For an incoming item i of appearance $e_{s_j t_k}$, PAlgorithm will check each node $\text{node}^{1}_{s_m}$ at level one and update i recursively if item i exists in $\text{node}^{1}_{s_m}$. After the recursive updating of i, PAlgorithm inserts or updates i with timestamp t_k into node $\text{node}^{1}_{s_j}$. By checking the items maintained by nodes, PAlgorithm recognizes the sub-paths of an item i and updates nodes through calling subroutine *PUpdate*. The sense of recursively calling PUpdate is to recognize each i's sub-paths, to concatenate each i's sub-paths with sj and to update each i's sub-paths to the latest timestamp. The item's timestamp of a node will be updated if the node's label equals the sensor id of the current appearance. Moreover, the timestamp(i, $\text{node}^{l}_{s_m}$) should be updated by the timestamp(i, $\text{node}^{l}_{s_m}$'s parent) instead of the current time period.

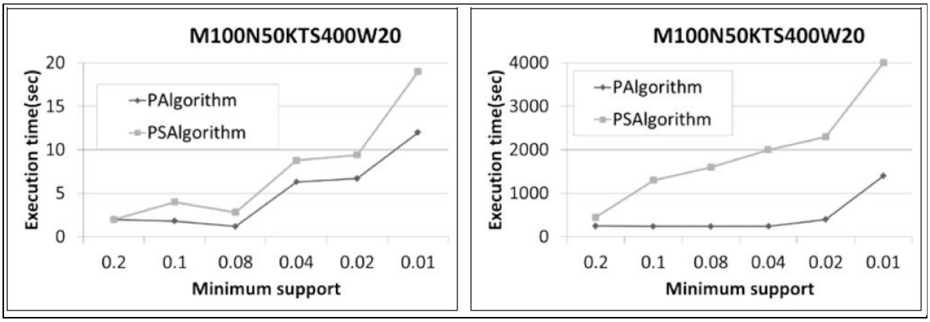
After the insertion, PAlgorithm will check nodes for the path extension by calling the subroutine *Path Extension*. The subroutine is triggered when the sup_count($\text{node}^{l}_{s_j}$) is no less than $\varepsilon \times n$. Moreover, the height of PTree is at most D and we can avoid the problem of delayed recognition. If the path extension process of a node $\text{node}^{l}_{s_j}$ is triggered, siblings with a support count of no less than ε (including $\text{node}^{l}_{s_j}$) will be identified and extended with each other to form nodes at the next level. An item i would not contains $\text{path}(\text{node}^{l}_{s_m})$ in time period of t_{k+1} whenever $\text{timestamp}(i, \text{node}^{l}_{s_m}) + D$ is no greater than t_{k+1}. Thus we drop items in nodes if their corresponding timestamp are expired. Moreover, we can delete item i from $\text{node}^{l}_{s_j}$'s sub-tree safely whenever $\text{timestamp}(i, \text{node}^{l}_{s_j})$ is expired in $\text{node}^{l}_{s_j}$. The sub-tree rooted at node $\text{node}^{l}_{s_j}$ would be deleted whenever sup_count($\text{node}^{l}_{s_j}$) is less than εn, owing to the path extension of ε-generation.

To avoid the space overhead of storing timestamps, we propose another algorithm, PSAlgorithm, to enumerate frequent paths with an estimated timestamp. The necessary information is timestamps of items in nodes at level one. The procedure of PSAlgorithm is almost the same as PAlgorithm. The difference between PAlgorithm and PSAlgorithm is the estimated timestamp, since PSAlgorithm gets an item's timestamp through an estimated method. Therefore, the PTree can be modified as follows: the nodes maintain a set of items instead of items and timestamps except nodes at level one. Because we evaluate an item's timestamp through an estimated method, it results in a loose guarantee of node's support count. The loose guarantee is a tradeoff to meet the compromise between accuracy and space.

4 Experimental Results

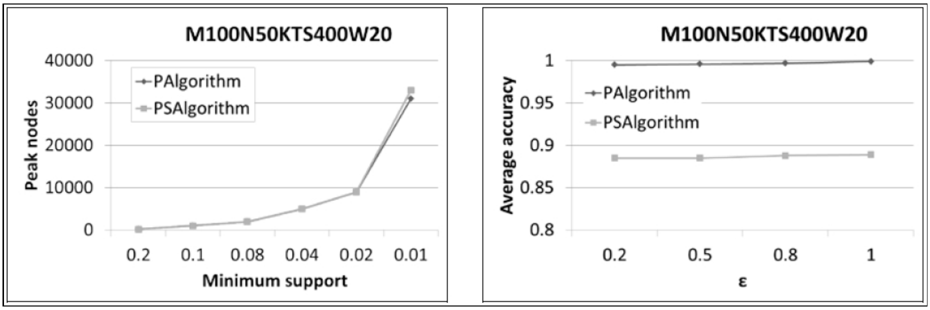
In the experiments, the synthetic data is generated by an IBM synthetic data generator. For example, M100N50KTS400W20 means the dataset is generated with regards to 100 sensors, fifty thousand items, during four hundred time periods and within a window size of 20.

In the first simulation, the relationship between execution time and minimum support is conducted. As shown in Fig. 3, the execution time grows as the minimum support decreases. Clearly, in Fig. 3(a), the execution time of PAlgorithm increases slightly in the range from min_sup = 0.2 to 0.02. Moreover, PAlgorithm outperforms PSAlgorithm in both datasets.



(a) (b)

Fig. 3. Execution time: (a) peak execution time of time periods; (b) total execution time



(a) (b)

Fig. 4. Average accuracy: (a) min_sup; (b) *ε*

Fig. 4(a) shows the average accuracies of frequent algorithm mined temporal paths over data streams. The average accuracy of the PAlgorithm is admirably perfect, since it nearly stores every timestamp of an item. The PSAlgorithm shows its exceptional competence with regard to memory usage and accuracy. However, the accuracy and speed of the PSAlgorithm is influenced by the distribution of the dataset. Referring to Fig. 4(b), the effect of ε is shown. As shown in the result, the accuracy is very stable as the error parameter increases. The stable curve of the PSAlgorithm shows that the PSAlgorithm has competitive behavior under situations of silent distribution.

5 Conclusions

In this paper, the problem of mining temporal frequent paths over data streams is discussed. A data structure, PTree, is introduced to maintain the possible paths of users. Using the PTree, two algorithms are proposed to deal with these problems of frequent path mining. PAlgorithm provides an efficient way for mining results with dramatic accuracy, owing to the error bounded PTree. To further ease the maintenance overhead, the PSAlgorithm is proposed to mine frequent paths with estimated timestamp. The Experiment's results indicate that the PSAlgorithm can achieve an acceptable behavior of accuracy through the estimation.

References

1. Chen, Y.C., Lee, G.: Mining Sequential Association Rules Efficiently by Using Prefix Projected Databases. Journal of Computers 22(2), 33–47 (2011)
2. Chang, L., Wang, T., Yang, D., Luan, H.: SeqStream: Mining Closed Sequential Patterns over Stream Sliding Windows. In: ICDM, pp. 83–92 (2008)
3. Dai, B.R., Jiang, H.L., Chung, C.H.: Mining Top-K Sequential Patterns in the Data Stream Environment. In: International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 142–149 (November 2010)
4. Li, H., Lee, S., Shan, M.: On Mining Webclick Streams for Path Traversal Patterns. In: International World Wide Web Conference on Alternate Track Papers & Posters (2004)
5. Li, H., Lee, S., Shan, M.: DSM-TKP: Mining Top-K Path Traversal Patterns over Web Click-Streams. In: IEEE/WIC/ACM International Conference on Web Intelligence (2005)
6. Yang, S.Y., Chao, C.M., Chen, P.Z., Sun, C.H.: Incremental Mining of Closed Sequential Patterns in Multiple Data Streams. Journal of Networks 6 (2011)

Finding Leaders with Maximum Spread of Influence through Social Networks

Tsung An Yeh[1], En Tzu Wang[2], and Arbee L.P. Chen[3]

[1] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
s9862659@m98.nthu.edu.tw

[2] Cloud Computing Center for Mobile Applications, Industrial Technology Research Institute, Hsinchu, Taiwan
m9221009@em92.ndhu.edu.tw

[3] Department of Computer Science, National Chengchi University, Taipei, Taiwan
alpchen@cs.nccu.edu.tw

Abstract. Social influence is an important phenomenon in social networks. A user is said to be influenced by his/her friends if this user performs the same actions after his/her friends. The problem of influence maximization is to find a small set of users that maximize the spread of influence throughout the social network. Many approaches are proposed to solve this problem under different influence cascade models. In this paper, we propose another solution based on a pruning strategy in which influence boundaries for users are computed to effectively reduce the number of users who have chances to be seeds, thus making the proposed solution efficient. A series of experiments are performed to evaluate the proposed approach and the experiment results reveal that our approach outperforms the previous works.

Keywords: influence maximization, social networks, independent cascade model.

1 Introduction

A social network is a network consisting of many users with interactions to each other. In a social network, information and messages are quickly propagated. A user performs a specific action, which may affect his/her friends to perform the same action. This phenomenon is so-called the *social influence*, which can be applied in the online marketing strategies. We call a user who initially starts the information propagation a *leader*. Distinct leaders make distinct influences to spread over distinct communities in the social network. Under a given budget, a company chooses a limited set of leaders for helping product promotion, which is the problem we are solving of selecting k leaders with maximum spread of influence in the social network.

Kempe et al. [6] first formulate the above problem as an optimization problem. A social network is modeled as a directed graph with nodes representing users and edges representing the relationships between a pair of users. Each edge in the graph is with a *propagation probability*, which is the probability that the corresponding one user will

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 269–279.
DOI: 10.1007/978-3-642-35452-6_29

influence the other user through this relationship. When a user performs a specific action and starts to influence the others, the associated node in the graph is denoted *active*. Thus, the nodes in the graph are either active or inactive. Given a social network graph and an integer of k, the influence maximization problem is defined as finding k nodes in the graph, which expect to "influence" the most other nodes in the graph. These k nodes, referring to leaders as mentioned, are denoted *seeds* and the set of the k nodes is called *seed set*. Two widely-studied diffusion models including *independent cascade* model [2][1][6] [8] and *linear threshold* model[3][5] [9][10] are proposed. Besides, another diffusion model named *general threshold model* is proposed in [4]. We only focus on the independent cascade model in this paper.

Kempe et al. prove that optimization of this problem is NP-hard and propose a greedy algorithm to find an approximate solution with a guarantee of 1 – 1/e from the optimal solution. The greedy algorithm works as follows. Choose a node from the graph, which influences the most nodes as a seed. Thereafter, update all of the unchosen nodes' influence spread by considering the nodes in the seed set to be all activated. Repeat the iteration until there are k seeds in the seed set. The greedy algorithm is not efficient since it needs to update almost all nodes' influence spread after selecting a seed. Therefore, many studies focus on improving the efficiency of the greedy algorithm, such as [2][1][8]. In [1], Chen et al. propose the *maximum influence arborescence* (MIA) model to evaluate the influence spread of a user. By using the index structures *maximum influence in-arborescence* (MIIA) and *maximum influence out-arborescence* (MIOA), Chen et al. quickly update the users to be influenced by seeds. In other words, it need not update all users' influence spread and thus is much faster than the original greedy algorithm.

In this paper, we consider the *power-law distribution* and the property of the MIA model to design a new algorithm with a strong pruning strategy, effectively eliminating the users unable to be a seed. By this pruning strategy, we only need to update the influence spread of the users who have potential to become seeds, i.e. the *candidate nodes*. The number of the candidate nodes is much less than that of the total nodes, making the whole process much faster than the previous works. The remainder of this paper is organized as follows. The preliminaries are introduced in Section 2. The proposed method is detailed in Section 3. Thereafter, the performance evaluations are described in Section 4 and finally, we conclude this work in Section 5.

2 Preliminaries

A directed graph is denoted $G = (V, E)$, where V denotes a set of nodes to represent users and E denotes a set of edges to represent the relationships of pairs of users. Each edge $(v_1, v_2) \in E$, where $v_1, v_2 \in V$, is with a weight within a range of [0, 1], denoted $pp(v_1, v_2)$ and representing the propagation probability that v_2 is activated by v_1 through the edge of (v_1, v_2). v_2 is activated by v_1 means v_1 is active and influences v_2 to become an active node. Following the definition in [1], a path from v_1 to v_2, denoted $P(v_1, v_2)$, is represented by a sequence of nodes, e.g. $P(v_1, v_2) = \langle v_1, v_i, v_j, v_2\rangle$, where v_i, $v_j \in V$. To activate v_2, v_1 needs to activate all the nodes in $P(v_1, v_2)$. The propagation probability of $P(v_1, v_2)$, denoted $pp^{path}(v_1, v_2)$ and representing that v_2 is activated by v_1 through $P(v_1, v_2)$, is defined as $\mathrm{pp^{path}(v_1, v_2) = pp(v_1, v_i) \times pp(v_i, v_j) \times pp(v_j, v_2)}$.

Moreover, we ignore the influence through a path $P(v_1, v_2)$ when the propagation probability of the path $pp^{path}(v_1, v_2)$ is smaller than a given *influence threshold θ.*

Definition 1 (*Out-path Node and In-path Node*). Given a directed graph G, if node v_1 has a path to node v_2, we define v_2 as an *out-path node* of v_1 and v_1 as an *in-path node* of v_2. The set of out-path nodes and the set of in-path nodes of node v are denoted $N^{out\text{-}path}(v)$ and $N^{in\text{-}path}(v)$, respectively. In addition, the set of out-neighbors and the set of in-neighbors of v are denoted $N^{out}(v)$ and $N^{in}(v)$.

Definition 2 (*Activation Probability*). According to [1], given a seed set S, the *activation probability* of a node v, representing the probability that v is activated by the influence from its in-neighbors, is defined as

$$ap(v, S) = \begin{cases} 1 & \text{, if } v \in S \\ 0 & \text{, if } N^{in}(v) = \emptyset \\ 1 - \prod_{v' \in N^{in}(v)} \left(1 - ap(v', S) \times pp(v', v)\right) & \text{, otherwise} \end{cases} \tag{1}$$

Example 1. As shown in Fig. 1, given a node v and its in-neighbors nodes v_1, v_2, and v_3 with propagation probabilities equal to $pp(v_1, v) = 0.7$, $pp(v_2, v) = 0.3$, and $pp(v_3, v) = 0.5$, respectively, if v_1, v_2, and v_3 are in S, the activation probability of v, i.e. $ap(v, S)$, is equal to $1 - ((1 - 1 \times 0.7) \times (1 - 1 \times 0.3) \times (1 - 1 \times 0.5)) = 0.895$.

Definition 3 (*Competition Node*). Given two nodes v_1 and v_2 and v_1 is an in-path node of v_2, the competition node of v_1 with respect to v_2 are the other in-path nodes of v_2. The set of competition nodes of v_1 with respect to v_2 is denoted $N^{competition}(v_1, v_2) = \{ v' \mid v' \in V \wedge v' \in N^{in\text{-}path}(v_2) \backslash \{v_1\} \}$.

Definition 4 (*Propagation Probability Considering Competition*). The propagation probability considering competition from node v_1 to node v_2 through the edge (v_1, v_2), representing the probability that v_2 is activated *only* from the influence of v_1 through the edge (v_1, v_2), is defined as

$$ppc(v_1, v_2, S) = \begin{cases} 0 & \text{, if } v_2 \in S \\ pp(v_1, v_2) \times \prod_{v' \in N^{in}(v_2) \backslash \{v_1\}} \left(1 - ap(v', S) \times pp(v', v_2)\right) & \text{, otherwise} \end{cases} . \tag{2}$$

The propagation probability considering competition from v_1 to v_m through $P(v_1, v_m) = \langle v_1, v_2, \ldots, v_m \rangle$, representing the probability that v_m is activated *only* from the influence of v_1 through $P(v_1, v_m)$, is defined as

$$ppc^{path}(v_1, v_m, S) = \begin{cases} 0 & \text{, if } v_i \in S \text{ for } i = 2 \text{ to } m \\ \prod_{i=1}^{m-1} ppc(v_i, v_{i+1}, S) & \text{, otherwise} \end{cases} . \tag{3}$$

The definition of propagation probability considering competition is the same as the lemma of *influence linearity* described in [1].

Suppose v_2 is an out-neighbor of v_1. According to Definition 2, $ap(v_2, S)$ is equal to $1 - (1 - ap(v_1, S) \times pp(v_1, v_2)) \times \prod_{v' \in N^{in}(v_2) \backslash \{v_1\}} (1 - ap(v', S) \times pp(v', v_2))$. Considering the condition that suppose v_1 is activated, i.e. $ap(v_1, S) = 1$, we represent the activation probability of v_2 under this condition as $ap'(v_2, S)$, which is equal to $1 - (1 - 1 \times pp(v_1, v_2)) \times \prod_{v' \in N^{in}(v_2) \backslash \{v_1\}} (1 - ap(v', S) \times pp(v', v_2))$. We define the difference between

$ap(v_1, S)$ and $ap'(v_2, S)$ as the *incremental influence* from v_1 to v_2, denoted $IncInf^{out}(v_1, v_2, S) = \left(1 - ap(v_1, S)\right) \times ppc(v_1, v_2, S)$. Moreover, $ppc(v_1, v_2, S)$ is replaced by $ppc^{path}(v_1, v_2, S)$ to get $IncInf^{out\text{-}path}(v_1, v_2, S) = \left(1 - ap(v_1, S)\right) \times ppc^{path}(v_1, v_2, S)$.

Definition 5 (*Incremental Influence Spread of a Node*). The *incremental influence spread* [1] of a node v is defined as the total incremental influence from v to its out-path nodes, denoted $IncInf(v, S) = \sum_{v' \in N^{out\text{-}path}(v)} IncInf^{out\text{-}path}(v, v', S)$.

Definition 6 (*Influence of Seed Set*). Given a seed set S, the influence spread of S is defined as the total influence spread of nodes in S, denoted $Inf(S) = \sum_{v \in V} ap(v, S)$.

Given a directed graph $G = (V, E)$, an influence threshold θ, and an integer of k, the problem of influence maximization is to find a seed set $S \subseteq V$ with a size of k, which has the maximum influence spread.

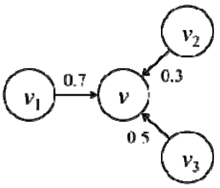


Fig. 1. An illustration of Example 1

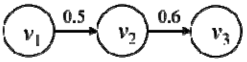


Fig. 2. An illustration of Example 2

Definition 7 (*Correlated*). Given two nodes v_1 and v_2, we defined v_1 is *correlated* with v_2 iff v_1 is an out-path node or in-path node or competition node of v_2.

Definition 8 (*Greedy Order*). The greedy order of a node v is the order that v is chosen as a seed in the greedy algorithm. The greedy order of node v_1 denoted $GO(v_1)$ is smaller than that of node v_2 denoted $GO(v_2)$ iff v_1 is activated before v_2.

Lemma 1. Given two nodes v_1 and v_2, if v_1 is correlated with v_2, then $IncInf(v_1, S)$ in the condition of $GO(v_1) < GO(v_2)$ is larger than $IncInf(v_1, S)$ in the condition of $GO(v_2) < GO(v_1)$.

Proof. We first consider the cases of out-neighbors and in-neighbors. Notice that we use $IncInf(v_1, S)$ and $ap(v_1, S)$ to represent the condition of $GO(v_1) < GO(v_2)$ and $IncInf'(v_1, S)$ and $ap'(v_1, S)$ to represent the condition of $GO(v_2) < GO(v_1)$.

Case 1 (v_2 is an out-neighbor of v_1): $IncInf(v_1, S)$ is equal to the summation of incremental influence to v_1's out-path node and $IncInf^{out}(v_1, v_2, S)$ is equal to $(1 - ap(v_1, S)) \times ppc(v_1, v_2, S)$. $ppc(v_1, v_2, S)$ is equal to 0 if node v_2 is activated, i.e. $GO(v_2) < GO(v_1)$. Furthermore, the incremental influence from v_1 to the out-neighbors of v_2 is limited by v_2. If $GO(v_2) < GO(v_1)$, $IncInf^{out}{}'(v_1, v_2, S)$ and the incremental influence to the out-neighbors of v_2 are equal to 0, making $IncInf(v_1, S) > IncInf'(v_1, S)$.

Case 2 (v_2 is an in-neighbor of v_1,): $ap(v_1, S)$ is related to v_1's in-neighbors and $IncInf^{out\text{-}path}(v_1, v', S)$ is equal to $(1 - ap(v_1, S)) \times ppc^{path}(v_1, v', S)$, where v' is an out-path node of v_1. We can infer that $ap'(v_1, S)$ is larger than $ap(v_1, S)$ if v_2 is activated, i.e. $GO(v_2) < GO(v_1)$. Thus, if $GO(v_2) < GO(v_1)$, $(1 - ap(v_1, S))$ is larger than $(1 - ap'(v_1, S))$, making $IncInf(v_1, S) > IncInf'(v_1, S)$.

Case 3 (v_2 is a competition node of node v_1): If $GO(v_2) < GO(v_1)$, the propagation probability considering competition is smaller than the propagation probability without considering competition, making $IncInf(v_1, S) > IncInf'(v_1, S)$. ■

Example 2. As shown in Fig. 2, v_2 is the out-neighbor of v_1 with $pp(v_1, v_2) = 0.5$ and v_3 is the out-neighbor of v_2 with $pp(v_2, v_3) = 0.6$. That is, v_3 is an out-path node of v_1 with $pp^{path}(v_1, v_3) = 0.5 \times 0.6 = 0.3$. If v_1 is activated before v_2 ($GO(v_1) < GO(v_2)$), $IncInf(v_1, S) = (1 - 0) \times 0.5 + (1 - 0) \times 0.3 = 0.8$ and $IncInf(v_2, S) = (1 - 0.5) \times 0.6 = 0.3$. If v_2 is activated before v_1 ($GO(v_2) < GO(v_1)$), $IncInf(v_1, S) = 0$ and $IncInf(v_2, S) = 0.6$. At the view point of v_1, v_2 is correlated with v_1 and $IncInf(v_1, S)$ in the condition of $GO(v_1) < GO(v_2)$ is larger than $IncInf(v_1, S)$ in the condition of $GO(v_2) < GO(v_1)$.As the view point of v_2, v_1 is correlated with of v_2 and $IncInf(v_2, S)$ in the condition of $GO(v_2) < GO(v_1)$ is larger than $IncInf(v_2, S)$ in the condition of $GO(v_1) < GO(v_2)$.

Definition 9 (*Maximum Influence and Minimum Influence*). The *maximum influence* of a node v denotes the incremental influence spread of v while v is activated before any other nodes in V, i.e. the first (*best*) greedy order, defined as

$$MaxInf(v) = \{IncInf(v, S) \mid GO(v) < GO(v'), \forall v' \in V\}. \qquad (4)$$

The *minimum influence* of v is the incremental influence spread of v, considering the *worst* greedy order. That is, for any node v' correlated with v, we assume $GO(v') < GO(v)$ and the incremental influence spread of v in this condition is defined as the minimum influence of v.

3 Finding Leaders

```
Algorithm 1: The ECE Algorithm
$S$ = {}, $Candidate$ = $V$
for each node $v \in V$ do
    compute $MaxInf(v)$
sort all nodes by their own $MaxInf$
Compute $LB$
prune the candidates with $MaxInf$ < $LB$
if |$Candidate$| == $k$    /* optimal pruning in initial */
     return Candidate as $S$
while |$S$| < $k$ do
    $u$ = arg max{$IncInf(u, S)$ | $u \in Candidate \setminus S$}
    $S$ = $S \cup \{u\}$, Candidate = Candidate \ {$u$}
    for each node $v$ in $Candidate$ do
        compute $IncInf(v, S)$
        if $IncInf(v, S)$ < $LB$ then
              prune $v$ from $Candidate$
    sort all nodes in $Candidate$ by $IncInf$
    /* optimal pruning */
    if |$Candidate$| == $k$ - |$S$| then
        $S$ = $S \cup Candidate$
return $S$
```

Our approach to solving the influence maximization problem, named *ECE* (for **E**fficient **C**andidates **E**limination), is shown in Algorithm1. Conceptually, we estimate the minimum influence of a node and then select the k^{th} largest minimum influence to be the lower bound (denoted *LB*) of seed set. For a node v, if $MaxInf(v) < LB$, v cannot be selected as a seed, thus being pruned. In the following, we detail how to compute *LB*.

Step 1. We compute $MaxInf(v)$ for each node v and sort them according to their maximum influences to find the node with the largest maximum influence, say v. $MinInf(v)$ is set equally to $MaxInf(v)$.

Step 2. As we know $MinInf(v)$, the out-path nodes of v are marked as *unstable*. Notice that a node v' being marked as unstable means v' is correlated with a seed and the greedy order of the seed is prior than $GO(v')$. According to Lemma 1, the incremental influence spread of v' is smaller than its maximum influence because a seed that is always activated before node v' exists. Therefore, we do not estimate the minimum influence of an unstable node since the incremental influence spread of the node may dramatically decrease, caused by the seed making it marked as unstable.

Step 3. Therefore, we derive the next node with the largest maximum influence and not marked as unstable, say node v_x. Given a node v_y correlated with v_x, if we cannot guarantee whether $GO(v_x)$ is smaller than $GO(v_y)$, we assume $GO(v_y) < GO(v_x)$, i.e. $ap(v_y, S) = 1$, to decrease $IncInf(v_x, S)$. For all nodes correlated with v_x, we repeatedly check them for decrease $IncInf(v_\mathrm{x}, S)$. Eventually, we can compute the minimum influence of v_x. Different cases for updating $IncInf(v_\mathrm{x}, S)$ are discussed as fallows.

Case 1 (v_y is an out-path node of v_x and $IncInf(v_x, S) < MaxInf(v_y)$): This case happens in two situations. 1) $MaxInf(v_x) < MaxInf(v_y)$ and v_y has been marked as unstable or 2) $MaxInf(v_x) > MaxInf(v_y)$ and $IncInf(v_x, S)$ is decreased by assuming $ap(v_y, S) = 1$. Since $IncInf(v_x, S) < MaxInf(v_y)$, v_y may have a chance to be activated before v_x. Accordingly, we compute $IncInf(v_\mathrm{x}, S)$ by assuming $GO(v_\mathrm{y}) < GO(v_\mathrm{x})$, i.e. $ap(v_\mathrm{y}, S) = 1$. In addition, under this assumption, since the influence from v_x through the path with v_y is blocked by v_y, we need to eliminate the incremental influence from v_x to the out-neighbors of v_y.

Case 2 (Suppose v_y is an out-path node of v_x and $IncInf(v_x, S) > MaxInf(v_y)$): We consider the condition that there exists at least one seed to be a competition node of v_x with respect to v_y, making v_y unstable. v_x is definitely activated before v_y due to $IncInf(v_x, S) > MaxInf(v_y, S)$. The influence from v_x to v_y is decreased due to the competition nodes that mark v_y as unstable. Therefore, we consider $IncInf(v_x, S)$ with the condition of $GO(v') < GO(v_x)$, for any user $v' \in N^{competition}(v_x, v_y)$ and v' being marked as unstable. Notice that we do not compute the exactly $ap(v', S)$, but to consider *the worst case* of competition, i.e. $ap(v', S) = 1$.

Case 3 (Suppose v_y is an in-path node of v_x and $IncInf(v_x, S) > MaxInf(v_y)$): For each competition node v' of v_x with respect to v_y, we will check if $IncInf(v_x, S) < MaxInf(v')$. This happens in two situations. 1) $MaxInf(v_x) < MaxInf(v')$ and v' has been marked as unstable or 2) $MaxInf(v_x) > MaxInf(v')$ and $IncInf(v_x, S)$ has be decreased due to the estimation in the iteration. Since $IncInf(v_x, S) < MaxInf(v')$, v' may have a chance to be activated before v_x. Therefore, we consider $IncInf(v_x, S)$ with the condition of $GO(v') < GO(v_x)$.

Case 4 (Suppose v_y is an in-path node of v_x and $IncInf(v_x, S) < MaxInf(v_y)$): We consider $IncInf(v_x, S)$ with the condition of $GO(v_y) < GO(v_x)$. Then, the influence from the in-neighbors of v_y to v_x through v_y is then blocked by v_y. As a result, we do not need to check the in-neighbors of user v_y in the next iteration.

Step 4. Steps 2 and 3 mentioned above are repeated to continuously maintain the k^{th} largest minimum influence. The process is terminated when the current k^{th} largest minimum influence is larger than the next largest maximum influence. Eventually, *LB* is obtained.

4 Performance Evaluation

In this section, we perform a series of experiments on six real datasets to evaluate the performance of our approach. NetHEPT and NetPHY also used in [2][1][6] are both academic collaboration networks where nodes represent authors and edges represent coauthor relations. Epinions [7] is a who-trust-whom online social network where nodes represent users and edges represent that one user trust another. Slashdot0902 [7] is extracted from a technology-related news website where nodes represent users and edges represent friendship among users. Amazon0302 and Amazon0601 [7] are both product co-purchasing network data where nodes represent products and edges represent that one product is often purchased with another. Epinions and Amazon-0302 are also used in [1]. The detailed information of the datasets is listed in Table 1.

Table 1. Detailed information of real-world network datasets

	NetHEPT	NetPHY	Epinions	Slashdot 0902	Amazon 0302	Amazon 0601
Node	15,233	37,154	75,888	82,168	262,111	403,394
Edge	32,235	180,826	508,837	948,465	1,234,877	3,387,388
Maximal Out Degree	44	131	1,801	2,511	5	10
Maximal In Degree	60	152	3,035	2,553	420	2,751
Average Degree	2.12	4.87	6.71	11.54	4.71	8.40

To simulate the network with non-uniform propagation probabilities, we use the *weighted cascade* model proposed in [6]. Given an edge (v_1, v_2), the propagation probability of the edge, $pp(v_1, v_2)$, is defined as $1/in\text{-}degree(v_2)$. We compare ECE with Greedy [6], CELF [8], and PMIA [1] in the experiments. All of these methods are implemented in C++ and the experiments are performed on a PC with the Intel

Core 2 Quad 2.66GHz CPU, 8GB of main memory, and under the Ubuntu 11.04 operating system.

The performance of the pruning strategy in ECE is shown in Fig.5. In this experiment, NetHEPT is used to be the test dataset and the criteria of *candidate rate* is equal to $\frac{\text{candidate size}}{\text{\# of total nodes}}$. The curve of *Initial* represents the condition that we first compute the lower bound of the seed set. The curve of *Final* represents the condition that we have found all seeds. The curve of *Optimal* represents the condition that meets the optimal pruning, i.e. candidate size is equal to *k*. As shown in Fig. 3, the candidate rates are all less than 4%, which means ECE prunes more than 96% of nodes and the pruning strategy is very useful.

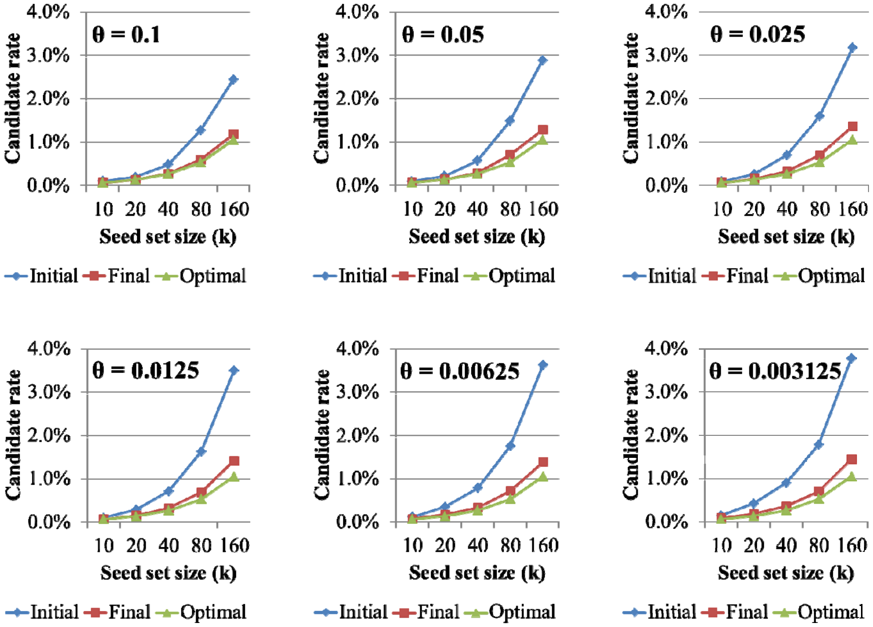


Fig. 3. Evaluation of the pruning strategy of ECE in NetHEPT dataset

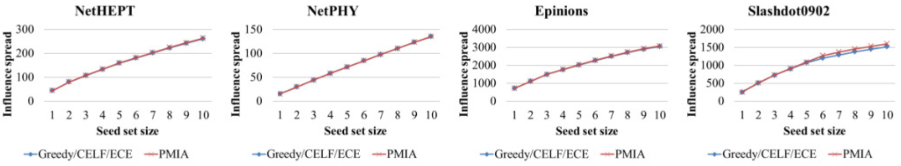


Fig. 4. Influence spread results in different datasets

The performance evaluation of each algorithm is as follows. We use NetHEPT, NetPHY, Epinions, and Slashdot0902 to be the test datasets since PMIA cannot be performed on Amazon0302 and Amazon0601 due to memory space limitation. *k* and

θ are set to 10 and 0.1, respectively. The influence spread of the seed set is shown in Fig. 4. The curves of the influence spread of each approach are close which means the qualities of the seed set found in each approach are similar. The running time of each approach on different datasets is shown in Fig.5. Each vertical bar is composed of different time blocks. The first time block *Initial* represents the processing time before finding the first seed. The other blocks represent the processing time on finding each seed. Notice that the real running time of Greedy is twice as long as shown in Fig. 5. Greedy does not have initial process and needs to re-compute the incremental influence spread of all nodes after finding a seed. Therefore, the processing time on finding each seed in Greedy is close. PMIA only needs to update the incremental influence spread of nodes correlated with the seeds after finding a seed, thus making the processing time of PMIA much smaller than that of Greedy. Since PMIA requires building index structures, its initial time is long. The concepts of CELF and ECE are similar. After finding a seed, both of them only re-compute the incremental influence spread of nodes that have potential to be seeds. Thus, their running time is much less than that of Greedy and PMIA.

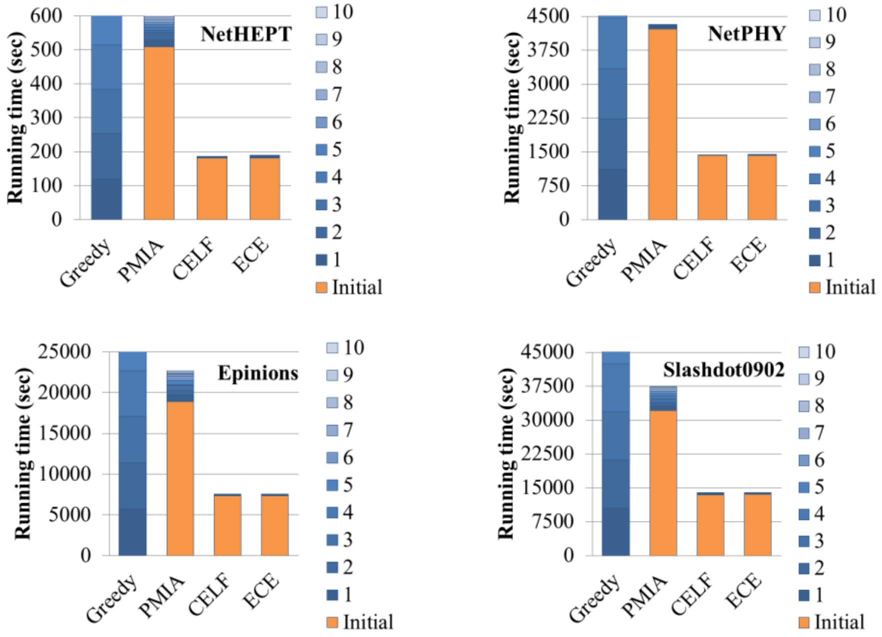


Fig. 5. Running time of different approaches in different datasets

The performance of ECE in each dataset is shown in Table 2. k and θ are set to 10 and 0.1, respectively. ECE achieves the optimal pruning in four datasets including NetHEPT, Epinions, Slashdot0902, and Amazon0302. In Fig. 5, the running time of CELF and that of ECE are very close. For each dataset, the average number of nodes being re-computed after finding a seed in CELF is 1.7 in NetHEPT, 1.0 in NetPHY, 1.3 in Epinions, 1.4 in Slashdot0902, 1.0 in Amazon0302, and 1.4 in Amazon0601. On the other hand, those of ECE are 4.8, 5.5, 0, 0, 0, and 6.8, respectively. Notice that in Epinions, Slashdot0902, and Amazon0302, ECE does not need to re-compute for any nodes after finding a seed. ECE can run faster than CELF in the datasets which make it to achieve the optimal pruning at initial.

Table 2. Pruning performance of ECE

	NetHEPT	NetPHY	Epinions	Slashdot 0902	Amazon 0302	Amazon 0601
Node	15,233	37,154	75,888	82,168	262,111	403,394
Initial Candidates	15	11	**10**	**10**	**10**	13
Pruning Rate	99.90%	99.97%	99.99%	99.99%	99.99%	99.99%
Time	0.69%	0.16%	1.42%	1.23%	0.01%	0.01%
Optimal Pruning	O	X	O	O	O	X

5 Conclusions

In this paper, we solve the problem of influence maximization in social networks. Finding the optimal solution is proved to be NP-hard. The original greedy algorithm needs to update almost all users' influence spread after selecting a user to the seed set. We propose an efficient approach ECE, based on a pruning strategy to avoid updating all users' influence spread after selecting a seed. The experiment results show that ECE prunes more than 96% of nodes in most of the cases. The running time of ECE is as efficient as CELF and much better than that of Greedy and that of PMIA. Since the efficiency of ECE is related to its pruning power if ECE achieves the optimal pruning in initial, i.e. the best performance of pruning, it is much faster than CELF.

References

1. Chen, W., Wang, C., Wang, Y.: Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 1029–1038 (2010)
2. Chen, W., Wang, Y., Yang, S.: Efficient Influence Maximization in Social Networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 199–208 (2009)
3. Chen, W., Yuan, Y., Sheu, S.T.: Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In: Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, pp. 88–97 (2010)
4. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: Learning Influence Probabilities in Social Networks. In: Proceedings of the Third International Conference on Web Search and Web Data Mining, New York, USA, pp. 241–250 (2010)

5. Goyal, A., Lu, W., Lakshmanan, L.V.S.: SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. In: Proceedings of the 11th IEEE International Conference on Data Mining, Vancouver, Canada, pp. 211–220 (2011)
6. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the Spread of Influence through a Social Network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 137–146 (2003)
7. Leskovec, J.: Stanford Large Network Dataset Collection, `http://snap.stanford.edu/data`
8. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective Outbreak Detection in Networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, pp. 420–429 (2007)
9. Pathak, N., Banerjee, A., Srivastava, J.: A Generalized Linear Threshold Model for Multiple Cascades. In: Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, pp. 965–970 (2010)
10. Narayanam, R., Narahari, Y.: A Shapley Value-Based Approach to Discover Influential Nodes in Social Networks. IEEE Transactions on Automation Science and Engineering 8(1), 130–147 (2011)

A Hybrid Prediction Algorithm for Traffic Speed Prediction

Bo-Wei Huang, Kun-Wei Wang, Ling-Yin Wei, and Wen-Chih Peng

National Chao-Tung University, Hsinchu, Taiwan, R.O.C.
{wallman.cs99g,kwwang.cs99g,lywei.cs,wcpeng}@nctu.edu.tw

Abstract. Many types of data can be regarded as time series data. Therefore time series data predictions are applied in a wide range of domains, such as investment, traffic prediction, etc. Traffic status prediction can be used for congestion avoidance and travel planning. We solve the problem of predicting traffic status by time series prediction. The time series data prediction problem is that given a query time and time series data, we intend to predict the data value at the query time. Usually, a query time will be a future time. In this paper, we propose a hybrid prediction algorithm which exploits regression-based and clustering-based prediction methods. Explicitly, regression-based prediction is accurate when the query time is not too far from the current time. Note that time series data may have some similar shapes or trends. To capture the similar shapes hidden in this data, we utilize clustering concepts. Using these clusters, we could further discover their sequential relationships. As such, if the query time is far away from the current time, we utilize the above cluster sequential relationships to predict the possible similar cluster. From the similar cluster, the data value at the query time is obtained. Note that the hybrid algorithm aggregates the above two methods using one threshold that decides which method to use. If the time difference between the query time and the current time is smaller than the prediction length threshold, hybrid prediction uses regression-based prediction. Otherwise, our hybrid algorithm uses clustering-based prediction. To prove our proposed methods, we have carried out a set of experiments on real data sets to compare the accuracy of the methods. The results of the experiments prove that our proposed methods are both accurate and practical.

1 Introduction

With the e-revolution, many businesses and organizations are now generating massive amounts of data which are constituted of values and time [12]. We call these data sets time series data. Therefore time series data predictions are applied in many domains, such as stock control, sales forecasting, financial risk management, traffic prediction, weather reporting, etc [2]. In this paper, we intend to predict the speed value in the future.

The common application of traffic status prediction is congestion avoidance. We can know which roads experience traffic jams from the prediction results,

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 281–296.
DOI: 10.1007/978-3-642-35452-6_30

and then we can take a road that is not congested. This technique can also be used in navigation. The navigation algorithm can consider the predicted traffic status, and can thus find a faster route. To predict the traffic status in the future, we must know the features of the traffic status data. It is almost a regular curve every day. The rising and falling of traffic speed results from human behavior, and the heaviest traffic flow is due to commuting which occurs on weekdays. Another important feature of traffic status data we observed is that congestion will not happen suddenly, as while it is happening, the traffic speed slows down gradually. Therefore, traffic status can be predicted by both short-term and long-term prediction methods.

In this work, we want to predict the value of traffic status in the future. To simplify our problem, we only focus on traffic speed. Assume that the traffic status data have patterns resulting from human behavior. Our proposed method can discover the patterns, and use these patterns to predict the time series value.

However, we do not focus on one of the applications. Thus we should solve the requirement using both short-term and long-term prediction. For example, short-term prediction is needed while driving on the road, and long-term prediction is needed while planning travel. Therefore we should deal with the disadvantages of both short-term and long-term prediction.

In this paper, we propose a hybrid prediction algorithm for time series data. This algorithm explores two types of prediction methods: regression-based and clustering-based. Regression-based prediction uses linear regression to construct the prediction model for a time series. This method is only accurate when the query time is close to the current time because regression-based prediction can only construct a linear model. The linear model goes farther and further from the time series with increasing time. For example, if a linear model of predicting traffic status is a decreasing line, the predicted value is a minus value when the query time is far away from the current time. However, a minus value for speed is not reasonable. Clustering-based prediction is a long-term prediction method because it uses subsequence of time series as its prediction unit. When using a clustering algorithm as a prediction method, however, some issues should be considered. These issues include: sequence segmenting, similarity function, clustering algorithm, and prediction method. The input time series data should be segmented before clustering because a single sequence cannot be clustered. Then, the clustering algorithm aggregates the similar sequences into a cluster according to the similarity function. The choice of clustering algorithm is important because different algorithms generate extremely different clusters. Finally the clusters can be used to predict values, including how to find frequent pattern and how to predict the value. However, clustering-based prediction methods are not more accurate than regression-based methods when the query is close to the current time. Thus, we propose a hybrid prediction method that exploits different prediction methods to avoid the disadvantages of previous approaches.

In the hybrid prediction method , different prediction methods are used according to different parameters. Linear regression is used for short-term prediction because it can make precise prediction when the query time is not too far from the current time. For long-term predictions, we use clustering-based prediction methods. The reason for choosing clustering-based prediction methods is that the data results from human behaviors are periodic. For example, people need to sleep every day, so obviously the loading of servers that provide services for humans is heavier in the daytime, which is a periodic pattern of resource utility for servers. Another example is that some people check the news and their e-mails when they get up. In fact the data we collected reflects those behaviors. Clustering methods can be used to aggregate similar time series together. Thus each cluster can be regarded as a pattern of data. Afterwards, we can use these patterns to predict the time series value in the future.

To summarize, the main contributions of this paper are as follows:

- Proposing two clustering-based prediction methods MCF and MPST which can predict the future values of time series in the long-term
- Proposing a hybrid prediction method which can choose the better prediction method from a regression-based prediction method and a clustering-based prediction method
- Using a clustering method on patterns and using the pattern sequence to predict time series data
- Carrying out performance evaluation of the proposed methods to prove the performance

The rest of this paper is organized as follows. We discuss the existing works and their disadvantages in Section 2. In Section 3, we define our problem and some terms more formally and clearly. In Section 4, we explain how our methods predict the data. In Section 5, we do some experiments to prove that our proposed methods show obvious improvement. Finally, we summarize the paper in Section 6.

2 Related Work

There are many existing works about time series data analysis or prediction. These works can be categorized as: time series prediction and other time series analysis. Works about time series prediction can be further divided into short-term prediction only and both short-term and long-term prediction. Short-term and long-term predictions differ in their accuracy for different prediction lengths. Short-term predictions are greatly affected by prediction length in that their accuracy decreases rapidly with the increase in prediction length.

In [6] and [20], the authors use neural networks to predict the future values of time series. Neural networks can construct a prediction model and do error correction for the model. However they can only predict the next value of a time series. In [17], Wang et al. proposed a two-phase iterative prediction approach which uses the pattern-based hidden Markov model. In these works, the methods can only predict the next state of data. In contrast, our proposed method can predict any prediction length. However, the concept of the hidden Markov model is like our proposed clustering-based prediction method MPST. The difference is that MPST uses subsequences as the prediction unit, while the hidden Markov model uses value as the prediction unit. Thus MPST can predict more than the hidden Markov model.

In [19], Xiong et al. proposed a resource analysis module and a resource allocation module using regression, a regression tree and boosting to analyze the resource usage in a cloud environment. Their approach uses additional information such as memory to predict the CPU usage. But in our problem we only want to use one dimensional time series data logs. Furthermore, this is a regression-based prediction method, so its accuracy is affected by the prediction length.

There are some works for long-term prediction such as [18] and [12]. In [18], Xing et al. proposed a method that classifies time series by prefix, and then calculates how to find the minimal prefix length. We can use this classification method to classify time series and predict the remaining values by the class's average series. However, this method needs a long prefix, and the required length of the prefix is not a fixed value. Therefore it cannot always give a prediction result, as it cannot make a prediction when the prefix length is not long enough. In [12], Ruta et al. proposed a generic architecture that can generate a prediction model based on any time series prediction method. This method can be short-term or long-term prediction method. It is decided by the method we give to the training process. However, it needs a great amount of time to train many models and to choose better ones to do the combination.

In [9], Ha et al. proposed a method to predict stocks. Their prediction is based on data increasing and decreasing. Therefore this method can be used for prediction of data which is affected by increasing and decreasing. However, not all time series data have such a pattern. For example, stocks and temperatures can be predicted using this method, but traffic speed cannot.

In our proposed clustering-based prediction method MPST, we try to symbolize time series. This problem is called "symbolic time series analysis" in other works. There are many advantages of symbolization, such as reducing space, noise filtering, and getting more temporal information[13]. The methods of symbolic time series analysis can be categorized into two domains: quantization and temporal segmentation. The main differences between these two domains are the breakpoints. The breakpoints of quantization are in the amplitude domain, while for temporal segmentation they are in the temporal domain. The existing quantization methods include SAX [7] and PERSIST [10]. These methods symbolize the time series by amplitude, but they cannot distinguish the shape of the time series. For example, the increasing and decreasing line can be marked with the same symbol. Therefore these methods cannot be applied to our problem. In the temporal segmentation method ACA [21], Zhou et al. used clustering to symbolize the time series, as we do in this work.

To the best of our knowledge, we are the first to consider the time difference between "the query time" and "the current time". If the query time is close to the current time, a regression method has better accuracy. Furthermore, none of the existing methods we mentioned above can solve this problem. Therefore we propose a hybrid prediction algorithm which exploits both the regression prediction method and the clustering-based prediction method.

3 Preliminaries

Before describing our methods, we should define the notations we used and the problem want to solve. In this section we explain the definition of time series, the problem, and issues.

Definition 1. *A univariate time series is a sequence of pairs (value,timestamp). The timestamps in the sequence are ordered in ascending order, meanings that the time point of the accompanying value is reduced. The values are reduced from continuous data such as traffic status, temperature, or from discrete data such as sales data. We denote x_i as the value at timestamp i in time series x.*

For example, a traffic status time series y= {(40, 0), (50, 15), (60, 30), (60, 45)} is a sequence reduced every 15 minutes from a continuous traffic status. We can present the first value as $y_0 = 40$, the second value as $y_{15} = 50$, and other values can be presented in this way.

Definition 2. *A subsequence $x_{i,j}$ is a subset of x which means the sequence of values from the time series x which has a timestamp bigger than i and smaller than j in ascending order.*

For example, $y_{0,30}$=(40, 50, 60) is a subsequence of y. The time series y= {(40, 0), (50, 15), (60, 30), (60, 45)} can be segmented into two subsequences $y_{0,29}$=(40, 50) and $y_{30,45}$=(60, 60).

Definition 3. *Time series period means a time interval within which some pattern in the time series appears.*

We will segment the time series sequence into subsequences by the *time series period* in our proposed methods. All the segmented subsequences have the same length of the *time series period.* For example, the time series z= {(40, 0), (50, 15), (60, 30), (60, 45), (40, 60), (50, 75), (60, 90), (60, 105)} has a *time series period* of 45. The pattern (40, 50, 60, 60) appears two times in this time series. Therefore we can segment z into two subsequence $z_{0,45}$ and $z_{46,105}$. We can find that these two subsequences have the same value sequence. This is why we need to find the *time series period* and segment the time series.

Definition 4. *Prediction length $l_{predict}$ is the distance from the current time t_c to the query time t_q. Therefore, $l_{predict} = t_q$ - t_c.*

Prediction length is an important parameter in the hybrid prediction method. Many prediction methods have different degrees of accuracy for different prediction lengths.

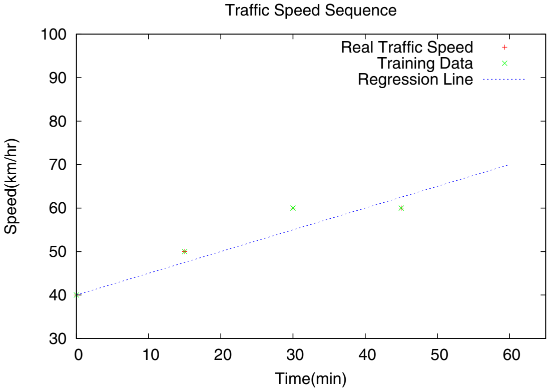


Fig. 1. Regression prediction for traffic status

3.1 Problem Definition

Given a univariate time series data of traffic status S, the current time t_c, and the query time t_q, predict the value S_{t_q} at the query time t_q. To achieve this goal, we assign the training data length l_{train} to construct a prediction model. Therefore the total inputs are as follows: time series data logs S, length of training data l_{train}, current time t_c and query time t_q. The output is the value at the query time S_{t_q}.

Definition 5. *Current time t_c is the timestamp of the last value of training data S.*

For example, given the time series z= {(40, 0), (50, 15), (60, 30), (60, 45), (40, 60), (50, 75), (60, 90), (60, 105)}, the current time t_c is 105. In the prediction process, we predict the value from this time point. In other words, t_q should be bigger than t_c.

For example, given a time series of traffic status log y= {(40, 0), (50, 15), (60, 30), (60, 45)}, then the current time is 45. If we predict the speed at 60 using training data length of 45 minutes, the predicted speed at time 60 is 70, as shown in Fig.1.

4 Prediction Methods

We propose three prediction methods in this paper. These methods can be categorized as regression-based, clustering-based and hybrid. In this section, we will explain the reasons for using these methods and their details.

4.1 Regression

The regression-based methods are well-known solutions for finding the nearest polynomial of the log, and are applied in time series analysis. There are some different regression algorithms such as simple linear regression, multiple linear regression, multilevel linear regression and piecewise linear regression.

Simple linear regression is the simplest regression method that can only be used on a univariate time series. Simple regression constructs the regression line, where the sum of the distances between data points and the regression line is the smallest. The formula of the regression line of simple linear regression is shown as Equation 1, where y in the equation means the value and x means the time. The coefficients α and β are to eliminate the sum of the distances between data points and the regression line. For example, Figure 1 shows two different regression lines that have the smallest distances to the traffic speed log time series.

$$y = \alpha x + \beta \tag{1}$$

Although there are many regression-based prediction methods, we use the simple linear regression in this paper. The reason is that we focus on univariate time series, and all we need is a short-term prediction method. Thus we choose the simplest method of regression. This method is the best method to represent the features of regression-based methods.

4.2 Clustering-Based Prediction

Clustering-based prediction can predict values far away from the current time because does not use a monotonic line to make a prediction. Thus, the problem of regression-based methods does not occur. In clustering-based prediction methods, we find the frequent patterns which are segments of time series data which appear many times. These frequent patterns are always produced by human behaviors, such as traffic congestion resulting from going to work everyday.

Definition 6. *A frequent pattern is a cluster of subsequences whose size is more than a specific value minimum support.*

Intuitively, a frequent pattern is a *subsequence* that appears many times (or at least more than the *minimum support*). However, in practice, a a time series pattern does not always have the same values. For example, the speed values of 60 and 61 are regarded as the same in a *frequent pattern*. Thus we use the cluster to institute using subsequences. When we need the values of the *frequent pattern*, we can use the average values of all subsequences in the cluster.

The flow of our prediction method is as follows. First, the time series data is segmented into subsequences with the length of the *time series period*. The time series data log is only one sequence of values, but clustering needs many sequences to be assigned to a cluster, so we need to segment the time series. In this paper, we use a fixed period to segment the time series data log, whereby each subsequence has the same length. The period decides whether these subsequences represent the time series pattern. To find the period of the time series data log, we can use the method proposed in [16]. However, how to find the time series period is not our focus. To get the best performance to prove the practicality of our proposed methods, we use an experiment to find the time series period.

Another important issue is how to compute the similarity between time series sequences. The similarity is like the distance between points, so we also call the similarity the distance function. When we cluster data points on a plane, we use the distance between the data points as the similarity function, and aggregate close data points into the same cluster. In other words, we want to calculate the distance between time series subsequences. There are many existing methods for calculating time series similarity, such as dynamic time warping (DTW) [1], longest common subsequences (LCSS) [15], edit distance on real sequence (EDR) [4], and edit distance with real penalty (ERP) [3]. After our experiments, we decided to use EDR as the similarity function in this paper.

After we have the distance function, we should choose a clustering algorithm. There are many existing clustering algorithms. K-means[8], density-based spatial clustering (DBSCAN) [5], and hierarchical clustering are commonly seen clustering algorithms. K-means is a classical centroid-based clustering algorithm. K-means divides the data points into circles, and minimizes the summation of distances from points to each circle's center. DBSCAN is a density-based clustering algorithm, which expands clusters by density, and can therefor generate clusters of any shape. But if we want to use a clustering algorithm to find a pattern, DBSCAN is not an ideal choice, because the maximal distance of points in the clusters is very large in some shapes. For example, if the DBSCAN algorithm generates a long-thin cluster, the points in its two extremities will be very far apart. This causes the time series in the same cluster to not always be similar to each other. Therefore, we use K-means as our clustering algorithm in this paper.

Finally, we have clusters generated from the clustering algorithm. We should decide how to use the clusters to predict the future values. How to use the clustering result is the final issue of clustering-based prediction. In the naive method, we can use the cluster which has the maximal size to predict the time series because it has the maximal occurrence probability. Based on this method, we propose the scheme maximal cluster first (MCF). Another method is based on probabilistic suffix trees (PST). we propose A scheme multi-item in a time probabilistic suffix tree (MPST). We explain the details of these methods in the next subsections.

4.3 Scheme MCF

Scheme MCF is a clustering-based prediction algorithm. In our observation, most time series have different patterns in one period. For example, people need to go to work, eat lunch, go home and sleep everyday. Therefore we segment the traffic status time series of one day into a number of time slots, and then we cluster each time slot individually. After clustering, there are maximal clusters in each time slot. MCF finds the maximal cluster of each time slot. Thus if we want to predict the value of the query time, MCF computes which time slot the query time belongs to, then uses the maximal cluster of the time slot to predict the value of the query time.

For example, given the traffic status data from 5/1 to 5/14, we segment one day into 4 time slots (in other words, each time slot is 6 hours). If we want to predict the value for 5/15 9:00, MCF finds the maximal cluster of the time slot 6:00 11:59 because 9:00 is in this slot, and then MCF uses the value of 9:00 in the average of the cluster as the predicted value. If we predict the value for 5/16 14:00, MCF chooses the maximal cluster of the time slot 12:00 17:59 to make a prediction.

However, MCF cannot be applied for every type of data. MCF can predict the data with patterns which occur at fixed intervals, but cannot predict the patterns occur at other intervals. Therefore MCF fits the data which is affected directly by human behaviors.

4.4 Scheme MPST

Even though we have MCF prediction, to use this prediction method we have to find a period during which some patterns will repeat. Therefore MCF is not a general method. We need a more general prediction method, so we propose MPST prediction, which is based on probabilistic suffix tree (PST), to find the pattern of *pattern sequences*. MPST is a more general clustering-based method, and we explain the details of this prediction method in this section.

Probabilistic suffix trees (PST) were proposed by Dana Ron in 1996 [11]. PST is a suffix tree in which the nodes present the suffix pattern and each node has a probability list which presents the occurrence probability of the next element. For example, we have already constructed a PST which is shown in Fig. 2. All the elements are a, b, c. If we see a string ”*abcabca*”, the last element is ”*a*”, so we go from the root to the node ”*a*”. The probability of elements are 0.1, 0.4 and 0.5. If we want to use a longer suffix string to predict the next element, we go from ”*a*” to ”*ca*” and then to ”*bca*”.

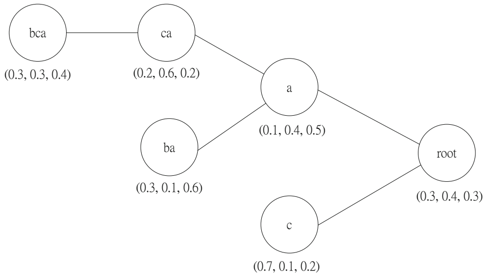


Fig. 2. An example of PST

PST can predict the next element of the sequence; therefore we can mark each subsequence with a pattern ID and use PST to predict the sequence. However, the subsequence is not always similar to only one pattern. In this paper we modified PST to be a multi-element in a time PST (MPST). MPST can predict a pattern sequence which has more than one pattern in a time slot.

Definition 7. *A pattern sequence is a sequence of pattern IDs. Pattern ID is defined by a frequent pattern finding algorithm. A pattern sequence can have more than one pattern ID in one time point if that subsequence is similar to more than one pattern.*

For example, for a *pattern sequence* that is denoted as "$a(ab)cab$", "(ab)" means the second subsequence is similar to patterns a and b. In our method, the pattern sequence is generated from the clustering algorithm. After the clustering process, we can see the average time series of each cluster as a pattern. Thus we can prune the infrequent patterns by the cluster size. In our experiments, we found that the cluster size is either extremely high or extremely low, and therefore we can prune the infrequent patterns easily. We then compute each distance of subsequences to the patterns. If the distance is small enough, the subsequence will be marked with corresponding pattern IDs. After all the time series is processed, the sequence of pattern IDs is the pattern sequence of this time series.

To generate the pattern sequence, we should use the clustering result. The parameters of the clustering algorithm affect the result. However, we observed that if the number of clusters is more than the number of patterns in the time series, there are many clusters with a size of 1. We use K-means as our clustering algorithm, so we can control the number of clusters we generate by parameter k. Therefore the value of k is not important if k is big enough.

The difference between PST and MPST is that MPST can predict the sequential data with multi-elements in one time point. When MPST predicts the sequential data, it goes through every node that may match the suffix string. For example, if MPST sees a suffix string "$b(bc)a$", it will go to the "bba" and "bca" nodes and summarize the probabilities. To do this, we modify the probability attribute to the number of occurrences so we can add the numbers of occurrence to compute the probability of the next element.

4.5 Hybrid Prediction

We found that regression is not accurate when the query time is far from the current time. When it is close to the current time, regression-based prediction is better than clustering-based prediction. Therefore we propose hybrid prediction which can combine the advantages of regression-based and clustering-based prediction.

Hybrid prediction is a combination of regression-based prediction and clustering-based prediction. It uses clustering-based prediction when the query time is far from current time and uses regression-based prediction when it is close to the current time. We define a threshold ε to decide which prediction method. When the time difference between the query time t_q and the current time t_c is bigger than ε, the hybrid prediction algorithm uses the clustering-based prediction method MPST. Otherwise our hybrid prediction algorithm uses the regression prediction method.

The value of ε is decided by experiments. We can make an automatic ε trainer by performing predictions with different t_q. The cross point where clustering-based prediction becomes better than regression-based prediction can be found out easily. However in this work we set ε manually.

5 Performance Evaluation

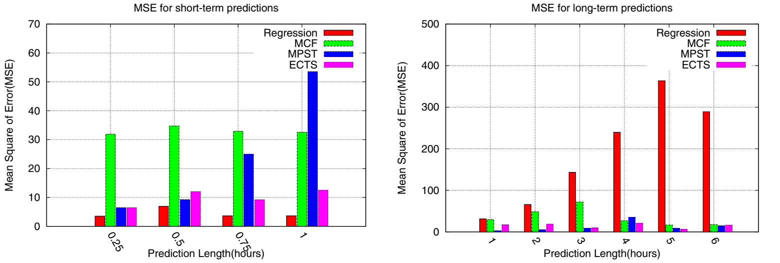
In order to evaluate the accuracy and efficiency of the proposed methods, we have conducted extensive experiments on real data sets under a number of configurations. In this section, we have done many experiments to prove that our proposed methods are practical and accurate for real data. We also compare our proposed method with one existing method, ECTS, proposed in [18].

We used a real traffic data set in our experiments. The data set is the traffic speed log of a freeway segment in Taiwan. This segment is from Hsinchu to Jubei which has the highest throughput in Taiwan. There is a traffic burst at commuting time on weekdays. Therefore this data set has the patterns we expected. We spent more than three months obtaining the data from government sensors.

However, the training data length is not ”the longer the better”. Too long training data that is too long contains data which is out of date. In our experiments, we used two weeks of traffic status as the training data for the clustering-based methods, while for the regression, we used the same prediction length as the training data length.

In Fig. 3, we compare the prediction methods for short-term and long-term prediction. In these two experiments, the parameters of MCF and MPST are the best parameters in the experiment, as shown in Fig. 6. The difference between short-term and long-term is the prediction length, and the prediction is short-term if it is less than 1 hour. This breakpoint is decided from our data set. The sampling rate of our data set is 1 minute. In the short-term predictions, each is a continuous prediction for the values of 15 minutes after the prediction length. For example, the prediction length of 0.5 hour means to predict the values at $t_c + 31$ $t_c + 45$. As the results show in Fig. 3(a), regression is the best prediction method, as we mentioned, for short-term predictions. For long-term predictions, each experiment is continuous predictions for the values of 1 hour. Although MPST is not always the most accurate prediction method, it is better than regression, MCF and ECTS on most occasions. Although the accuracy of MPST is close to that of ECTS, our MPST is better than ECTS on the tracing length. The tracing length is how far we should trace back the time series to predict a value. In Fig. 5, we compared the average tracing length of MPST and ECTS. In this experiment, ECTS needs a longer tracing length, and the needed tracing length is not fixed. Therefore our MPST uses less data than ECTS, and the accuracy of MPST is close to that of ECTS.

To summarize the above results, we did an experiment of 6 hours continuous prediction, the results of which are shown in Fig. 4. In this experiment, the parameters of the prediction methods are the same as in the above experiments. The hybrid prediction in this experiment is the most accurate prediction method.



(a) Prediction results for short-term prediction

(b) Prediction results for long-term prediction

Fig. 3. Prediction accuracy for traffic status

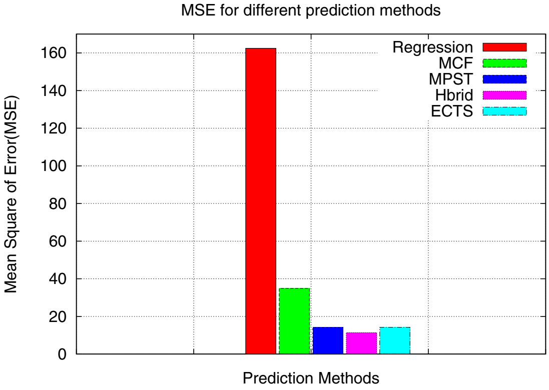


Fig. 4. Prediction results for 6 hours continuous prediction

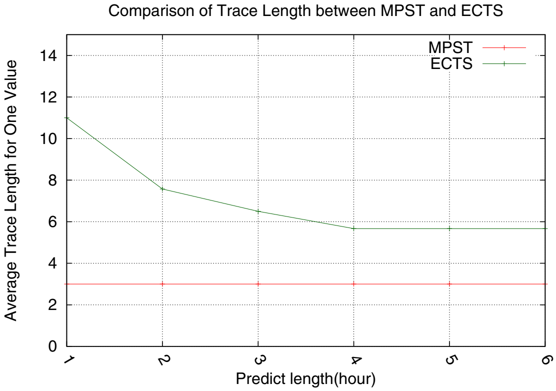
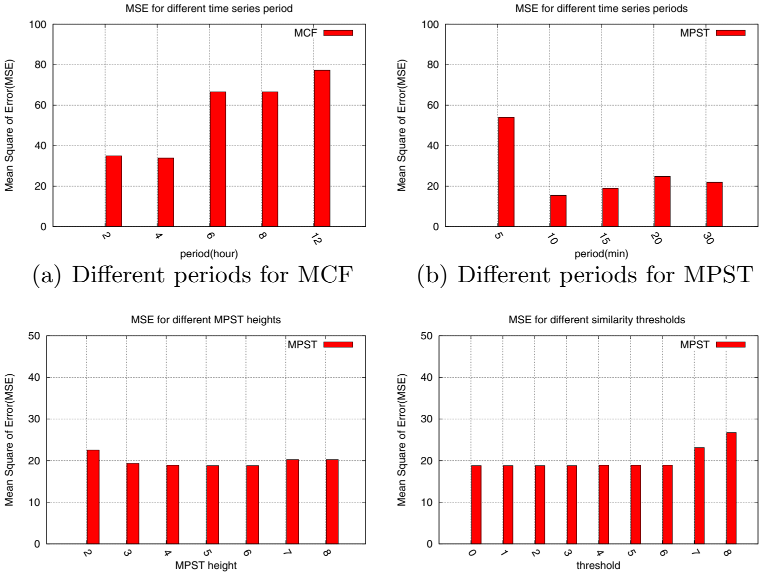


Fig. 5. Comparison average trace length of MPST and ECTS

In this experiment the threshold of hybrid prediction is 1 hour; therefore we use regression when the prediction length is less than 1 hour and MPST when it exceeds 1 hour.

We also carried out the experiment using our clustering-based methods. In Fig. 6, we use different parameters for the clustering-based prediction methods MCF and MPST to investigate the effect of the parameters. In MCF, there are only two parameters: k and time series period. The parameter k is for clustering algorithm k-means. The value of k affects the clustering result of both MCF and MPST; therefore we will discuss this issue with MPST. The experiment result for the time series period of MCF is shown in Fig. 6(a). As can be seen, the best time series period for MCF when using our data set is 4 hours. In Fig. 6(b), we used different time series periods for MPST. Because it predicts the future value using several recent patterns, the period is shorter than in MCF. If the period is 30 minutes and the MPST height is 3, MPST will use the recent data of 1.5 hours to predict the next 30 minutes. In this experiment, we found that 10 minutes is the best time series period for MPST when using our data set. The height of MPST is another important parameter. We mentioned that the prediction result is decided from the recent patterns. The height of MPST affects the maximal patterns we use to make predictions. The effect of MPST heights is shown in Fig. 6(c). The effect is not obvious in this experiment because we used



(a) Different periods for MCF

(b) Different periods for MPST

(c) Different tree heights for MPST

(d) Different pattern similarity thresholds for MPST

Fig. 6. Effects of parameters for clustering-based prediction methods

the number of occurrences to make the predictions. The number of occurrences is lower and lower in the lower nodes; therefore the effectiveness is also lower. Thus the MPST height is not so important as period. The last parameter of MPST is the pattern similarity threshold. This is the parameter which decides whether the subsequence is like the pattern in the symbolization process. The value of the pattern similarity threshold means: if the distance between the subsequence and the pattern is lower than the threshold, the subsequence is similar to this pattern. As shown in Fig. 6(d), the effect of the threshold is stepwise. When the threshold is greater than 6, the prediction result is inaccurate. This is because a threshold that is too big causes MPST to use patterns which are not similar to the subsequence to predict future values.

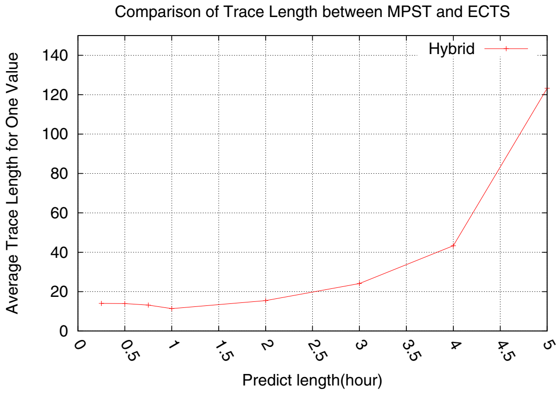


Fig. 7. Different thresholds for the hybrid prediction

For the hybrid prediction method, we tried different threshold values. The experiment results are shown in Fig. 7, The best threshold is 1 hour, therefore we use 1 hour as the breakpoint of the short-term and long-term predictions. In this experiment, a bigger threshold means using much regression; thus accuracy is low with a big threshold.

In these experiments, we verified the performance of the proposed methods. We also prove that the regression-based method fits for short-term prediction, clustering-based prediction fits for long-term prediction.

6 Conclusion

In this paper, we studied the existing time series analysis works, and found that they cannot predict the value of different data types. Therefore we proposed 3 prediction methods: regression-based prediction, clustering-based prediction and hybrid prediction. Regression-based prediction is accurate when the query time is not too far from the current time, while clustering-based predictions are more accurate when the query time is far from the current time. To combine the advantages of the above two methods, we proposed hybrid prediction using a prediction length threshold that decides which method to use. We also did many experiments on many real data sets such as traffic status logs to prove the accuracy of our proposed methods. In the experimental results, we verified the correction of the properties of the regression-based prediction and clustering-based prediction methods. We also proved that our hybrid prediction is more accurate than other prediction methods.

References

1. Bemdt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop, pp. 229–248 (1994)
2. Chatfield, C.: Time-Series Forecasting. Chapman and Hall/CRC (2001)
3. Chen, L., Ng, R.T.: On the marriage of lp-norms and edit distance. In: VLDB (2004)
4. Chen, L., Ozsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: SIGMOD, pp. 491–502 (2005)
5. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD (1996)
6. Giles, C.L., Lawrence, S., Tsoi, A.C.: Noisy time series prediction using recurrent neural networks and grammatical inference. Machine Learning (2001)
7. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Mining and Knowledge Discovery 15(2), 107–144 (2007)
8. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory (1982)
9. Ha, Y.-M., Park, S., Kim, S.-W., Won, J.-I., Yoon, J.-H.: Rule discovery and matching in stock databases. In: IEEE International Computer Software and Applications Conference (2008)
10. Morchen, F., Ultsch, A., Hoos, O.: Extracting interpretable muscle activation patterns with time series knowledge mining. International Journal of Knowledge Based Intelligent Engineering Systems 9(3), 197 (2005)
11. Ron, D., Singer, Y., Tishby, N.: The power of amnesia: Learning probabilistic automata with variable memory length. Machine Leraning (1996)
12. Ruta, D., Gabrys, B., Lemke, C.: A generic multilevel architecture for time series prediction. TKDE (2010)
13. Sant'Anna, A., Wickstrom, N.: Symbolization of time-series: An evaluation of sax, persist, and aca. In: 2011 4th International Congress on Image and Signal Processing (CISP), vol. 4, pp. 2223–2228. IEEE (2011)
14. Tsay, R.S.: Analysis of Financial Time Series. John Wiley&Sons (2002)
15. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: ICDE, pp. 673–684 (2002)
16. Vlachos, M., Yu, P., Castelli, V.: On periodicity detection and structural periodic similarity. In: SDM (2005)
17. Wang, P., Wang, H., Wang, W.: Finding semantics in time series. In: SIGMOD (2011)
18. Xing, Z., Pei, J., Yu, P.S.: Early classification on time series. In: KAIS (2012)

19. Xiong, P., Chi, Y., Zhu, S., Moon, H.J., Pu, C., Hacıgümüs, H.: Intelligent management of virtualized resources for database systems in cloud environment. In: SDM (2005)
20. Yuan, H., Liu, J., Pu, H., Mao, J., Gao, S.: Prediction of chaotic ferroresonance time series based on the dynamic fuzzy neural network. In: ECAC (2012)
21. Zhou, F., Torre, F., Hodgins, J.K.: Aligned cluster analysis for temporal segmentation of human motion. In: 8th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2008, pp. 1–7. IEEE (2008)

Online Forum Thread Retrieval Using Pseudo Cluster Selection and Voting Techniques

Ameer Tawfik Albaham and Naomie Salim

Faculty of Computer Science and Information System,
Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
ameer.tawfik@gmail.com, naomie@utm.my

Abstract. Online forums facilitate knowledge seeking and sharing on the Web. However, the shared knowledge is not fully utilized due to information overload. Thread retrieval is one method to overcome information overload. In this paper, we propose a model that combines two existing approaches: the Pseudo Cluster Selection and the Voting Techniques. In both, a retrieval system first scores a list of messages and then ranks threads by aggregating their scored messages. They differ on what and how to aggregate. The pseudo cluster selection focuses on input, while voting techniques focus on the aggregation method. Our combined models focus on the input and the aggregation methods. The result shows that some combined models are statistically superior to baseline methods.

Keywords: Forum thread search, Voting techniques.

1 Introduction

Online forums are user-generated content platforms, which enable users to build virtual communities. In these communities, users interact with each other to seek and share knowledge. The interaction happens through exchanging information in a form of discussions. A user starts a discussion by posting an initial message. Then, replies to it are contributed by the other users. A pair of an initial message and the set of its reply messages builds a thread.

Thread retrieval helps forums' end users to find information satisfying their needs. However, the challenge is that threads are not text; they are collections of "messages"—the initial and the reply messages. Therefore, given a user query, a retrieval system needs to utilize the text of the messages to rank threads. The problem of the thread retrieval is similar to the problem of the blog site retrieval[11,6]. In blog site retrieval, given a query, we leverage the blogs' postings in order to rank blogs. An analogy between these two retrieval problems is that threads are the blogs, and messages are the postings.

Because of the thread retrieval's resemblance to the blog site retrieval, researchers, as in [5] and [10], have adapted techniques from blog distillation, such as [4,11], to thread search; and, the adapted models performed well. Motivated by this fact, we propose to combine two known techniques on blog site retrieval:

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 297–306.
DOI: 10.1007/978-3-642-35452-6_31

Pseudo Cluster Selection(PCS)[11] and Voting Techniques(VT) [6]; then, we apply the combined model to thread search.

Both methods first rank a list of postings and then aggregate the postings' scores to rank their parent blogs. However, they differ in two aspects: the input and the aggregation method. PCS focuses on the top k ranked postings from each blog, whereas VT considers all ranked postings. Furthermore, PCS uses the geometric mean to fuse scores, while VT adapts various data fusion methods, such as CombMAX[12] and expCombSUM[9], to the blog distillation task. In using CombMAX, blogs are ranked based on their best scoring posting; in using expCombSUM, a blog is scored based on the sum of the exponential values of the blog's ranked postings' relevance scores. It was reported that the voting methods that favor blogs with highly ranked postings performed better than the other voting methods[6]. This characteristic is similar to PCS's emphasis on highly ranked postings because it considers only the top k postings.

In thread retrieval, [3] reported that scoring threads using the maximum score of their ranked messages is superior to using the arithmetic mean of the scores; and, PCS is statistically superior to both methods. Note that the arithmetic mean will be affected by the messages with low scores, whereas, the maximum score method favors threads with highly ranked messages. In addition, CombMAX is a special case of PCS($k = 1$)[5]. In other words, focusing on highly ranked messages improve the retrieval performance. Therefore, in this study, we hypothesize that voting methods can achieve better performance by focusing on only the top k ranked messages.

2 Related Work

The voting approach to the blog distillation task is inspired by works on data fusion[12,9] and expert finding [7]. A data fusion technique aims to combine several ranked lists of documents generated by different retrieval methods into a unified list [1]. In addition, retrieval methods that retrieved a document are voters for that document. Then, a data fusion technique is used to fuse these votes. Data fusion techniques are categorized into score based and rank based aggregation methods. The score based methods — such as CombMAX[12], use the relevance scores of documents, whereas the rank based methods, such as BordaFuse [1], utilize the ranking positions of these documents.

The expert finding task is defined as retrieving a list of people who are experts in the topic of the user query[7]. To estimate the expertise of a person, methods in this task leverage the documents that are associated with or written by that person[7]. [7] models the problem of expert finding as a data fusion problem. Motivated by the success of the voting model in the expert finding task[7], [6] models the blog site retrieval as voting process as well. The connection between these tasks is that: in both, we first rank a list of documents with respect to a user query using an underlined text retrieval model, then, each ranked document is considered as a vote supporting the relevance of its asscoiated "object"— the person or the blog. Indeed, in both tasks, the voting approach was found to be statistically superior to baseline methods [7,6]. Similarly, in the Pseudo Cluster Selection(PCS) method [11], we first rank a list of postings, and then an aggregation method is applied. However, in PCS, only the top k ranked postings from each blog are used to estimate the blog's relevance. If the number of the blog's ranked postings is less than k , then a padding is supplied by using the minimum score of all ranked postings.

In thread retrieval, similar approach has been applied as well [5,3,10]; that is ranking threads by aggregating their message relevance scores. [5] proposed two strategies to rank threads: inclusive and selective. The inclusive strategy utilizes evidences from all messages in order to rank parent threads. Two models from previous work on blog site retrieval [4] were adapted to thread search: the large document and the small document models. The large document model creates a virtual document for each thread by concatenating the thread's message text, then it scores threads based on their virtual document relevance to the query. In contrast, the small document model defines a thread as a collection of text units (messages). Then, it scores threads by adding up their messages' similarity scores. In contrast to the inclusive strategy, [5]'s selective strategy treats threads as a collection of messages; and it uses only few messages to rank threads. Three selective methods were used. The first one is scoring threads using only the initial message relevance score. The second method scores threads by taking the maximum score of their message relevance scores. The third method is based on PCS. Generally, it was found that the selective models are statistically superior to the inclusive models [5,3] especially the PCS method. Our work extends the Pesudo Cluster Selection method by investigating more aggregation methods.

Another line of research is the multiple context retrieval approach [10]. This approach treats a thread as a collection of several "local contexts" — types of self-contained text units. Four contexts were proposed: posts, pairs, dialogues and the entire thread. The thread and post contexts are identical to [5]'s virtual document and message concepts respectively. The pair and the dialogue contexts exploit the conversational relationship between messages to build text units. To rank threads using the post, pair and dialogue contexts, PCS was used. It was observed that retrieval using the dialogue context outperformed retrieval using other contexts. Additionally, the weighted product between the thread context and the post, pair or the dialogue contexts achieved better performance than using individual contexts. In our work, we are focusing on how to combine the ranked contexts' scores— using the message context. Therefore, our work is complementary to [5]'s work.

The third line of work is the structure based document retrieval [2]. In this approach, a thread consists of a collection of structural components: the title, the initial message and the reply messages set. In this representation, the thread relevance to the user query is estimated using [8]'s inference network framework. Our work can be applied to [2]'s representation as well. We could use [2]'s inference based relevance score the same way the thread context score was used in [10].

3 Combined Models

In this work, given a query $Q = \{q_1, q2, ..., q_n\}$, we first rank a list of messages R_Q with respect to Q. Then, we score threads by aggregating their top k ranked messages' scores or ranks. [10] stated that the pair and the dialogue contexts must be extracted using thread discovery techniques, and retrieval using inaccurate extracted contexts will hurt the performance significantly. As a result, we use only the message context.

In estimating the relevance between Q and a message M, we employ the query language model assuming term independence, uniform probability distribution for M and Dirichlet smoothing[13] as follows:

$$P(Q|M) = \prod_{q \in Q} \left(\frac{n(q, M) + \mu P(q|C)}{|M|} \right)^{n(q,Q)} \tag{1}$$

where q is a query term, μ is the smoothing parameter. $n(q, M)$ and $n(q, Q)$ are the term frequencies of q in M and Q respectively, $|M|$ is the number of tokens in M and $P(q|C)$ is the collection language model.

To rank threads, the twelve aggregation methods proposed by [7] are adapted: Votes, Reciprocal Rank(RR), Bordafuse, CombMIN, CombMAX, CombMED, CombSUM, CombANZ, CombMNZ, expCombSUM, expCombANZ and expCombMNZ. In addition to these methods, this study uses "CombGNZ"— the geometric mean of the relevance scores. We use this method because it is the aggregation method employed by the Pseudo Cluster Selection method [11].

Note that our combined models aggregate only the top k ranked messages from each thread in order to infer a thread's relevance score. Let R_T denote the set of all ranked messages from a thread T, and $R_{T,k}$ denote the set of the top k ranked messages. However, if the size of R_T is less than k, then the "set with less members" is used— either R_T or $R_{T,k}$, we denote to it by $R_{T,L}$. Using $R_{T,L}$, we can score threads using any of the following methods:

$$\text{Votes}_k(Q, T) = |R_{T,L}| \tag{2}$$

$$\text{RR}_k(Q, T) = \sum_{M \in R_{T,L}} \frac{1}{\text{rank}(Q, M)} \tag{3}$$

$$\text{BordaFuse}_k(Q, T) = \sum_{M \in R_{T,L}} |R_Q| - \text{rank}(Q, M) \tag{4}$$

$$\text{CombMIN}_k(Q, T) = MIN_{M \in R_{T,L}} P(Q|M) \tag{5}$$

$$\text{CombMED}_k(Q, T) = Median_{M \in R_{T,L}} P(Q|M) \tag{6}$$

$$\text{CombSUM}_k(Q, T) = \sum_{M \in R_{T,L}} P(Q|M) \tag{7}$$

$$\text{CombANZ}_k(Q,T) = \frac{1}{|R_{T,L}|} \times \sum_{M \in R_{T,L}} P(Q|M) \tag{8}$$

$$\text{CombGNZ}_k(Q,T) = \left(\prod_{M \in R_{T,L}} P(Q|M) \right)^{\frac{1}{|R_{T,L}|}} \tag{9}$$

$$\text{CombMNZ}_k(Q,T) = |R_{T,L}| \times \sum_{M \in R_{T,L}} P(Q|M) \tag{10}$$

$$\text{expCombSUM}_k(Q,T) = \sum_{M \in R_{T,L}} \exp(P(Q|M)) \tag{11}$$

$$\text{expCombANZ}_k(Q,T) = \frac{1}{|R_{T,L}|} \times \sum_{M \in R_{T,L}} \exp(P(Q|M)) \tag{12}$$

$$\text{expCombMNZ}_k(Q,T) = |R_{T,L}| \times \sum_{M \in R_{T,L}} \exp(P(Q|M)) \tag{13}$$

where rank(Q,M) is the rank of the message M on R_Q, $|R_Q|$ is the size of R_Q and $|R_{T,L}|$ is the size of $R_{T,L}$. This rest of this paper reports the retrieval performance using these methods.

4 Result and Discussion

Thread retrieval is a new task, and the number of test collections is limited. In this study, we used the same corpus used by [2]. It has two datasets from two forums—Ubuntu[1] and Travel[2] forums. The statistics of the corpus is as follows. In the Ubuntu dataset, there are 113277 threads, 676777 messages, 25 queries and 4512 judged threads. In the Travel dataset,there are 83072 threads, 590021 messages, 25 queries and 4478 judged threads. The same relevance protocol was followed: a thread with 1 or 2 relevance judgement is considered as relevant, while a relevance of 0 is considered as irrelevant. Text was stemmed with the Porter stemmer and no stopword removal was applied. In conducting the experiments, we used the Indri retrieval system [3].

As for evaluation, for each query, we calculated Precision at 10 (P@10), Normalized Discounted Cumulative Gain at 10 (NDCG@10) and Mean Average Precision (MAP). In addition, We used the virtual document model(VD)[5] as our baseline. We used VD because it has been used as a strong baseline in most previous studies[5,10]. In addition to VD, we used the basic aggregation methods. In the basic methods, all ranked messages are included in the aggregation process. That enables us to make a fair judgement about the performance of the combined methods.

[1] ubuntuforums.org

[2] http://www.tripadvisor.com/ShowForum-g28953-i4-NewYork.html

[3] http://www.lemurproject.org/indri.php

As for parameter estimation, we have three parameters to estimate: the smoothing parameters μ for the virtual document and message language models, the size of the initial ranked list of messages R_Q and the value of k. To estimate μ, we varied its value from 500 up to 4000; adding 500 in each run. To estimate the size of R_Q, we varied its value from 500 up to 5000 adding 500 in each run. To estimate k— the number of top ranked messages, we varied its value from 2 up to 6 adding 1 in each run. Then, an exhaustive grid search was applied to maximize MAP using 5-fold cross validation.

5 Result and Discussion

This section reports the result of fusing only the top k messages. Table 1 and Table 2 presents the retrieval performance of these methods in the Ubuntu dataset and the Travel dataset respectively. The symbols △ and ▽ denote statistically

Table 1. Retrieval performance of fusion top k messages on the Ubuntu dataset

	Method	MAP	P@10	NDCG@10
Baseline	VD	0.3437	0.4200	0.3284
Basic models	Votes	0.2749 ▽	0.4680	0.3551
	RR	0.3313	0.4600 △	0.3428
	BordaFuse	0.3153	0.5080 △	0.3778
	CombMIN	0.1779 ▽	0.2600 ▽	0.1849 ▽
	CombMAX	0.3074 ▽	0.4480	0.3257
	CombMED	0.2212 ▽	0.2760 ▽	0.1927 ▽
	CombSUM	0.3100	0.4720	0.3633
	CombMNZ	0.3108	0.4880	0.3720
	CombANZ	0.2314 ▽	0.2800 ▽	0.1991 ▽
	CombGNZ	0.2272 ▽	0.2760 ▽	0.1971 ▽
	expCombSUM	0.3088	0.4840	0.3676
	expCombANZ	0.2315 ▽	0.2800 ▽	0.1991 ▽
	expCombMNZ	0.3088	0.4840	0.3676
Top k models	Votes$_k$	0.2699 ▽	0.4600	0.3327
	RR$_k$	0.3359	0.4600 △	0.3425
	BordaFuse$_k$	0.3355 ▲	0.4800	0.3659
	CombMIN$_k$	0.2372 ▽▲	0.3320 ▽▲	0.2559 ▽▲
	CombMED$_k$	0.2644 ▽▲	0.3680 ▲	0.2698 ▽▲
	CombSUM$_k$	0.3575 ▲	0.4760 △	0.3851 △
	CombMNZ$_k$	0.3544 ▲	0.4800 △	0.3866 △
	CombANZ$_k$	0.2644 ▽▲	0.3680 ▲	0.2698 ▽▲
	CombGNZ$_k$	0.2668 ▽▲	0.3840 ▲	0.2832 ▲
	expCombSUM$_k$	0.3539 ▲	0.4800 △	0.3865 △
	expCombANZ$_k$	0.2644 ▽▲	0.3680 ▲	0.2698 ▽▲
	expCombMNZ$_k$	0.3539 ▲	0.4800 △	0.3865 △

significant improvements or degradations over the virtual document model (VD) respectively, whereas, ▲ and ▼ denote statistically significant improvements or degradations of a top k model over its basic model, e.g CombSUM$_k$ over CombSUM. All significant tests are conducted using ttest at $p < 0.05$. The upper part from the tables contains retrieval performance of the virtual document (VD) and the basic aggregation methods, while the lower parts contain the performance of the combined methods.

Generally, the data on these tables supports the result from previous researches. In the basic mode, BordaFuse, CombSUM, CombMNZ, expCombSUM, expCombSUM and expCombMNZ were able to produce better or comparable result with respect to VD. These methods favor threads with highly ranked messages. In contrast, CombGNZ, CombMED, CombANZ, CombMIN and expCombANZ might be affected by threads that have a lot of low scored messages. In the combined models, all ranked based methods have almost similar results to their results on the basic mode. In contrast, the score based methods benefit largely from fusing only the top k messages. Among them, methods favouring threads with highly ranked messages brought significant improvements over baseline methods— see the performance of CombSUM, CombMNZ, expCombSUM and expCombMNZ. In addition, CombANZ, CombMIN, CombMED and expCombANZ benefits as well. Therefore, focusing on top k messages is a good strategy to improve data fusion performance on thread retrieval.

Discussion

The improvement might be due to the removal of noise introduced when all ranked messages are considered. By focusing on the top k messages, low score messages are discarded leading to improving retrieval. In fact, ranking using only one message, e.g. CombMAX, might not be enough to capture the topical relevance of threads, while considering all ranked messages might introduce irrelevant messages. Therefore, focusing on the top k ranked messages will balance these two aspects. This premise explains, to some extend, why no improvements were observed among ranked based methods. Rank based methods inherently address the issue of low scoring messages. Furthermore, the near similar performance of CombSUM, CombMNZ and their exponential variants suggests that: once we focus on the top k messages, extra emphasis on these messages does not help. In other words, focusing on the top ranked messages is the main reason behind the observed improvements.

To confirm this hypothesis, a study was conducted to investigate how does the performance change as k increases? As Figures 1 and 2 show, the optimal value of k for each method ranges between 2 to 5 messages. Going beyond the optimal value, rank based methods tend to have consistent performances that are similar to their result on the basic mode. In contrast, score based methods performed badly as k increases. That further supports the proposed hypothesis. Another interesting observation is the performance of CombGNZ. The result shows the inferiority of CombGNZ to the aforementioned good performing methods. However, [5,3] reported that PCS— using the geometric mean, outperformed

Table 2. Retrieval performance of fusion top k messages on the Travel dataset

	Method	MAP	P@10	NDCG@10
Baseline	VD	0.3774	0.4800	0.3549
Basic models	Votes	0.2996 ▽	0.5280	0.4190
	RR	0.3155 ▽	0.4520	0.3432
	BordaFuse	0.3630	0.5640 △	0.4350 △
	CombMIN	0.1574 ▽	0.3040 ▽	0.2199 ▽
	CombMAX	0.2727 ▽	0.4360	0.3216
	CombMED	0.2004 ▽	0.3480 ▽	0.2389 ▽
	CombSUM	0.3668	0.5560	0.4440 △
	CombMNZ	0.3575	0.5280	0.4205
	CombANZ	0.2065 ▽	0.3400 ▽	0.2346 ▽
	CombGNZ	0.2000 ▽	0.3320 ▽	0.2319 ▽
	expCombSUM	0.3513	0.5200	0.4109
	expCombANZ	0.2065 ▽	0.3400 ▽	0.2346 ▽
	expCombMNZ	0.3513	0.5200	0.4109
Top k models	Votes$_k$	0.2864 ▽▼	0.5280	0.3943
	RR$_k$	0.3126 ▽▼	0.4520	0.3400
	BordaFuse$_k$	0.3639	0.5520	0.4380 △
	CombMIN$_k$	0.1918 ▽▲	0.3360 ▽▲	0.2381 ▽
	CombMED$_k$	0.2478 ▽▲	0.3960 ▽	0.2881 ▽
	CombSUM$_k$	0.3736	0.6080 △▲	0.4788 △
	CombMNZ$_k$	0.3608	0.5720 △	0.4587 △
	CombANZ$_k$	0.2478 ▽▲	0.3960 ▽	0.2881 ▽▲
	CombGNZ$_k$	0.2330 ▽▲	0.3760 ▽	0.2754 ▽
	expCombSUM$_k$	0.3600	0.5720 △	0.4559 △
	expCombANZ$_k$	0.2479 ▽▲	0.3960 ▽	0.2882 ▽▲
	expCombMNZ$_k$	0.3600	0.5720 △	0.4559 △

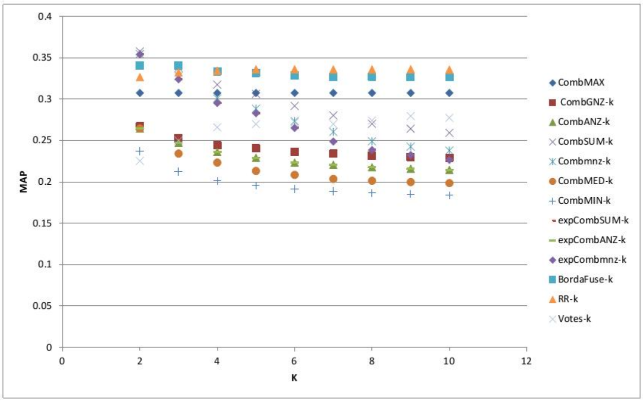


Fig. 1. The performance of the combined methods as k changes on the Ubuntu dataset

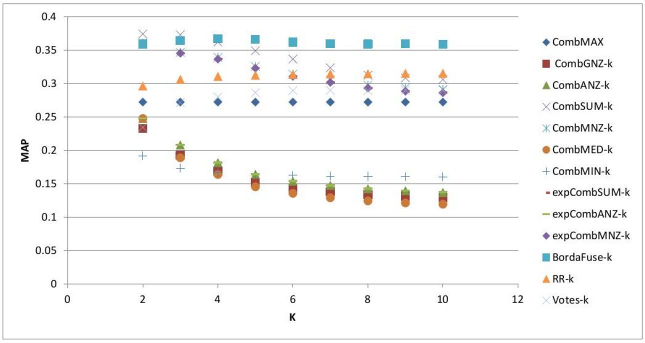


Fig. 2. The performance of the combined methods as k changes on the Travel dataset

CombMAX. That does not contradict our findings. In [5,3], PCS adds padding step if the number of ranked messages is less than k, whereas we do not apply this step. We plan to study the effect of the padding step in future works.

6 Conclusion and Future Works

In this paper, we addressed the problem of online forums thread retrieval. We conducted experiments to investigate the performance of combining the pseudo cluster selection and voting techniques approaches. The results showed that the combined models can improve the performance of score based techniques significantly in various measures.

Our future work has two directions. First, we will apply the voting methods as aggregation methods on the other thread representations. Second, we will study the effect of the padding step in the performance of combined models.

Acknowledgements. This work is supported by Ministry of Higher Education (MOHE) and Research Management Centre (RMC) at the Universiti Teknologi Malaysia (UTM) under Research University Grant Category (VOT Q. J130000. 7128. 02H99).

References

1. Aslam, J.A., Montague, M.: Models for metasearch. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2001, pp. 276–284. ACM, New York (2001)
2. Bhatia, S., Mitra, P.: Adopting inference networks for online thread retrieval. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, Georgia, USA, July 11-15, pp. 1300–1305 (2010)
3. Elsas, J.L.: Ancestry.com online forum test collection. Technical Report CMU-LTI-017, Language Technologies Institute, School of Computer Science, Carnegie Mellon University (2011)

4. Elsas, J.L., Arguello, J., Callan, J., Carbonell, J.G.: Retrieval and feedback models for blog feed search. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, pp. 347–354. ACM, New York (2008)
5. Elsas, J.L., Carbonell, J.G.: It pays to be picky: an evaluation of thread retrieval in online forums. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, pp. 714–715. ACM, New York (2009)
6. Macdonald, C., Ounis, I.: Key blog distillation: ranking aggregates. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 1043–1052. ACM, New York (2008)
7. Macdonald, C., Ounis, I.: Voting techniques for expert search. Knowl. Inf. Syst. 16(3), 259–280 (2008)
8. Metzler, D., Croft, W.B.: Combining the language model and inference network approaches to retrieval. Inf. Process. Manage. 40(5), 735–750 (2004)
9. Ogilvie, P., Callan, J.: Combining document representations for known-item search. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 143–150. ACM, New York (2003)
10. Seo, J., Croft, W.B., Smith, D.: Online community search using conversational structures. Information Retrieval 14, 547–571 (2011), doi:10.1007/s10791-011-9166-8
11. Seo, J., Croft, W.B.: Blog site search using resource selection. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 1053–1062. ACM, New York (2008)
12. Shaw, J.A., Fox, E.A.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2), pp. 243–252 (1994)
13. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004)

Genetic-Evolved Bayesian Networks in a Biomedical Application

Chih-Chiang Wei[*]

Department of Digit Fashion Design, Toko University
No. 51, Sec. 2, University Rd., Pu-Tzu City, Chia-Yi County 61363, Taiwan
d89521007@ntu.edu.tw

Abstract. This study presents genetic algorithm (GA) for discovering Bayesian network structure. The algorithm is applied to a medical datasets for vertebral column. Data set containing values for six biomechanical features is used to classify patients into three categories: disk hernia (DH), spondylolisthesis (SL), and normal (NO) or two categories: abnormal (AB), and NO. On ten-fold cross-validation run, the average AUC (the area under the ROC curve) measures of 0.874 and 0.923 for two and three categories are obtained, respectively. Results indicate that GA is relatively effective algorithm. Consequently, the GA-evolved BN is powerful tool for knowledge representation and inference because the causality relationship can be observed.

Keywords: Bayesian network, Genetic algorithm, Medicine, Classification.

1 Introduction

Forecasting the behavior of complex systems has been a broad application domain for machine learning. As known, Bayesian networks (BN) is a graphical knowledge representation tool. The merit of BN is its conditional dependencies can be observed by a directed acyclic graph (DAG). The BN based on an approach in probability theory is a powerful knowledge representation and reasoning tool under conditions of uncertainty [1].

In recent years, BN has drawn considerable attention owing to its high predictive ability for a wide range of applications, such as [2-4]. As a result, BNs are widely used in problems where uncertainty is handled using probabilities [5]. Unfortunately, finding the optimal network structure requires searching through all possible structures [6]. Although promising results have been obtained using global search techniques, their computation cost makes them unfeasible [7,8].

Recently, it is found that modern heuristic methods such as genetic algorithm (GA) [9-11] can be used for large combinational optimization problem. For GA algorithm, it works by having a population of BN structures and allow them to mutate and apply cross over to get offspring. The best network structure found during the process is

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 307–315.
DOI: 10.1007/978-3-642-35452-6_32

returned [12]. This BN learning approach using evolutionary techniques might be employed to find a well scoring Bayes network structure.

This paper describes GA evolutionary algorithm that uses to discover BN network structures from data. This study considers a biomedical case which is the pathologies of the vertebral column. The vertebral column is a system composed by a group of vertebras, invertebrate discs, nerves, muscles, medulla and joints. This complex system can suffer dysfunctions that cause backaches with very different intensities. Disc hernia and spondylolisthesis are examples of pathologies of the vertebral column that cause intense pain. The details of pathologies of the vertebral column can be found in Rocha Neto et al. [13].

2 Algorithm

2.1 Bayesian Network

A BN arrows signify the existence of direct causal influences from variables represented by parent nodes [14]. BN can be defined by a tuple (X, S, P), where $X = \{X_1, X_2,\ldots, X_\mathrm{n}\}$ is a set of variables, S is the structure of the BN that defines the causal influence among the variables in X, and the strength of these influences are quantified by the conditional probabilities, P [15]. Each variable X_i, $i \in (1,\ldots,n)$ corresponds to a node, and an arc from variable i to variable j denotes the cause-effect relationship between them, where i is a parent of j. A node can have a number of parents if the node is influenced by a number of variables (causes).

The influences are measured by a set of conditional probabilities, $p=\left(X_i \middle| Pa_i\right)$ for each state of a child variable X_i, given a configuration of its parent variables, denoted by Pa_i. If a child and its parents are discrete variables, the conditional probability is represented by an $m \times n$ conditional probability table, where m is the number of possible configurations of the parents' states and n is the number of states of the child node. As stated, the structure S describes independence and dependence between variables. Given a sample of X, $\mathbf{x} = \{x_1, x_2,\ldots, x_n\}$, the joint probability can be obtained by

$$p\left(X=\mathbf{x}\right)=\prod_{i=1}^{n} p\left(X_i = x_i \middle| Pa_i = \pi_{ij}\right) \tag{1}$$

where π_{ij} denotes the state of Pa_i in $\mathbf{x}$. If node i has no parent, then $p\left(X_i = x_i \middle| Pa_i = \pi_{ij}\right) = p\left(X_i = x_i\right)\cdot$

2.2 Structure Learning Algorithm

In this paper, the BN learning approach uses heuristic algorithms for finding a well scoring BN structure. A heuristic is a procedure which will give a good solution- not necessarily the optimal- to problems [16]. The GA is described in the following.

GA [17] is adaptive method that can be used for solving problems of search and optimization. It is based on the genetic process of living organisms. Through generations, the populations evolve in nature according to the principles of natural selection and survival of the fittest postulated by Darwin. The fundamental principle of GA is based on natural selection. To solve an optimization task, GA typically involves five steps [18]:

Step 1: Initialization. The GA starts from a population of randomly selected chromosomes (a coded feature string). Each chromosome consists of a number of genes (features) and corresponds to a possible solution to the problem.

Step 2: Evaluation. An assessment function is applied to evaluate the performance of the entire population of chromosomes. This step typically is the most difficult and computationally the most costly.

Step 3: Selection. With use of different selection methods (e.g., roulette wheel selection [17]), the chromosomes with better performance can expand to include a larger fraction of the population.

Step 4: Search. After the population has adjusted itself to take advantage of the better-performing chromosomes, search operators are used to recombine or create a new generation of chromosomes. The most frequently used operators are crossover and mutation. Crossover exchanges genes between two chromosomes, and mutation introduces random changes to select genes.

Step 5: Termination. The GA evolves continually, until a predetermined termination condition is met.

Imitating this process, GA is capable of creating solutions for real world problems [16]. Here, the score of Bayesian metric is employed to determine the measure used to judge the quality of a network structure. The Bayesian metric of a BN structure B_D for a database D is

$$Q_{Bayes}\left(B_S, D\right) = p\left(B_S\right) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma\left(N'_{ij}\right)}{\Gamma\left(N'_{ij} + N_{ij}\right)} \prod_{k=1}^{r_i} \frac{\Gamma\left(N'_{ijk} + N_{ijk}\right)}{\Gamma\left(N'_{ijk}\right)} \tag{2}$$

where $p(B_S)$ is the prior on the network structure, q_i denotes the cardinality of the parent set of x_i in B_S $(1 \leq i \leq n)$, r_i is the cardinality of x_i, $\Gamma(.)$ is the gamma-function, N'_{ij} and N'_{ijk} represent choices of priors on counts restricted by $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$.

3 Experiment

3.1 Data

The database applied in this work was supplied by Dr. Henrique da Mota, who collected it during a medical residence in spine surgery at the Centre Medico-Chirurgical de Readaptation des Massues, Lyon, France [19]. This database contains data about 310 patients obtained from sagittal panoramic radiographies of the spine. Each patient is represented in the data set by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine (see Table 1). The correlation between the vertebral column pathologies and these attributes was originally proposed in Berthonnaud et al. [20].

In this experiment, the first sub-dataset consists in classifying patient status as belonging to one out of three categories: disk hernia (DH) (60 patients), spondylolisthesis (SL) (150 patients), and normal (NO) (100 patients). For the second sub-dataset, the categories DH and SL were merged into a single category labeled as "abnormal." Thus, the second dataset consists in classifying patients as belonging to one out of two categories: abnormal (AB) (210 patients), and NO (100 patients).

Table 1. Attributes used in vertebral column dataset

Denotation	Description	Type	Range
PI	Pelvic incidence	Numeric	26.15~129.83
PT	Pelvic tilt	Numeric	-6.55~49.43
LA	Lumbar lordosis angle	Numeric	14.00~125.74
SS	Sacral slope	Numeric	13.37~121.43
PR	Pelvic radius	Numeric	70.08~163.07
GS	Grade of spondylolisthesis	Numeric	-11.06~418.54

3.2 Model Parameters Learning

This study employs the WEKA environment, a popular suite of machine learning software written in Java, for the implementation [12]. In order to assess the ability of such forecasts, it is necessary to have accurate ways of quantifying forecast skill. To calculate these measures, the 2×2 decision matrix is illustrated in Table 2.

Three skill scores are defined here. Firstly, the true-positive rate (TPR) (also called "sensitivity") is calculated as ($TP/TP + FN$). TPR measures the proportion of diseased patients correctly identified as positive [21]. Secondly, the false positive rate (FPR) is calculated as ($FP/TN + FP$). FPR measures the proportion of disease-free patients incorrectly identified as positive. Thirdly, the Precision score is the proportion of the examples which truly have the specific class among all those which were classified as the specific class [12]. For prediction of abnormal, Precision is as calculated as ($TP/TP + FP$), while for prediction of normal, Precision equals to ($TN/FN + TN$).

GA search works by having a population of BN structures and allow them to mutate and apply crossover to get offspring. The best network structure found during the process is returned. The size of the population of descendants that is created each generation is set at 100. The number of generations of BN structure populations is set at 20. The size of the population of network structures that is selected each generation is tested here. By the sensitivity analysis of the population size in the interval of [1, 40], the TPR, FPR, and Precision are calculated and the results plotted in Fig. 1. From the figure, the suitable population size is set to 17.

Table 2. Decision matrix

Prediction outcome	Presence of disease		Total
	Present	Absent	
Abnormal	*TP*	*FP*	*TP* + *FP*
Normal	*FN*	*TN*	*FN* + *TN*
Total	*TP* + *FN*	*FP* + *TN*	

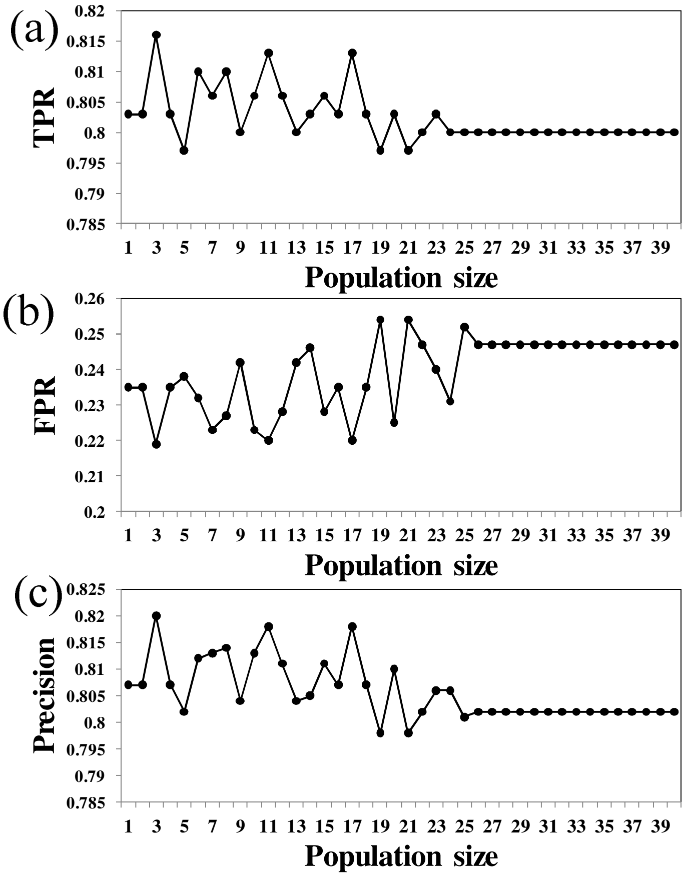


Fig. 1. Sensitivity analysis of population size on the TPR, FPR, and Precision

3.3 Results and Discussions

Based on the learned BN, a causality relationship is observed. The resulting networks are shown in Fig. 2(a) and (b) for two categories of {AB, NO}, and {DH, SL, NO}, respectively. The networks display complicated causality relationships among attributes.

To provide the most comprehensive description of diagnostic accuracy, this study employs the receiver operating characteristic (ROC) analysis. The ROC curve is a popular tool in medical and imaging research. It conveniently displays diagnostic accuracy expressed in terms of Sensitivity against (1 – Specificity) at all possible threshold values [22,23].

By changing the cut-off value, several different pairs of (1 – Specificity) and Sensitivity can be obtained [21]. These pairs are plotted as the *X* and *Y* coordinate values on the ROC curve. Both axes of the graph range in value from 0 to 1. Accordingly, the ROC curves for two and three categories are given in Figs. 3 and 4, respectively.

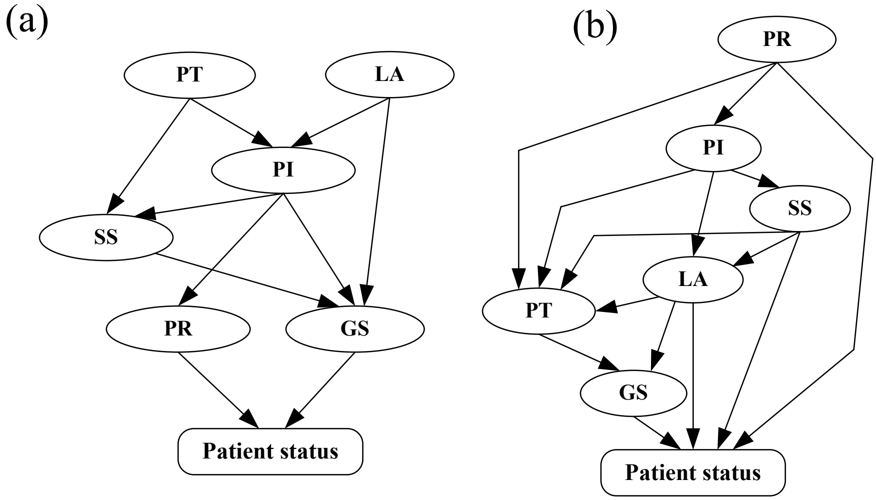


Fig. 2. The resulting networks using evolutionary algorithms for (a) two categories of {AB,NO}, and (b) for three categories of {DH,SL,NO}

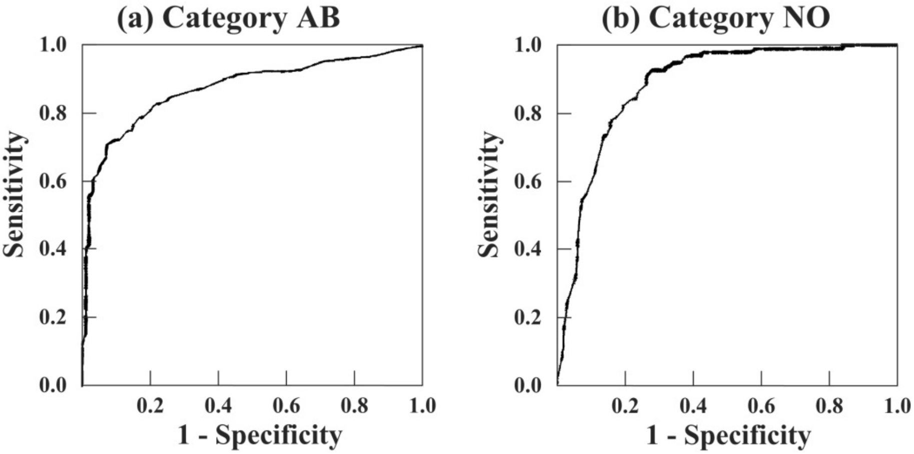


Fig. 3. ROC curve for two classes in (a) AB, and (b) NO

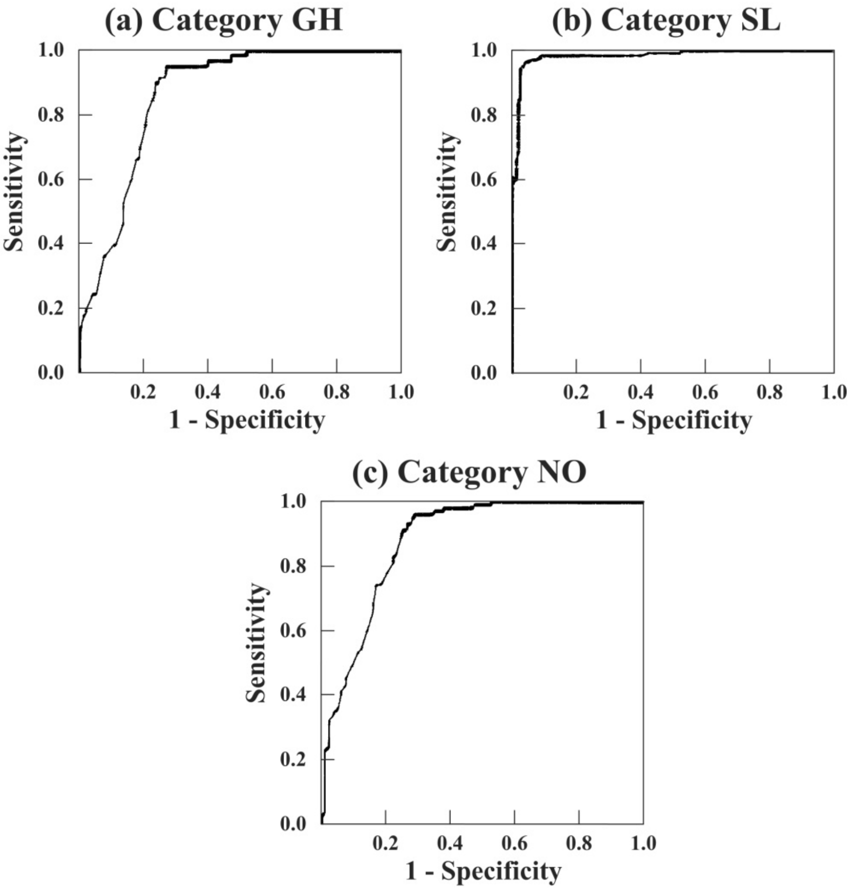


Fig. 4. ROC curve for three classes in (a) GH, (b) SL, and (c) NO

The area under the ROC curve (AUC) can give an indication of the accuracy of the test. If the ROC curve follows the diagonal line, the implication is that this diagnostic test is not useful and the area under this curve is 0.50, indicating that the test does not perform any better than random guessing [24].

On ten-fold cross-validation run, the average AUC measures of 0.874 and 0.923 for two and three categories are obtained, respectively. As mentioned in [25], the AUC could distinguish between non-predictive (AUC = 0.5), less predictive (0.5 < AUC < 0.7), moderately predictive (0.7 < AUC < 0.9), highly predictive (0.9 < AUC < 1) and perfect prediction (AUC = 1). Accordingly, the results belong to moderately predictive and highly predictive for two categories and three categories, respectively.

4 Conclusions

This study has successfully presented the use of the genetic evolutionary algorithms for learning BN structures, and the GA-evolved learning algorithm for BN was empirically evaluated. This study tested a biomedical application which is the pathologies of the vertebral column. Data set containing values for six biomechanical features used to classify patients into three categories of {DH, SL, NO} or two categories of {AB, NO}.

On ten-fold cross-validation run, the average AUC measures of 0.874 and 0.923 for two and three categories are obtained, respectively. Overall results indicate that GA is relatively effective algorithms. Consequently, the GA-evolved BN is powerful tool for knowledge representation and inference because the causality relationship can be observed.

Acknowledgements. The support under Grant No. NSC101-2119-M-464-001 by the National Science Council, Taiwan is greatly appreciated. Also, the author is grateful to UC Irvine (UCI) machine learning repository for providing the data for the experiment.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, San Francisco (1988)
2. Guo, S., Xu, G., Zhang, H., Li, C.: A real-time flood updating model based on the Bayesian method. Methodology in Hydrology 311, 210–215 (2007)
3. Verron, S., Li, J., Tiplica, T.: Fault detection and isolation of faults in a multivariate process with Bayesian network. Journal of Process Control 20, 902–911 (2010)
4. Balov, N.: A Gaussian mixed model for learning discrete Bayesian networks. Statistics and Probability Letters 81, 220–230 (2011)
5. Inza, I., Larranaga, P., Sierra, B.: Feature subset selection by Bayesian networks: a comparison with genetic and sequential algorithms. International Journal of Approximate Reasoning 27(2), 143–164 (2001)
6. Chickering, M., Geiger, D., Heckerman, D.: Learning Bayesian Networks is NP-hard, Technical Report MSR-TR-94-17, Microsoft Research, Redmond (1994)
7. Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Yurramendi, Y.: Learning Bayesian network structures by searching for the best ordering with genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics-Part A 26(4), 487–493 (1996)
8. Wong, M.L., Lam, W., Leung, K.S.: Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(2), 174–178 (1999)
9. Zheng, B., Chang, Y.H., Wang, X.H., Good, W.F., Gur, D.: Feature selection for computerized mass detection in digitized mammograms by using a genetic algorithm. Academic Radiology 6(6), 327–332 (1999)
10. Moussa, A., El-Gammal, M., Abdallah, E.N., Attia, A.I.: A genetic based algorithm for loss reduction in distribution systems. IEEE Transactions on Power Delivery 4(2), 447–453 (2000)
11. Aiyar, R.S., Gagneur, J., Steinmetz, L.M.: Identification of mitochondrial disease genes through integrative analysis of multiple datasets. Methods 46(4), 248–255 (2008)
12. Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., Scuse, D.: WEKA Manual for Version 3-7-3. University of Waikato, New Zealand (2010)

13. da Rocha Neto, A.R., Sousa, R., de A. Barreto, G., Cardoso, J.S.: Diagnostic of Pathology on the Vertebral Column with Embedded Reject Option. In: Vitrià, J., Sanches, J.M., Hernández, M. (eds.) IbPRIA 2011. LNCS, vol. 6669, pp. 588–595. Springer, Heidelberg (2011)
14. Jensen, F.V.: Introduction to Bayesian Networks. Springer, Berlin (1996)
15. Zhu, W.: Using Bayesian network on network tomography. Computer Communications 26, 155–163 (2003)
16. Sierra, B., Serrano, N., Larranaga, P., Plasencia, E.J., Inza, I., Jimenez, J.J., Revuelta, P., Mora, M.L.: Using Bayesian networks in the construction of a bi-level multi-classifier. A case study using intensive care unit patients data. Artificial Intelligence in Medicine 22(3), 233–248 (2001)
17. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
18. Hluck, G.: Genetic algorithms. In: Liebowitz, J. (ed.) The Handbook of Applied Expert Systems. CRC, Boca Raton (1997)
19. Frank, A., Asuncion, A.: UCI Machine Learning Repository University of California, School of Information and Computer Science, Irvine, CA (2010), `http://archive.ics.uci.edu/ml`
20. Berthonnaud, E., Dimnet, J., Roussouly, P., Labelle, H.: Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. Journal of Spinal Disorders & Techniques 18(1), 40–47 (2005)
21. Han, U.K., Kim, Y.H.: Determination of Class II and Class III skeletal patterns: receiver operating characteristic (ROC) analysis on various cephalometric measurements. American Journal of Orthodontics and Dentofacial Orthopedics 113(5), 538–545 (1998)
22. Zou, K.H.: Comparison of correlated receiver operating characteristic curves derived from repeated diagnostic test data. Academic Radiology 8(3), 225–233 (2001)
23. Metz, C.E.: Receiver operating characteristic analysis: A tool for the quantitative evaluation of observer performance and imaging systems. Journal of the American College of Radiology 3(6), 413–422 (2006)
24. Daya, S.: Diagnostic test: Receiver operating characteristic (ROC) curve. Evidence-based Obstetrics and Gynecology 8, 3–4 (2006)
25. Swets, J.A.: Measuring the accuracy of diagnostic systems. Science 240(4857), 1285–1293 (1988)

Feature Reduction and Selection for Automatic Image Annotation

Guan-Bin Chen and Been-Chian Chien

Department of Computer Science and Information Engineering
National University of Tainan, Tainan, Taiwan
fatex2790@hotmail.com, bcchien@mail.nutn.edu.tw

Abstract. Automatic image annotation for large collections of images is a challenging problem. For labeling images precisely, more various features including low-level image features, EXIFs, textual tags of images are expected to be used. However, not all features contribute useful information for each concept. The high-dimension problem causing by combining all features is detrimental to the concept learning. In this paper we propose the feature reduction and selection method to improve the performance of annotating images. The proposed feature reduction methods extract informative features to reduce the dimensions. While the feature selection method based on the wrapper model can select effective features from miscellaneous features. The experimental result shows that the proposed feature reduction method improves the efficiency of concept learning. The developed feature selection method also increases the labeling precision and recall of images.

Keywords: Image Annotation, Image Classification, Feature Extraction, Feature Selection, Feature Reduction.

1 Introduction

In recent years, along with the development of social networks, people often share their personal experiences on the Internet, such as travel logs, food recipes, family activities, etc. Although the shared contents and events may be different for users, photos are the common media used for recording a glorious writing task. Since digital photos can be easily produced by different mobile devices, the number of images is extremely increasing on webs. The modern search engines like Google, Yahoo, and Bing can provide efficient retrieval of relevant textual documents and visual images based on keywords. However, the search results are generally not dependent on the meaning of target documents, but the texts appearing in the documents. Especially, an image is usually unstructured and scattered in its representation format. The semantics of images are usually obtained from their filenames or textual description pages which are given by creators or users. However, the image annotation is subjective and tedious work for a large amount of images. In order to search and manage images in a large image dataset using semantic keywords, an effective systematic image annotation approach is important and necessary. The automatic image annotation task is to

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 317–326.
DOI: 10.1007/978-3-642-35452-6_33

assign suitable labels to an unknown image according to a given set of images and their corresponding annotated labels.

For annotating a large number of image data, several difficulties should be outlined and solved. First, visual perception is one of the important features while annotating images. However, the different visual features express all sorts of visual perceptions, like color, texture, shape, etc. A single low level visual feature may not represent a semantic concept of an image enough. Further, a semantic concept might be usually synthesized by various visual features. How to select the proper features for effective annotating concepts is an important issue. Second, to select and fuse visual features effectively, many multidimensional visual features need to be taken and analyzed. A number of high-dimension features are another problem of concept annotation. The process of high-dimension data is generally time consuming and harmful for learning effective annotation models.

In this paper, we propose the feature reduction and selection methods to improve the performance of image annotation. First, we propose a statistical method to reduce the dimension of visual features. The dimensions of textual features are reduced by applying the discriminant coefficient [1]. Then, a feature selection method based on wrapper model [2] is developed to select the proper features for each concept. We also study and compare the effectiveness of single model and mixed models with different combination of features. The experimental result shows that the proposed feature reduction method can improve the efficiency of concept learning. The developed feature selection method does increase the effectiveness of image annotation.

The rest of this paper is organized as follows. In Section 2, we describe the related work in image annotation and feature selection. We propose the feature reduction and the feature selection methods in Section 3. Section 4 shows the experimental results on Flickr image web. Finally, we conclude this paper in Section 5.

2 Related Work

The problem of image annotation has been studied for decades. Many annotation models were proposed in the previous researches. Generally, the image annotation models can be divided into semi-automatic and automatic annotation. The traditional semi-annotation model is inspired from the content-based image retrieval. With the help of image retrieval systems and relevance feedback methods, the user assigns concept labels to the unlabeled images semi-automatically [3]. Such a model needs proceed to repeat feedback processes to achieve good annotation results.

While automatically annotating images, two categories of methods can be used: the classification-based method and the retrieval-based method. The classification-based methods treat the annotation process as the classification problem [4, 5]. The labeling classifiers are learned from the training set and unknown images are labeled by the classifiers. A probabilistic based learning model, the cross-media relevance model, proposed by Jeon et al. [6] used join probability of words, blobs and images to predict the labels of unknown images. The retrieval-based method takes an unlabeled image as the query image. The query image is then employed to search relevant images by an image retrieval system from the annotated image database. Finally, the annotation words from the textual descriptions of the relevant images are ranked and identified [7]. Although such a method does not need to create labeling models in advance, it requires a large number of labeled images in database.

The function of feature reduction can be accomplished by feature extraction or feature selection. Feature extraction tries to convert or transform the original features into a new set of reduced features but reserve the information in original set as much as possible. While feature selection is to find useful features and take out the useless ones. Further, feature selection can be classified into two types of models: filters and wrappers [2]. The filters model generally measures the importance of features by calculating the data dependence of features in the dataset, e.g. information gain [8], entropy and so on. The wrappers model generates different combinations of features and classifiers to measure the effectiveness of the subset of features. The filters model is usually more efficient than the wrappers model. However, the selected features by the wrapper model are more effective than the filters model's.

3 Feature Reduction and Selection Methods

3.1 The Feature Reduction Methods

Visual Features Reduction. Given a set of images **I** with size of n, $I_i \in$ **I**, $1 \le i \le n$. We assume that a visual feature of an image I_i is represented as $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ with m dimensions. Let C_1, C_2, …, C_p be the p predefined concepts of images. The feature reduction process first computes the mean μ_{lj} and the deviation σ_{lj} of the jth dimension in the visual feature for the class l, as follows:

$$\mu_{lj} = \frac{\sum_{I_i \in C_l} x_{ij}}{|C_l|}, \quad \sigma_{lj} = \sqrt{\frac{\sum_{I_i \in C_l} (x_{ij} - \mu_{lj})^2}{|C_l|}} \quad , 1 \le j \le m \,, 1 \le l \le p, \tag{1}$$

where $|C_l|$ is the number of images belonging to the concept C_l. The *concept similarity* of the visual feature $\mathbf{x}_i$ corresponding to the concept C_l for the image I_i defined as:

$$f_{li} = \prod_{j=1}^{m} \exp\left[-\left(\frac{x_{ij} - \mu_{lj}}{\sigma_{lj}} \right)^2 \right]. \tag{2}$$

Hence, a m-dimension visual feature $\mathbf{x}_i$ of an image I_i will be transformed into a p dimensional features $\mathbf{f}_i = [f_{li}]$, $0 \le f_{li} \le 1$, $1 \le l \le p$. This feature reduction method will be applied when $m > p$. If $m \le p$, we keep the original feature and do nothing in this feature reduction process.

Textual Features Reduction. The textual feature reduction method is based on the discriminant coefficient proposed by Lin & Chien [1]. The method defined the discriminant coefficients calculated by multiplication of the differences between the statistics of keywords belonging to two classes. We briefly present the reduction method in the following.

Given a textual term frequency $\mathbf{w}_i = [w_{ij}]$, $1 \leq j \leq m$, for an image $I_i \in \mathbf{I}$ and m is the number of dimensions of textual features. The $\mathbf{w}_i$ is first normalized by the maximum of term frequency in each image and transform these images into the concept domain, as follows:

$$u_{lj} = \frac{\sum_{I_i \in C_l} w_{ij} \Big/ \max_{1 \leq j \leq m} \{w_{ij}\}}{|C_l|}. \tag{3}$$

We define the *relative discriminant variable* v^l_x on the jth term for the concept l against the other concept x, as follows:

$$v^l_{xj} = \begin{cases} 1 & \text{if} \quad u_{xj} = 0, x \neq l, \\ |u_{lj} - u_{xj}| & \text{if} \quad u_{xj} \neq 0, x \neq l. \end{cases} \tag{4}$$

Then, the discriminant variable v^l_x is adjusted by logarithmic scale to avoid multiplicative underflow. The *logarithmic discriminant coefficient* $\mathbf{Z} = [z_{lj}]_{p \times m}$ is defined as

$$z_{lj} = \begin{cases} 1 & \text{if} \quad u_{lj} = 0 \quad \text{or} \quad \prod_{1 \leq x \leq p, x \neq l} v^l_{xj} = 0, \\ \ln\left(\prod_{1 \leq x \leq p, x \neq l} v^l_{xj} \right) & \text{others.} \end{cases} \tag{5}$$

Finally, the logarithmic discriminant coefficient $\mathbf{Z}$ is normalized to be the *log-scaled discriminant matrix* $\boldsymbol{J} = [J_{lj}]_{p \times m}$, where

$$J_{lj} = \begin{cases} 0 & \text{if} \quad z_{lj} = 1, \\ \dfrac{\min_{1 \leq l \leq p} \{z_{lj}\} - 1 - z_{lj}}{\min_{1 \leq l \leq p} \{z_{lj}\} - 1} & \text{if} \quad z_{lj} \neq 1, 1 \leq l \leq p. \end{cases} \tag{6}$$

The range of J_{lj} is between 0 and 1. A larger J_{lj} represents that the jth term is more distinguishable in the concept C_l than other concepts.

Let $\boldsymbol{W} = [w_{ij}]_{n \times m}$ be term frequency matrix of all images in $\mathbf{I}$. The feature reduction can be done by multiplication of the term frequency matrix $\boldsymbol{W}$ and the log-scaled discriminant matrix $\boldsymbol{J}$

$$\boldsymbol{W} \cdot \boldsymbol{J}^{\mathrm{T}}, \tag{7}$$

where $\boldsymbol{J}^{\mathrm{T}}$ is the transpose of the matrix $\boldsymbol{J}$.

3.2 Feature Selection and Learning Model

In this paper we propose the feature selection method based on wrapper models. The wrapper model generally uses internal classifiers to classify the subset of features and estimate their performances. We integrate visual, textual features and the Random Forest [9] to build internal classifiers. Basically, the Random Forest has been reported that this model got precise results in previous research of image annotation [5].

The feature selection approach is described in the following. First, a validation set of images $\hat{\mathbf{I}}$ is selected from the training set **I**. The left training images are used to build the Random Forest classifiers RF_{lj} according to the individual *j*th feature for each concept C_l. Then, the validation set $\hat{\mathbf{I}}$ is tested on the classifiers RF_{lj} to estimate the effectiveness of the *j*th feature for the concept C_l. To select the effective features for the concept C_l, we define the threshold θ_l as

$$\theta_l = \max_j \{\xi_{lj}\} - 2\sigma_l \,, \tag{8}$$

where ξ_{lj} is the *microF*1 measures of *j*th feature for the concept C_l, and σ_l is the standard deviation of ξ_{lj} for the concept C_l. The *j*th feature is selected to be one of the candidates in S_l of classifying the concept C_l if the condition $\xi_{lj} \geq \theta_l$ successes. However, four features are selected at most for each concept. The detailed feature selection algorithm is shown as Table 1.

After the features are selected for each concept, the selected features are combined and used to rebuild the classifier by Random Forest. Owing to images are annotated by multi-label, *p* classifiers are learned totally using *one-versus-all* (OVA) formulation. The OVA is that the *i*th classifier is trained by the examples in the *i*th class as positive and all other examples as negative.

Table 1. The feature selection algorithm

```
Algorithm: Feature Selection
Input: The image set I, the validation features set $\hat{\mathbf{I}}$.
Output: The selected features $S_l$ for each concept $C_l$.
 for each class $C_l$
     for the $j$th feature in I
         $RF_{lj} = RandomForest(j, \mathbf{I} - \hat{\mathbf{I}})$;
         $\xi_{lj} = \mathrm{MicroF1}(RF_{lj}, \hat{\mathbf{I}})$;
     endfor
     $\theta_l = \max_j \{\xi_{lj}\} - 2\sigma_l$;
     for each $\xi_{lj} \geq \theta_l$
         $j \in S_l$;
     endfor
     if $|S_l| > 4$
         keep the largest four $\xi_{lj}$ features in $S_l$ and remove others;
     endif
 endfor
```

Table 2. The used visual features

	Color	Shape	Texture	Spatial	Dimensions
Auto Color Correlogram	×			×	256
CEDD	×	×	×		144
Color Layout	×				120
Edge Histogram		×			80
FCTH	×	×	×		192
Gabor			×		60
Jpeg Coefficient Histogram	×	×			192

4 Experiments and Evaluation

4.1 Experimental Environments and Data Set

The experiments were conducted on a computer with Intel Core i7 Quad-Core 3.4 GHz with 16GB RAM. The main programming tool is MATLAB 7.13.546 (R2011b).

The entire MIRFLICKR collection [10] contains about one million images from the social photo sharing website Flickr.[1] In our experiments, we used the subset of MIRFLICKR collection which comes from ImageCLEF 2012 photo annotation task.[2]

This image set contains 15,000 images and each image provides metadata (ex. EXIF) and tags. The images are annotated by 94 pre-defined visual concepts which can be classified into five categories. The concepts are diverse and wide ranges. Some of the concepts are clear and concrete, such as “cat,” dog,” “coast,” etc. However, some of the concepts are vague and abstract, for example, “family,” “strangers,” “happy,” and so on.

The preprocessing of image data includes the extraction of various visual features and textual process on tags. The used visual features and the characteristics of textual features in our experimental dataset are depicted in the following.

Visual Features. Generally, low level visual features extract different visual perceptions such as color, shape, texture, and spatial relationship. The visual features used here try to cover different visual perceptions. We extracted 7 visual features including Auto Color Correlogram [11], Color Layout [12], CEDD [13], Edge Histogram [14], FCTH [15], Gabor [16], and Jpeg Coefficient Histogram [17]. The dimensions and related attributes for the used features are listed in Table 2. All of the above visual features can be extracted by using the codes in LIRE JAVA library.[3]

Textual Features. The textual features are the description tags for each image. The tags describe related semantics and objects about the image. However, the original tags are multi-lingual. In fact, more than 60 different languages are found in tags for

[1] `http://press.liacs.nl/mirflickr`

[2] `http://imageclef.org/2012/photo-flickr`

[3] `http://www.semanticmetadata.net/lire`

all images. Synonyms of terms also need to be unified. To solve these problems, we first used the Google translation tools to translate terms into English, check spelling, and get their synonym terms.[4] Then, the WVTool was applied to remove the stop words and retrieve the stemmer of each term.[5] The number of the final tags is 36702 words in English. The term frequency for each tag was also counted and recorded.

4.2 Evaluations

The 15,000 images in the benchmark were divided into two parts: 10,000 training images and 5,000 testing images. For performing the proposed feature selection method, the 10,000 training images were further separated into a set of 7,500 images for learning and a set of 2,500 images for validating the internal classifiers. After the features were selected, all 10,000 training images were used to generate the final classifiers for all concepts. The 5,000 testing images then were tested on the final classifiers to evaluate their performances.

In order to demonstrate the effectiveness of the proposed feature reduction methods and the selection method, the experiments were arranged as two parts. The first part is to observe the performance of each single original feature. The features of this part are evaluated as rows of visual and textual in Table 3. The visual row contains 7 different features. The tags are the unique source in textual row. The results in Table 3 show their measurements on *MicroP*, *MicroR*, and *MicroF*1.

The second part is the combinations of feature reduction and feature selection methods. The goal of this part is to analyze properties and compare the results of the proposed methods. The mixed row in Table 3 shows the six possible combinations of the original features, the reduced features, and the feature selection method. The OL means that all 7 visual features in their original sizes and the reduced textual feature were used to learn classifiers. The RL is that the classifiers are learned from 7 reduced visual features and reduced textual features. The OS-OL composed of the feature selection with original features and the learning strategy in the OL. The OS-RL represents the composition of applying the original features to the feature selection and learning classifiers from the reduced features. The RS-OL used the reduced features to perform feature selection and the original features to learn classifiers like the OL. Finally, the RS-RL used the reduced features in both the feature selection method and the learning of classifiers.

From the evaluation of Table 3, some notable results are observed:

- Generally, the results of annotation using mixed features are better than the results using a single feature. It shows that there exists complementarity among the image features.
- The results of the methods using feature selection are better than the method without using feature selection. For example, OS-OL and RS-OL outperform the OL method. Both OS-RL and RS-RL are superior to the RL method.

[4] http://translate.google.com

[5] http://sourceforge.net/projects/wvtool

Table 3. The experimental results

	Features	*MicroP*	*MicroR*	*MicroF*1
Visual	AutoColorCorrelogram (ACC)	0.4963	0.3977	0.4416
	CEDD	0.5225	0.4086	0.4586
	ColorLayout (CL)	0.4805	0.4023	0.4379
	EdgeHistogram (EH)	0.4283	0.4649	0.4459
	FCTH	0.4814	0.4271	0.4526
	Gabor	0.3446	0.3518	0.3482
	JpegCoefficientHistogram (JCH)	0.5027	0.4330	0.4652
Textual	Tags_reduction	0.5010	0.4702	0.4851
Mixed	Original features (OL)	0.5562	0.4714	0.5103
	Reduced features (RL)	0.4990	0.4973	0.4982
	OS-OL	0.5439	0.5048	0.5236
	OS-RL	0.4905	**0.5241**	0.5068
	RS-OL	**0.5577**	0.4974	**0.5258**
	RS-RL	0.4774	0.5212	0.4983

- As comparison on the RS and the OS, the effectiveness of the reduced features is not degraded so much. However, the efficiency of feature selection and concept learning can be evidently improved.
- While observing the learning of features, OL and RL, the results using the reduced features are worse than the original features. Obviously, the feature reduction will lose some discernible information in the original data. In summary, it is reasonable that the RS-OL method has the best annotation result in this dataset.

We also analyze the results of feature selection in Table 4 and Fig. 1. Table 4 shows the selection rates of features using different feature sets in the proposed feature selection method. The features with larger $F1$-measure values have higher selection rates (e.g. Tag, FCTH, CEDD, and JCH). To analyze the effect of features used in each concept, we also list the concepts whose $F1$-measure differences between the OS and the RS are greater than 5% in Fig. 1. The average number of selected features for all concepts using the OS and the RS are 1.88 and 2.05, respectively. However, we found that the OS results in higher $F1$-measure than the RS in these concepts with large difference. The reason is that the number of images in these concepts is very few. The concepts with less number of images will loss more information during the processing of feature reduction.

Table 4. The selection rates of image features

	Tags	FCTH	CEDD	JCH	ACC	CL	EH	Gabor
OS	0.42	0.10	0.12	0.10	0.07	0.06	0.08	0.05
RS	0.32	0.16	0.14	0.13	0.11	0.05	0.05	0.04

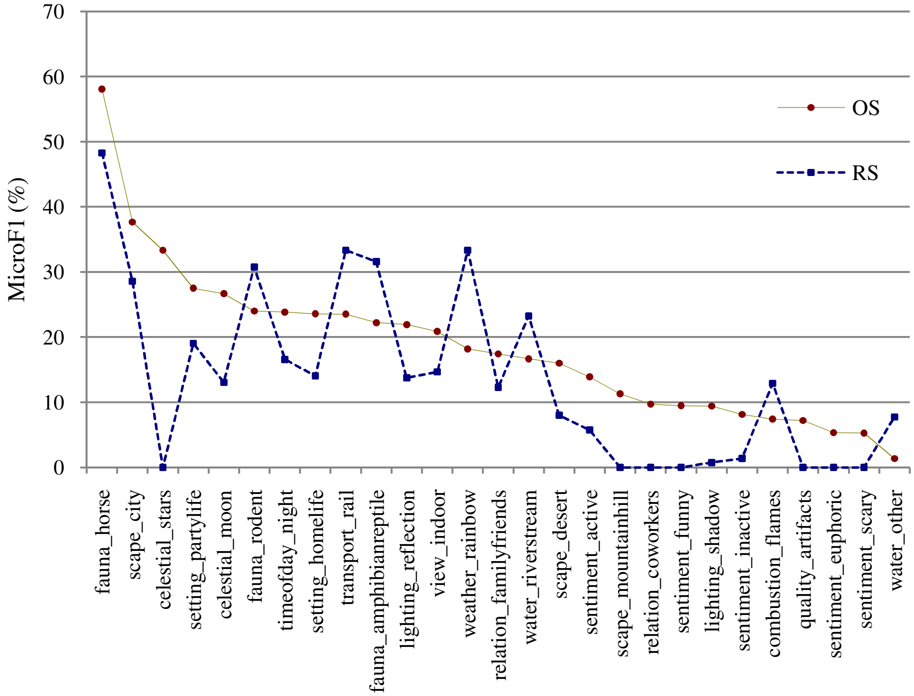


Fig. 1. The concepts with differences are larger than 5%

5 Conclusion

Automatic image annotation needs integrate a lot of heterogeneous image features for labeling images precisely. However, the numerous multi-dimension features derive the problem of high-dimension computation. In this paper, we developed the feature reduction and feature selection methods to improve the performance of image annotation. The feature reduction methods can not only reduce the dimensions of visual features and textual features but also speed-up the feature selection based on the wrapper model. The proposed feature selection method also can improve the effectiveness of image annotation. From the experimental results of evaluation, the composing method RS-OL can enhance the performance on both efficiency and effectiveness.

Although the proposed framework works in the Flickr dataset, further studies on advanced issues are addressed in the following. First, the dimensions of the reduced features are dependent on the number of concepts. Second, the small set of concept examples is difficult to handle and discover the important features. Third, how to get more image features to increase the precision of labeling images. All above issues are worth investigating in the future.

Acknowledgements. This research was supported in part by the National Science Council of Taiwan, R.O.C. under the grant NSC 101-2815-C-024-020-E.

References

1. Lin, Y.X., Chien, B.C.: Efficient Feature Reduction for High-Precision Text Classification. In: Proceedings of National Computer Symposium on Databases, Data Mining, and Information Retrieval, Chia-Yi, Taiwan (2011)
2. Kohavi, R., John, G.: Wrappers for Feature Subset Selection. Artificial Intelligence 97, 273–324 (1997)
3. Liu, W.Y., Dumais, S., Sun, Y., Zhang, H.J., Czerwinski, M., Field, B.: Semi-Automatic Image Annotation. In: Human-Computer Interaction, pp. 326–333 (2001)
4. Carneior, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised learing of semantic class for image annotation and retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 394–410 (2007)
5. Escalante, H.J., Hernández, C.A., Gonzalez, J.A., López-López, A., Montes, M., Morales, E.F., Sucar, L.E., Villaseñor, L., Grubinger, M.: The segmented and annotated IAPR TC-12 benchmark. Computer Vision and Image Understanding 114, 419–428 (2010)
6. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. In: 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 119–126 (2003)
7. Wang, X.J., Zhang, L., Li, X., Ma, W.Y.: Annotation Images by Mining Image Search Result. IEEE Transactions of Pattern Analysis and Machine Intelligence 30(11), 1919–1932 (2008)
8. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: 14th International Conference on Machine Learning, pp. 412–420 (1997)
9. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Techinical Report. no. 666, Department of Statistics, University of Berkely (2004)
10. Huiskes, M.J., Lew, M.S.: The MIR Flickr Retrieval Evaluation. In: ACM International Conference on Multimedia Information Retrieval (MIR 2008), Vancouver, Canada (2008)
11. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image Indexing Using Color Correlograms. In: Conference on Computer Vision and Pattern Recognition, CVPR 1997, San Juan, Puerto Rico, pp. 762–768 (1997)
12. Chang, S.F., Sikora, T., Puri, A.: Overview of the mpeg-7 standard. IEEE Transactions on Circuits and Systems for Video Technology, 688–695 (2001)
13. Chatzichristofis, S.A., Boutalis, Y.S.: CEDD: Color and Edge Directivity Descriptor: A Compact Descriptor for Image Indexing and Retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 312–322. Springer, Heidelberg (2008)
14. Won, C.S., Park, D.K., Park, S.J.: Efficient Use of MPEG-7 Edge Histogram Descriptor. Electronics and Telecommunications Research Institute 24(1), 23–30 (2002)
15. Chatzichristofis, S.A., Boutalis, Y.S.: Fcth: Fuzzy color and texture histogram a low level feature for accurate image retrieval. In: 9th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2008, Klagenfurt, Austria, pp. 191–196 (2008)
16. Manjunath, B.S., Ma, W.Y.: Texture Features for Browsing and Retrieval of Image Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(8), 837–842 (1996)
17. Wallace, G.K.: The JPEG Still Picture Compression Standard. Communications of the ACM - Special Issue on Digital Multimedia Systems 34, 30–44 (1991)

A Discriminative Multi-Objective Programming Method for Solving Network DEA

Han-Ying Kao* and Chieh-Yu Chan

Department of Computer Science and Information Engineering,
National Dong Hwa University, Hualien, Taiwan
{teresak@mail,610021005@ems}.ndhu.edu.tw

Abstract. This study proposes the multi-objective programming (MOP) method for solving network DEA models. In the proposed method, the efficiency of each division (within a DMU) and the overall efficiency of the DMU are formulated as different objective functions. By the fuzzy approach, the overall as well as the divisional efficiencies can be computed in a cohesive framework. We adopt information entropy to estimate the discriminating power of various DEA models. The results show that this study obtains satisfactory and discriminating efficiency scores compared with related studies.

Keywords: Network DEA, BCC models, multi-objective programming, information entropy.

1 Introduction

Data envelopment analysis (DEA) [3,5] has been widely used in assessing the relative efficiencies of decision making units (DMUs). Compared to conventional DEA models that operate by a black-box or a separation approach, network DEA arises due to its competence in evaluating the performance without neglecting internal interactions within DMUs.

Recently Tone and Tsutsui [8] propose a general slack-based network DEA approach called Network SBM that can deal intermediate products formally. Cook et al. [6] review a special type of two-stage network DEA models where all the outputs from the first stage are the only inputs to the second stage and classify all methods in two-stage DEA into Stackelberg and cooperative methods. Besides Cook et al. [6] examine the general problem of an open multistage process by presenting the overall efficiency as an additive weighted average of the efficiencies of the individual components or stages. In response to Tone and Tsutsui's work [8], Fukuyama and Mirdehghan[7] propose a network DEA approach for identifying the efficiency status of each DMU and its divisions so that the inappropriate decision due to multiple optima can be avoided.

This study proposes a multi-objective programming (MOP) method for solving the network DEA problems. Based on the CCR model [3] or BCC model [1], the overall

* Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 327–335.
DOI: 10.1007/978-3-642-35452-6_34

as well as divisional efficiencies within a DMU are defined as different objective functions to be measured cohesively. We formulate the example in Tone and Tsutsui [8] by the BCC MOP model and compare the compromise and non-compromise solutions from the MOP. To compare the discriminating power of various network DEA methods, the entropy of different solutions are analyzed. The rest of this paper is organized as follow. In section 2, we develop the BCC MOP and use Zimmermann's approach [9] to solve the model. In section 3, we solve the illustrative example and compare the results with the related methods. Finally the concluding remarks are given in section 4.

2 The Model

This work follows the notation systems from Tone and Tsutsui [8]. Consider n DMUs (j = 1,…, n) consisting of K divisions (k = 1,…, K). Let m_k and r_k be the numbers of inputs and outputs to Division k, respectively. We denote the link streaming from Division k to Division h by (k,h) and the set of links by L. The observed input resources to DMU$_j$ at Division k are $\{\mathbf{x}_j^k \in R_+^{m_k}\}$ (j = 1,…, n; k = 1,…, K); the output products from DMU$_j$ at Division k are $\{\mathbf{y}_j^k \in R_+^{r_k}\}$ (j = 1,…, n; k = 1,…,K); the linking intermediate products from Division k to Division h are $\{\mathbf{z}_j^{(k,h)} \in R_+^{t_{(k,h)}}\}$ (j = 1,…, n; $(k,h) \in L$) where $t_{(k,h)}$ is the number of items in Link (k,h). The variable returns to scale (BCC) model evaluating for a general network structure is proposed as follow.

BCC MOP model

$$Max \quad E_o^k = \frac{\sum_{r=1}^{r_k} u_r^k Y_{ro}^k + \sum_{\forall(k,h)} \sum_{p=1}^{t_{(k,h)}} \mu_h^k Z_{op}^{(k,h)} - \alpha_k}{\sum_{i=1}^{m_k} v_i^k X_{io}^k + \sum_{\forall(g,k)} \sum_{q=1}^{t_{(g,k)}} \omega_g^k Z_{oq}^{(g,k)}} \qquad k = 1,\ldots,K$$

$s.t.$

$$\frac{\sum_{r=1}^{r_k} u_r^k Y_{rj}^k + \sum_{\forall(k,h)} \sum_{p=1}^{t_{(k,h)}} \mu_h^k Z_{jp}^{(k,h)} - \alpha_k}{\sum_{i=1}^{m_k} v_i^k X_{ij}^k + \sum_{\forall(g,k)} \sum_{q=1}^{t_{(g,k)}} \omega_g^k Z_{jq}^{(g,k)}} \le 1 \tag{1}$$

$$j = 1,2,\ldots,n; k = 1,2,\ldots,K$$

$$u_r^k, v_i^k, \mu_h^k, \omega_g^k \ge \varepsilon > 0, \alpha_k \text{ unrestricted in sign}, k = 1,2,\ldots,K$$

$$r = 1,2,\ldots,r_k; i = 1,2,\ldots,m_k; all\ (k,h),(g,k) \in L$$

In the BCC MOP model for the network DEA, the objective function E_o^k measures the efficiency of Division *k* at DMU*o*, where the weighted links outgoing from Division *k*, $\mu_h^k Z_{jp}^{(k,h)}, \forall (k,h), p = 1,\ldots,t_{(k,h)}$, are regarded as the (intermediate) outputs of Division *k*, and the incoming inputs to Division *h*. To ensure its solvability, the extra constraint $\sum_{i=1}^{m_k} v_i^k X_{io}^k + \sum_{g=1}^{t_{(g,k)}} \omega_g^k Z_{ro}^{(g,k)} = 1$ is added for each division to transform the BCC model into the linear BCC model [2]. The overall efficiency of DMU*o*, E_o, will be defined as the weighted mean of all divisional efficiencies within DMU*o*. The network DEA in (1) is designed as a cooperative model; that is, the strategic resources are allocated collaboratively by each division as well as the enterprise decision level.

We adopt the fuzzy approach proposed by Zimmermann [9] and develop the following algorithm to solve the model in (1).

Step 1: Get the ideal solution of every objective.

To obtain the ideal solution, every objective is optimized independently regardless of other objectives. For DMU$_o$, we maximize $E_o^k, k = 1,\ldots,K$ individually to acquire their ideal objective values, $E_o^{k*}, k = 1,\ldots,K$.

Step 2: Get the anti-ideal solution of every objective.

To obtain the anti-ideal solution, every objective is computed in the opposite way regardless of other objectives. Now, we minimize $E_o^k, k = 1,\ldots,K$ to acquire their anti-ideal solutions $E_o^{k-}, k = 1,\ldots,K$.

Step 3: Define the membership function of every objective by its ideal and anti-ideal solutions as follow.

$$\mu_{E_o^k}(E_o^k) = \frac{E_o^k - E_o^{k^-}}{E_o^{k^*} - E_o^{k^-}} \tag{2}$$

The membership function evaluates the fulfillment level of every objective.

Step 4: Two types of solutions will be obtained according to the decision setting: *compromise* and *non-compromise* solutions.

The compromise solution maximizes the weighted mean membership function of all objective functions as follow.

Compromise

$$\begin{aligned} &Max \quad \lambda = \sum_{k=1}^{K} w_k \mu_{E_o^k}(E_o^k) \\ &s.t. \quad \text{all original constraints} \end{aligned} \tag{3}$$

where w_k is the weight for Division *k*. On the other hand, the *non-compromise* solution maximizes the minimum of the membership functions for all objective functions.

Non-compromise

$$\begin{aligned} &Max \quad \lambda = \min_{k=1,\ldots,K} \{\mu_{E_o^k}(E_o^k)\} \\ &s.t. \quad \text{all original constraints} \end{aligned} \tag{4}$$

where λ is the minimum of all divisional efficiencies to be maximized. In (3) and (4), every division in DMU$_o$ can be evaluated simultaneously, and then the overall score E_o is measured for DMU$_o$. This work will implement both compromise and non-compromise models for comparison.

3 The Illustrative Example

This study demonstrates the example of electronic power companies in Tone and Tsutsui [8]. The business process model is in Fig. 1 and the sample data are exhibited in Table 1.

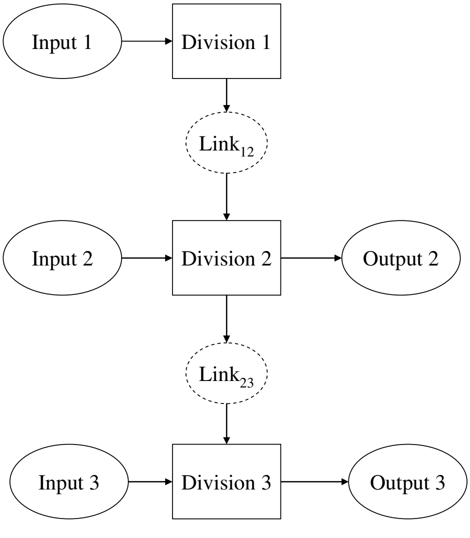


Fig. 1. The illustrative process model

We solve the fixed-link cases using the BCC MOP model in (1) by using the algorithm described in section 2. The results of the BCC MOP models are compared with those of other methods in Table 2 to Table 5 and Fig. 2 to Fig. 5.

In Fig. 2, the trends for both BCC MOP models are approximately similar; while separate and SBM reveal another pattern. In Table 2, there is one efficient DMU

Table 1. The sample data

DMU	*Div1*	*Div2*		*Div3*		*Link*	
	Input1	Input2	Output2	Input3	Output3	Link12	Link23
A	0.838	0.277	0.879	0.962	0.337	0.894	0.362
B	1.233	0.132	0.538	0.443	0.18	0.678	0.188
C	0.321	0.045	0.911	0.482	0.198	0.836	0.207
D	1.483	0.111	0.57	0.467	0.491	0.869	0.516
E	1.592	0.208	1.086	1.073	0.372	0.693	0.407
F	0.79	0.139	0.722	0.545	0.253	0.966	0.269
G	0.451	0.075	0.509	0.366	0.241	0.647	0.257
H	0.408	0.074	0.619	0.229	0.097	0.756	0.103
I	1.864	0.061	1.023	0.691	0.38	1.191	0.402
J	1.222	0.149	0.769	0.337	0.178	0.792	0.187

Table 2. Comparisons of the overall scores

DMU	Compromise BCC MOP	Non-compromise BCC MOP	Separate	SBM(fixed)
A	0.844	0.708	0.659	0.478
B	0.739	0.678	0.657	0.739
C	0.909	0.840	0.984	0.968
D	0.943	0.857	0.719	0.719
E	1.000	1.000	0.547	0.456
F	0.677	0.541	0.844	0.719
G	0.901	0.881	0.855	0.947
H	0.965	0.932	0.893	0.969
I	0.768	0.470	0.915	0.832
J	0.862	0.857	0.64	0.59
Entropy	**2.122**	**2.522**	**2.171**	**2.171**

(E_E =1) in both BCC-MOP and no efficient DMU in separate and SBM models. Due to the significant difference between two groups of models for the score E_E, we refer to the results of the black-box approach in Tone and Tsutsui [8], where E_E =1 (efficient). Unlike the black-box approach by which 8 in 10 DMUs are efficient, BCC MOP models reveal more discriminating in evaluation. To compare the discriminating power of different methods, we use the information entropy

$$-\sum_{i=1}^{c} p(E_o \in class_i)\log_2 p(E_o \in class_i) \quad (5)$$

where c is the number of data classes and the class intervals are 1, [0.9000, 0.9999], [0.8000, 0.8999], [0.7000, 0.7999], and so on. The entropys show that two BCC MOP models are more discriminating in computing overall scores and can avoid absence of efficient DMUs.

Table 3. Comparisons of the scores of Div 1

DMU	Compromise BCC MOP	Non-compromise BCC MOP	Separate	SBM(fixed)
A	0.448	0.448	0.633	0.633
B	0.261	0.177	0.26	0.349
C	1.000	0.840	1.000	1.000
D	0.254	0.254	0.297	0.297
E	0.202	0.202	0.202	0.263
F	0.528	0.541	1.000	1.000
G	0.595	0.627	0.712	1.000
H	0.721	0.733	0.787	0.922
I	0.419	0.470	1.000	1.000
J	0.218	0.225	0.263	0.288
Entropy	**2.122**	**2.646**	**1.846**	**2.046**

Table 4. Comparisons of the scores of Div 2

DMU	Compromise BCC MOP	Non-compromise BCC MOP	Separate	SBM(fixed)
A	0.609	0.538	0.662	0.339
B	0.329	0.649	0.763	1.000
C	1.000	0.840	1.000	1.000
D	1.000	1.000	1.000	1.000
E	1.000	1.000	1.000	1.000
F	0.668	0.687	0.635	0.403
G	0.965	0.881	1.000	1.000
H	0.916	0.859	0.89	1.000
I	1.000	0.555	1.000	1.000
J	0.855	0.733	0.672	0.377
Entropy	**2.122**	**2.246**	**1.685**	**1.157**

Table 5. Comparisons of the scores of Div 3

DMU	Compromise BCC MOP	Non-compromise BCC MOP	Separate	SBM(fixed)
A	0.977	0.691	0.684	0.393
B	0.675	0.678	1.000	1.000
C	0.773	0.840	0.959	0.919
D	1.000	1.000	1.000	1.000
E	0.959	0.959	0.665	0.377
F	0.689	0.593	0.792	0.596
G	0.920	0.914	0.926	0.868
H	1.000	0.932	1.000	1.000
I	0.992	0.984	0.786	0.581
J	0.825	0.857	1.000	0.825
Entropy	**1.193**	**1.458**	**1.685**	**2.246**

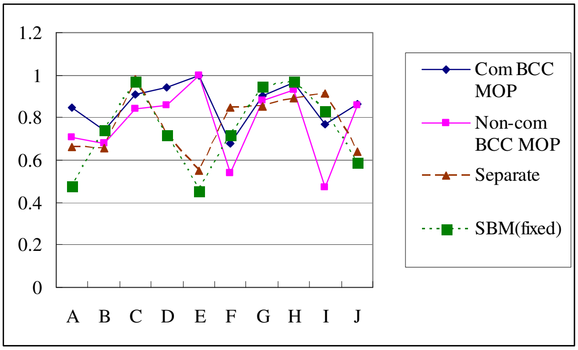


Fig. 2. Comparisons of the overall scores

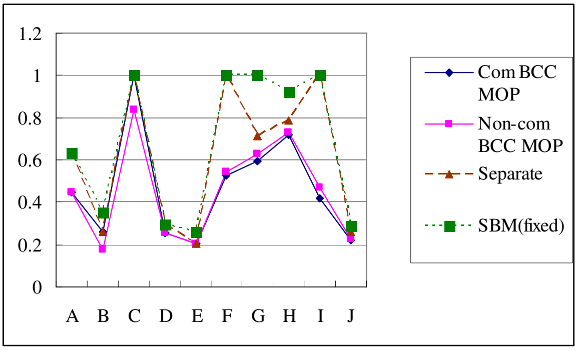


Fig. 3. Comparisons of the scores of Division 1

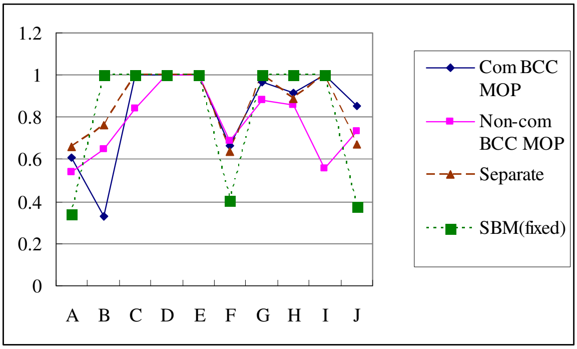


Fig. 4. Comparisons of the scores of Division 2

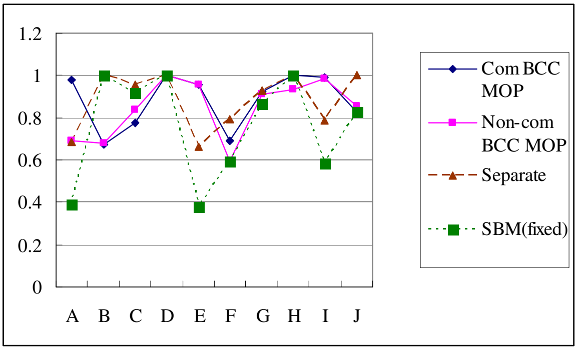


Fig. 5. Comparisons of the scores of Division 3

Observing the divisional efficiencies, two BCC MOP models obtain higher entropy except for Division 3. In Table 5, the separate and SBM identify 4 and 3 efficient divisions, respectively, while the compromise and non-compromise BCC MOP models classify 2 and 1 efficient divisions, respectively. In short, the BCC MOP models, especially the non-compromise type, identify relatively less efficient divisions and are more discriminating than other methods.

4 Conclusions

In this study, we propose the alternative approaches and solutions for network DEA. In our compromise and non-compromise BCC MOP models, the overall as well as divisional efficiencies for DMUs are evaluated in a cohesive way. The results show that both BCC MOP models obtain useful and discriminative outcomes for assessing DMUs and the divisions within the DMUs.

Acknowledgement. The authors are indebted to the comments and suggestions from the anonymous reviewers. This study is support by the National Science Council, Taiwan (Project no. 99-2410-H-259-045-MY2).

References

1. Banker, R.D., Charnes, A., Cooper, W.W.: Some models for estimating technical and scale inefficiencies in data envelopment analysis. Management Science 30, 1078–1092 (1984)
2. Charnes, A., Cooper, W.W.: Programming with linear fractional functionals. Naval Research Logistics Quarterly 9, 181–186 (1962)
3. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2, 429–444 (1978)
4. Cook, W.D., Liang, L., Zhu, J.: Measuring performance of two-stage network structures by DEA: A review and future perspective. Omega 38, 423–430 (2010)
5. Cook, W.D., Seiford, L.M.: Data envelopment analysis (DEA) – Thirty years on. European Journal of Operational Research 192, 1–17 (2009)
6. Cook, W.D., Zhu, J., Bi, G., Yang, F.: Network DEA: Additive efficiency decomposition. European Journal of Operational Research 207, 1122–1129 (2010)
7. Fukuyama, H., Mirdehghan, S.M.: Identifying the efficiency status in network DEA. European Journal of Operational Research 220, 85–92 (2012)
8. Tone, K., Tsutsui, M.: Network DEA: A slacks-based measure approach. European Journal of Operational Research 197, 243–252 (2009)
9. Zimmermann, H.J.: Fuzzy programming and linear programming with several objective functions. Fuzzy Sets and Systems 1, 45–55 (1978)

A Bibliometric Analysis of the Theses and Dissertations on Information Literacy Published in the United States and Taiwan

Pao-Nuan Hsieh[1], Tao-Ming Chuang[2], and Mei-Ling Wang[3,*]

[1] Department of Library and Information Science, National Taiwan University
pnhsieh@ntu.ed.tw
[2] Department of Information and Communications, Shih Hsin University, Taipei, Taiwan
tmchuang@cc.shu.edu.tw
[3] Graduate Institute of Library, Information and Archival Studies,
National ChengChi University
meilingw@nccu.edu.tw

Abstract. The purpose of this study is to explore the characteristics of the theses and dissertations on information literacy from 1988 to 2010 in the United States and Taiwan, such as publishing universities, paper growth, author/ advisor productivity, type of literacy, and research methods. The comparison of theses and dissertations on information literary research is made between those completed in the United States and Taiwan. This study investigates and maps the trends in information literacy research by applying bibliometric analysis to the 767 theses and dissertations in the field of information literacy in the United States and Taiwan. The study reveals that theses and dissertations on information literacy in Taiwan grew rapidly (502, 65.45%) and more were published than in the United States (265, 34.55%), although the first doctoral dissertation published in the United States was in 1988 while the first master thesis published in Taiwan was in 1996.The rates at which they dealt with information literacy, media literacy, and digital literacy were respectively 54.57%, 30.59%, and 14.84%, there are significant differences between the United States and Taiwan in the three types of literacy research. Furthermore, the type of methodology implemented in theses and dissertations in the United States is different from that used in Taiwan.

Keywords: information literacy, media literacy, digital literacy, United States, Taiwan.

1 Introduction

As the information society has been approaching in the twentieth century, faced with highly- developed computer and communication technology, and the information explosion, people encounter the difficulties of how to find information to solve

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 337–348.
DOI: 10.1007/978-3-642-35452-6_35

problems. On the other hand, higher education is also seeking to reform, coupled with the promotion of the concept of lifelong education and learning, and so has contributed to the information literacy research. (Wang, et al., 2012) In 1989, the American Library Association first proposed the important document entitled "The American Library Association Presidential Committee on Information Literacy Final Report, "in which it advocated that every person and organization should be information literate. Subsequently, in the United Kingdom, Australia, Europe, and Taiwan, as well as many countries, a variety of projects and activities related to information literacy were held, including seminars, research projects, and related curricula.(American Library Association, 1989)

The definition of information literacy developed by the American Library Association is the ability to "recognize when information is needed and have the ability to locate, evaluate, and use effectively the needed information." (American Library Association, 2012) The ERIC (Education Resources Information Center) Database defines information literacy as "the ability to access, evaluate, and use information from a variety of sources." (ERIC, 1992) According to these definitions, one could conclude that information-literate people are those who are in possession of the necessary knowledge, skills, and attitudes to apply information technology effectively to collect, analyze, assess, organize and synthesize information for solving problems encountered in their living and working needs.

Information literacy is a multifaceted dimensional concept, commonly referred to as "information skills." The terms related to information literacy in the ERIC database include access to information; computer literacy; information seeking; information skills; information utilization; librarian teacher cooperation; library instruction; library skills; multiple literacy; online searching; scientific literacy; search strategies; technological literacy; and users (information). (ERIC, 1992) McClure (1994) identified information literacy as "the ability to solve information problems is a high-level literacy covering four kinds of literacy, including traditional literacy in reading and writing, computer literacy, media literacy, and network literacy."

Information literacy research has developed since 1990, governments have promoted national information literacy competency via inclusion in national basic education policy through a variety of plans or policies in the UK, Australia, Canada, France, Spain, and Taiwan. Information literacy research with diverse and rich content covers exploring the definition and connotations of information literacy, how to cultivate information literacy, how to provide information literacy education and curricula, as well as various levels of information literacy competency such as those of primary and secondary school students, undergraduates ,adults, workers, professionals, etc.

Bruce (2000) analyzed the literature on information literacy during 1980-2000 and found information literacy researchers beginning to develop a collective consciousness that represents the newly-appearing territory of information literacy research. This study divides information literacy research into four phases: precursor's period, experimental period, exploring period, and evolving period. In the precursor's period (1980s), information literacy research was to create a period when people explored information skills and bibliographic education. In the experimental period (1990-1995), many schools provided information literacy education and discussed research issues. The United Kingdom and Australia also began to study information literacy. In the exploring period (1990-1995), information literacy research was marked by the identification and exploration of different paradigms for information literacy research, and the offering of multiple research agendas. While most research was conducted within, and for, the educational sector, interest in workplace-based research began to emerge. In the evolving period (since 2000), information literacy research has come to include the development of a community of researchers and research teams; growth in research beyond the educational sector, particularly in the workplace and community; attention has been paid to a wider variation of cultural settings.

In 1993, T. C. Lee of the National Taiwan University, and G. Wang and S. T. Wu, of the National Cheng-Chi University, started to undertake fundamental study on information literacy from different research approaches with funding support from the National Science Council of Taiwan, the former from the perspective of library science, and the latter form the perspective of media communication. (Lee, 1994; Wang & Wu, 1994). Since 2001, the number of Internet users in Taiwan has grown rapidly, people using the Internet have become younger, under such circumstances, information literacy gradually received attention. Many discussions were held regarding the information literacy competency of IT teachers, IT researchers, librarians, and university faculty (Chuang, 2009). As the trend of network development emerged in Taiwan, many universities, libraries, departments and institutes of Information Science, Education, Information Management, Business Administration, Mass Communication, Information and Communication began to engage in information literacy research. So far, Taiwan's information literacy research has developed for more than ten years, but the lack of review research represents a gap in information literacy. The study mainly explores information literacy research in the United States and Taiwan with bibliometric analysis of theses and dissertation during 1988-2010.

The purpose of this study is to explore the characteristics of the theses and dissertations on information literacy, such as publishing universities, literature growth and author/advisor productivity, type of literacy, and research methods. The comparison of theses and dissertations on information literary research is made between those published in the United States and Taiwan, in addition, a detailed analysis and comparison of information literacy, media literacy, and digital literacy between the two countries is also proposed. This topic was identified as being of importance to cultivate information-literate citizens and to promote information literacy research in the Internet age.

2 Method

This study, based on the above concept of information literacy, concludes that an information literacy model should include (1) information literacy: skills for acquiring, analyzing, organizing, and evaluating information; (2) digital literacy: the ability of digitization and utilization of information resources; (3) media literacy: understanding the creation, analysis, and evaluation of media messages.

For the present work, the National Digital Library of Theses and Dissertations in Taiwan and the ProQuest Dissertations & Theses have been employed to retrieve bibliographic data of theses and dissertations on information literacy from 1988 to 2010. All of the master theses and doctoral dissertations on information literacy submitted to National Central Library in Taiwan were searched from the online database: The National Digital Library of Theses and Dissertation in Taiwan (NDLTD, 臺灣博碩士論文知識加值系統) from 1996 (the date of the first theses presented in the NDLTD) to 2010 were located and in total 502 theses were analyzed.

An international sample (N=265) of doctoral dissertations primarily from North America from 1988 to 2010 forms the second part of the material for this analysis. This sample was derived from a keyword search (for "information literacy" OR "digital literacy" OR "media literacy" OR "internet literacy" In Title, Subject, Keyword) in an online subscription database: ProQuest Dissertations & Theses (PQDT, most of the contents of which come from the United States and Canada). Titles and abstracts from the results of the initial search were examined in order to exclude results which were clearly unrelated to information literacy and theses/dissertations for which abstracts were not provided.

3 Results

3.1 The Growth of Theses/Dissertations on Information Literacy Production

The most important finding of this research is that Taiwan (which has published 502 theses/dissertations, 65.45%) published more theses/dissertations than United States (265, 34.55%) up to the date to which we carried out our study, although the first doctoral dissertation published in United States was in 1988 while the first master thesis published in Taiwan was in 1996. This research also reveals that the years when the most research was submitted were 2008 in United States, when 35 papers were submitted, and 2009 in Taiwan, when 61 papers were published. In Taiwan, two theses were prepared in 1996 but at doctoral degree level, no dissertations were prepared between the years of 1996-1999, and only seven doctoral dissertations were written on information literacy between 1996 and 2010.

It can be observed that a very big increase in terms of theses/dissertations published took place in both the United States and Taiwan during the years of 2000-2010. Figure 1 shows the evolution of production since 1988. There has been an ongoing increase in the number of theses/dissertations written on information literacy. From annual paper analysis, the development of theses/dissertations on information literacy in Taiwan can be divided into three stages. In the first stage, before the year 2000, less than 10 theses/dissertations were approved, at the second stage, 2001-2003, the number of papers increases from 10-40; after 2004, the number of papers remains higher than 40, showing a steady growth trend. Overall, the theses/dissertations growth rate is in a steadily up-growing trend since 2000 in the U.S. and in Taiwan.

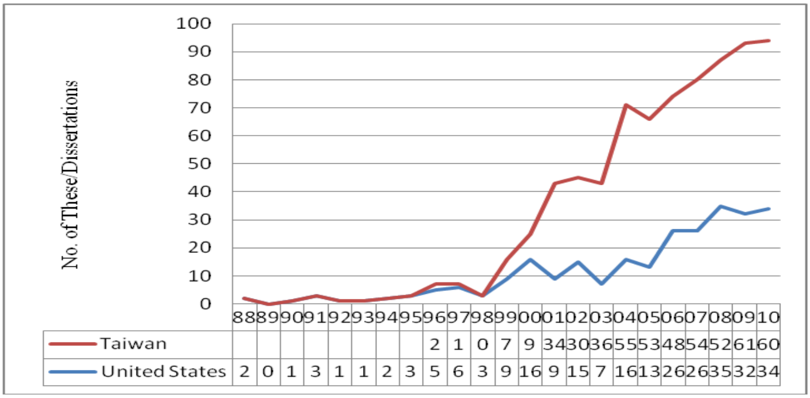


Fig. 1. Evolution of theses/dissertations on information literacy between 1988 and 2010

3.2 Some Bibliometric Characteristics of Theses/Dissertations on Information Literacy

The 265 theses and dissertations on information literacy in the United States come from 159 universities, additionally, the 502 theses and dissertations on information literacy in Taiwan come from 71 universities, as shown in Table 1.

Table 1. Theses/dissertation production of information literacy, 1988-2010

	Theses	Dissertation	Subtotal	Universities
United States	NA	265	265(34.55%)	159
Taiwan	495	7	502(65.45%)	71
Total	495	272	767	230

The 25 most productive universities are listed in Table 2. It should be noted that a university may be a single department/institute or may consist of several departments/institutes. Among them, there are 25 universities which have produced more than 91 dissertations on information literacy in the U.S. The Florida State University and University of Pittsburgh are the first locations for the submission of six dissertations on information literacy to PQDT respectively. Temple University and University of Massachusetts Amherst are the second, making up five dissertations respectively. Table 2 shows the universities which have registered more than three dissertations on information literacy in PQDT.

Table 3 lists the 16 most productive universities in Taiwan which have registered more than ten theses/dissertations on information literacy in Taiwan from 1996-2010. It should be noted that a university may be a single department/institute or may consist of several departments/institutes. The National Taiwan Normal University is the largest contributor, approving 29 theses/dissertations. The National Kaohsiung Normal University is the second-largest contributor, approving 26 theses/dissertations.

Table 2. The most productive universities of information literacy theses/dissertations in the U.S., 1988-2010

University	Dissertations	Percentage
The Florida State University	6	2.3
University of Pittsburgh	6	2.3
Temple University	5	1.9
University of Massachusetts Amherst	5	1.9
Capella University	4	1.5
Nova Southeastern University	4	1.5
The University of North Carolina at Chapel Hill	4	1.5
The University of Wisconsin - Madison	4	1.5
University of North Texas	4	1.5
University of Toronto (Canada)	4	1.5
Columbia University Teachers College	3	1.1
New York University	3	1.1
Oklahoma State University	3	1.1
State University of New York Empire State College	3	1.1
Syracuse University	3	1.1
The University of Alabama	3	1.1
The University of Iowa	3	1.1
The University of New Mexico	3	1.1
University of California, Los Angeles	3	1.1
University of Central Florida	3	1.1
University of Florida	3	1.1
University of Maryland, College Park	3	1.1
University of South Carolina	3	1.1
University of Virginia	3	1.1
Walden University	3	1.1

Table 3. Universities in Taiwan which have registered more than ten theses/dissertations on information literacy in Taiwan, 1996-2010

University	Theses/diss.	Percentage
National Taiwan Normal University	29	5.8
National Kaohsiung Normal University	26	5.2
National Taipei University of Education	24	4.8
National Changhua University of Education	23	4.6
National Chiayi University	22	4.4
National Chengchi University	20	4.0
National University of Tainan	19	3.8

Table 3. (*continued*)

Tamkang University	18	3.6
National Chung Cheng University	15	3.0
National Taichung University of Education	13	2.6
Shih Hsin University	12	2.4
National Pingtung University of Education	12	2.4
Taipei Municipal University of Education	11	2.2
Shu-Te University	11	2.2
Fu Jen Catholic University	10	2.0
National Taitung University	10	2.0

3.3 Advisor Productivity

There are 225 professors from 159 universities who have advised doctoral dissertations on information literacy in North America. Only nine professors have been the advisors of two doctoral dissertations, as shown in Table 4.

Table 4. Professors who have advised two doctoral dissertations written on information literacy in the U.S

Advisor	No. of dissertations
Austin, Erica Weintraub	2
Colvin, Carolyn	2
Cooks, Leda	2
Hill, Clifford	2
Hobbs, Renee	2
Ragsdale, Ronald	2
Schwarz, Gretchen	2
Truglio, Rosemarie	2
Yan, Wenfan	2

There are 356 professors who have advised theses/dissertations on information literacy in Taiwan. Table 5 shows there have been 15 advisors of at least four master theses/doctoral dissertations on information literacy in Taiwan. The most productive advisor is Professor Lin Jing (林菁) of National Chiayi University, who advised 14 papers. Regarding the schools with which these advisors were affiliated, most

Table 5. Professors who have advised more than four theses/dissertations written on information literacy in Taiwan

Advisor	University	No. of dissertations
Lin Ching Chen(林菁)	National Chiayi University(國立嘉義大學)	14
Yuanling Lai(賴苑玲)	National Taichung University of Education (國立臺中教育大學)	11
Jia-Rong Wen(溫嘉榮)	National Taichung University of Education (國立臺中教育大學)	10
Tao-Ming Chuang(莊道明)	Shih Hsin University(世新大學)	7
Tsuey Jen Wu(吳翠珍)	National Chengchi University(政治大學)	5
Chih-Lung Lin(林志隆)	National Pingtung University of Education (國立屏東教育大學)	5
Mei-Chun Yin(尹玫君)	National University of Tainan(國立臺南大學)	5
An-Kuo Chiang(蔣安國)	Fo Guang University(佛光大學)	4
Chang-bin Wang(王昌斌)	Nanhua University(南華大學)	4
Yau-Jane Chen(陳姚真)	National Chung Cheng University(國立中正大學)	4
Wen-Shyong Dai(戴文雄)	National Changhua University of Education (國立彰化師範大學)	4
Chia-Jung Lin(林佳蓉)	National Taipei University of Education(國立臺北教育大學)	4
Pao-Nuan Hsieh(謝寶煖)	National Taiwan University(國立臺灣大學)	4
Chao-chen Chen(陳昭珍)	National Taiwan Normal University(國立臺灣師範大學)	4
Hou-Long Yen(顏火龍)	National University of Tainan(臺南師範學院)	4

specialized in education, library and information science, management information systems and communication.

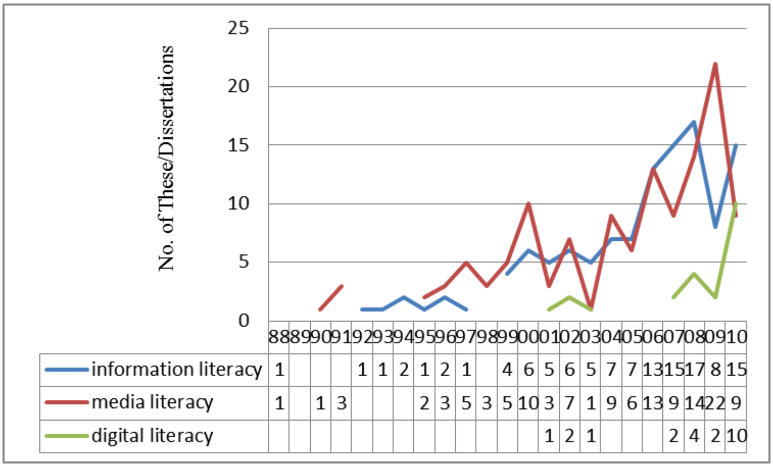
3.4 Distribution of Theses/Dissertations on Information Literacy, Media Literacy and Digital Literacy

In terms of all theses/dissertations (n=767), the percentage of those which dealt with information literacy, media literacy, and digital literacy was respectively 54.57%, 30.59%, and 14.84%. There are significant differences between the papers published in the United States and those published in Taiwan. The number of dissertations focusing on media literacy in the United States is slightly higher than those in Taiwan, but those theses/dissertations focusing on information literacy are dominant with a percentage of about 59%. Table 6 compares the difference of the distribution of three types of literacy between the United States and Taiwan. The distribution of theses/dissertations among the three type of literacy in the United States and in Taiwan is shown in Figure 2 and Figure 3.

Table 6. The distribution of theses/diss. on information literacy, media literacy and digital literacy in the U.S. and Taiwan

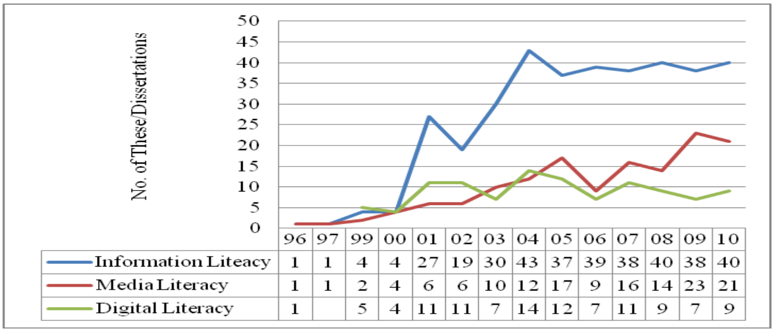
Type of literacy	United States	Taiwan	Total
information literacy	117 (44.2%)	361 (59.08%)	478 (54.57%)
media literacy	126 (47.5%)	142 (23.24%)	268(30.59%)
digital literacy	22 (8.3%)	108 (17.68%)	130(14.84%)
Total	265	611*	876

*Among the total of 502 literacy research papers in Taiwan, there are 99 papers whose research subject covers two or more literacies.



	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10
information literacy	1				1	1	2	1	2	1		4	6	5	6	5	7	7	13	15	17	8	15
media literacy	1		1	3				2	3	5	3	5	10	3	7	1	9	6	13	9	14	22	9
digital literacy														1	2	1				2	4	2	10

Fig. 2. The distribution of theses/dissertations between three types of literacy in the United States



	96	97	99	00	01	02	03	04	05	06	07	08	09	10
Information Liteacy	1	1	4	4	27	19	30	43	37	39	38	40	38	40
Media Literacy	1	1	2	4	6	6	10	12	17	9	16	14	23	21
Digital Literacy	1		5	4	11	11	7	14	12	7	11	9	7	9

Fig. 3. The distribution of theses/dissertations between three types of literacy in Taiwan

3.5 Research Method Adopted by Theses/Dissertations on Information Literacy

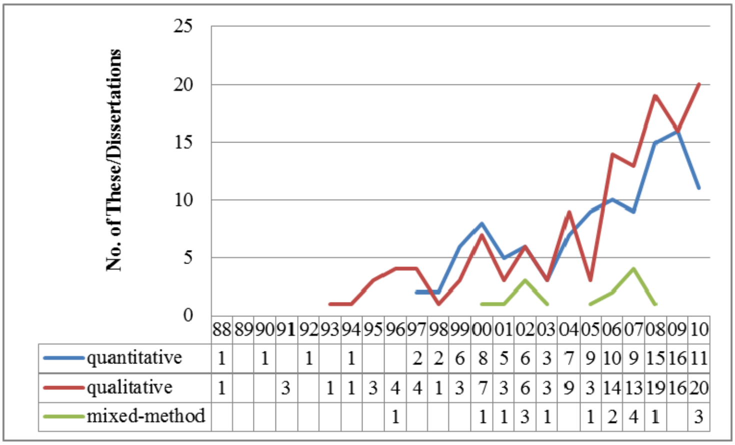
In analyzing the type of methodology implemented in theses/dissertations, there are significant differences between the United States and Taiwan, as shown in Table 7.

The majority of doctoral dissertations on information literacy in the United States used qualitative research designs (n=133, 50.2%), although quantitative designs (114, 43%) and mixed-method designs (18, 6.8%) were also used. Figure 4 shows the distribution of research methodology adopted by theses/dissertations on information literacy in the United States.

On the other hand, most of the master theses (and seven doctoral dissertations) on information literacy in Taiwan adopted quantitative research designs (431, 85.9%), and only 14.1% adopted qualitative research designs. Within the 431 quantitative theses/dissertations, survey research (384, 76.5%) appears to be the predominant method of inquiry. Furthermore, it appears that a minority of theses/dissertations used experimental and quasi-experimental designs (25, 5%) to investigate their research questions. Within the 71 qualitative research designs, there were 25 (5%) which used observation, 17 (3.4%) used interviews, and 11 (2.2%) used action research.

Mixed methods is defined as research in which the researcher collect and analyzes data, integrates the findings, and draws inferences using both qualitative and quantitative approaches or methods in a single study(Creswell, 2008). According to Creswell, mixed methods research has attracted substantial interest during the past 20 years. In our research, the first dissertation adopting mixed methods approach appeared in 1996, and the second dissertation with mixed methods arrived three years later.

In Taiwan, information literacy research is highly related to information education research, as well as extending to other disciplines including information technology, information management, library and information science, and communication studies. Regarding study themes, information literacy research has been guided by government policies, including those of the Ministry of Education, to promote IT education and media literacy education in elementary and junior high schools, and the Research and Development Committee, to promote the digital divide plan which contributes to a growing trend of information literacy research by year.



	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10
quantitative	1		1		1		1			2	2	6	8	5	6	3	7	9	10	9	15	16	11
qualitative	1			3		1	1	3	4	4	1	3	7	3	6	3	9	3	14	13	19	16	20
mixed-method									1				1	1	3	1		1	2	4	1		3

Fig. 4. The distribution of research method adopted by theses/dissertations on information literacy in the United States

Table 7. Methodology adopted by theses/dissertations written on information literacy

Research method	The United States				Taiwan			
	Information literacy	Media literacy	Digital literacy	Total	Information literacy	Media literacy	Digital literacy	Total
Qualitative research	**43**	**80**	**11**	**133**	**40**	**29**	**2**	**71**
Action research	4	10		14	4	6	1	11
Case study	15	21	4	40				
Comparative study		1		1				
Content analysis	3	9		12	1	4		5
Ethnographic study		6	1	7				
Grounded theory	2	2		4				
Interview	4	4		8	12	5		17
Literature review		1		1	8	3		11
Observation					13	11	1	25
Phenomenological study	2	3		5				
Qualitative research	11	17	6	34	2			
Rhetorical analysis		3		3				
Theoretical investigation	2	3		5				
Quantitative research	**65**	**41**	**7**	**113**	**321**	57	53	431
Experiment	13	26	1	40	14	9	2	25
Survey	48	15	6	69	306	47	51	404
Delphi study	4			4	1	1		2
Mixed-methods	**9**	**5**	**4**	**18**				
Total	117	126	22	265	361	86	55	502

4 Conclusions

The present work explores the characteristics of theses and dissertations on information literacy in the United States and Taiwan from 1988 to 2010 based on the databases of NDLTD and ProQuest Dissertations & Theses using bibliometric analysis. The study reveals that theses and dissertations on information literacy in Taiwan grew more rapidly (502, 65.45%) and were published more frequently than in the United States (265, 34.55%), although the first doctoral dissertation was published in the United States in 1988 while the first master thesis was published in Taiwan in

1996.The theses/dissertations growth rate has displayed a steadily up-growing trend since 2000 in the United States and in Taiwan.

The 265 theses and dissertations on information literacy in the United States come from 159 universities, furthermore, the 502 theses and dissertations on information literacy in Taiwan come from 71 universities. The most productive universities and advisors of theses and dissertations on information literacy in the United States and in Taiwan are identified.

In terms of all theses/dissertations, the rates at which information literacy, media literacy, and digital literacy are handled are respectively 54.57%, 30.59%, 14.84%, in addition, there are significant differences between the United States and Taiwan. There are more theses and dissertations on media literacy in the United States than in Taiwan, while there are more theses and dissertations on digital literacy in Taiwan than in the United States.

The present study also indicates that there are significant differences between the type of methodology implemented in theses and dissertations in Taiwan, and those implemented in the United States. The majority of doctoral dissertations on information literacy in the United States used qualitative research designs, while most of the master theses on information literacy in Taiwan adopted quantitative research designs and only 14.1% adopted qualitative research designs.

References

1. American Library Association. American Library Association Presidential Committee on Information Literacy Final Report (1989), `http://www.infolit.org/docoents/89Report.html` (retrieved August 20, 2012)
2. Bruce, C.S.: Information literacy research: Dimensions of the emerging collective consciousness. Australian Academic and Research Libraries (AARL) 31(2), 91–109 (2000)
3. Chuang, T.M.: The state-of-art in adult information literacy research in Taiwan: Thesis and dissertation. Library and Information Service 53(3), 9–14 (2009)
4. Creswell, J.W.: Mixed methods research. In: Given, L.M. (ed.) The Sage Encyclopedia of Qualitative Research Methods, Los, vol. 2, pp. 526–529 (2008)
5. ERIC. Information literacy (1992), `http://www.eric.ed.gov/ERICWebPortal/thesaurus/thesaurus.jsp` (retrieved: August 2, 2012)
6. Lee, T.C.: Guidelines and evaluation indicators of information literacy education in library information services (NSC86-2413-H002-031) (1994), Retrieved from National Science Council website:
   `https://nscnt12.nsc.gov.tw/was2/award/AsAwardMultiQuery.aspx`
7. McClure, R.C.: Network literacy: A role for libraries? Information Technology & Libraries 13(2), 115–125 (1994)
8. OECD. The definition and selection of key competencies (2008),
   `http://www.oecd.org/dataoecd/47/61/35070367.pdf`
   (retrieved August 20, 2012)
9. Wang, G., Wu, S.T.: Research on the cognition, value and importance assessment related to the information literacy (NSC83-0111-S004-002-TL) (1994), Retrieved from National Science Council website:
   `https://nscnt12.nsc.gov.tw/was2/award/AsAwardMultiQuery.aspx`
10. Wang, M.L., et al.: Library and Information Literacy Education. Open University of the ROC, Taipei (2012)

Using Mobile Annotation System in Public Health Practice Course Based on Project-Based Learning

Ting-Ting Wu

Department of Information Management, Chia-Nan University of Pharmacy and Science, Taiwan
wutt0331@mail.chna.edu.tw

Abstract. This study designed a mobile Annotation System for Health Education to assist nursing students in health education practice course. Learning strategy used PBL for planning the educational activities and assisting nursing students in internalizing professional knowledge and developing critical thinking. According to the experimental results, introduction of information technology in public health nursing practice course can really promote learning effectiveness. Furthermore, the majority of nursing students and nursing educator are showing positive attitude towards this annotation system, and looking forward to using again in related practice courses in the future.

Keywords: Public health practice course, Nursing students, Project based learning, Mobile devices.

1 Introduction

The clinical internship course during the schooling period is the most important part of nursing education (Dolan, 2003). Nursing students can understand, internalize, and form concepts toward nursing theory through real-life situation operation, and familiarizes, and accurately applies nursing skills by using clinical internship experience (Christy, 1980).

The development of mobile devices for learning has brought a new direction (Jeng, Wu, Huang, Tan and Yang, 2010), and its convenience and instantaneous features gives the users the initiative to acquire knowledge, and also the immediate access to critical knowledge to solve pressing problems (Kynaslahti, 2003). In recent years, the study of using mobile devices to support nursing students in clinical internship has become even more developed (Criswell and Parchman, 2002; Fowler, Hogle, Martini and Roh, 2002). The introduction of mobile devices in nursing practice can save manpower, minimize errors, and provide rapid access to information. In addition, prompt feedback and core knowledge support enhances nursing students' professional skills (White et al., 2005). Carroll and Christakis (2004) suggested that the introduction of mobile devices reinforced the precision of practical procedures. White et al (2005) showed that the use of mobile devices increased the students' confidence in applying their professional knowledge. Thus, this study uses thin, easy to control and large

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 349–357.
DOI: 10.1007/978-3-642-35452-6_36

screen tablet PCs as a learning tool, and introduces it in the public health nursing practice course. The instant support and services of the device can provide nursing students with a wide range of learning content and teaching strategies expand nursing practice in an innovative learning environment.

Recently, Nursing Internship courses have more emphasis on nursing students' critical thinking, clinical judgment, and problem-solving abilities (Girot, 2000). The teaching strategies during the learning activities will affect the learner's message choice, acquisition, and structure, and further affect their behavior and thinking during the process. (Weinstein and Mayer, 1986) In order to achieve the learning objectives of the internship course and integrate traditional public health teaching method, project-based learning is perfect for this learning environment.

Project-based learning was initially designed for medical students, with the purpose of training learners how to think and seek solutions. The development to date is no longer limited to medical education. It has been used on other professional areas (Barrows, 1996). Project-based learning uses exploration-oriented learning method, which choices the authentic, integrated, and challenging problems to draw conclusions and present results by asking questions, gathering information, analyzing, sorting, and using inductive reasoning (Blumenfeld, Soloway, Marx, Krajcik, Guzdial and Palincsar, 1991). This allows the learners to structure their own knowledge system and provide actual experience with the pursuit of solving real problems. This learning pedagogy focuses on cultivating the ability of self-learning, and solving problems (Krajcik, Czerniak, and Berger, 1999).

Therefore, this research used the project-based learning as the base of strategic design and constructed the Annotation System for Health Education (ASHE) on the tablet PCs to fit public health nursing practice courses. The nursing students divided into group for analysis and exploration the client data by utilizing the tablet PCs. The whole learning activities adopted constructing teaching method for teaching strategies. Through teamwork, the nursing students can plan exploration direction, collect the required information, establish decision-making action, and the last published works, which not only learn by doing but also learn from research process (Thomas, 2000). The students also use technology-assisted to facilitate information search skills and grasp the progress of work effectively, and further promote students to develop learning metacognition, critical thinking and problem-solving ability (Solomon, 2003). Moreover, public health nursing practice courses combine project-based learning. In addition to teaching nursing students professional knowledge and skilled expertise, it also focuses on allowing the nursing students to learn multi-direction thinking and to deal with the complexity of nursing practice process. Thus, students can learn to take responsibility for their own behaviors, and cultivate a humanized professional care attitude and behavior (Watson and Foster, 2003).

2 Mobile Annotation System for Health Education (ASHE)

In order to effective use of technology, match the goal of public health nursing practice courses, and fit the traditional teaching activity's content, this research has discussed with the nurse educator regarding the system design, user requirements, operating procedures, and accessibility features before the annotation system establish. The system is developed according to the discussion results on both sides. The overall system architecture is shown in Fig. 1.

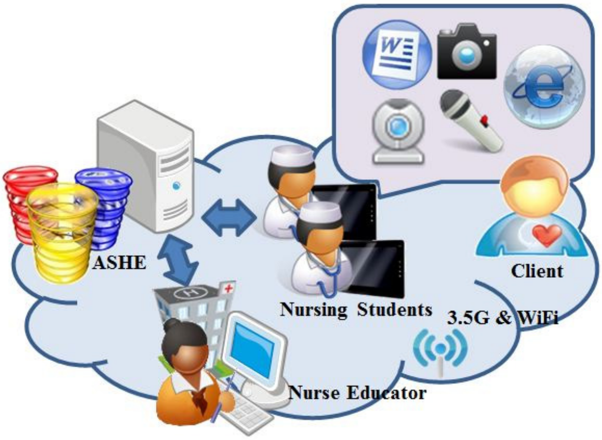


Fig. 1. System architecture of ASHE

Before the practice activities beginning, technical personnel installed the ASHE on the tablet PC. Through the tablet PC, the nursing students can operate system functions and proceed with practice activities. The function designed in ASHE not only can read normal text files and word file, but can also re-arrange and organize the page layout. For the annotation function, in addition to general text and graphics annotation, it can add audio and video files to be the annotation. Those resources read except through the tablet PC's storage and SD cards, but also can gather more multi-element and related information as annotated contents from internet. Moreover, the annotation system comes with audio and video recording functions, which allow nursing students to record the complete learning process during activities before the home visits or outside health education practice. The recorded data store in the backend database and is beneficial to follow-up analysis and discussion.

The design of the system database is divided into three parts. The first part is the nursing student's profile database which stores the profile information for each nursing students, the courses they are enlisted in, and the school achievement they got. Another part is portfolio database which stores process of operation system, status of practice activity, and records of documents and file during the process of public health nursing practice. Nurse educator can use the portfolio stored in the system as the basis for future teaching and guidance, as well as nursing students can use as the guidance for self-reflection and self-regulated learning. The last part is client database which include the patient's private information and related disease, and those data was allowed and provided by the department of regional health center.

In tradition practice activities, nursing students use paper-based method to record and collect information, which not only increase costs and time-consuming, but also easier to leak private information out. Therefore, the system records information by using digital way. Through digitized content can effectively strengthen information storage, transmission, management and use, thus it also increases information sharing, exchange, and interaction between peers or teachers and students. Before home visit for health education, nursing student should design and plan annotated contents, and those prepared information are stored in the portfolio database. When having home visit, the nursing students can use rich and variety contents showing on the tablet PC's to proceed with explanation and presentation of the health education. The annotated text, images, and audio/video that prepared before home visit becomes a great assistant tool for explanation during the health education process. Regarding the individual client having health education, using the tablet PC can present diverse and lively health education content can attract the attention of the client being taught, deepen the memories of teaching content, and improve the services of health care. In addition, nurse educator can use the learning portfolio data stored in the database to monitor and understand each nursing students' learning conditions and give guidance and feedback at the appropriate time. Moreover, further analysis of learning portfolio can provide nurse educator a reference for adjustment or design teaching strategy.

3 Research Design

3.1 Participants

The participants of this study were nursing students from the fourth year of 5-year nursing college, with a total of 36 students in the class. All of them attend the public health nursing practice courses and volunteer to participate in experiment activities. The experiment followed the practice echelon which was divided into two practice echelon of 36 nursing students by the college. The first batch of nursing students was set as the control group (18 people), and the second was the experimental group (18 people). The control group used the traditional PBL teaching, and nursing students used pen and paper to record and exchange information during the entire learning activity. On the other hand, the nursing students in the experimental group used the tablet PC as a learning device, and carried out PBL teaching activities through ASHE. The nurse educator for both groups is the same person. In addition, to ensure that each nursing student have good execution ability and learning process during the experiment, the grades are listed in this semester's performance for this subject.

3.2 Learning Procedure

The total time of practice training for public health care in the region health center is four weeks, seven hours a day. In the first two weeks, the nursing students stay in the health centers in order to understand the content and function of primary health care and organize and prepare health education materials for home visits. During the last two weeks, they start going out to intern home visits and to participate in community health nursing work. The research of this study is still ongoing. This paper focuses on experiment and exploration of data collection and preparation of health education in the first two weeks.

On the first day of practice activities, the nurse educator introduces main information to all the nursing students, which includes: the work role and function of the public health center nurses, activity program and implementation processes of project-based learning, operating instructions and use standardized of ASHE. In order to allow nursing students to familiarize with the system functions, this experiment reserved time for students to actually operate system, and to achieve the PBL teaching strategy, this experiment grouped students into six (3 persons per group). The second day began PBL teaching mode. Based on curriculum objectives and learning concepts, this study proposed 5 teaching steps to match the goal of public health nursing practice course: preparation, implementation, presentation, evaluation, and reflection. The experiment activity is shown in Fig. 2.

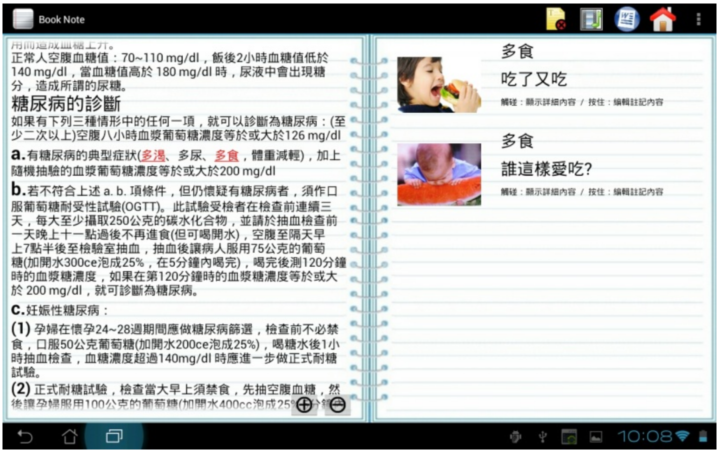
Fig. 2. Preparation of health educator materials

- Preparation

Through nurse education guidance, each group of the nursing students selected and confirmed the study's client by the process from diffused to aggregated thinking.

- Implementation

The experimental group of nursing students used their cognitive knowledge as basis, and prepared health education materials for home visit on the ASHE through the information technology and internet assisted. Each group of nursing students collected and organized materials according to the study's client they were selected, and utilize the system to annotate the parts that is important and need special interpretation for the client. The system interface is shown in Fig. 3. The left side of system interface is health education content part, and right-hand side is shown the annotation. The top of right side is other provided functions in the ASHE.



用而造成血糖上升。
正常人空腹血糖值：70~110 mg/dl，飯後2小時血糖值低於140 mg/dl，當血糖值高於 180 mg/dl 時，尿液中會出現糖分，造成所謂的尿糖。

糖尿病的診斷

如果有下列三種情形中的任何一項，就可以診斷為糖尿病：(至少二次以上)空腹八小時血漿葡萄糖濃度等於或大於126 mg/dl

a.有糖尿病的典型症狀(多渴、多尿、多食，體重減輕)，加上隨機抽驗的血漿葡萄糖濃度等於或大於200 mg/dl

b.若不符合上述 a. b. 項條件，但仍懷疑有糖尿病者，須作口服葡萄糖耐受性試驗(OGTT)。此試驗受檢者在檢查前連續三天，每大至少攝取250公克的碳水化合物，並請於抽血檢查前一天晚上十一點過後不再進食(但可喝開水)，空腹至隔天早上7點半後至檢驗室抽血，抽血後讓病人服用75公克的葡萄糖(加開水300ce泡成25%，在5分鐘內喝完)，喝完後測120分鐘時的血漿糖濃度，如果在第120分鐘時的血漿糖濃度等於或大於 200 mg/dl，就可診斷為糖尿病。

c.妊娠性糖尿病：

(1) 孕婦在懷孕24~28週期間應做糖尿病篩選，檢查前不必禁食，口服50公克葡萄糖(加開水200ce泡成25%)，喝糖水後1小時抽血檢查，血糖濃度超過140mg/dl 時應進一步做正式耐糖試驗。

(2) 正式耐糖試驗，檢查當大早上須禁食，先抽空腹血糖，然後讓孕婦服用100公克的葡萄糖(加開水400cc泡成25

多食
吃了又吃
觸碰：顯示詳細內容 / 按住：編輯註記內容

多食
誰這樣愛吃?
觸碰：顯示詳細內容 / 按住：編輯註記內容

Fig. 3. System interface of ASHE

- Presentation

When the implementation step was finished, the group members used information software to conduct a meaningful conversion of the collected data, annotated information, and the process of organized information, and used the diversified way to present the report contents. After each group finished their presentation, peers should give immediate feedback and suggestions regarding the contents of the report.

- Evaluation

The PBL evaluation goals are the assessment of learning outcomes and the learning process. Therefore, this study used self-evaluation (25%), peer evaluation (25%), and teacher evaluation (50%) as the assessment method for PBL learning outcome.

- Reflection

The nursing students reflected on the improvement in the presented project according to the feedback from the evaluation step, and self-review the communication, coordination, cooperation, judgment ability during the project-based learning activities. Thus, the nursing students can enhance the introspective potential and self-adjustment learning ability.

When the nursing students engaged in project-based learning activities, in the same time, they participated in primary health care activities in the regional health center and recognized the scope of public health nursing services, operation, and development from these activities.

4 Result and Discussion

4.1 Evaluation of Learning Effectiveness

Each nursing student in both the experimental group and the control group received a total score by proportionately distributed from three evaluation approaches after the end of the PBL activities. Thus, this study used the independent sample t-test to evaluate the learning performance, and used descriptive statistics for description. The results were shown in Table 1. The value of Levene's test was greater than .05 (Levene =.422 > .05), so homogeneity of variance was accepted, which means experimental group and control group had normal distribution and similar discrete state. Moreover, *p* value was smaller than .05 (*p* = .000 < .05), which shown that the learning performance of the experimental group and control group had significant difference. Further explored the average scores of the two groups, the average score of the experimental group was significantly higher than the control group. This result exhibited that although the both groups engaged in PBL activities, the experimental group that used ASHE had better learning performance than the control group. Furthermore, according to the statistics results can know that introducing information technology into traditional public health nursing practice course can improve the drawbacks of traditional learning and effectively promote the learning effectiveness of practice course.

Table 1. The t-test results of learning effectiveness

	Mean	N	Std. Deviation	Levene	t	df	p
Experimental	88.300	18	2.226	.422	-5.395	34	.000*
Control	84.375	18	2.093				

*p < .05.

4.2 Evaluation of Cognitive Load

In terms of reliability analysis of cognitive load scale, the Cronbach α of experimental group was .863, and the Cronbach α of control group was .802. The Cronbach α of the both groups were greater than .07, which means the reliability of the questionnaire reached a certain level.

Then, the independent sample t-test was used to analyze the cognitive load scale. Table 2 shows that *p* value is greater than .05 (p = .237 > .05), which indicate that cognitive load had no significant difference between the experimental group and control group. Therefore, the introduction of the ASHE did not increase the cognitive load of the nursing students. Further analysis of the average value, it was shown that average score of the experimental group was slightly higher than the control group, but the gap was small. This result expressed that introduction of the system did not significantly cause learning pressure on the nursing students. Although the score of the cognitive load was slightly higher than the control group, it is still in acceptable range for the nursing students in the experimental group.

Table 2. The t-test results of cognitive load

	Mean	N	Std. Deviation	t	df	p
Experimental	4.111	18	.758	-1.204	34	.237
Control	3.833	18	.618			

4.3 Analysis of Interview

The experimental group of nursing students expressed satisfaction with the ASHE, and they thought the practice course can make some conversions with the transformation of technologies and learning styles. Through the proposed learning system, students felt that data collection and preparation can be accelerate, and important and specific points can be marked immediately and search on the internet for a variety of explanatory documents. Those useful functions not only can assist them to prepare the materials conveniently, but also can support to do the explanation and display when they have the health education activity. In addition, students thought using digitized information is not only advantage of storage, exchange and management, video and audio recording of learning portfolio but also as review and reflection tool after the school.

The nursing students in the control group expressed that the feedback, evaluation and reflection are the feature of the PBL strategy. Through feedback and evaluation, they can understand their own deficiencies they have not discovered and learn to accept multiple perspectives. Moreover, through the reflection, the nursing students can use different viewpoints and orientations to discuss and consider the problem, and they believed the reflection can help them achieve self-regulated learning, develop critical thinking, and promote the acquisition of knowledge is a non-inert knowledge.

The introduction of the ASHE can reduce nurse educator's burden. The nurse educator expressed that they only needed to play the role of the mentor and gave guidance and assistance at the appropriate time. In addition, nurse educator emphasized they can immediately understand the learning situation and process of nursing students and effectively grasp the progress of the overall teaching and learning activities through the recorded learning portfolio. Assistant of information technology solve the problem of not easily getting immediate feedback and situation that exist in traditional practice course. Moreover, nurse educator said PBL is a structured learning strategy and can promote students' cooperation, communication, independent, and critical thinking skills.

5 Conclusion and Future Work

This study constructed the ASHE system to match the public health practice course. Through the system, nursing students can organize and prepare health education data for home visits, and achieve the learning objectives by using PBL strategy. According to the results of statistical analysis, it indicates that importing the ASHE system can enhance the effectiveness of nursing student learning. Through the system, nursing students can speedily collect and integrate health education information, and proceed with various annotations and presenting for explanation the important and special part to the cases. Moreover, based on the observation of nurse educator, the highly autonomous PBL teaching strategy can further enhance nursing students' learning motivation, cooperation method, problem-solving ability, resource management skills, and critical thinking. Because students have daily use and operating experience of the information system, the introduction of the ASHE system will not be a direct burden on nursing students' learning. Students' cognitive load will gradually decrease with the increase of the number operations and familiar the system.

The participator of this study were all females, future studies can use different gender and internship subject as exploration subject. In addition, because this experiment is still ongoing, follow-up analysis can be further analyze the gathered information from home visit for health education. Furthermore, more detailed and in-depth exploration can be done on nursing students' learning portfolio, individual characteristics, and different teaching strategies.

References

1. Carroll, A., Christakis, D.: Pediatricians' use of and attitudes about personal digital assistants. Pediatrics 113, 238–242 (2004)
2. Christy, T.E.: Clinical practice as a function of nursing education: An historical analysis. Nursing Outlook 28, 493–497 (1998)
3. Criswell, D.F., Parchman, M.L.: Handheld computer use in U.S. family practice residency programs. Journal of the American Medical Informatics Association 9(1), 80–86 (2002)
4. Dolan, G.: Assessing student nurse clinical competency: Will we were get it right? Journal of Clinical Nursing 12(1), 132–141 (2003)
5. Fowler, D.L., Hogle, N.J., Martini, F., Roh, M.S.: The use of personal digital assistant for wireless entry of data into a database via the Internet. Surgical Endoscopy 16, 221–223 (2002)
6. Jeng, Y.L., Wu, T.T., Huang, Y.M., Tan, Q., Yang, S.J.H.: The Add-on Impact of Mobile Applications in Learning Strategies: A Review Study. Educational Technology & Society 13(3), 3–11 (2010)
7. Krajcik, J.S., Czerniak, C.M., Berger, C.: Teaching children science: A Project-based approach. McGraw-Hill College, Boston (1991)
8. Kynaslahti, H.: In search of elements of mobility in the context of education. In: Kynaslahti, H., Seppala, P. (eds.) Mobile Learning, pp. 41–48 (2003)
9. Solomon, G.: Project-Based Learning: a Primer. Technology & Learning 23, 20–30 (2003)
10. Thomas, J.W.: A review of research on project-based learning. Autodesk, San Rafael (2000)
11. Watson, J., Foster, R.: The Attending Nurse Caring Model: Integrating theory, evidence and advanced caring-healing therapeutics for transforming professional practice. Journal of Clinical Nursing 12(3), 360–365 (2003)
12. White, A., Allen, P., Goodwin, L., Breckinridge, D., Dowell, J., Garvy, R.: Infusing PDA technology into nursing education. Nurse Education. Nurse Education 30, 150–154 (2005)

Conceptualizing Citizen's Digital Literacy through Everyday Internet Use

Mei-Mei Wu[1] and Ying-Hsang Liu[2]

[1] Graduate Institute of Library & Information Studies
National Taiwan Normal University
Taipei 10610, Taiwan, R.O.C.
meiwu@ntnu.edu.tw
[2] School of Information Studies
Charles Sturt University
Wagga Wagga NSW 2678, Australia
yingliu@csu.edu.au

Abstract. Digital opportunity which is the solution for digital divide offers three levels of accessibility to the citizen, namely, physical accessibility of computer hardware and network, the content accessibility and the intellectual accessibility in which the cognitive as well as operating skills are critical to digital literacies. The purpose of the study is to propose a framework for citizen's digital literacy to fill the gap of the current model of digital opportunity which emphasizes physical accessibility without specific action plan for intellectual accessibility. By analyzing the daily activities of Internet use and identifying the groups of these activities, we propose a conceptual framework of citizen's digital literacy. Secondary data from a national-wide survey, the fifth phase of the fourth year survey of the TSCS Research Project (Taiwan Social Change Survey (TSCS), 2009) is used. Descriptive statistics, chi-squared, logistic regression and multidimensional scaling analyses are conducted to reveal the citizen's use of information and communication technologies (ICTs) over time and of which the relationship with the citizen's demographic data. The results suggest an overview of primary online information activities engaged by citizens, including (1) gaining information of traveling, information technology, news and knowledge acquisition, (2) consuming and finance, and (3) social networking and communication. Although information activities for learning and work purposed in the survey have not been significant which may be due to the design of the survey questions, findings from multidimensional scaling suggest that there seems to be a hierarchy of information needs, similar to Maslow's (1943) basic needs, manifested in Internet use activities. Different from the traditional approach of information literacy focusing on information seeking sills, digital literacy goes beyond to the territory of doing, actual conducting of daily activities by using ICTs. A conceptual framework as well as a working model for citizen's digital literacy is suggested for future research.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 359–371.
DOI: 10.1007/978-3-642-35452-6_37

1 Introduction

Digital literacy and its application to the areas of digital divide and digital opportunities are one of the critical information policy issues in people's lives. The conceptualization of digital opportunities is concerned with the physical accessibility of computer hardware and network, the content accessibility and the intellectual accessibility. In terms of the intellectual accessibility, the cognitive and operational abilities and skills are particularly important to digital literacies.

According to the definition of digital literacy proposed by American Library Association (ALA) [2], a digitally literate person:

1. Possesses the variety of skills–technical and cognitive–required to find, understand, evaluate, create, and communicate digital information in a wide variety of formats;
2. Is able to use diverse technologies appropriately and effectively to retrieve information, interpret results, and judge the quality of that information;
3. Understands the relationship between technology, life-long learning, personal privacy, and stewardship of information;
4. Uses these skills and the appropriate technology to communicate and collaborate with peers, colleagues, family, and on occasion, the general public; and
5. Uses these skills to actively participate in civic society and contribute to a vibrant, informed, and engaged community.

Simply put, digital literacy is an extent of concerns of information literacy that people in the digital era will be functioning properly to do their daily activities, such as shopping, commuting, work and study by using ICTs. A historian and freelance writer, Gilster [10] wrote fifteen years ago, states that the important issues in digital literacy include literacy for the Internet, the nature of digital literacy, content evaluation, from hypertext to context, searching the virtual library, knowledge assembly and a future for the digital literate. Gilster stresses that people are no longer passive receivers of information in the networked digital environment. People need to actively discover, evaluate and disseminate information in this new environment. Likewise, Eshet-Alkalai [9] propose a framework for understanding digital literacy in the digital environment in which photo-visual literacy, reproduction literacy, branching literacy, information literacy and socio-emotional literacy are important skills to function effectively . That illustrates digital literacy as not only skills, technology related but also legal, ethical and social responsibility bounded.

Originated from the field of sociology, the notion of everyday life has been effectively applied to human information behavior (HIB) research by Savolainen's research program on people's everyday life information seeking (ELIS) [16]. This line of research overall has suggested a wide variety of information needs in specific contexts (see e.g., [1, 17]) partly because of the predominant use of qualitative research methods, such as critical incident techniques and the focus on extraordinary situations. In this sense, citizens' everyday ordinary activities may not be well represented in these studies.

Citizens' information needs were investigated in early user studies [5, 7], with particular reference to library and information services. More recently, research within this area has been concerned with the use of ICTs and how they affect ordinary people's information seeking behaviors (see e.g., [13, 14, 18, 19]). In Savolainen and Kari's study [18], they proposed the notion of information source horizon to characterize channels and sources of information. One of the major characteristics of citizens' information behavior is the use of a wide variety of sources of information affected by perceived accessibility in information-seeking contexts. The goals or purposes of ELIS in the context of Internet use were investigated in Hektor's characterization of how people use Internet [12]. Overall, these studies have suggested that ordinary people are engaged with Internet information seeking activities for the various purposes, such as information gathering or monitoring. The use of Internet for self-development is another important characteristics of ELIS (e.g., [12, 18]), but has not been studied in details. Furthermore, Internet use activities may relate to social and communication functions as well as consuming and finance functions, in additional to information source horizon. Those are everyday life related and consider necessity skills for surviving in the digital society.

The need and use for information is ubiquitous in our everyday life. The Pew Internet & American Life Project [3] offers 23 topics including banking, blogs, communities, dating, decision making, education, gaming, government, health, identity, libraries, music, news, podcasting, politics, religion, safety, science, search, shopping, social networking, video, and work, and none of them is information-free. Yet those citizen's daily activities of information behavior deserves a framework that would be beneficial to the design of a digital literacy program. The purpose of the study does not aim at designing a digital literacy program but to identify a conceptual framework for citizen's digital literacy.

The specific research questions addressed in this paper are as follows:

1. What are the daily activities of Internet use of Taiwanese people?
2. How age, gender, income and education influence Internet use of Taiwanese people?
3. How to group the daily activities of Internet use in order to propose a framework for citizen's digital literacy?

2 Methods

2.1 Overview

The study takes a quantitative analysis of a secondary data set approach. To explore citizens' information activities over time, we analyzed the data sets from large-scale longitudinal survey project, entitled "Taiwan Social Change Survey (TSCS)" [4]. Descriptive statistics was used to reveal the overall trend of citizens' use of ICTs, such as mobile phone devices, personal computers and Internet connections. Cross-tabulation, chi-squared and logistic regression were used to determine the relationship between the demographic variables and the frequency of Internet use. As such, we were able to characterize the citizens' four major domains of information use (everyday life, learning, leisure and work) from the analysis of longitudinal survey data.

2.2 Survey Data Sets

In this study we performed secondary data analyses on the large-scale survey data from the 2008 TSCS in order to characterize citizens' information-related activities regarding the use of ICTs in everyday information use. Data analyzed in this paper were collected in the fifth phase's fourth year survey of the TSCS research project [4]. The project was conducted by the Institute of Sociology, Academia Sinica and sponsored by the National Science Council, Taiwan. Specifically, drawing largely from the 2008 TSCS mass communication module that is repeated every five years and previous survey data collected in 1998 and 2003, we were able to understand social changes in the area of media use.

The survey consisted of 325 items that cover Internet use and other important issues in mass communication, and the influence and functions of mass media were added to the module in 2008. Drawing from the citizens aged over 18, the survey used the PPS (probability proportional to size) sampling method to be representative of the age, gender and geographic location, resulting in a sample of 4,604. A total of 1,980 completed questionnaires were obtained by onsite interviews, with a completion rate of 43%Ḟor the globalization and culture module, there were a total of 2,067 completed questionnaires, with a 45% completion rate. The average interview time was 50 minutes, with a range of 36 minutes to two hours. For the purposes of this study, we focused the analysis on the question items regarding communication behavior of mobile phone use and Internet use from the communication module, as well as the global media of TV and computer use from the globalization and culture module.

2.3 Data Analysis

We used descriptive statistics to reveal the citizens' use of ICTs over time. Cross-tabulation, chi-squared and logistic regression were used to determine the relationship between the demographic variables and Internet use. To explain whether one has used Internet or not, logistic regression with demographic variables of age, education and personal income was used. Multidimensional scaling (MDS) analysis of Internet activities was conducted to reveal the distance among these online activities. Our conceptualization of citizens' everyday life information use behavior was then derived from the examination of MDS plot, as well as our understanding of human information behavior research literature. We used open source R statistical computing software package for data analysis [8], because it provides flexibility for customizing approaches of data analysis and presentation of graphics.

3 Results

This section reports our findings of citizens' information activities, including the use of ICTs, Internet use and the relationship the demographic variables and the Internet use. Particular reference to the use of Internet over time is made, where appropriate.

3.1 ICT Use

The primary purposes of using mobile phones are interpersonal communication and recreation. The results from 2003 and 2008 survey data showed that mobile phones are largely used for the purposes of making phone calls (97.3% and 82.4% respectively) and using SMS (Short Message Service), with 55.2% and 42.9% of responses in 2003 and 2008. This suggested the primary purpose of interpersonal communication in mobile phone use. However, the 2008 survey revealed the increasing use of mobile phone for recreational purposes, such as mobile phone ring tone (32.4%), photo sharing (24.4%), listening to music (14.4%) and gaming (11.0%).

3.2 Internet Use

We summarize the main results from Internet use as follows:

1. The frequency of Internet use every day is increasing. The results demonstrated that the frequency of Internet use every day has increased from 5.9% in 1998 and 24.3% in 2003 to 41.6% in 2008. In the 2003 sample the proportion of people who spent less than three, four or six hours online is 72.9%, 81.8% and 91.9% respectively. A total of 84.4% people spent less than four hours every day on the Internet, excluding the purposes of work and learning.

2. Primary online information activities include traveling, information technology, consumer finance, social news and knowledge acquisition. The results from 1998 and 2003 showed that traveling, information technology and consumer finance are ranked top 3 online information activities. The sub- divided categories of information activities that included social news, consumer finance, traveling information and knowledge acquisition were popular online activities in the 2008 survey. Overall, traveling, information technology, consumer finance, social news and knowledge acquisition were primary online information activities.

3. Internet facilitates people's knowledge expansion. The 2003 survey results indicated that more than half people perceive that the use of Internet has influenced their knowledge expansion (87.66%), self- efficacy (54.6%), Internet addiction (56.8%), preferences of anonymities (52.5%) and expanding interpersonal relationships (60.9%).

4. Seeking and acquiring knowledge as well as recreation and time killing are primary motivations of Internet use. In addition to learning and looking for

information, people also used the Internet for recreational activities and keeping up-to-date. The results from 2003 survey revealed the top 5 motivations of Internet use include looking for useful information, learning new things, recreation, time killing and keeping up-to- date.

5. The adoption of email facilitates interpersonal communication. The results from 2003 survey indicated that more than three-fourth people had email accounts (77.7%) and used email for interpersonal communication (75.6%). This suggested that email has become the primary means of interpersonal communication.

6. E-commerce and the use of digital media increase over time. A comparison of online information activities in 2003 and 2008 showed that people were more engaged with browsing and collecting information and interpersonal communication (using email). It was also found that e-commerce transactions increased between 2003 and 2008, particularly in stock market transactions (from 9.3% to 19.3%) and online shopping (from 16.6% to 18.8%). Importantly, new online activities arose from the application of digital media, such as blogging (26.8%), online video watching (20.5%) and online TV viewing (11.8%).

The data revealed the popularity of online shopping, as indicated in the frequency of looking for product or service related information. A total of 37.9% did not look for product information in the past three months in 2003, while 16.6% people did not do so in the past one month in 2008. People still used product or service related information from TV shopping channels, suggesting that people collect product information from a variety of sources.

7. The polarity of time spent on TV watching. The results from 2008 data indicated that almost all people (98%) spent time watching TV at least once a month, and 87.6% watched TV every day. This suggested that people frequently used TV for information related activities, such as recreation, news watching and knowledge acquisition. The results from globalization and culture module showed that there was an increase in the people who seldom watch TV (never or less than 6 minutes), while there was also an increase in the people who watch TV more than four hours a day. Most people watched TV between one and two hours a day.

8. The time spent on Internet using personal computer increases. The 2008 results showed that 64.2% of people used personal computer in which 42.2% connected to the Internet. The survey data suggested that the time spent on Internet increases over time and the number of hours concentrate on one to three hours.

9. Information use for learning or work purposes in everyday life has not been widespread among citizens. The results from a comparison of TV watching and Internet use suggested that people spent long hours and the time spent concentrated on two hours. However, people primarily were engaged with such information activities as interpersonal communication, e-commerce and recreation. The information activities for learning or work purposes have not been widespread among citizens in their everyday life.

Table 1. Demographic variables and Internet use by citizens in 2008

	Internet Use	
Variables	Yes	No
Age (%)		
18-29	38.9	1.9
30-39	25.5	5.4
40-49	20.1	20
50-59	12.6	26.9
60+	3	45.9
$\chi^2 = 880.4 * **^a$	1075	905
Education (%)		
< Primary School	0	12.6
Junior High School	6	59.2
Vocational School	49.7	26.6
Technical Institute	8.3	0.3
College +	36.1	1.2
$\chi^2 = 1025.6 * **$	1075	905
Personal Income[b] (%)		
< NT$ 10K	20	44.9
< NT$ 20K	11.6	18.2
< NT$ 30K	19.3	13.5
< NT$ 40K	26	13.9
> NT$ 50K	21.7	7.2
$\chi^2 = 227.3 * **$	1060	884

[a] Significance code: *** p < .001.

[b] Personal income is the amount by month. NT$ = New Taiwan Dollars. A total of 36 cases belong to the 'not applicable' category for personal income.

Table 2. Logistic regression analysis of Internet use by demographic variables

Predictor	Coef	S.E.	Wald	*df*	*p* value
Intercept	0.20	0.19	1.07	1	.28
Age	2.55	0.17	15.45	1	* * **
Education	-2.89	0.16	-17.81	1	* * **
Income	-1.19	0.19	-6.32	1	* * **
Test				df	p value
Overall model evaluation					
Likelihood ratio test			1337.25	4	* * **
Wald test			621.54	4	* * **

Note. Significance code: **** $p < .0001$.

3.3 Demographic Variables and Internet Use

The results from cross-tabulation and chi-squired analysis between demographic variables and Internet use indicated extremely significant relationships in terms of the age ($\chi^2 = 880.5, df = 3, p < .001$), education ($\chi^2 = 1025.6, df = 4, p < .001$) and personal income ($\chi^2 = 227.3, df = 5, p < .001$). A summary of results is presented in Table 1.

Not surprisingly, a large portion of the young people (38.9%) aged below 29 have used Internet, while only 3.0% of the senior citizens aged above 60 have done so. People with low levels of education have had little use of Internet (59.2% for junior high school graduates), while only 1.2% college graduates did not use Internet. A total of 44.9% of low-income earners (less than 10K) had not used Internet before, but they also had a relatively high level of Internet use (20.0%), compared with people earning less than 20K (11.6%). We speculated that this discrepancy is in part due to the young Internet user groups who have not entered into the job market.

To explain whether one has used Internet or not, a binary logistic regression was used [11]. The predictors of age, education and personal income were transformed into binary variables, with cut-off age of 40, high school education and personal income of 40K. The results demonstrated that age, education and personal income are all extremely good predictors of whether people use Internet or not (see Table 2). That is, people who are less than 40 years old, have more than high school education, and personal income of 40K are more likely to use Internet for information-seeking activities. This model of Internet use can account for about two-thirds of variations ($R^2 = 0.66$).

3.4 Dimensions of Internet Information-Seeking Activities

The multidimensional scaling (MDS) was intended to reveal the underlying dimensions of Internet activities from the perspectives of ELIS and basic human needs. MDS analysis with rectangular matrices for all Internet information-seeking activities from the survey data revealed the conceptual distances among the Internet activities [6]. Specifically, several groupings of Internet activities were derived from MDS scaling joint configuration plot (Figure 1), with distances within 0.1.

To be precise, the groupings of Internet activities included: (a) Basic daily life activities in group (3, 5, 9): Weather forecast, Transportation and Food; (b) Keeping up-to-date information in group (10, 26, 27): TV, Information technology and specific topic knowledge; (c) Consumption in group (6, 7): Shopping and Travel; (d) International affairs informational in group (2, 18): International news, Sports; (e) Health informational in group (4, 19): Op-ed, and Health (f) Recreation in group (12, 15): Music and Movies; (g) Others (13, 14, 16, 17, 20, 21, 22, 23, 24, 25, 28, 29, 30, 31, 32, 33, 34): soap opera, gaming, household, cartoon, art and culture, literature, religion, divine, classic music, classic performance, scholar and thinking, teaching and learning, government affairs,

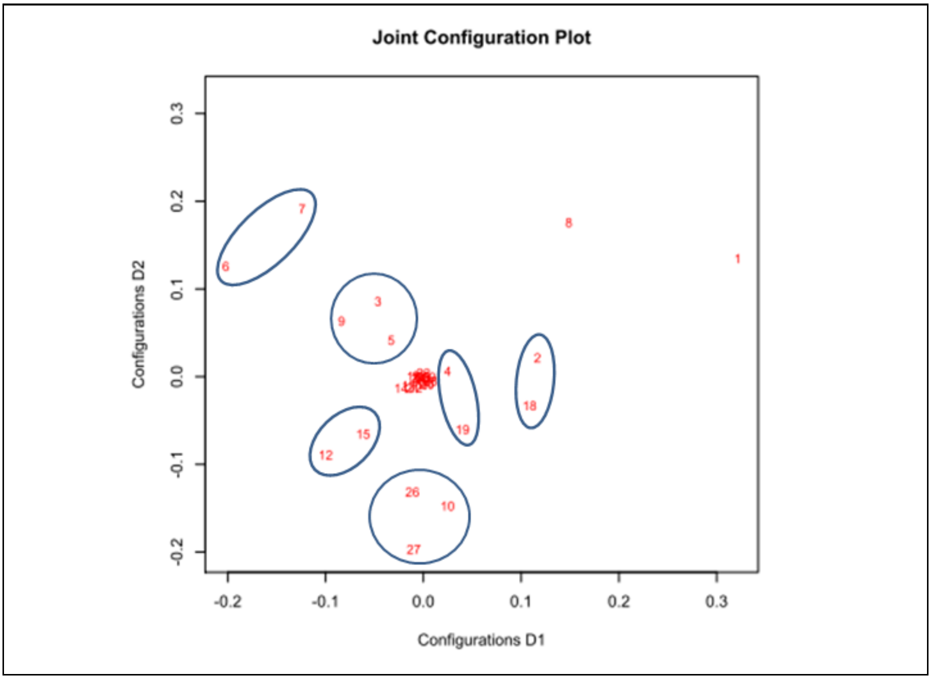


Fig. 1. Multidimensional scaling plot of 32 Internet activities from [4], N = 1076. Note: Internet activities shown in the plot are 1. Social news; 2. International news; 3. Weather forecast; 4. Op-ed; 5. Transportation; 6. Shopping; 7. Travel; 8. Consumer finance; 9. Food; 10. TV; 12. Music; 15. Movies; 18. Sports; 19. Health; 26. Information technology; 27. knowledge e.g. National Geography.

law services, marketing, recreational news. And consumer finance (8) and social news (1) were the out-lying categories in the plot.

These findings suggested that Marslow's a theory of human motivation of which composed by a hierarchy of five basic needs, namely, physiological, safety, love, esteem, and self-actualization seems to fit with the groupings of the characterized Internet activities: mostly physiological and safety needs (Food, Weather forecast, Transportation, Sports and Health), love and belonging and esteem (Music, Movies, Shopping, Travel, International news and Op-ed etc.) to self-actualization (all possible information activities related to push for one's potential). It should be noted that the information-seeking activities characterized here might be affected by the use of ICT and sociocultural factors, as represented in these activities. As Maslow [15] stated that "while behavior is almost always motivated, it is also almost always biologically, culturally and situationally determined as well." Our findings suggested that the underlying structure of information needs reflects the hierarchy of basic needs, as manifested in Internet information-seeking activities.

4 Discussion and Conclusions

Digital literacy is a paramount concern of the ordinary people as well as the government in successfully running everyday life and fulfilling social responsibility. This study attempts to depict a concept of digital literacy by scrutinizing the existing survey data. Three research questions are represented accordingly as follows.

What are the daily activities of Internet use of Taiwanese people?
The survey data shows that seeking and acquiring knowledge as well as recreation and time killing are primary motivations of Internet use, while traveling, information technology, consumer finance, social news and knowledge acquisition are primary daily activities of Internet use of Taiwanese people. The results from 1998 and 2003 showed that traveling, information technology and consumer finance are ranked top 3 online information activities. The sub-categories of information activities that included social news, consumer finance, traveling information and knowledge acquisition were popular online activities in the 2008 survey. In addition to learning and looking for information, people also used the Internet for recreational activities and keeping up-to-date. 2003 survey reveals that more than half people perceive that the use of Internet has influenced their knowledge expansion (87.66%), interpersonal relationships expanding (60.9%), and self-e?cacy (54.6%). Internet use shows a positive influence on Taiwanese people. Yet, Internet addiction (56.8%)and preferences of anonymities (52.5%) deserve attention.

E-commerce and the use of digital media increase over time may evoke and require the new version of survey questions. Between 2003 and 2008, particularly in stock market transactions and online shopping behavior increase. New online activities also arose from the application of digital media which indicates that Taiwanese people incline to adopt innovation.

How age, gender, income and education influence Internet use of Taiwanese people?
Age income and education are major factors that explain Internet use. Young people aged between 18-29 are the largest population that used Internet, while only 3.0% of the senior citizens aged above 60 have done so. People with less education have had little use of Internet. Also the largest portion of Internet users are those with a income higher than NT$40k (26%). It is also found that lowest income earners (less than NT$10K) also had a relatively high level of Internet use (20.0%). Compared with people earning less than 20K (11.6%), this discrepancy may in part due to the young Internet user groups who have not entered into the job market.

The Pew research [3] conducted in the states analyzes a demographics of Internet use,85% of American men and women are internet users. Almost all of people (99%) with high household income ($75,000+)use internet. And 75% of those whose income is less than$30,000/yr also are internet users. Education is also a significant indicator of internet users. Only 61% of those American adults without high school diploma use internet while 97% American adults with college and above use internet. Digital divide is an international phenomenon from the two studies running east and west. Findings from demographics and Internet use may suggest policy implications for digital opportunity program that GIS can be useful tool to locate underdeserved people.

How to group the daily activities of Internet use in order to propose a framework for citizen's digital literacy?
Internet use reflects citizens' everyday life activities. The plot analysis reveals a map of seven groups of internet everyday activities, including basic daily life activities, such as Weather forecast, Transportation and Food; keeping up-to-date Information, such as TV, Information technology and specific topic knowledge; consumption behavior, such as Shopping and Travel; international affairs information, such as International news, Sports; health related informational, such as Op-ed, and Health; recreation, such as Music and Movies; and others, such as soap opera, gaming, household, cartoon, art and culture, literature, religion, divine, classic music, classic performance, scholar and thinking, teaching and learning, government affairs, law services, marketing, recreational news; and consumer finance and social news were the out-lying categories in the plot.

These findings suggested that Marslow's a theory of human motivation of which composed of a hierarchy of five basic needs to fit with the groupings of the characterized Internet activities, mostly physiological and safety needs (Food, Weather forecast, Transportation, Sports and Health), love and belonging and esteem (Music, Movies, Shopping, Travel, International news and Op-ed etc.) to self-actualization (all possible information activities related to push for one's potential).

The Pew research also shows a result of what American Internet users do online, the activities applied by over 50% of the American people include use a search engine to find information, send or read e-mail, look for information on a hobby or interest, search for a map or driving directions, check the weather, look for health/medical information, look for information online about a service or product one is thinking to buy, get news, go online just for fun or to pass the time, buy a product, watch a video on a video-sharing site like YouTube or Vimeo, search for information about someone you know or might meet, look for how to , do it yourself or repair information, visit a local, state or federal government website, use a social networking site like Facebook, Linkedin or Google Plus, buy or make a reservation for travel, do any banking online, look online foe news or information about politics, look online for information about a job, look for information on Wikipedia, use online classified ads or sites like Craigslist, get news or information about sports, take a virtual tour of a location online, do any type of research for your job, etc.

The results of both surveys suggest that internet activities include information seeking as well as information doing. Buying things online, investing and banking online, making reservation, managing schedule online, that is, besides tradition information seeking and receiving online, there are all kinds of online doings. It implies an important territory of digital literacy concept to include an online daily doings in the map of digital literacy.

5 Future Research

The prevalent of digital environment escalates everyday activity online. Not only shopping, banking, social networking is online, but also applying for a job, learning and working. Not only information as sources but also information as doings. People need new set of skills to be survived in the digital era. This research applying two sets of secondary data and suggest a conceptual framework based on Maslow's basic needs and found most everyday online activities are physiological and safety related. Yet, there are more work to be done. In order to minimize digital divide, digital literacy education program emphasizing intellectual accessibility, the proper use of internet for everyday life, study and work are vital. Two related issues are suggested for future study:

Firstly, one interesting result of the study has been the plot chat of which demonstrated that consumer finance and social news were the out-lying categories in the plot. Why is it the case and how this pattern related to the demographic data is one research issue deserves further investigation.

Secondly, based on the foundation of citizen's daily life internet use behavior and the theory of Maslow's human need, this study has suggested a concept of digital literacy to include information doings as well as information seeking activities. It suggests that there are fundamental everyday activities that require the capability of using information and digital tools properly. A conceptual framework for digital literacy as well as a working model for digital literacy education are suggested for future research.

Acknowledgments. Data analyzed in this paper were collected by the research project "Taiwan Social Change Survey (TSCS)" sponsored by the National Science Council, Taiwan. This research project was carried out by the Institute of Sociology and directed by Dr. Yi-Yun Chang (Principal Investigator). The Center for Survey Research of Academia Sinica is responsible for the data distribution. The authors appreciate the assistance in providing data by the institutes and individuals aforementioned. The views expressed herein are the authors' own. This study was sponsored by National Science Council of Taiwan (NSC 99-2511-S-003-028).

References

[1] Agosto, D.E., Hughes-Hassell, S.: Toward a model of the everyday life information needs of urban teenagers, part 1: Theoretical model. Journal of the American Society for Information Science and Technology 57(10), 1394–1403 (2006)

[2] American Library Association. Digital literacy definition (2011), `http://connect.ala.org/node/181197`

[3] Pew Research Center. Pew Research Center's Internet & American Life Project (2012), `http://www.pewinternet.org/Trend-Data-(Adults).aspx`

[4] Chang, Y.-Y.: Taiwan Social Change Survey, TSCS (2009), `http://www.ios.sinica.edu.tw/sc/`

[5] Chen, C., Hernon, P.: Information seeking: Assessing and anticipatinguser needs. Neal-Schuman Publishers, NY (1982)

[6] de Leeuw, J., Mair, P.: Multidimensional scaling using majorization: Smacof in r. Journal of Statistical Software 31(3), 1–30 (2009)

[7] Dervin, B.: The everyday information needs of the average citizen: A taxonomy for analysis. Information for the Community, 19–38 (1976)

[8] R Development Core Team. R: A language and environment for statistical computing (2010)

[9] Eshet-Alkalai, Y.: Digital literacy: A conceptual framework for survival skills in the digital era. Journal of Educational Multimedia and Hypermedia 13(1) (2004)

[10] Gilster, P.: Digital literacy. Wiley Computer (1997)

[11] Harrell, F.E.: R-package rms: Regression Modeling Strategies (R package version 3.5-0) (2012), `http://cran.r-project.org/web/packages/rms/index.html`

[12] Hektor, A.: Information activities on the internet in everyday life. The New Review of Information Behaviour Research 4(1), 127–138 (2003)

[13] Marcella, R., Baxter, G.: The information needs and the information seeking behaviour of a national sample of the population in the united kingdom, with special reference to needs related to citizenship. Journal of Documentation 55(2), 159–183 (1999)

[14] Marcella, R., Baxter, G.: The impact of social class and status on citizenship information need: The results of two national surveys in the uk. Journal of Information Science 26(4), 239–254 (2000)

[15] Maslow, A.: A theory of human motivation. Psychological Review 50(4), 370–396 (1943)

[16] Savolainen, R.: Everyday life information seeking: Approaching information seeking in the context of "way of life". Library & Information Science Research 17(3), 259–294 (1995)

[17] Savolainen, R.: The role of the internet in information seeking: Putting the networked services in context. Information Processing & Management 35(6), 765–782 (1999)

[18] Savolainen, R., Kari, J.: Placing the internet in information source horizons. A study of information seeking by internet users in the context of self-development. Library and Information Science Research 26(4), 415–433 (2004)

[19] Savolainen, R., Kari, J.: Conceptions of the internet in everyday life information seeking. Journal of Information Science 30(3), 219–226 (2004)

Can Internet Usage Positively or Negatively Affect Interpersonal Relationship?

Chih-Hung Lai[1,*], Chunn-Ying Lin[2], Cheng-Hung Chen[3],
Hwei-Ling Gwung[1], and Chia-Hao Li[4]

[1] Department of Computer Science and Information Engineering,
National Dong Hwa University, Taiwan
[2] Department of Early Childhood Education, National Dong Hwa University, Taiwan
[3] Department of Educational Administration and Mangement,
National Dong Hwa University, Taiwan
[4] Department of Curriculum Design and Human Potentials Development,
National Dong Hwa University, Taiwan
1, Sec. 2, Da Hsueh Rd., Shou-Feng, Hualien, Taiwan, 974, R.O.C.
laich@mail.ndhu.edu.tw

Abstract. Many past studies showed that Internet addiction negatively affected the interpersonal relationship. However, new functions on the Internet provide more online interactions, especially some social websites such as Facebook which enables individuals to establish new relationships with acquaintances, as well as maintain close relationships with friends. This drives us to the questions whether the Internet is detrimental to one's interpersonal relationship or whether, instead, it might enhance one's interpersonal relationship. This study examined the association among Internet addiction, various Internet usage, and interpersonal relationships. In total, 444 valid copies of questionnaires were collected from a university. The results indicated that the Internet functions on social interaction, video watching, and information seeking can enhance interpersonal relationship while porn-website surfing and game playing cannot directly affect interpersonal relationship. On the other hand, the social interaction, porn-website surfing, and video watching led to poor interpersonal relationship mediated by Internet addiction.

Keywords: Internet addiction, Interpersonal relationship, Internet usage.

1 Introduction

With quick development of computer technologies, Internet has become an indispensable tool for modern people, especially for college students. Despite the numerous benefits associated with Internet use [1], Internet addiction has become a vital issue. The previous research point out that college students are more vulnerable to developing Internet addiction than other people [2], due to a mixture of reduced level of parental monitoring, free access to the Internet, and the great availability of free time [3].

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 373–382.
DOI: 10.1007/978-3-642-35452-6_38

Internet addiction leads to negative impacts on the academic, relationship, financial, and occupational aspects of many lives [4][5][6]. Much studies have specially pointed out that Internet addicted users decreased social interaction with real people [7], which made interpersonal relationship poor [3][8][9]. Besides, communication on the Internet is easier and more comfortable than traditional forms of personal communication, which giving Internet addicted users feel controllable when they interact online [4]. This situation, which Internet users spend more time online instead of face-to-face interaction with others, made their interpersonal relationship worse and worse. However, the Internet provides many communication tools, such as MSN, Skype, email, discussion board. With new functions of Internet hit in these years, especially Web 2.0, the Internet provides more online social networking. College students are increasingly inclined to cultivate their virtual social relationships and virtual life on existing prevalent social networking websites such as Facebook, Plurk, and MySpace [10]. The Internet supports an alternative tool to develop interpersonal relationships [7][11]. Making friends seems to be one of the main activities in which individuals engage when they use the Internet [12].

In short, it is a debatable point whether Internet usage can hinder or enhance the interpersonal relationship. To explain the contradiction from the existing research, this study, therefore, aimed to examine the effect of Internet addiction on the interpersonal relationship through the perspective of various Internet usages.

2 Literature Review

Internet addiction has widely drawn scholars' attention science Young [13] firstly proposed the notion [3]. Internet addiction was defined as "an individual's inability to control their Internet use, which in turn leads to feelings of distress and functional impairment of daily activities" [14]. There were various terms associated with the concept of Internet addiction, including "Problematic Internet Use," "Pathological Internet Use," "Internet Addiction Disorder (IAD)," "Excessive Internet Use," and "Compulsive Internet use" [15]. Although Internet addiction was studied for a couple of decades, there is no standardized definition of Internet addiction [4]. Based on the previous research, Internet addiction consists of five components: (1) withdrawal, including feelings of anger, tension, and/or depression when the computer is inaccessible [16][17], (2) tolerance that necessitates the need for better equipment, more software [18], (3) compulsive use, often associated with a loss of sense of time or a neglect of basic drives [19], (4) time management problem that lacks managing hours of use Internet [18], and (5) health management problem [7][20].

Despite the numerous benefits associated with Internet use [1], Internet addiction leads to many problems, such as declines in the size of social circle, depression [9][21], loneliness [22][23], and low interpersonal relationship [17][21] and negative effects on poor academic achievement, negative effects on social and psychological welfare [24]. Among these issues, the relationship between interpersonal relationships and Internet addiction has received a fair amount of attention in scholarly literature. However, the previous research findings were not consistent. More studies showed that Internet overuses have negative effect on interpersonal relationship [23][25][26]. That is, Internet addicted users have a lower mean quality in interpersonal relationships [3]. The scholars argued that Internet overuses make the time interacting with real people be replaced by the Internet usage. It decreases the size of one's social circle [27] and increases in depression [9][21] and loneliness [22][23][27]. Besides, the Internet became a refuge from interpersonal interactions. Individuals who feel extreme anxiety over establishing real interpersonal relationships, may use the Internet as a substitute for real world social contacts [26]. Internet surfing has been suggested to provide a compensation for loneliness and rejecting and conflicting relationships" [28].

On the other hand, other research also indicated that the Internet provides social and psychological support and enhances interpersonal relationship [29]. Shaw & Gant [30] showed that increased Internet usage was associated with decreased levels of loneliness and depression and increased levels of social support and self-esteem. Morahan-Martin and Schumacher [31] indicated that the Internet does not only provide a vastly expanded social network, but also provides altered social interaction patterns online. These researchers contend that the Internet support and maintain friendships rather than as an alternative to offline friendships [32]. The Internet also provides online social interactions to allow users to perceive themselves as more secure and more at ease than in traditional face-to-face interactions [3]. Therefore, Internet overuse decreases actual time spent with real people. At the same time, it increases the time spent with the cyber friends that may make the relationships with online friends grow stronger [7].

As shown in the above literature review, existing research in the association between the Internet addiction and interpersonal relationship were contradictious in whether the Internet is detrimental to one's interpersonal relationship or whether, instead, it might enhance one's interpersonal relationship. These variations in results could be attributed to the changing nature of the Internet, the greater numbers of individuals who use it, as well as its increased accessibility, affordability and availability [32]. With the quick evolvement of usage functions, the Internet can provides more social interactions through using synchronous computer mediated communication functions such as MSN and Skype, and asynchronous communication, such as email, message boards [29] and social website (e.g., Facebook, Plurk), which may influence the relationship of Internet usage and interpersonal relationship. As a result, perhaps such contradictious research results might be explained by the different Internet usage which made the Internet addiction. It is therefore the intent of the present study to examine the effect on the interpersonal relationship from various Internet usages.

3 Method

3.1 Data Collection

Paper-based questionnaires were delivered to 490 students who enrolled in various undergraduate programs in a university of eastern Taiwan. Deducting the returns with missing demographic data and incomplete responses, a total of 444 valid copies were collected. Of the participants, 167 (37.6%) were freshmen, 68 (15.3%) were sophomore, 144 (32.4%) were junior, and 65 (14.6%) were senior. One hundred and ninety-nine participants (44.8%) were males and 245 (55.2%) were females.

3.2 Measures

For the collection of the data, the adapted Internet Addiction Scale (IAS) was used to measure the Internet addiction extents of the individuals; and the Information Usage Questionnaire (IUQ) was used to obtain personal information and the extents about the Internet usages.

The IAS was adapted based on previous Internet addiction scale (e.g., [33][34]), which comprises five subscales: tolerance, compulsive use, withdrawal, health management, and time management. It consisted of 16 items using 5-point Likert scale with 5 indicating “strongly agree” and 1 “strongly disagree.” (e.g., I feel like missing something without using the Internet for a while; Every time I log off in order to do something else, I cannot help but getting online again.) CFA was then conducted to assess the validity and reliability.

The interpersonal relationship scale (IRS) was adapted from existing research [3][11]. There were eighteen items in the IRS in terms of social acceptance, peer relationship, teacher-student relationship, and parent-child relationship. These items were measured with five-point Likert scales ranging from ‘strongly disagree’ to ‘strongly agree’. (e.g., I talk to me family about my feeling and mood; I have intimate friends to share my thoughts with.)

The IUQ was developed for the present study referring to previous research (e.g., [35][36]). The scale includes 5 items with a 5-point Likert scale. The five items were to understand the extent which students used Internet in terms of five functions: social interaction (I chat or make new friends online frequently), porn-website surfing (I visit porn websites frequently), game playing (I play online games frequently), video watching (I watch video clips and/or listen to music online frequently) and information searching (I search information through online search engines frequently).

3.3 Reliability and Validity Analysis

To address the research purposes, this study conducted two-step structural equation modeling (SEM) process [37]. In the first step, confirmatory factor analyses explored the measurement model by examining the fit of the indicators to their latent constructs. In the second step, the measurement model was used to determine the relationships among the latent constructs in the structural model. Besides, Cronbach’s alpha was used to evaluate the internal reliability of the scales. The AMOS 18.0 and SPSS 18.0 were used to perform the analysis.

For the scales, the results of reliability and validity analysis were summarized in Table 1. All Cronbach’s alphas are above the 0.7 level, indicating good reliability [38]. The composite reliability (CR) for each construct is above 0.7, which means adequate internal consistency of the measurement model [39]. The average variance extracted (AVE) for every construct is above 0.5, which means the scales have good convergent validity [40]. The Cronbach's alpha coefficients of IAS and IRS were 0.922 and 0.931respectively, indicating good internal consistency.

The manifest variables of the constructs of the measurement model were examined for the goodness of fit by CFA. Overall measurement of model fit was assessed using five indices: chi-square (χ^2) statistics; chi-square–degrees of freedom ratio (χ^2/df); the comparative fit index (CFI); the root-mean square error of approximation (RMSEA); and the standardized root-mean-square residual (SRMR). The value of χ^2 is contingent on the sample size, and an insignificant χ^2 or χ^2/df value of less than 5 indicates a good fit [41]. A CFI value greater than 0.9, an RMSEA value less than 0.08 and an SRMR less than 0.05 may be taken as indicators of a good model fit [42]. The goodness of fit indices of the CFA for Internet addiction scale (χ^2/df = 3.0766, CFI = .954, GFI = .925, AGFI = .892, RMSEA = .068, and SRMR = .0493) and interpersonal relationship scale (χ^2/df = 4.864, CFI = .912, GFI = .860, AGFI = .815, RMSEA = .093, and SRMR = .0541) indicated that the model was moderately acceptable. Although χ^2/df is above the threshold of 3, it would be affected by the large sample which usually leads to enlarge χ^2 [43].

3.4 Structural Model Test

AMOS was used to test the structural model. To evaluate the proposed model, the score for each subscale among IAS and PRIS was the mean of items for each subscale as observed indictors. We tested our research model and summarized the results with AMOS coefficients in Fig. 2. All fit indices (χ^2/df = 2.582, GFI = .940, AGFI = .897, CFI = .922, RMSEA = .070, and SRMR = .0538) are within acceptable ranges indicated that the model fitted well. The proportions of variances explained are 24% for Internet addiction and 12% for interpersonal relationship.

4 Results

From the model test in Fig. 1, interpersonal relationship was positively affected by Internet usage in terms of social interaction, video watching, and information seeking (β= .16, .18, and .19, $p < .05$, respectively). However, the effects of porn-website surfing and game playing on interpersonal relationship was not significant (β = -.03 and .04, $p > .05$, respectively). It meant that more Internet usage on social interaction, game playing, and information seeking can enhance the users' interpersonal relationship.

Besides, Internet addiction negatively affect interpersonal relationship (β = -.18, $p < .05$). Internet addiction was associated with the Internet usage on social interaction, porn-website surfing, game playing, and video watching (β= .29, .11, .26 and .13, $p < .05$, respectively). Contrarily, Internet addiction was not affected by information seeking (β = .05, $p > .05$). That is, addicted in the social interaction, porn-website surfing, game playing, and video watching on the Internet led to poor interpersonal relationship.

So far, many scholars have explored the association between the Internet addiction and interpersonal relationship [8][11][18][44][45]. However, there were no consistent

Table 1. Instrument validity and reliability

Scale	Construct	Item	Factor loading	Composite reliability (CR)	Average variance extracted (AVE)	Cronbach's α
Internet addiction scale						0.922
	Tolerance	TO1	0.80	0.8576	0.6679	0.859
		TO2	0.86			
		TO3	0.79			
	Compulsive use	CU1	0.83	0.8605	0.6729	0.854
		CU2	0.79			
		CU3	0.84			
	Withdrawal	WI1	0.75	0.8621	0.6107	0.856
		WI2	0.86			
		WI3	0.74			
		WI4	0.77			
	Health management	HM1	0.72	0.8402	0.6378	0.886
		HM2	0.85			
		HM3	0.82			
	Time management	TM1	0.73	0.8650	0.6825	0.861
		TM2	0.89			
		TM3	0.85			
Interpersonal relationship scale						0.929
	Social acceptance	SA1	0.80	0.8971	0.6859	0.895
		SA2	0.85			
		SA3	0.87			
		SA4	0.79			
	Peer relationship	PR1	0.76	0.8700	0.5730	0.868
		PR2	0.78			
		PR3	0.70			
		PR4	0.73			
		PR5	0.81			
	Teacher-student relationship	TS1	0.75	0.8521	0.5909	0.848
		TS2	0.84			
		TS3	0.75			
		TS4	0.73			
	Parent-child relationship	PC1	0.86	0.9323	0.7339	0.932
		PC2	0.88			
		PC3	0.88			
		PC4	0.79			
		PC5	0.87			

conclusions. As stated by Whitty and McLaughlin [32], the reasons can be attributed to the changing nature of the Internet in these years. However, few studies have examined the effect of various function of Internet usage on the interpersonal relationship. This paper is intended as an investigation of the mediating effect of Internet addiction between Internet usage and the interpersonal relationship.

This study divided the Internet usage into five functions: social interaction, porn-website surfing, game playing, video watching, and information seeking. A total of 444 valid copies were collected from a university. The results indicated that different natures and contents of the Internet functions play different roles in the associations with the interpersonal relationship [28]. At the same time, the results also revealed that the Internet functions on social interaction, video watching, and information seeking can enhance interpersonal relationship while porn-website surfing and game playing cannot directly affect interpersonal relationship. On the other hand, with Internet addiction, the social interaction, porn-website surfing, and video watching led to poor interpersonal relationship mediated by Internet addiction. In conclusion, information seeking on the Internet can advance the interpersonal relationship. Moderate Internet usage in terms of social interaction and video watching can enhance interpersonal relationship, whereas they can also negatively affect interpersonal relationship with

Internet addiction. Although previous studies affirmed that social interaction functions on the Internet [28][46][47], such as email and chat room, and video watching [48] can facilitate interpersonal relationship, this study reminded that too extensively using Internet on these two functions still may decrease the quality of interpersonal relationship. Besides, the results in this study indicated that the porn-website surfing and online game playing may be detrimental to interpersonal relationship due to Internet addiction. Most past researchers asserted that game playing is associated with poor interpersonal relationship (e.g., [26][28][49]). They thought that game playing is one of the sources of interpersonal difficulties [26]. This study further explained that restrained game playing demonstrated no effect on the interpersonal relationship, but with the increase of the Internet addiction, the game playing will inversely correlate with the degree the game playing.

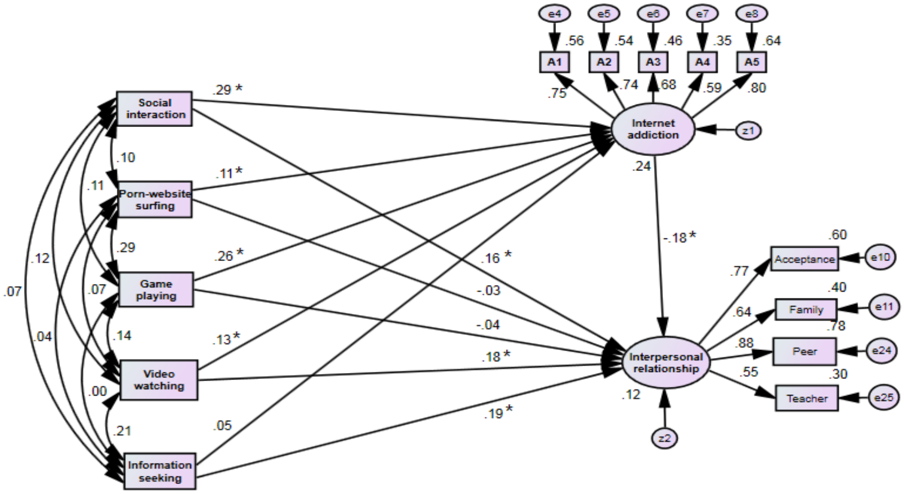


Fig. 1. Results of the model test

In sum, this study proposed a model of the structural association between Internet usage, Internet addiction, and interpersonal relationship. This model can explain the controversy from the previous research as to why Internet usage hinder or enhance interpersonal relationship. However, interpersonal relationship may consist of parent-child, teacher-student, peer, and even cyber-friend relationships. The interpersonal interactions are different from the various relationships. In the future, we will examine the effect of Internet usage on different interpersonal relationships.

References

1. Rodgers, S., Chen, Q.M.: Internet community group participation: Psychosocial benefits for women with breast cancer. Journal of Computer-Mediated Communication 10(4) (2005)

2. Lin, M.P., Ko, H.C., Wu, J.Y.W.: The role of positive/negative outcome expectancy and refusal self-efficacy of Internet use on Internet addiction among college students in Taiwan. Cyberpsychology & Behavior 11(4), 451–457 (2008), doi:10.1089/cpb.2007.0121
3. Milani, L., Osualdella, D., Di Blasio, P.: Quality of Interpersonal Relationships and Problematic Internet Use in Adolescence. Cyberpsychology & Behavior 12(6), 681–684 (2009), doi:10.1089/cpb.2009.0071
4. Chou, C., Condron, L., Belland, J.C.: A review of the research on Internet addiction. Educational Psychology Review 17(4), 363–388 (2005)
5. Chou, C., Hsiao, M.C.: Internet addiction, usage, gratification, and pleasure experience: the Taiwan college students' case. Computers & Education 35(1), 65–80 (2000), doi:10.1016/s0360-1315(00)00019-1
6. Griffiths, M.: Does Internet and computer "addiction" exist? Some case study evidence. Cyberpsychology& Behavior 3(2), 211–218 (2000)
7. Douglas, A.C., Mills, J.E., Niang, M., Stepchenkova, S., Byun, S., Ruffini, C., Blanton, M.: Internet addiction: Meta-synthesis of qualitative research for the decade 1996-2006. Computers in Human Behavior 24(6), 3027–3044 (2008)
8. Iskender, M., Akin, A.: Social self-efficacy, academic locus of control, and internet addiction. Computers & Education 54(4), 1101–1106 (2010)
9. Young, K., Rodgers, R.C.: The relationship between depression and Internet addiction. CyberPsychology & Behavior 1, 25–28 (1998)
10. Yu, A.Y., Tian, S.W., Vogel, D., Kwok, R.C.W.: Can learning be virtually boosted? An investigation of online social networking impacts. Computers & Education 55(4), 1494–1503 (2010)
11. Wang, C.C., Chang, Y.T.: Cyber relationship motives: Scale development and validation. Social Behavior and Personality 38(3), 289–300 (2010), doi:10.2224/sbp.2010.38.3.289
12. Katz, J., Rice, E.R.: Falling into the net: Main street America playing games and making friends online. Communications of the ACM Archive 52(9), 149–150 (2009)
13. Young, K.S.: Internet addiction: the emergence of a new clinical disorder. Cyber Psychology & Behavior 1, 237–244 (1998)
14. Shapira, N.A., Goldsmith, T.D., Keck, P.E., Khosla, U.M., McElroy, S.L.: Psychiatric features of individuals with problematic Internet use. Journal of Affective Disorders 57(3), 267–272 (2000)
15. Widyanto, L., Griffiths, M.: Internet addiction: A critical review. International Journal of Mental Health and Addiction 4(1), 31–51 (2006)
16. Beard, K.W., Wolf, E.M.: Modification in the proposed diagnostic criteria for internet addiction. Cyberpsychology and Behavior 4(3), 377–383 (2001)
17. Ko, C.H., Yen, J.Y., Chen, C.C., Chen, S.H., Yen, C.F.: Gender differences and related factors affecting online gaming addiction among Taiwanese adolescents. Journal of Nervous and Mental Disease 193(4), 273–277 (2005), doi:10.1097/01.nmd.00001583373.85150.57
18. Kwon, J.H., Chung, C.S., Lee, J.: The Effects of Escape from Self and Interpersonal Relationship on the Pathological Use of Internet Games. Community Mental Health Journal 47(1), 113–121 (2011), doi:10.1007/s10597-009-9236-1
19. Block, J.J.: Issues for DSM-V: Internet addiction. American Journal of Psychiatry 165, 306–307 (2008)
20. Tomer, J.F.: Addictions are not rational: A socio-economic model of addictive behavior. Journal of Socio-Economics 30(3), 243–261 (2001)

21. Yen, J.Y., Yen, C.F., Chen, C.C., Chen, S.H., Ko, C.H.: Family factors of internet addiction and substance use experience in Taiwanese adolescents. Cyberpsychology & Behavior 10, 323–329 (2007)
22. Nalwa, K., Anand, A.: Internet addiction in students: A cause of concern. CyberPsychology & Behavior 6(6), 653–656 (2003)
23. Whang, L.S., Lee, S., Chang, G.: Internet over-users' psychological profiles: A behavior sampling analysis on internet addiction. Cyberpsychology & Behavior 6(2), 143–150 (2003)
24. Iskender, M., Akin, A.: Self-compassion and Internet addiction. Turkish Online Journal of Educational Technology 10(3), 215–221 (2011)
25. Anderson, K.J.: Internet use among college students: An exploratory study. Journal of American College Health 50(1), 21–26 (2001)
26. Lo, S.K., Wang, C.C., Fang, W.C.: Physical interpersonal relationships and social anxiety among online game players. Cyberpsychology & Behavior 8(1), 15–20 (2005), doi:10.1089/cpb.2005.8.15
27. Kraut, R., Lundmark, V., Patterson, M., et al.: Internet paradox: a social technology that reduces social involvement and psychological well-being? American Psychologist 53, 1017–1031 (1998)
28. Punamaki, R.L., Wallenius, M., Holtto, H., Nygard, C.H., Rimpela, A.: The associations between information and communication technology (ICT) and peer and parent relations in early adolescence. International Journal of Behavioral Development 33(6), 556–564 (2009), doi:10.1177/0165025409343828
29. Campbell, A.J., Cumming, S.R., Hughes, I.: Internet use by the socially fearful: Addiction or therapy? Cyberpsychology & Behavior 9(1), 69–81 (2006)
30. Shaw, L.H., Gant, L.M.: Defense of the internet: The relationship between Internet communication and depression, loneliness, self-esteem, and perceived social support. Cyberpsychology & Behavior 5(2), 157–171 (2002), doi:10.1089/109493102753770552
31. Morahan-Martin, J., Schumacher, P.: Incidence and correlates of pathological Internet use among college students. Computers in Human Behavior 16(1), 13–29 (2000), doi:10.1016/s0747-5632(99)00049-7
32. Whitty, M.T., McLaughlin, D.: Online recreation: The relationship between loneliness, Internet self-efficacy and the use of the Internet for entertainment purposes. Computers in Human Behavior 23(3), 1435–1446 (2007), doi:10.1016/j.chb.2005.05.003
33. Demtrovics, Z., Szeredi, B., Rozsa, S.: The three-factor model of internet addiction: The development of the Problematic Internet Use Questionnaire. Behavior Research Methods 40(2), 563–574 (2008), doi:10.3758/brm.40.2.563
34. Nichols, L.A., Nicki, R.: Development of a psychometrically sound Internet addiction scale: A preliminary step. Psychology of Addictive Behaviors 18(4), 381–384 (2004), doi:10.1037/0893-164x.18.4.381
35. Bonebrake, K.: College students' internet use, relationship formation, and personality correlates. Cyberpsychology & Behavior 5(6), 551–557 (2002)
36. Li, S.M., Chung, T.M.: Internet function and Internet addictive behavior. Computers in Human Behavior 22(6), 1067–1071 (2006), doi:10.1016/j.chb.2004.03.030
37. Anderson, J.C., Gerbing, D.W.: Structural equation modeling in practice: a review and recommended two-step approach. Psychological Bulletin 103(3), 411–423 (1988)
38. Nunnally, J.C.: Psychometric theory. McGraw-Hill Press, New York (1978)
39. Fornell, C., Larcker, D.: Evaluating structural equation models with unobservable variables and measurement error. Journal of Marketing Research 18(1), 39–50 (1981)

40. Baggozi, R.P., Yi, Y.: On the evaluation of structural equation models. Journal of the Academy of Marketing Science 16(1), 74–94 (1988)
41. Bollen, K.A.: Structural equations with latent variables. John Wiley, New York (1989)
42. Hu, L.T., Bentler, P.M.: Cutoff Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives. Structural Equation Modeling-a Multidisciplinary Journal 6(1), 1–55 (1999), doi:10.1080/10705519909540118
43. Zhao, L., Lu, Y.B., Wang, B., Huang, W.N.: What makes them happy and curious online? An empirical study on high school students' Internet use from a self-determination theory perspective. Computers & Education 56(2), 346–356 (2011), doi:10.1016/j.compedu.2010.08.006
44. Batigun, A.D., Hasta, D.: Internet addiction: an evaluation in terms of loneliness and interpersonal relationship styles. Anadolu Psikiyatri Dergisi-Anatolian Journal of Psychiatry 11(3), 213–219 (2010)
45. Schmit, S., Chauchard, E., Chabrol, H., Sejourne, N.: Evaluation of the characteristics of addiction to online video games among adolescents and young adults. Encephale-Revue De Psychiatrie Clinique Biologique Et Therapeutique 37(3), 217–223 (2011), doi:10.1016/j.encep.2010.06.006
46. Franzen, A.: Does the internet make us lonely? European Sociological Review 16, 427–438 (2000)
47. Morahan-Martin, J., Schumacher, P.: Loneliness and social uses of the Internet. Computers in Human Behavior 19(6), 659–671 (2003), doi:10.1016/s0747-5632(03)00040-2
48. Bou-Franch, P., Lorenzo-Dus, N., et al.: Social Interaction in YouTube Text-Based Polylogues: A Study of Coherence. Journal of Computer Mediated Communication 17(4), 501–521 (2012)
49. Ho, S.M.Y., Lee, T.M.C.: Computer usage and its relationship with adolescent lifestyle in Hong Kong. Journal of Adolescent Health 29(4), 258–266 (2001), doi:10.1016/s1054-139x(01)00261-0

A Study of Course Assessment on C++ Programming

Chen-Kuei Yang[1], Pei-Yu Tsai[2], and Tsu-Feng Ho[1]

[1] Department of Computer Science and Information Engineering
Ming Chuan University, Taipei, Taiwan
{Ckyang,tfho}@mail.mcu.edu.tw
[2] Institute of Education, Xiamen University, China
tsaipamela@hotmail.com

Abstract. The aim of the study is to enhance learning effectiveness, promote students' willingness, and their constantly learning ability. Curriculum design is based on the action research method and the spiraling cycle policy. Teaching activities are designed with multi-fun activities in order to induce students' interest and motivation in learning, and urge students to study hard. Diversified evaluations such as quality observations and quantity tests are adopted for learning assessment. Learning processes and learning attitudes are also considered. The evaluation of the effectiveness of the course rests on the measurement results through competence inventory tests and programming qualify examination. As a result of C++ programming qualify examination, 87.5% of students in the sampling class are passed. The best grade point average in School of Information Technology is obtained. All common competences are promoted except stress tolerance competence. Six of competences are grow up significantly.

Keywords: Integration teaching, Curriculum design, Competence inventory, Common competence, Course assessment.

1 Introduction

In recent years, Taiwan's educational reform has stressed the subjectivity of student-oriented learning, and even now our university educational reform is based on "student-centered" teaching. Apparently, learners' "learning" is the key point, and "teaching" needs to guide “learning” to the right way. Hodson [1] believes that teachers who are in student-centered could listen to students and focus on students' learning experience. With the development of multi-society, our value is rapidly changing. Accordingly, the significance of university education is in a confusing predicament especially in teaching targets and strategies, which cause students' low performance and willingness. Moreover, it may result in self-abnegation or re-examination. If teachers' teaching and effort are not recognized by students, the interaction of the teaching process, the knowledge of the instruction, and even the teaching attitude of teachers will be effected more or less, which impact on the effectiveness of student education. Consequently, the issues in many challenges of educational reform are worthy topics to study, for example, how to strengthen the programming in computer

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 383–392.
DOI: 10.1007/978-3-642-35452-6_39

education, and how to reduce the gap of human resource between the input of workplace selection and the output of school education. The purpose of this study is to explore the curriculum design and effectiveness of the integrated common competences into C++ programming course in the core course of the School of Information Technology.

1.1 Research Objectives

Teaching students with common functions required by the workplace is the basic foundation of the comprehensive and balance development of the university [2-3]. Therefore, the curriculum design is used "integrated common competences into professional course", and conduct teaching evaluation in order to help teachers to review each teaching activity for their improvement. Meanwhile, through common competence measurements, this study will explore and detect the effectiveness of the integrated four or more common competences into professional course.

1.2 Research Objects

Studies objects are two professors and forty-eight students from department of Computer Science and Information Engineering (abbrev. as CSIE) of Ming Chuan University (abbrev. as MCU) on C++ programming course during 2011-2012 academic year. Students are freshmen whose ages are between 18 and 19 and without the foundation of the programming. Students are randomly compiled at class B of CSIE when they registered into MCU. The numbers of boys are thirty-eight, and girls are ten.

1.3 Research Method

The course of the study is in accordance with the concept of the action research methods and the procedure of the spiraling cycle policy [4]. It mainly analyzes the content of C++ programming course, the interactive, contest, practice, peer- and self-learning etc. strategies are considered. Therefore, the curriculum design rests on the moodle system [5] in order easily to conduct the teaching activities.

There are four main sources for data collection: (1) Teaching logs in reflection. After each lesson, the instructors would immediately carry on 10-15 minutes discussion and reflection, which record the class of procedure, atmosphere, teaching skills and student reaction. A log is for a week and totally thirty-two logs；(2) There are total seventy-eight tests from the college students' psychological common competence pretests and posttests at 2011-2012 academic year；(3) Student surveys during the midterm of each semester, a total of hundred parts (including redo students)；(4) Programming Qualify Examination (abbrev. as PQE) which held each year in late May by the School of Information Technology (abbrev. as SIT) of MCU.

2 Related Literature

To perform the "low interaction" programming language and support the development of the major software project, C or C++ programming must be translated and compiled. Generally, this programming language is constructed in an integrated development environment (IDE, Integrated Development Environment), which includes a lot of developmental program functions, but most beginners will not use them. Felleisen, Findler, Flatt, Krishnamurthi[6]and Kolling, Quig, Patterson, Rosenberg[7] think that beginners in IDE use the "professional" and "low interaction" programming languages to learn programming design will more likely to produce learning problems such as complex syntax, misunderstanding the error message, and slow learning. For beginners, focusing on fast interaction, problem solving and logical thinking is more important than learning grammar and developmental tools. Hence, medical education [8-10], service learning and character education [11] which based on the theories of problem-based learning (PBL) and integrated teaching have fruitful achievements. Teachers should mention how to use language tools to practice problem solving in order to avoid students low interest and response due to traditional teaching types. Indeed, we need to use PBL and integrated teaching to instruct C++ programming.

Bruner[12] emphasizes the role of teachers is to build a proper learning environment for students to enhance their independent learning methods and self-learning interests, satisfying their knowledge and curiosity, and play their ability to achieve the learning objectives rather than teach full range of knowledge. Edwards [13] mentions teachers could build up learning standards, deal with students learning behaviors, and supervise teaching activities through creating learning environment. Additionally, teachers should apply regulations such as rewards and punishments to maintain an effective learning and efficient learning environment, and a good teacher-student relationship.

College students whose ages are between fifteen and twenty five are in the exploring stage. At the stage, they are trying to explore the concept of self, career and social roles [14-15]. Students will carefully choose their learning paths, which may affect their lives and careers in the future. Therefore, the career planning and counseling at this stage is significantly important. National Youth Commission [16] surveyed and found that employers favor graduates with eight core job skills and features, which are good working attitudes, stabilities, resisting pressure, communication, learning willingness, malleability, team work, expertise and technology, exploring and problem solving, and foreign languages. Cheng, Lin, and Cheng [17] studied on the employment of college students and concluded nine common competences which are required in the workplace. According to the literature review of the common competences dimensions, it concluded fifteen dimensions, and with twenty-one experts' opinions by survey to compile nine common competences dimensions. The psychological test system of competence inventory is followed by Cheng et al [17] proposed, which installed in the system of Student Affairs web server to provide for students free measurements and usages. The individual measurement will automatically remit to the students' e-portfolio for student reference. Four common competences are focused to promote in our course design such as teamwork, communication, problem solving, and professional learning.

3 Teaching Strategy

The first week of the beginning of the course, announce lecture rules and describe grouping rule which randomized grouping members at the first semester and in accordance to their grades at the second semester. The psychological competence test has been described and surveyed in the programming lab. Then performing group interactive learning, stimulating self-learning willingness, and promoting the learning opportunity from each other are our strategies. Conducted by group discussions to analyze problems, understand problems and solve problems during the programming practice lessons. Each student should write, test, and debug codes by self and may ask a help from the crew if a dilemma case happen. Once the program is fully completed, it needs to upload program to moodle system. The system will record the uploaded time and compute the average uploaded time of the group. Each group reports their program how to design and shares the programming experience. Professors will amend the teaching activities according to the teaching log discussion, group code review performance and their average grades of quizzes.

(1) Five to ten-minute beginning quiz and ending quiz
The multiple-choice, fill in the blank or cloze question may be included into one quiz. At least one to three questions are in the quiz which easily created by moodle system [5]. The period of quiz is presetted at the expected time from the moodle system, and students must on time go to the class and do the test.

(2) Lectures
First, explain the answers of each quiz. Second, encourage students with good answers. Then, describe teaching topic focus that includes four steps which are defining the problems, analyzing the problems, searching the necessary tools (language instruction or function), and finding the solutions. Finally, use interactive Q&A to observe and record each group performance.

(3) Problem oriented teaching
In teaching process, the teacher first cites the example of the problem according to the problem definition, analysis of the problem, the necessary tools, and solution of the problem. A detailed explanation of those main four-steps is required. Subsequently cite two or three similar problems in accordance with the mentioned above problem solving steps. Open a panel discussion of daily similar problems and randomly selected a group to report their problem solving steps. The problem-solving experiences are as mediums to integrate common competence into teaching such as pursuit programming logical thinking (initiative), finding a solution (innovation), deriving an algorithm (problem solving), coding (professional learning), compiling (professional learning), debugging(team work), and doing group code review (communication).

(4) Encouragement and guidance
During the lecture lessons, the teacher is based on the reaction of the students to take a listen, smile, and affirmation. To observe and concern students' learning and promote the willingness of learning are the teacher's focus. Group code review is a useful method to understand the idea of others. Encourage students to share self-learning experience to others if they done a good solution. Teachers should timely clarify

students' doubts and cultivate the whole programming concepts. The capacity of problem solving can be enhanced by repeatedly drills of diversity problems.

(5) Experience sharing

Teachers should initiative to share their own life of professional service experiences to students. Group code review is a good approach for experience sharing and peer learning, which is easier for students to experience programming skills, and is willing to share their own experiences. Some common competences (professional learning and communication, for example) may be integrated into doing the experience

Table 1. Teaching activities of C++ programming

Content	Responsibility of student	Responsibility of teacher
presentation	Making a note	Give questions or answer questions
Q&A contest	Report the solution after group discussion	A teacher gives feedback comments immediately and record their grade
Code review	Review the codes and give suggestions	A teacher gives rewards to student for their good coding and record grade
Sharing coding experience	Each group share the coding experience	A teacher shares the professional rules and encourages students with good coding skills to share their experiences
Program Contest	Upload to moodle system when group finish the program	A teacher reviews the codes and records the upload time
Beginning quiz	Review the textbook	Record the grade and explain the quiz
Ending quiz	Students pay attention in the class	Record the grade and explain the quiz
Midterm and final examination	Open books on line test	A teacher gives high or low grades, which are based on the skills of coding and parts of questions they are completed.
Report homework solution	Review codes and give suggestions	A teacher feedbacks comments immediately, gives rewards for good codes and records their grade
Report intern project	Report their intern project and discuss	Remind students to summarize report or query; Let poor oral presentation of students have more chance to report; Record the grade for each group

sharing. Teacher will amend the teaching activities according to the students' performance on group code review or the average grades of quizzes to check the coding skills they learned.

(6) Creative learning

Teachers flexibly provide creative learning such as riddles, story-telling, and a piece of movie to ease the mood of tension and anxiety of the students in the learning process, which let their thoughts be temporarily precipitation, re-staring and relink their learning. Encourage students to participate in students' societies or clubs, lectures, and reading computer-related novels, comic books, magazines.

(7) Mapping figure for conceptualization

Encourage students to do conceptualization mapping to issues, theories and programming methods. First, from the input, process, output (IPO) programming concept mapping to the daily matter transaction process is the same as control things by handled its IPO. Second, the structure programming concept is mapping to the document signed process or hierarchical management of organization. Third, the functional programming concept is mapping to the functions of digital camera. Finally, the object oriented programming concept is mapping to the principle of Lego bricks. All conceptualizations are modeling that can be seen everywhere in the daily living image. The cultivation of those conceptualizations to students, and the integration of the common competences through teaching activities are also considered in the curriculum design. The related teaching activities are shown on Table 1.

4 Course Assessment and Reflection

Teaching evaluation is the use of scientific methods and techniques to analyze students' behavior and achievement, includes a series of works of analysis, research, and judgment according to the teaching goals on students' learning performance[18]. Diversified assessment is adopted for the evaluation of the students' learning effectiveness. Assessments include quizzes (30%), team cooperation contest performance (10%), program code reading and experience sharing (10%), proactive respond (10%), midterm (10%) and final test (30%). The evaluation of the curriculum effectiveness in two ways, one is based on the outcome of the PQE, the other one is based on college students common competence psychological test results that proposed by Cheng et al [17].

Since 2011-2012 academic year, the PQE is the requirement before students graduation from SIT of MCU. Totally, eleven classes in SIT. There are 48 students in Class B of CSIE. Three students are absent on PQE and three students are un-passed on PQE. Therefore, forty-two students pass the PQE, and the pass rate is 87.5% (93.3% if absents are not considered). The grade point average of PQE is 81.13 that shown on Table 2. The best grade point average in SIT is obtained. Compared with the worst pass rate's class in SIT, the pass rate is higher than 56.1%. And compared with the lowest grade point average of PQE in SIT, there exists 37.97 highly difference.

Table 2. C++ PQE results of SIT of MCU in academic year 2011-2012

Class ID	num	absent	un-pass	pass	pass%	average
CSIE-A	52	2	8	42	80.8%	76.48
CSIE-B	48	3	3	42	87.5%	81.13
CICE-A	49	1	26	22	44.9%	60.28
CICE-B	52	8	6	38	73.1%	80.45
MIS-A	55	5	27	23	41.8%	43.16
MIS-B	55	4	10	41	74.5%	59.27
MIS-C	51	4	31	16	31.4%	47.52
EE-A	57	2	11	44	77.2%	54.25
EE-B	47	3	26	18	38.3%	55.50
CCE-A	55	6	3	46	83.6%	79.83
CCE-B	54	3	9	42	77.8%	75.16

Common competence psychological tests were conducted at C ++ programming practical lessons on two semesters (be called pretest and posttest) in 2011-2012 academic year. The psychological consultant can give assistance to students when they are done the test. Forty-eight and thirty test results are obtained at pretest and posttest respectively, but only twenty-six test results are valid. Therefore, 54% of valid tests are obtained because some students drop the course, absent and incomplete the test when time out at the practical lesson.

From Table 3, eight common competences are improved except the stress tolerance, six of them which include initiative, innovation, Customer focus, communication, Team work, Professional learning (the last three competences are our focus competences to integrate into our course design) are significantly improved because of 3.0 average difference between pre- and post- test. Only the problem solving competence is slight improved. The reason may due to our amendment the practice problems more complicated at the second semester for object oriented programming learning. Most students can't understand why need to create an object and how to define a based class those result in slight improvement in problem solving competence. Fig. 1 is a radar map to indicate the average scale of common competences that is generated from table 3. Mining the stress tolerance scale drop's reason and heavier freshman courses, such as calculus, linear algebra, introduction to computer science are taken. Those courses are incomprehensible and result in higher frustrated degrees. Some appropriate remedial courses or other reinforcement courses may need to provide at the near future. Regarding to the problem-solving and achievement orientation competences (notwithstanding the progress but not significant), may be considered reasonably phenomenon because of the programming beginner of freshman. The logical thinking capacity is subject to the subsequent courses of continuous 3-year university learning in order to significantly enhance these two common competences.

Although the growth and performance of the common competence are not contribution only from one course, but the results from tables 2, 3 and fig. 1, which shows teachers bother deliberation course design to help student improving and learning is an indisputable fact. Also to prove curriculum arrangements for freshman in Department of CISE of MCU does contribute to the cultivation of student learning and common competences growth. Nine common competences except the stress tolerance are improved. The drop phenomenon on stress tolerance may need to provide some appropriate remedial or other reinforcement courses at the near future.

Table 3. Nine common competences test results from class B of CSIE freshmen

Competence Test	Initiative	Stress tolerance	Innovation	Communication	Team work	Customer focus	Problem solving	Professional	Achievement
Pre test	52.808	60.538	62.423	54.808	64.308	61.654	68.077	69.692	58.577
Post test	62.385	49.154	69.808	57.808	69.538	66.192	70.192	75.731	59.308

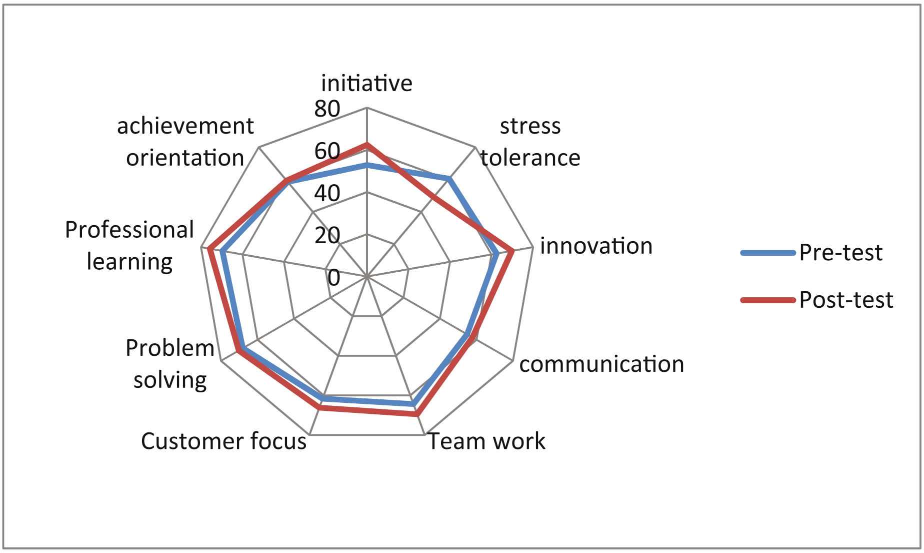


Fig. 1. Nine common competences of pre-test and post-test results from the sampling class

Reflections of teaching can be learned from the final questionnaire. (1) Student attendance rate in practical lesson is higher than in the regular course, possible reason is that the period of the regular course is in the early mornings (09:10-12:00) and practical lesson is in the afternoons (12:50-14:40). The student can't get up in the early mornings. (2) The teaching classroom selection, 67% of students prefer the general classroom to the computer room. It is shown on a good learning place results in providing students' initiative, positive and helpful learning, and students give mostly positive assessment. (3) The number of questions and the time period of the quiz are 1-2 entitles in 5-10 minutes and test at the beginning or at the ending of class. The purpose of the beginning quiz is to enhance the attendant rates and expected to students can do a preview study. The ending quiz try to remind the students reaching the content of the class by quiz and enhance students to pay their attention. Grades of quizzes must be accounted for a certain percentage of the final grade, which will result in a positive effect on student learning. From statistical analysis of the students' views on the quiz, students give a positive support (a 3.6 scale is obtained from 1 to 5 scales). (4) The view of teaching materials, the poor logical thinking skills and the English dyslexia will result in the lower willingness of learning, most students would like to read the Chinese textbook and hope teachers supplemented with English terminology in Chinese-based explanation. (5) In teaching strategy, teachers use groups learning like a hen take care with chicks method, and conduct peer learning, which students with learning difficulties would know who can give a help. Students give a positive support (over 3.5 are obtained).

5 Conclusions and Discussions

The course is designed to focus on student-centered teaching that is the main teaching focus. This study is designed by multiple interactions to find the solution, derived algorithm trigger to learn, and code review sharing, peer to convey as based on the spirit of the teaching policy. The main purpose is caused by the willingness of students to take the initiative to learn and contribute to the ability to continue to learn through peer learning. In one academic year psychological test at two different times, it is found that six common competences are growth but the stress tolerance ability is drop down. Of Course, most common competences of growth achievements can't be attributed to a single course teaching. However, it can be used to inspire the feasibility and effectiveness of the integration some common competences in professional courses teaching. Meanwhile, the other assessment of the course is evaluated the PQE as the reference basis. The results of 87.5% students of the sampling class pass test and the best class of grade point average in the SIT of MCU is obtained. Professional knowledge growth, the effectiveness of the course and learning activities design are also proved.

This study results are indeed available references for professional courses' teachers and curriculum committee members. Some competences should be carefully planned and discussed how and what to be integrated into the professional courses. We also recommended that each student before graduation should give at least three times of the common competence psychological tests, testing times are suggested on freshman, sophomore and senior year before graduation. Simultaneously, view all requirement and elective courses can be cultivated which common competences to student, and integration links to the elective course system. System can be automatically give customized recommended list of the elective courses according to each student's test results, so students will elective on some of suggestion courses, but not lightly or random elective on nutrient credits.

Acknowledgment. This work was partially supported by National Science Council, Republic of China under Grant NSC101-2221-E-130-025. Dr. Yang would like to thank the referees sincerely, whose valuable comments and suggestions have helped improving the presentation of this paper.

References

1. Hodson, K.K.: Multiple Perspectives, The Changing Faces of Student- Centered Teaching: Refiguring the Center. In: The 53rd Annual Meeting of the Conference on College Composition and Communication, Chicago, pp. 1–15 (2002)
2. Wood, R., Payne, T.: Competency-based recruitment and selection. John Wiley & Sons, New York (1998)
3. Harvey, L., Locke, W., Morey, A.: Enhancing employability, recognizing diversity. Universities, London (2002)
4. Wu, M.L.: Introduction to education action research - theory and practice. Wunan books, Taipei (2001)
5. Yang, C.K., Chia, T.L., Liu, J.J.: An Exploring of Applying Moodle System at Ming Chuan University. In: Proc. of 2011 International Conference on Information Technology and Pratical Theory Ming Chuan University, Taipei, pp. 77–85 (2011)
6. Felleisen, M., Findler, R.B., Flatt, M., Krishnamurthi, S.: The Teach Scheme! Project: Computing and programming for every student. Computer Science Education 14(1), 55–77 (2004)
7. Kolling, M.B., Patterson, Q.A., Rosenberg, J.: The Blue J system and its pedagogy. Computer Science Education 13(4), 249–268 (2003)
8. Albanese, M.A., Mitchell, S.: Problem-based learning: A review of literature on its outcomes and implementation issues. Academic Medicine 68(1), 52–81 (1993)
9. Boud, D., Feletti, G.: The Challenge of Problem Based Learning, 2nd edn. Kogan Page Limited, London (1997)
10. Shiau, S.J., Chiang, Y.T., Huang, Y.C., Chiu, P.R.: To Explore the Embracing 3C in the Master Course of Nursing Theory. Fu-Jen Journal of Medicine 7(1), 9–16 (2009)
11. Pan, C.D., Lee, C.Y.: Practice and Reflection ofthecharacter education. Taiwan Journal of General Education 4, 93–120 (2009)
12. Bruner, J.S.: Toward a theory of instruction. W.W. Norton, New York (1966)
13. Edwards, C.H.: Classroom principles and management. Macmillan, New York (1993)
14. Super, D.E.: Career Education and the Meaning of Work. US Dept. of Health, Education, and Welfare, Washington D.C. (1976)
15. Zhou, C.M., Shen, C.H.: An Analysis of the Conditions of Employment for Technological and Vocational School Graduates Majoring in Commercial Subjects. Bulletin of Educational Research 50(2), 147–178 (2004)
16. National Youth Commission: Colleges to enhanceyouth employmentforce. Executive Yuan, Taipei (2009)
17. Cheng, N.S., Lin, P.C., Cheng, Y.C.: Developing the Competence Inventory for College Students. Psychological Testing 56(3), 397–430 (2009)
18. Chien, M.F.: Multi assessments idea and methods for learning and teaching. Modern Education Forum 7, 189–197 (1999)

Challenges for Promoting the Decision-Making Processes Based on Spatial Data Analysis

Elzbieta Malinowski

Department of Computer and Information Sciences
University of Costa Rica
elzbieta.malinowski@ucr.ac.cr

Abstract. Spatial Data Infrastructures (SDIs) provide members of society with access to and sharing of spatial and conventional data related to different aspects of human activities, e.g., social, economic, environmental, etc. However, to make this data available to a wide variety of users, several challenges must be overcome. These challenges refer to different aspects, e.g., the expansion of the vision about the purpose of SDIs that allow citizens to improve the information literacy level, the establishment of a governmental strategy and national standards to publish and consume data, the awareness of issues related to data quality and integration as well as the changes that are necessary to implement at the educational and social levels. In this article, we refer to the role of SDIs and the challenges that need to be faced, especially in developing or newly-industrialized countries to shorten the gap with developed countries.

Keywords: Decision making, SDI, spatial data, developing countries.

1 Introduction

Geographic or spatial data forms part of every-day human activities all around the world. It is used with different purposes from leisure activities to decision-making processes, e.g., to obtain directions through different mobile devices, to promote different tourist attractions, to exchange data for preventing disasters, or to support decision-making processes related to agriculture, transportation, or business issues, among others. Therefore, the traditional production and usage of geographic data by geo-experts, e.g., geographer, cartographer, surveyors, is now being subordinated to the business issues confronted by government and commercial organizations [19].

The widespread use of spatial data can improve the decisions taken at the personal, local, or global levels, because it is well known that the presentation of data in a space helps to discover patterns that otherwise would be difficult to find. For example, on a personal level a user can determine if the land that s/he wants to buy is not close to a garbage disposal or high tension towers; on a local level, municipalities could rely on this data to prepare plans for cantonal human development in order to improve living conditions and strengthen territorial planning; on a global level one could make better

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 393–402.
DOI: 10.1007/978-3-642-35452-6_40

decisions to create risk prevention plans for areas of hazardous risks (e.g., flood, corrosion, earthquake) considering population, house distribution, road availability, etc.

Promoting the culture of decision making based on data requires several issues to be addressed. Firstly, currently the focus among world-wide researchers and the Web-related community is on big data. However, issues related to big data exist mainly in developed countries where the level of information literacy is high and users produce data and need it during decision-making processes. In developing or newly-industrialized countries, there is still a need to resolve problems as well as promote and educate citizens to use and analyze data before making decisions, i.e., improving their information literacy level and making them aware of the importance that data and analytics tools have in today´s competitive world. Secondly, there is a need for providing infrastructures that facilitate data access and allow easy integration of data originated from different sources, e.g., from research centers, government, or other institutions/organizations responsible for their collection. Thirdly, spatial data should be considered as "first-class citizens" in the decision-making processes and new tools that do not require specialized knowledge related to geographic information systems (GISs) should be highly accessible. Currently, obtention and integration of data is not an easy task and requires expertise in the field of computer science to implement processes for data cleaning, transformation, and integration. This might change if the development of a Spatial Data Infrastructure (SDI) considers the geomatics vision, i.e., an integrated vision that represents needs of every potential user.

This paper refers to the challenges related to promote a vision of making decisions based on spatial and conventional data, especially in developing countries. Section 2 describes the role that a SDI should play, while Section 3 refers briefly to knowledge and tools that are required for spatial data analysis. Section 4 presents different challenges that should be overcome to help developing countries to improve their information literacy level. Section 5 concludes this article.

2 Role of the Spatial Data Infrastructure

A SDI is defined as "the technology, policies, standards, human resources, and related activities necessary to acquire, process, distribute, use, maintain, and preserve spatial data" [12]. It can be seen as a fundamentals for a Information Society willing to use and share spatial data over the Internet that "allow nations to better address social, economic, and environmental issues" [6]. There are many successful initiatives of the SDIs, e.g., the Canadian Geospatial Data Infrastructure, the American National Data Infrastructure, or the Infrastructure for Spatial Information in Europe [15]. Furthermore, the Global SDI Association leads several initiatives to support international collaboration [6]. This global tendency of developing SDIs can be seen over the world, e.g., Asia [13] with Taiwan in its 10-years SDI plan [10], or Latin America [4].

Costa Rica is a small country in Central America, where several public institutions are interested in having spatial data for different purposes. Currently, the main responsibility in spatial data acquisition and regularization relies on the Cadaster and Registry Regularization Program [17] approved in December 2001 by the Costa Rican

Legislative Assembly and supported by the Inter-American Development Bank through a US$65 million loan for an amount up to US$98 million in 2010. The program's objective is to develop a more secure and modern system of land property rights registration and administration. The necessity to have this system arose from the irregular situation in the Costa Rican Cadaster, having 1.6 million plans with 1.2 million parcels and exceeding the Costa Rican territory for about 20%. The new system will provide data for another system called SNIT (National Land Information System) that is considered a Costa Rican SDI. Both systems will serve as a platform of spatial data exchange for government agencies, including municipalities that are at the lowest hierarchy level of administration, and other external users. Other government institutions, e.g., for electricity or census, are also collecting their spatial data.

A SDI should not only facilitate access to and sharing of spatial data over the Internet, but also help promote a culture to use this data in decision-making processes at different administration levels, e.g., municipalities, ministries, educational or health institutions, among others. This aspect is important especially in the developing countries because it could help better support local decisions, produce knowledge required for a country to grow, inculcate among youth another vision of reality, and put the country at the global development level.

Publication of spatial data in the SDI should also reflect a geomatics perspective, i.e., consider that the spatial data can be used for different purposes by different specialists. For example, the presentation of beaches or airports by points impedes the use of this data by travel agencies, where the length or area of beaches and airports are important aspects. In addition, the geomatics approach for creating the SDI allows one to consider spatial as well as related conventional data as useful elements in the decision making processes. In this way, the SDI could be a national provider for spatial and conventional data via standardized web (map) services based on international, and locally adapted, standards. This could reduce the acquisition costs of its own spatial data by different institutions and eliminate the need or facilitate the process of data integration. However, the development of SDIs usually relies on geo-experts that may limit their knowledge to the particular field of expertise, e.g., cartography, surveying, and do not consider and promote data reusability by different stakeholders.

3 Knowledge and Tools Required for Spatial Data Analysis

Traditionally, the manipulation and querying of spatial data rely on (often complex) GISs that require a geo-knowledge, e.g., knowledge about spatial reference systems, layers, storage formats, operations and/or functions necessary for spatial data analysis; this geo-knowledge is not always available to personnel responsible for making decisions, especially at lower administrative levels of municipalities or other administrative entities. In addition, traditional GISs handle data using the hybrid architecture, where the spatial and conventional data are stored separately [18]. One example is the particular format called shape file that includes (at least) two files: one for storing geometries and another one for conventional data. This type of architecture does not provide the capabilities to establish the control of data quality and prohibit providers to publish inconsistent data, e.g., to put a null value in a district code or to include a geometry of a district outside the limits of its corresponding county. Currently, the situation is improving and we can see a clear convergence of GISs and spatially-extended DBMSs, thus incorporating all known features of DBMSs and extending them with concepts related to spatial data storage and manipulation [18]. As a consequence these converging technologies could provide more control mechanisms and improve data quality.

Furthermore, even though GISs offer a big variety of analysis tools [5], they usually are directed to specialists in a particular geo-field and require advanced geo-knowledge to use them and understand them, as well as to interpret the results. Although these tools meet the expectations of different types of geo-experts, they represent a challenge to other specialists. Different possibilities of analysis should be available for users that need spatial data for improving their decisions but do not count on geo-knowledge. This situation is similar to the one we have had at the end of the 90s, where decision-making users were benefitted by On-Line Analytical Processing tools (OLAP) and could analyze the summarized and detailed data in a dynamic way without any programming effort. These kinds of tools are already extended to include a spatial component and are known as spatial OLAP (SOLAP) [14]. It is already demonstrated that SOLAP applications are much easier to use by non-expert users than pure GIS applications [14]. SOLAP tools provide spreadsheet-like environments enabled with spatial representations and allow one to develop Internet-based applications that are closer to informatics knowledge of users with the lack of or little geo-knowledge. Furthermore, for more advanced users spatial data mining could deliver knowledge that is difficult to find otherwise [8].

4 Problems Identified and Challenges to Overcome

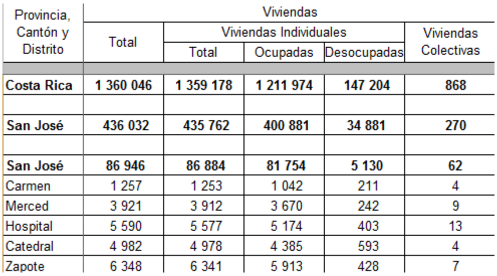
The variety of options that currently exists for analyzing spatial and conventional data can be seen as a positive indicator for creating a new information society. However, in many occasions this represents an intimidating and overwhelming factor if users are not adequately prepared for technological advances and do not see the advantages of having a variety of data to support decision-making processes. Different challenges must be overcome to change this situation, especially in developing countries.

4.1 Data Quality and Integration

Data quality is one of the most important challenges that developing countries should overcome to ensure that decisions are made based on data that represents real situations. Furthermore, the utility of spatial data is bigger when is accompanied with conventional data according to users expertise and problems to be resolved. However, in Costa Rica, and in other developing countries, procedures and formats for data publications and exchange are not well established. Different institutions may deliver data in whatever formats they prefer and users' responsibility is to deal with problems that data may have. The detected problems refer to spatial and conventional data.

Publicly available spatial data refers to a wide variety of topics, e.g., administrative distribution, health or economic regions, localization of beaches, bank agencies, schools, clinics, crop distribution, risk zones, national park distribution, etc. However, this data cannot always be processed directly and complex cleaning and transformation procedures must be developed and applied before using it. For example, even though Costa Rica has 475 districts, the shape file includes 849 tuples[1] and among them, 45 districts are represented by more than one tuple. In these repeated tuples the only different attribute is the one describing geometry, since each tuple includes a part of the geometry representing a district. If the user wants to calculate the total population of Costa Rica using the population of all districts as a starting point, s/he will obtain as result 7 million in the year 2000, even though the real population was 3.8 million. Other problems identified refer to (1) null values in the name of objects, e.g., among 31468 tuples representing rivers, 15730 do not have names and 1853 are represented more than once; (2) geometries that are outside Costa Rica's territory, e.g., some clinics; (3) incorrect spatial extensions of related objects, e.g., some counties expand between two provinces; (4) incorrect ring orientation, i.e., order of specifying coordinates of polygons that affect calculations and analysis.

Conventional data used together with spatial data can be obtained from different places. For example, the National Institute of Statistics and Census provides census data from different time periods; several research centers at the University of Costa Rica, that are in charge of monitoring the development and addressing important issues that allow the country to improve its situation, publish necessary data to improve social awareness; the Social Security system includes data about the medical and health situation in the country. Available data refers to different aspects, e.g., read/write literacy levels, death and birth rates, professional occupation, houses equipment, etc. It is usually represented in spreadsheet or flat file formats. The most commonly used spreadsheet format is represented in many occasions with a "nice" view that includes different levels of nesting or mixed information. For example, Fig. 1 shows data about different types of housing distributed by districts. However, the three first rows are nested and the data in the first column represents provinces (in bold with a blank row after), counties (in bold), and district (that follow each county).



Provincia, Cantón y Distrito	Viviendas				
	Total	Viviendas Individuales			Viviendas Colectivas
		Total	Ocupadas	Desocupadas	
Costa Rica	**1 360 046**	**1 359 178**	**1 211 974**	**147 204**	**868**
San José	**436 032**	**435 762**	**400 881**	**34 881**	**270**
San José	**86 946**	**86 884**	**81 754**	**5 130**	**62**
Carmen	1 257	1 253	1 042	211	4
Merced	3 921	3 912	3 670	242	9
Hospital	5 590	5 577	5 174	403	13
Catedral	4 982	4 978	4 385	593	4
Zapote	6 348	6 341	5 913	428	7

Fig. 1. Spreadsheet that requires transformation for integrating with spatial data

[1] Known as features in the GIS terminology.

Therefore, this data requires transformations before merging with spatial data that can be difficult to perform by non-expert users.

Problems related to spatial and conventional data increase when the integration processes start. Unfortunately, data from shape files and other sources do not use the same nomenclature. The differences are at the level of (not) using accent marks, capitalization, hyphens - situations relatively easy to solve. Equally easy to address is the exclusion of some words, e.g., area, from one origin or inclusion into another origin. Other problems may be more complex, but still possible to solve automatically, e.g., using fuzzy lookup, fuzzy search, among available options; however, many problems to solve may require a tedious manual inspection.

It should be clear that if the data providers understand the aspects of data quality and the problems of data integration, if they could have access to gazetteers and nation-wide dictionaries and understand that some formats could be easier to use for merging data, programming effort will decrease and data could be used more frequently. However, this is not always an easy task to accomplish and a significant effort and many years of development may be needed for creating a system as, for example, a TGOS (Taiwan Geospatial One-Stop) system to integrate data and reduce redundancy in data acquisition in order to facilitate data sharing and supply [10].

4.2 Assimilation of New Technologies

Changes in technology occur at very fast pace, however, global development not always has a direct effect on adopting new technologies in developing countries. Costa Rica may be considered the Central American “Silicon Valley” with the potential of becoming the leading producer of ICT per capita in Latin America [2]. Furthermore, the presence of many different high-tech international companies, e.g., Intel, Acer, Panasonic, HP, Siemens, Hitachi, among others, and investment of US$470 million in 2011 by this kind of companies [3] fortifies the Costa Rican position. However, the users’ preparation in ICT, especially in public institutions, not always goes at the same pace as the production sectors.

The above-mentioned project developed by the Cadaster, in its third stage, trained municipality staff in handling spatial and conventional data from the Cadaster. 220 users from municipalities around Costa Rica were trained during 165 hours in using this complex system with the purpose of improving municipalities’ revenues by applying more control over the land property taxes. Only geo-experts (surveyors, cartographer, and geographer) were trained, leaving non-expert users, who are responsible for decisions, without direct access to this data and use it in tools that do not require geo-expertise. Even though the person responsible for this program ensures that trained staff become the strategic advisors for the city council during the process of making decisions on land use [17], there are several issues to address: (1) geo-experts highly rely on GISs and do not consider other kinds of tools as mentioned in Sec.3, (2) only data from the Cadaster is used, (3) the integration of spatial and conventional data from different sources is difficult to achieve, (4) the vision of using data is limited to the control of taxes, and (5) there is a dependency from geo-experts that prevents decision-making users from analyzing directly data according to their needs.

Furthermore, occasional training does not stimulate changes and searchers for improvement. On the contrary, it can promote concerns due to the complexity of the system, which requires a high level of expertise and many hours of training. This is clearly in contrast with the global tendency that the complexity of tools and systems is hidden from non-expert users allowing them to be confident in looking for new data, convert it into new knowledge, and use it effectively for solving the issues or problems at hand, i.e., improving the information literacy in a country.

4.3 Expanding a Vision for Education

To provide different analysis options using spatial and conventional data by non-expert users, geomatics specialists are required. They should combine the basic geo- and informatics-knowledge to prepare the necessary platform for publishing data for analysis purposes. However, in many developing countries, there is still a gap between geo- and informatics-knowledge. Many courses in different geo-fields do not consider advances in informatics, leaving to the initiative of the students to explore them. On the other hand, informatics experts are not very skilled in applying their knowledge to spatial data handling, since spatial extension to current DBMSs is a relatively new technology. This lack of knowledge about spatial data features among informatics experts makes the communication process between them and geo experts a difficult task. As a consequence, even though currently the GIS and DBMS systems are clearly converging technologies, the informatics and geo knowledge does not merge at the same pace in order to face upcoming technological and social changes. This necessity of expanding knowledge related to spatial databases among different fields is clearly shown in several sources, e.g., [18, 19]. However, this requires preparing geoinformatics or geomatics experts who are willing to integrate knowledge from both fields: geo related (geographers, cartographers, surveyors, etc.) and informatics as already done in several universities around the world, e.g., Laval University in Quebec, Canada, California State University, USA, Politécnico de Milano, Italy.

The lack of a geomatics vision of data reuse among specialists from different fields forces institutions/organizations to incur in high cost of digitalizing the same territory using different scales and geometries to represent spatial extents and create separate systems. As mentioned before, travel agencies in Costa Rica could not obtain the data about the beaches' sizes or about the length of the landing tracks in airports, since the shape files were created by separate institutions unrelated to tourism, representing these spatial objects as points. Another example is the use of lines for spatial representation of risk zones that do not allow users to check whether other objects (schools, clinics, hospitals, etc.) are within these zones. Although the solution for using different spatial data types to represent the same object already exists in spatial databases referring to as multiple representations, e.g., [1], it is not widely known, particularly among geo experts.

A geomatics view was applied for developing spatial databases and spatial data warehouses for a Costa Rican institution using a set of 40 different shape files. Before the development of spatial databases, different geo-expert users manipulate subsets of these shape files according to their specialties. For example, meteorologists use the data related to weather, geologists exploit the shape files related to geomorphology, the school administration handle the data of different kinds of educational centers, health institutions focus on clinics and hospitals, and Ministry of Transportation looks for data about roads. These different shape files were located in separate systems and were difficult to share among interested parties.

Several projects were developed using different combinations of shape files to show the possibility of cross-spreading knowledge among different entities, e.g., related to geology and emergencies or transportation and river distribution, and to create new knowledge and awareness for preventing risks and taking actions for better planning. One of the projects considered many layers that at the beginning were not taken into account to be exploited together. We used the shape files of the fire stations, fire risk areas, road network (its extension and coverage), location and capacity of clinics and hospitals, as well as distribution of Costa Rica into districts, counties and provinces. We extended this spatial data with conventional data about housing and population for each district, as well as age range and gender. These data allowed users to create different analysis scenarios to determine if the number of fire stations is adequate for the amount of people that must be attended in the case of a fire. Also, this application helped determine several other aspects, e.g., the proximity of other fire stations in the case where the additional assistance is required or the locations of hospitals and clinics and the type of road coverage to access them in the case of emergencies.

4.4 Developing a New Social View

Improving the information literacy level does not mean including enough data and new technology to manage it. In many countries, it might represent the changes at a social level. High data availability may open the possibilities to see the aspects that society does not want to make visible, e.g., part of the country with high illiteracy or poverty levels, high crime rates, poor road systems, many risk zones, etc. Not always this data is welcome since making it visible requires an effort from corresponding entities to accept the situation and to take actions in order to solve it or improve it. Furthermore, in many developing countries the different decisions may be made based on political plots of events or personal interests of stakeholders. Therefore, the challenge to overcome may require changes in the society that could not be easy to make and could take a significant amount of time.

In addition, users locally responsible for the administration of territory, e.g., members of municipalities, should be able to not only query data, but also modify it. This is necessary since spatial data may change over time frequently, e.g., parcels split or merge, the land corrosion or other natural disasters can change land characteristics. Local administrative officers become aware of changes earlier than central governmental institutions and, therefore, the data needed to make decisions could be up-to-date without having to incur in expensive loans to modify nation-wide spatial data. This fact requires changes in the government and system implementers' mentality. The inclusion of different users that modify data in the interest of a specific community is not a new approach. It is already present in different Internet sites, e.g., Wikipedia or OpenStreetMap, and in the case of spatial data it is known as Crowdsourcing, Volunteered, or Collaborative Geographic Information Systems (Services) [16].

Participation of different users for incorporating more local data activates people awareness about the environment and neighborhood. In this way, local awareness may expand to nation-wide awareness and could help speed-up the development process. This could help to have more independent countries that make better decisions in local development that is in harmony with global trends.

4.5 Leaderships of Governmental Entities

The culture of sharing data and using it for decision-making processes only develops and expands when the government establishes local standards based on, or adapted from, already existing international standards [7, 11] for data acquisition and sharing. This issue is particularly important for spatial data due to different forms of referencing, collecting, and representing objects on the Earth. In this way, it would be possible to avoid the process of digitalizing the same territory by different institutions that is not only an expensive endeavor paid by citizens, but also a demanding task in terms of technology and human resources.

Furthermore, a governmental leadership should inspire the coordination among industrial, official, and academic circles [10] to envisage and solve may unexpected problems that might arise while building a SDI. This is a complex task and requires willingness and awareness of all parties involved.

5 Conclusions

New emerging technologies and increasing volume of available spatial and conventional data give opportunities for using this data in the decision-making processes. In order to achieve this and to create the high level of information literacy, i.e., "...the ability to know when there is a need for information, to be able to identify, locate, evaluate, and effectively use that information for the issue or problem at hand." [9], adequate infrastructure and standards must be utilized. A Spatial Data Infrastructure (SDI) that includes nation-wide spatial and related conventional data could be seen as a platform that qualifies for this purpose. However, building SDI in developing or newly industrialized countries is still a difficult task that requires society to overcome several challenges in order to succeed.

Challenges related to data quality and integration can be solved establishing a clear governmental strategy about data publications and consumptions based on nation-wide standards. Another challenge is a real assimilation of new technologies by the staff responsible for decisions at different administration levels. They should be able to use systems and request the necessary data without having high technological skills. This requires changes in education, preparing geo-specialists who are able to merge geo- and informatics-knowledge and hide system complexities for non-expert users. However, data and infrastructure are only the facilitators; the real challenge in developing countries is to change the social perception about the importance of having high quality data and use it for decision-making processes.

References

1. Bédard, Y., Bernier, E.: Supporting Multiple representations with Spatial Databases View: Management and the Concept of "VUEL". In: Proceedings of the Joint Workshop on Multi-Scale Representations of Spatial Data (2002)
2. Luciano, C.: Promoting Silicon Valleys in Latin America: Lessons from Costa Rica (Regions and Cities). Routledge (2012)
3. Daily News: Inside Costa, http://www.insidecostarica.com/dailynews/2012/january/19/costarica12011904.htm (accessed 2012)
4. Delgado, T., Crompvoets, J.: Infraestructuras de Datos Espaciales en Iberoamérica y el Caribe. Project CYTED-IDEDES 606P10294 (2006), http://redgeomatica.rediris.es/idedes/IDEs_en_Iberoamerica.pdf
5. ESRI. ArcGIS for Desktop Extensions, http://www.esri.com/software/arcgis/about/desktop-extensions (accessed August 2012)
6. GSDI: Global Spatial Data Infrastructure, http://www.gsdi.org/ (accessed August 2012)
7. ISO/TC211, Geographic information/Geomatics (2010), http://www.isotc211.org/ (accessed August 2012)
8. Miller, H., Han, J. (eds.): Geographic Data Mining and Knowledge Discovery. Taylor & Francis (2002)
9. National Forum on Information Literacy. What is NFL? http://infolit.org/about-the-nfil/what-is-the-nfil/ (accessed August 2012)
10. NGIS Taiwan, Overview: Evolution, http://ngis2.moi.gov.tw/2008ngis/eng2/7_detail.aspx?sn=20081126142415265 (accessed August 2012)
11. OGC. Standards and Specifications (2010), http://www.opengeospatial.org/standards
12. OMB (Office of Management and Budget). Circular No.A-2, http://www.whitehouse.gov/omb/rewrite/circulars/a016/a016_rev.html (accessed 2010)
13. PCGIAP, Permanent Committee on GIS Infrastructure for Asia & the Pacific, http://www.pcgiap.org/wgroup/bbs/115 (accessed August 2012)
14. Rivest, S., Bédard, Y., Proulx, M., Nadeau, M., Hubert, F., Pastor, J.: SOLAP Technology: Merging Business Intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. ISPRS Journal of Photogrammetry & Remote Sensing 60(1), 17–33 (2005)
15. Smits, P.C., Düren, U., Østensen, O., et al.: INSPIRE: Infrastructure for Spatial Information in Europe, position paper. European Commission. Joint Research Centre (2002)
16. Sui, D., Elwood, S., Goodchild, M. (eds.): Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice. Springer (2013)
17. UE (Unidad Ejecutora). Catastro, Costa Rica, http://www.uecatastro.org/ (accessed August 2012)
18. Worboys, M., Duckham, M.: GIS: A Computing Perspective. CRC (2004)
19. Yeung, A., Hall, G.B.: Spatial Database Systems: Design, Implementation, and Project Management. Springer (2007)

Using Facebook to Better Engage College Students in Learning

Wei-Bin Lee[1], Chun-Wen Teng[2], Nien-Lin Hsueh[1], and Yung-Hui Li[1]

[1] Department of Information Engineering and Computer Science, Feng Chia University
{wblee,nlhsueh,yhuili}@fcuoa.fcu.edu.tw
[2] Graduate Institute of Public Policy, Feng Chia University
cwteng@fcuoa.fcu.edu.tw

Abstract. When more and more people integrate Social Network Services (SNS) into their daily life, digital native students were more accustomed to communicate via SNS. Research shows SNS is beneficial to some academic goals, but research also shows most teachers hesitate about adopting SNS to assist teaching. Comparing to why should we integrate SNS into education, teachers are more interested in how to integrate SNS into education. In this study, two fomal college class students(N= 120) are examined the possible situation of praticing facebook assisted teaching may encounter. We found that even with the high information exchanging frequency made by facebook notification features, it's still hard to increase students' interaction without course interaction strategy. We also found the potential of creating issue group instead of class group to continue subject learning between students and alumni.

Keywords: Facebook, Computer mediated communication, Student engagment, College education.

1 Introduction

According to the prediction of its growth trend, the number of active users of Facebook will reach one billion in 2012 (Jeff, 2012). More and more users treat Facebook as an essential part of their daily lives. At the same time, more and more corporates treat Facebook as an important marketing tool to communicate with their customers (Isabel and Kamila, 2010). Although in the past, there were studies to verify the positive impact of Facebook service in the realm of education, most teachers still hesitate about whether to use Facebook in their class (Reynol, 2011). As living in a modern society where internet technology integrates highly to aspects of life, and facing digital native students who think and learn in a way completely different, teachers need not to know "why to couple Facebook in teaching" but "how to use it". If we can properly follow the trend of technology and learning habit of students, smartly interact with them in an environment that they are used to, not only can we enhance the involvement of the students in learning, but also open the possibility for students to integrate what they learn with what they encounter in real scenario. In this study, we experiment in the scenario where teachers used Facebook service to assist teaching, in

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 403–408.
DOI: 10.1007/978-3-642-35452-6_41

order to find out what they can do with Facebook to help their teaching. The results of this study can be used as a foundation for later (more detailed) study.

2 Literature Review

According to the theory of Zone of proximal development (ZPD), which is proposed by Vygotsky, learners learn better through the proper iteraction with their colleagues in a cooperative environment (Vygotsky,1978). In this process of teaching, every teacher is looking forward to enhance the positive interaction among students so that they can learn more and better when they share what they know to each other. As mobile technology becomes more important in learning environment, it greatly enhances the convenience for teachers and students to interact with each other and exchange their knowledge.

Facebook, which is an internet service launched in 2004, and is going to reach its one billion users in 2012, is the most popular social platform in the whole world. It now incorporates the function of mobile notification, which greatly facilitates the convenience for user interaction. According to the past research, students are happy to incorporate Facebook function into their learning process (Roblyer er al., 2012). Although Facebook may not be the best choice in the realm of Computer-Aided-Learning (CAL), it is still an appropriate choice for the purpose of group communication (Ana & Elisabeth, 2012). Recent studies have shown that it enhances the level of involvement of students in class when Facebook is used to facilitate the teaching process in college classrooms (Reynol, 2011).When using Facebook to supporting formal course, group members had better communicating frequency (Ana & Elisabeth, 2012).

Teachers believe that by following the steps of their students and interacting with them in the environment they like, it is promising to enhance the level of involvement of their students in the class. Most teachers suspect that using Facebook in their teaching process will bring positive influence to students. Although there are many advantages of using Facebook and teachers themselves use Facebook in their private domain, only very few teachers officially use Facebook as a tool to assist teaching in class (Moran, Seaman & Tinti-kane, 2011).

3 Experiment

The subjects involved in this study are the college students who took one of the elective courses in department of information engineering and computer science at Feng Chia University, Taichung, Taiwan. The experiment runs from March to June, 2012. Two different classes were chosen to be experiment group and control group. Each class is assigned one Teaching Assistant (TA). Both classes have set up “close group” on Facebook for students to interact with teachers online, and some of the senior students who already took this course last year are invited to involve into the online interaction. In our setup, the level of involvement of the students in experiment group will be quantified and evaluated as part of their final grade, and specific rules are made to ask students to get involved. For students in control group, they have freedom to decide whether they want to join the online Facebook group or the level of involvement. TA of the experiment group was asked to throw out course-related topics once in a while, but there is no such requirement for TA of the control group (as shown in Table 1).

Table 1. Description of group settings

	Experiment	Control
Course Name	Introduction To Information Security	Software Quality Assurance and Software Test
Students	76	44
Group Members	89	62
FB Engagement	Required	Not Required
FB Grading	Yes	No
TA Operation	Yes	No

Table 2. Experiment group facebook engagement rule

	Original			New		
	Max Point	Rule		Max Point	Rule	
		Requirement	Point		Requirement	Point
Post	3	1 Original Post	1	2	1 Original Post	1
					Single Post 10 comments	1
Comments	3	1 Comments	1	4	1 Basic Topic Comments	1
		2~5 Comments	2		2~5 Comments	2
		Over 5 Comments	3		5~10 Comments	3
					Over 10 Comments	4
Like	1	Over 15 Likes	1	1	Over 15 Likes	1
Bonus						
Post		Single Post ½ class Liked	3		Single Post ½ class Liked	3
Comments		Over 20 Comments	4		Over 20 Comments	4

The level of online interaction of the students of the experiment group is rated according to four phases throughout the whole semester, with the score during each phase weights 7%. Students can get additional 7% bonus score if they were highly participated. Therefore, the total weight of the online interaction gets up to 35% of the final grade. The level of online interaction is mainly evaluated by the number of the message a student posts, the number that he/she responds or pushes "like" button to other students' posts. The details of the rule of grading are listed in Table 2. During experiment, in order to increase the number of message involved in online discussion, the rule might have slightly modified. The online interaction of the students in control group was not considered at all for their final grade.

4 Results and Discussion

Table 3 listed the number of message posted, the number of responses and the number of "liked" for both Facebook group during the entire experiment period. The significant difference across all metric between these two groups proves that the requirement of online interaction greatly enhance the frequency of posts. Furthermore, by analyzing the data we found that the number of interactions among students has exceeds the original requirement. There is a maximal bonus points students can get with online interaction; thus, if the goal of online participation is only for getting higher score, they can just participate as much as being required. Therefore, it is reasonable to infer that during the process of online interaction, some students established substantial relationship with each other, which causes them to interact with each other much deeper.

Among all members, the one that posts most frequently in experiment group is the class TA, which weight 18.4% among all posts. The one that posts most frequently in control group is the class teacher, which weight 49.2% among all posts. From Figure 1 we can see that the distribution of the number of posts for experiment group is higher and flater compared to the control group.

Table 3. Facebook engagement results

	Experiment	Control
Posts	396	120
Comments	2790	645
Likes	13208	551
Post members	66	16
Single member highest posts	27	59

We randomly chose students, TA or teacher and interview with them. During the interview, both TA and students said that they were eager to know what other students responded to their previous posts when they were notified such event happened by their mobile devices. When being asked about whether online interaction with students increases their work load, TAs said that they do not think it is a heavy load for them because they themselves are interested in the course material and they treat the online interaction as a mean of interchanging information with friends. Similar responses are collected from students too.

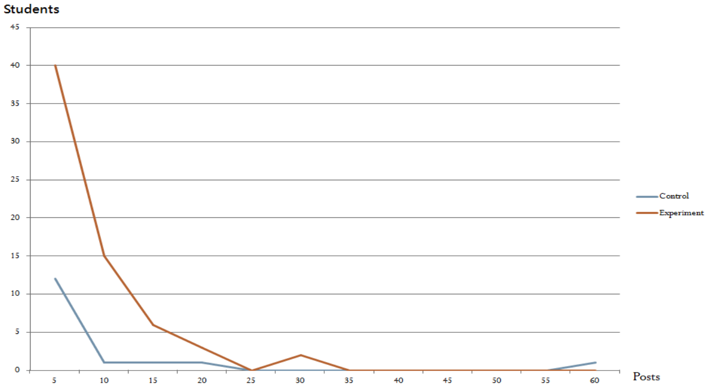


Fig. 1. Post distribution

During the experiment, for the control group, when it is getting to the second half of the semester, the number of online interaction approaches to zero. Even when the class teacher posted the summary of the keypoints of the class material, the number of "liked" and responses is also close to zero. For experiment group, the number and frequency of online interaction does not seem to drop through the whole semester. Sometimes when there is not enough time to discuss reference material, students would proactively share the related links on Facebook. After the semester is over, we found that the members of the experiment group continue sharing and discussing related topics on Facebook, rather than immediate dismiss. Although the number and frequency of online interaction is not as high as during the semester, some of the members still keep participating.

Based on such experimental results, we suggest that when coupling Facebook into teaching and learning activities, it is better to run the online group if the coordinator treats it as a specialized club of course-related domain knowledge. It is better to get students interested in the material and not being forces by the grading policy. It is good if it is possible to invite senior students who already took the course or department alumni to join the online group so that students have chance to further explore the potential application of what they learn during the class.

5 Conclusion

The goal of this study is to explore how to use Facebook to enhance the level of involvement of students in learning activities. Based on the experimental results, even if students use Facebook every day and see the new posts in the online group, when the group manager does not run the group properly, students will not be motivated to actively join the interaction. However, if proper strategy for managing the online group is used, such as providing a guideline, stimulating students to get involved, and helping them to link what they learn to field applications, it is highly possible to help them to learn and involve more throughout the whole semester. Taking advantage of the nature of knowledge sharing of Facebook services, it is also possible to synchronize the knowledge learning and field applications. But in order for that to happen, further experiment needs to be performed. This study is a pilot one, and we will go further in more detailed design of learning strategy in the future.

References

1. Coric, A., Pergler, E.: Team Communication via Facebook: Success or Failure. In: Central European Conference on Information and Intelligent Systems (2012)
2. Chang, C.Y., Sheu, J.P., Chan, T.W.: Concept and design of Ad Hoc and Mobile classrooms. Journal of Computer Assisted Learning 19, 336–346 (2003)
3. Isabel, A.Q., Kamila, B.: Marketing and Facebook: How Fashion Companies Promote Themselves on Facebook. Department of Business (2010)
4. Hughes, J.: Facebook projected to hit 1 billion active users by August. Digital Trends (January 12, 2012), `http://www.digitaltrends.com/international/facebook-projected-to-hit-1-billion-active-users-by-august/#ixzz28tYykJGh`
5. Moran, M., Seaman, J., Tinti-kane, H.: Teaching, learning, and sharing: How today's higher education faculty use social media. Research report published by Pearson, The Babson Survey Research Group, and Converseon (2011), `http://www3.babson.edu/ESHIP/research-publications/upload/Teaching_Learning_and_Sharing.pdf` (retrieved July 20, 2011)
6. Junco, R.: The relationship between frequency of Facebook use, participation in Facebook activities, and student engagement. Computers & Education (2011)
7. Vygotsky, L.S.: In: Cole, M., John-Steiner, V., Scribner, S., Souberman, E. (eds.) Mind in Society: the Development of Higher Psychological Processes. Harvard University Press, Cambridge (1978)

IP Address Exchanging Scheme for Vehicle Ad Hoc Networks

Chiu-Ching Tuan and Yi-Chao Wu

Graduate Institute of Computer and Communication Engineering,
National Taipei University of Technology, Taiwan, R.O.C.
{cctuan,t5419001}@ntut.edu.tw

Abstract. In vehicle ad hoc networks (VANET), vehicles require to process a unique IP address to access the Internet. While the number of vehicles across the different service areas is increased, the time and loading for IP addressing are increased. Moreover, the conflict of IP addressing or the invalid IP address is also increased. Therefore, an IP address exchanging scheme (IPAES) in VANET is proposed in this paper. In IPAES, it passes its IP address to another vehicle and receives a new IP address by another vehicle directly. Hence, IPAES could reduce the time and loading of IP addressing. Simulation results showed that IPAES had less time and loading of IP addressing. The time and loading of IP addressing do not increased once the number of service areas or number of vehicles is increased. Moreover, the time and loading of IP addressing are decreased while the moving speed of vehicles is increased.

Keywords: Vehicle ad hoc networks, IP addressing, IP address exchanging, Service Area, Moving Speed.

1 Introduction

Vehicle ad hoc networks (VANET) was composed of on-broad unit (OBU) of intelligent vehicle and roadside units (RSU). The bandwidth of VANET is allocated from 5.850 GHz to 5.925 GHz with 7 channels by Federal Communication Commission (FCC) in 1999. In VANET, the communication is divided into inter-vehicle communication (IVC) and roadside-to-vehicle communication (RVC). In 2004, IEEE 802.11p is proposed for VANET which the data rate is from 3 to 54 Mbps and the maximal communication radius is 1000 meter [1].

Most of researches in VANET focused on data transmission among vehicles in IVC. Hence, improving the current routing protocols in mobile ad hoc networks (MANET) and proposing a new routing protocol are the important issues in VANET [2]. However, some vehicles may require the information from the Internet [3-5]. Thus, the vehicles not only need to build a multi-hops path, such as AODV [6] or ROMSGP [7], but also require a unique mobile IP address to access the Internet, as shown in Fig. 1.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 409–418.
DOI: 10.1007/978-3-642-35452-6_42

Differed from the wire network, it is impossible to assign a static unique IP address to each mobile host in MANET, since the mobile host requires a dynamic IP address due to moving in different network areas. Thus, many dynamic IP addressing schemes are proposed to assign a unique dynamic IP address to a mobile host [8-10]. However, these schemes of MANET could not be applied for VANET, since the speed of vehicle is much faster than other mobile hosts [12-13]. Moreover, the variation of density in VANET is higher than that in MANET. Therefore, we proposed an IP address exchanging scheme (IPAES) for VANET in this paper.

Simulation results demonstrated that the efficiency of IPAES. The rest of the paper is in the following sections. Section 2 presented the related work. Section 3 stated IPAES. Section 4 presented the simulation results. Section 5 concluded this paper.

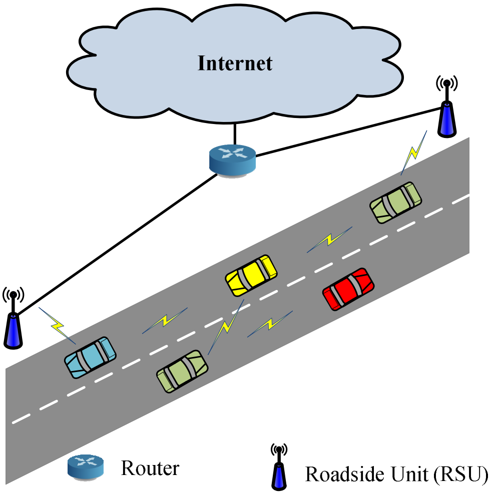


Fig. 1. Access the Internet in VANET

2 Related Work

IP address allocation is divided into static and dynamic IP addressing. In static IP addressing, each host needs to set the IP address, network mask, gateway address, and DNS address [13]. Thus, it is easy to manage for few of hosts, but it is hard for large number of hosts. Since the number of IP address in IPv4 is limited, static IP addressing is not applied for large number of hosts. In MANET, IP address in each host may be invalid once the host is moving. Therefore, many dynamic IP addressing schemes are proposed for MANET [8-10]. However, these schemes could not be used directly for VANET, since the moving speed of vehicle is faster and VANET has additional roadside units (RSU) to help to assign IP address for vehicles [11-12].

In centralized address configuration (CAC) [14], RSU acts as a relay agent between vehicles and DHCP server. Each vehicle could connect to the nearest RSU to get a unique mobile IP address with a limited time. CAC adjusts the limited time to reduce the conflict of IP addressing in each round. Hence, CAC could be applied for different moving models compared with VAC. However, each vehicle needs to broadcast four packets for IP addressing, as shown in Fig.2. Between DHCPCOVER and DHCPOFFER, a complex process, such as duplicate address detection, DAD, must be generated. Hence, the time of IP addressing is increased. Since IP addressing of CAC are responsible by RSU, loading of IP addressing is also increased. Therefore, the conflict of IP addressing and the invalid IP address are prone to be happened.

To address the issues of CAC, a mobile IP address exchanging scheme (IPAES) is proposed in this paper. In IPAES, RSU still acts as a relay agent of DHCP server, but RSU is divided into border RSU and inner RSU to manage the number of vehicles and the direction of vehicles. Hence, the insufficient IP address in CAC could be solved. The detail of IPAES is described in session 3.

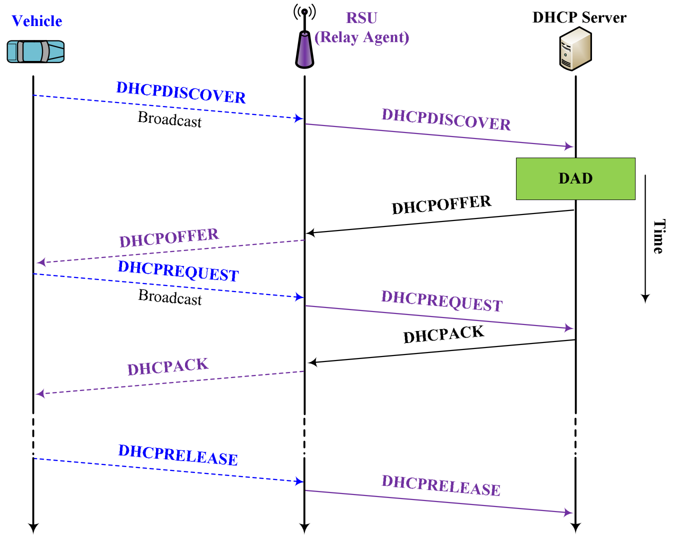


Fig. 2. CAC

3 IP Address Exchanging Scheme

In IP address exchanging scheme (IPAES), RSU is deployed in the border of service area, such as the intersection of road. $RSS_i^{current}$ denoted that a vehicle *i* (V_i) receives a received signal strength from RSU in the current service area. RSS_i^{out} denoted that V_i receives a received signal strength from RSU out of the current service area. Once $RSS_i^{current}$ and RSS_i^{out} are increased, it means that V_i will move out the current service. V_j broadcasts a HELLO packet to other vehicles within the transmission range. Once V_j in the different service area received the HELLO packet from V_i, it sends a GATE packet to V_i. If V_i receives the GATE packet in a predefined time, V_i and V_j exchange their IP address by sending a GARP packet; otherwise, V_j releases its IP address to the current RSU and receives a new IP address from another RSU by sending DHCPDOSCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK packets. Since IP address exchanging is happened between two vehicles without through RSU, IPAES could reduce the time and loading of IP addressing. The conflict of IP addressing and the invalid IP address also could be reduced.

For example, V_1 receives RSS_1^1 and RSS_1^2 , and broadcasts a HELLO packet, as shown in Fig. 3. Since there is no vehicle received the HELLO packet, V_1 releases its IP address to RSU$_1$ and receives a new IP address from RSU$_2$ after a predefined time. V_1 receives RSS_1^1 and RSS_1^2 , and broadcasts a HELLO packet. In this moment, V_2 receives the HELLO packet and sends a GATE packet to V_1. Since V_1 receives a GATE packet, it sends a GARP packet to RSU$_1$ and exchanges its IP address with the IP address of V_2, as shown in Fig. 4. The flow chart of IPAES is shown in Fig. 5.

In IPAES, while a vehicle could not receive an IP address from another vehicle by IP address exchanging, it needs to receive an IP address from RSU similar to CAC. In this case, four packets are needed, such as DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK. The size of these four packets is 1312 bytes. Moreover, the frequency of DAD may be higher to increase the conflict of IP addressing and the probability to receive an invalid IP address may be increased. Once a vehicle could receive an IP address from another vehicle by IP address exchanging, it only sends a GARP packet to RSU. Therefore, the time and the loading of IP addressing in IPAES are lower than those in CAC. The conflict of IP addressing and receiving the invalid IP address also could be reduced.

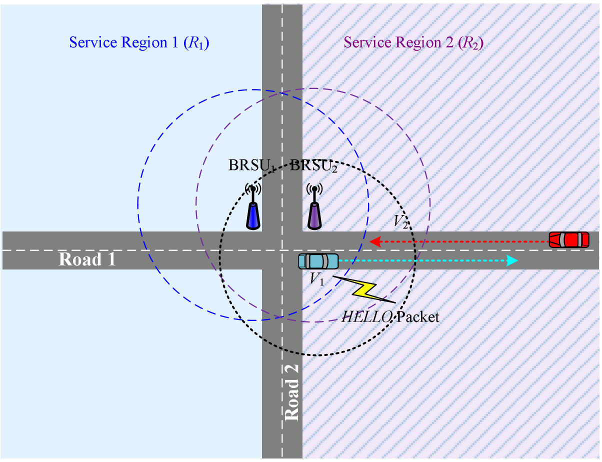


Fig. 3. IP Address Releasing and Assigning in IPAES

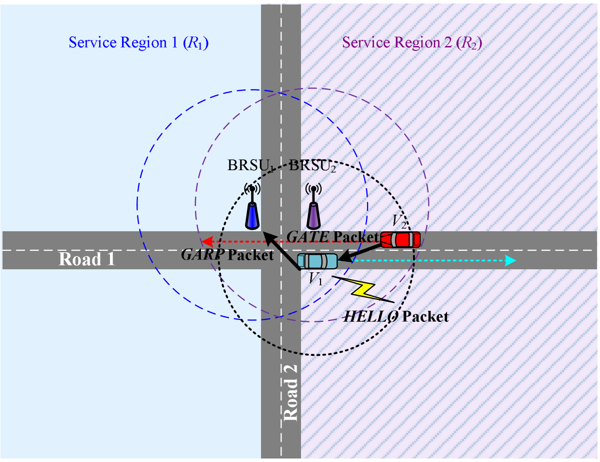


Fig. 4. IP Address Exchanging in IPAES

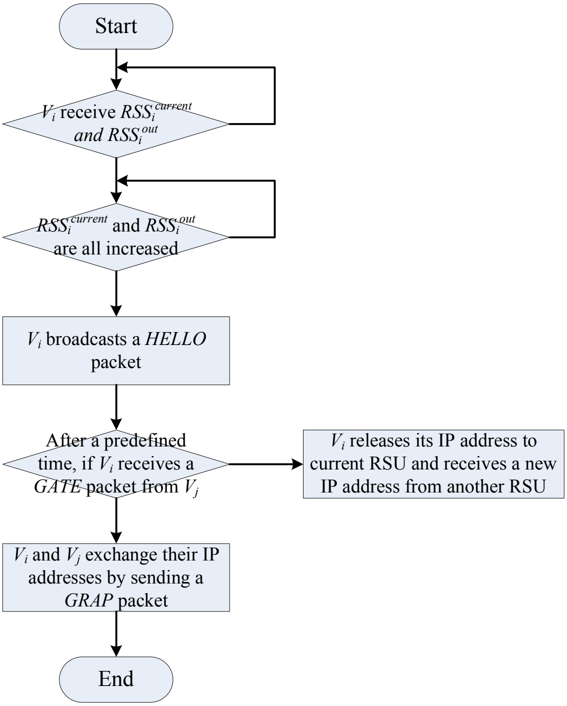

Start
V_i receive $RSS_i^{current}$ and RSS_i^{out}
$RSS_i^{current}$ and RSS_i^{out} are all increased
V_i broadcasts a *HELLO* packet
After a predefined time, if V_i receives a *GATE* packet from V_j
V_i releases its IP address to current RSU and receives a new IP address from another RSU
V_i and V_j exchange their IP addresses by sending a *GRAP* packet
End

Fig. 5. Flow Chart of IPAES

4 Simulation Results

The effectiveness of our approach is validated through simulation in a C++ custom simulator with the OMNeT++ 4.1 platform [15]. This section describes the simulation environment, performance metrics, and experimental results, respectively. The simulation parameters are listed in Table 1.

Table 1. Simulation parameters

Factor	**Value**
Simulation area	4000 × 4000 (m^2)
Number of service areas (N_s)	2, 4
Number of vehicles (N_v)	100, 250, 500, 750, 1000, 1500
Moving speed (V)	40-50 (km/hr)
Number of RSU	18, 29
MAC	802.11a
Data rate	6 (Mbps)
Communication radius of vehicle	150 (m)
Communication radius of RSU	250 (m)
DHCPDISCOVER packet size	328 (bytes/packet)
DHCPOFFER packet size	328 (bytes/packet)
DHCPREQUEST packet size	328 (bytes/packet)
DHCPACK packet size	328 (bytes/packet)
GARP packet size	28 (bytes/packet)
Simulation time	30 (minutes)

4.1 Performance Metrics

In the performance metrics, evaluate the average time of IP addressing (T_a) and the loading rate of IP addressing (R_L) in RSU under IPAES and CAC. Since CAC does not applied for IP address exchanging, the rate of IP address exchanging (R$_{ex}$) is only evaluated in IPAES. For T_a, it denotes the total time of receiving the new IP address divided by total vehicles as (1). T_i denotes the time of IP addressing by V_i moving across the different service area. It is better, while T_a is decreased.

$$T_a = \frac{\sum_{i=1}^{Nv} T_i}{N_v} \tag{1}$$

R_L in RSU denotes the difference between R_L of CAC and that of IPAES divided by R_L of IPAES as (2). S_{DHCP} denotes the total length of sending packets without IP address exchanging. S_{GARP} denotes the total length of sending packets by IP address exchanging. In this simulation, the length of S_{DHCP} is 1312 bytes included *DHCPDISCOVER*, *DHCPOFFER*, *DHCPREQUEST*, and *DHCPACK* packets. The length of S_{DHCP} is 28 bytes, such as *GARP* packet. P_i denotes the ratio of the times of IP address exchanging to that without IP address exchanging for V_i. While R_L is lower, the loading of RSU is decreased.

$$R_L = \frac{\sum_{i=1}^{i=Nv}(1-P_i)\times S_{DHCP} + \times P_i \times S_{GARP}}{N_v \times S_{DHCP}} \quad (2)$$

Rate of IP address exchanging (R_{ex}) denotes the times of IP address exchanging divided by the times of all IP addressing as (3), where N_{ex} denotes the number of vehicles which receive new IP address by IP address exchanging.

$$R_{ex} = \frac{N_{ex}}{N_v} \quad (3)$$

4.2 Experimental Results

To evaluate T_a, we estimates T_a versus N_v. T_a of IPAES is less 29% than that of CAC, as shown in Fig. 6. In Fig 6 we found that T_a is decreased while V is increased. It is caused that the probability of IP address exchanging is higher while V is higher. It proved that IPAES had less T_a than CAC.

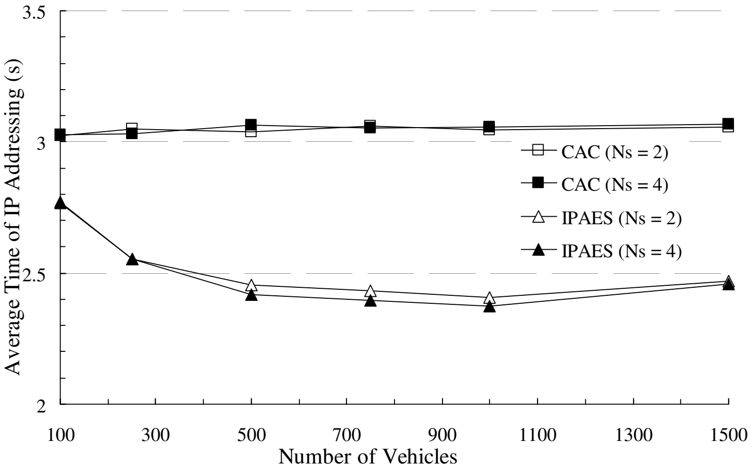


Fig. 6. Time of IP addressing

R_L of IPAES is 17% averagely, while N_v is more than 500, as shown in Fig. 7. Fig. 7 showed that R_L is increased while V is increased. It proved that IPAES had lower loading of RSU than CAC.

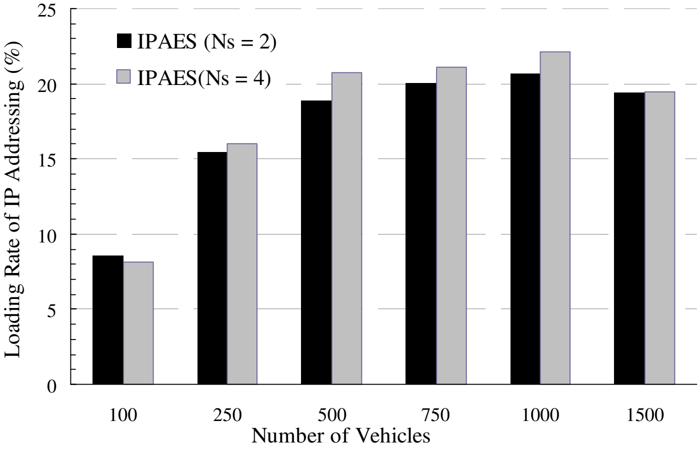


Fig. 7. Loading Rate of IP Addressing

R_{ex} of IPAES is 20% averagely, while N_v is more than 500, as shown in Fig. 8. It showed that R_{ex} is increased while V is increased. However, the rate of IP address exchanging in IPAES is still lower than that in CAC even if V is increased. It proved that IPAES had lower loading of RSU than CAC.

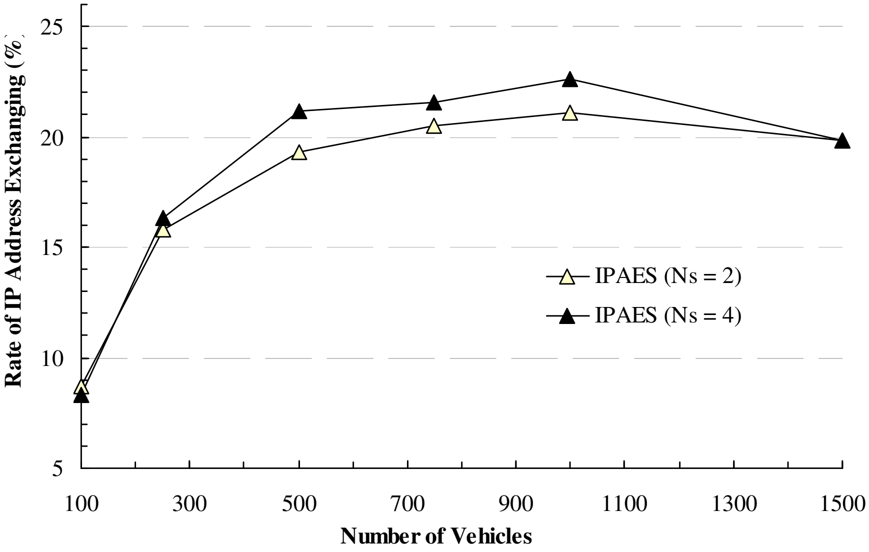


Fig. 8. Rate of IP Address Exchanging

In the simulation results, we found that IPAES had lower average time of IP addressing and loading rate of IP addressing. It proved that the conflict of IP addressing and the probability of receiving the invalid IP address could be reduced. Hence, vehicles could access the Internet more stably. Since the contribution of IPAES is IP address exchanging, we evaluate the rate of IP address exchanging in different number of service areas (N_s), number of vehicles (N_v), and the moving speed of vehicle (V). The simulation results show that R_{ex} does not decrease once N_s is changed. R_{ex} is increased while N_v or V is increased. From the simulation results, it proved that the IPAES had better performance than CAC.

5 Conclusions

In mobile ad hoc networks (MANET), IP addressing is an important issue since the mobile host needs a unique IP address to access the Internet. Among the proposals of IP addressing, IP addressing based on DHCP is the general solution because mobile hosts may move to another service area and require a new IP address to access the Internet. Vehicle ad hoc networks (VANET) is another kind of MANET. Since, in VANET, the moving speeding of vehicles is faster than that of mobile hosts in MANET and the power energy is unlimited. Thus, the proposals in MANET could not be applied directly. To address this issue, some IP addressing schemes in VANET. Among the IP addressing schemes in VANET, vehicular address configuration (VAC) and centralized address configuration (CAC) are the common solution. VAC is a distributed IP addressing scheme. However, VAC only be applied for linear topology and constant moving speed. Moreover, VAC could be applied for the large service area.

Hence, CAC is proposed. CAC is proposed based on DHCP. In CAC, each RSU is an agent of DHCP server. The communication is divided into inter-vehicle communication (IVC) and roadside-to-vehicle communication (RVC). Although CAC could be applied for varied topology, varied moving speed, and the large service area, the loading of IP address releasing and assigning focus on RSU. While the number of vehicles is increased or the moving speed of vehicles is varied, the conflict of IP addressing and the invalid IP address are increased.

To address these issues, a mobile IP address exchanging scheme (IPAES) is proposed. IPAES is based on CAC with a RSU deployment scheme and an IP address exchanging scheme we proposed. Hence, while a vehicle move out its current service area, IPAES starts the IP address reassignment or IP address exchanging by the received signal strength (RSS) from different RSU. Since IP address reassignment includes IP address releasing and assigning by DHCP, the average time and loading of IP address reassignment is more than those of IP address exchanging. In IPAES, the ratio of IP address reassignment to IP address exchanging is 75% to 25%. Hence, the average time and loading of IP addressing in IPAES is lower than those in CAC. Moreover, the average time and loading of IP addressing in IPAES do not increase while the number of vehicles or the number of service areas is varied. While the moving speed of vehicles is increased, the average time and loading of IP addressing are decreased. It proved that IPAES had better performance than CAC.

Since the moving model of vehicles affect average time and loading of IP addressing, we will investigate other moving models of vehicles and propose a more efficient IP addressing scheme.

References

1. Li, F., Wang, Y.: Routing in Vehicular Ad hoc Networks: A Survey. IEEE Vehicular Technology Magazine 2, 12–22 (2007)
2. Emmelmann, M., Bochow, B., Kellum, C.C.: Vehicular Networking: Automotive Applications and Beyond. John Wiley and Sons, NJ (2010)
3. Sharif, B.S., Blythe, P.T., Almajnooni, S.M., Tsimenidis, C.C.: Inter-Vehicle Mobile Ad hoc Network for Road Transport Systems. IET Intelligent Transport Systems 1, 47–56 (2007)
4. Chen, Y.-S., Cheng, C.-H., Hsu, C.-S., Chiu, G.-M.: Network Mobility Protocol for Vehicular Ad hoc Networks. In: IEEE International Symposium on Wireless Communications and Networking, pp. 1–6. IEEE Press, Budapest (2009)
5. Toor, Y., Muhlethaler, P., Laouiti, A.: Vehicle Ad hoc networks: Applications and Related Technical Issues. IEEE Communications Surveys and Tutorials 10, 74–88 (2008)
6. Perkins, C.E., Royer, E.M.: Ad Hoc on Demand Distance Vector Routing. In: IEEE International Symposium on Mobile Computing Systems and Applications, pp. 90–100. IEEE Press, LA (1999)
7. Taleb, T., Sakhaee, E., Jamalipour, A., Hashimoto, K., Kato, N., Nemoto, Y.: A Stable Routing Protocol to Support ITS Services in VANET Networks. IEEE Transactions on Vehicular Technology 56, 3337–3347 (2007)
8. Mohsin, M., Prakash, R.: IP Address Assignment in a Mobile Ad hoc Network. In: IEEE International Symposium on Military Communications, pp. 856–861. IEEE Press, California (2002)
9. Thoppian, M.R., Prakash, R.: A Distributed Protocol for Dynamic Address Assignment in Mobile Ad hoc Networks. IEEE Transactions on Mobile Computing 5, 4–19 (2006)
10. Kim, H., Kim, S.C., Yu, M., Song, J.K., Mah, P.: DAP: Dynamic Address Assignment Protocol in Mobile Ad-hoc Networks. In: IEEE International Symposium on Consumer Electronics, pp. 1–6. IEEE Press, Texas (2007)
11. Bugti, S.A., Xia, C., Li, W., Hussain, E.: Cluster based Addressing Scheme in VANET. In: IEEE International Symposium on Communication Software and Networks, pp. 450–454. IEEE Press, Xian (2011)
12. Chao, S.-J., Zhang, J.-M., Tuan, C.-C.: Hierarchical IP Distribution Mechanism for VANET. In: IEEE Symposium on Ubiquitous and Future Networks, pp. 349–354. IEEE Press, Jejuis land (2010)
13. Rooney, T.: IP Address Allocation. Wiley-IEEE Press, NJ (2011)
14. Mohandas, B.K., Liscano, R.: IP Address Configuration in VANET Using Centralized DHCP. In: IEEE International Symposium on Local Computer Networks, pp. 608–613. IEEE Press, Montreal (2008)
15. OMNeT++ Network Simulation Framework, http://www.omnet.org

Hierarchical Deficit Round-Robin Packet Scheduling Algorithm

Min-Xiou Chen and Shih-Hao Liu

Department of Computer Science and Information Engineering,
National Dong Hwa University, Hualien, Taiwan, R.O.C.
{mxchen,m9921080}@mail.ndhu.edu.tw

Abstract. In the paper, we propose Hierarchical Deficit Round-Robin (HDRR) packet scheduling algorithm for multiple classes of service to pro-vide a fair share of residual bandwidth with lower computation complexity. We design HDRR based on the Deficit Round-Robin packet scheduling algorithm, and can distribute bandwidth in proportion to weights of flows sharing the bandwidth, and share the unused bandwidth to backlogged flows according to the hierarchical architecture. In order to realize the performance differences between the proposed scheduling algorithm and the other related scheduling algorithms, we implement the pro-posed scheduling algorithm in our simulation. The simulation results show that the proposed algorithm provides well QoS guarantees while fully utilizing the system bandwidth and low computation complexity.

Keywords: Hierarchical Packet Scheduling, QoS, differentiated service, resource management.

1 Introduction

Supporting multiple classes of service with quality of service (QoS) guarantee for each user is a great challenge of computer network. In order to satisfy the requirements of the multimedia applications, the system has to provide some good resource management mechanisms. Among various resource management mechanisms, packet scheduling is the essential technique for providing multiple classes of service. Provided a minimum amount of bandwidth guarantee and a fair share of residual bandwidth are the two major objectives of the packet scheduling in computer networks. Many packet scheduling algorithms have been extensively designed based on the InterServ [1] or DiffServ [2] model, such as Weighted fair queueing (WFQ) [3], self-clocked fair queuing (SCFQ)[4], Worst-case fair weighted fair queueing (WF2Q) [5], Start-time fair queuing (STFQ) [6], Worst-case fair weighted fair queueing plus (WF2Q+) [7], Deficit Round-Robin (DRR) [8], and Surplus Round-Robin (SRR) [9].

Although these approaches [1-9] can pro-vide QoS guarantee for each flow, but less of them are implemented commercially due to the scalability problem. Thus, the packet scheduling algorithms designed based on the DiffServ mod-el received more attention. Moreover, traditional packet scheduling algorithms have two limitations:

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 419–427.
DOI: 10.1007/978-3-642-35452-6_43

1.limited resource management capability; 2. coupled delay and bandwidth allocation. Thus, it is important to design the packet scheduling algorithms to provide link sharing and different service classes simultaneously. The hierarchical packet scheduling algorithms [7][10-12] were proposed to provide a general and flexible framework to support hierarchical link sharing and traffic management for different service classes.

With a hierarchical packet scheduling algorithm, the unused bandwidth will be distributed following the hierarchy until it reaches the leaf nodes where there are some packets in their physical queues. WF2Q+ was a famous hierarchical packet scheduling algorithm, and can provide well QoS guarantee for each flow. Moreover, WF2Q+ also distributes the unused bandwidth in a hierarchical fashion. However, the computation complexity of WF2Q+ is very high, and may not be deployed commercially.

In this paper, we propose the Hierarchical Deficit Round-Robin (HDRR) packet scheduling algorithm based on the DRR packet scheduling algorithm. DRR is a famous packet scheduling algorithm that can share the resource based on the quantum size and deficit count, which can be used to provide QoS guarantee for each flow. The computation complexity of DRR is better than that of WF2Q, when the quantum size of each flow is larger than the maximal packet length.

Based on the DRR, we introduce the hierarchical packet scheduling architecture in order to distribute the unused bandwidth in a hierarchical fashion. According to our proposed algorithm, when the quantum size of each flow is larger than the maximal packet length, the computation complexity for each flow is O(1).

The structure of this paper is organized as follows. Section 2 proposes the HDRR scheduler algorithm. Section 3 presents the numerical results of HDRR scheduling algorithm. Finally, conclusions and areas of future research are given in Section 4.

2 Hierarchical Deficit Round-Robin Packet Scheduling Algorithm

The proposed scheduling algorithm, HDRR, is based on the DRR algorithms, which is known for its low computational complexity while maintaining reasonable fairness properties. Some major objectives were considered in the design of our schedulers:

- Supports differentiated QoS for each flow
- Makes efficient use of the network resource and provides high resource utilization.
- Distributes residual bandwidths to the backlogged users according to the hierarchical architecture.
- Isolates the well-behaved users from the ill-behaved users.

The system structure and algorithm of HDRR are modified from the DRR. In HDRR, the system determines the service amount of flow $i \in f$ is based on the quantum size, Qi, and the deficit count, DCi. At each round, the increased amount of DCi is its quantum size. The flow cannot transmit a packet at this round when the packet length is larger than its deficit count. Due to the reason that DCi cannot less than 0, the amount of Qi is very critical in the original DRR algorithm. In order to achieve O(1) per packet selection complexity, Qi must be larger than the maximal packet size. This constraint is impractical in the real system. The computation complexity of SRR proposed in [29] is also as O(1). SRR is derived from DRR, the major difference between these two scheduling algorithms is that the value of deficit count can be less 0. Although, in HDRR, the value of deficit count cannot be less 0, but the algorithm can be rewritten based on SRR with ease.

Moreover, in HDRR, the system capacity is C, and $\sum_{i\in f} DC_{\mathrm{i}} = C$. Let P_i^k be the kth packet of flow i, D_{xy} denotes a yth group in xth layer of HDRR, and Flow(D_{xy}) is the associated flow of group D_{xy}. P_i^{top} denotes the first packet in the queue of flow i and length(P_i^{top}) is the packet length of P_i^{top}. $STFV_i$ is the normalized short-term fairness value of flow i.

The system architecture is shown in Figure 1. In each D_{xy}, the structure of group contains prev_group, next_group, child_group_list, Boolean_table, parent_group, shared_leaf_group_id, and associated_node_ptr. prev_group and next_group denote the link to previous sibling group and next sibling group of D_{xy}, respectively. child_group_list denotes the set of child group of D_{xy}. Boolean_table used to indicate backlog or non-backlog leaf groups, and parent_group is linked to the parent group. leaf_group_list contains all the leaf group of the group. shared_leaf_group_id be indicated the previous flow that is allocated residual quantum, and associated_node_ptr is linked to the associated node in SQL.

There are three major lists in HDRR. Backlog List (BL) is a list to contain all backlog flows. Non-backlog List (NL) contains all non-backlog flows. Shared Quantum List (SQL) is used to share the residual Quantum. At initial, SQL=NL. The node structure in these list contain q_{share}, which denotes the residual quantum, flow_list, which has all the non-backlog flows, group_id and associate_group, which is link to associated group.

When a packet P_i^k arrived, if queue of G(i) is empty, the node i will be moved from NL to BL, Boolean_table of G(i) is set as 1, and performs Set_associate_node(i).

When a packet P_i^k departed, if queue of G(i) is empty, the node i will be moved from BL to NL, Boolean_table of G(i) is set as 0, and performs Set_associate_node(i).

The Set_associate_node(i) function is used to set the group i's associated_node_ptr to associated node in NL or BL, and set the associated node's associate_group to group i.

The main algorithm is shown in Figure 2. For example, suppose that all the quantum size of flows in Figure 1 is 1, the order of BL is D31, D33, D30, and D35, and the order of NBL is D32, D37, D38, D36, D39, and D34. At initial, SQL=NL. Thus, the order of SQL is D32, D37, D38, D36, D39, and D34. According to the algorithm shown in Figure 2, the deficit count of the nodes in the BL will be added its quantum size. Next, the nodes in the SQL will distribute its quantum size in order. So, D32 will be retrieved from SQL, and the system finds the sibling node, D30 is at BL. Thus, qshare=q_{32}-q_{30} and $STFV_{D30}$ = $STFV_{D30}$ +1. Because that qshare =0, the system doesn't find the sibling node of D32, and the next node, D37 is retrieved from SQL. the system finds the sibling node, D36 is at NBL. Thus, the system will retrieve D36 from SQL, combines D36 and D37 to D22, and inserts D22 into SQL.

Next, D38 will be retrieved from SQL, combines D39 into D23, and inserts D23 into SQL. At that time, the order of SQL is D34, D22, and D23. Thus, D34 is retrieved from SQL, and the system finds the sibling node, D33 is at BL. Thus, qshare=q_{34}-q_{33} and $STFV_{D33}$ = $STFV_{D33}$ +1. Because that qshare =0, the system doesn't find the sibling node of D34, and the next node, D22 is retrieved from SQL. Because that D22's sibling group, D23, also is in SQL, D22 and D23 will be combined into D11, and D11 is inserted in to SQL.

Finally, D11 is retrieved from SQL, and qshare = qD11 = qD36+ qD37+ qD38+ qD39. The qshare distributed order is D30, D31, D33, and D35. The qshare distribution method doesn't need to consider the virtual time, and the computation complexity for n flow is O(n). Moreover, in order to provide fair share of qshare, we introduce the normalized short-term fairness value, STFVi, to limit the distribution amount for each backlogged sibling flow. The updated equations of normalized short-term fairness value are shown in the following, when qshare>0:

$$STFV_i = \begin{cases} STFV_i + 1 & \text{If } q_{share} \geq q_i \\ STFV_i + q_{share} / q_i & \text{Otherwise} \end{cases}$$

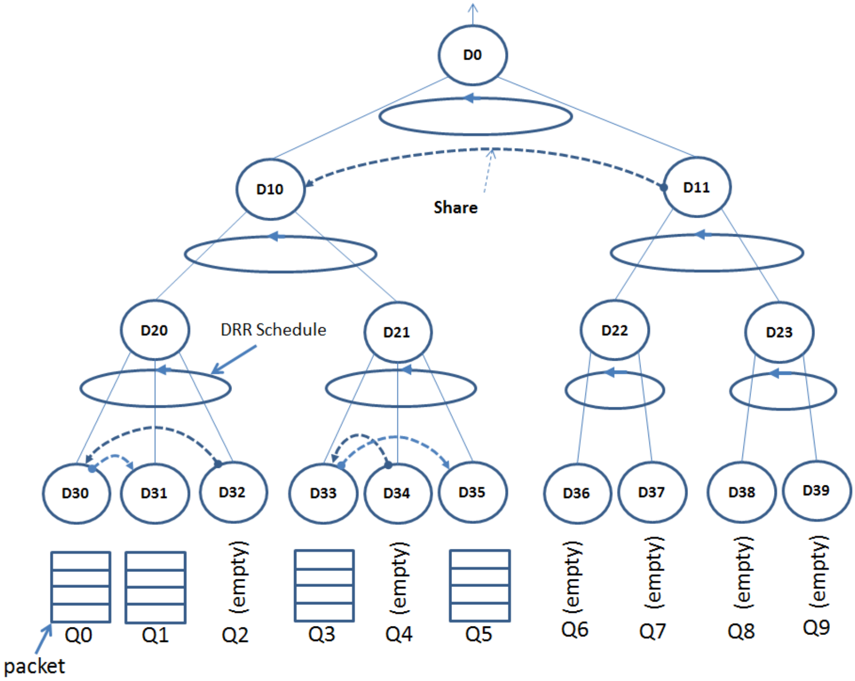


Fig. 1. System Architecture

In addition, in order to provide long-term fairness, we use the shared_leaf_group_id to indicate the previous flow that is allocated residual quantum, and the system can select the next shared flow based on the shared_leaf_group_id and leaf_group_list.

```
Initial: SQL=NL
Step 1: Serves all the flow in BL.
Step 2: Selects a node a from SQL. Let's gid deontes the node's group id and pidid = parent_group(gid).
    Step 2-1. If next_group of gid is not in SQL, and pid's the next shared_leaf_group_id is k-1, selects next flow k and updated shared_leaf_group_id.
        if $q_{share} >= q_k$, $q_{share} = q_{share} – q_k$, $DC_k = DC_k + q_k$, $STFV_k = STFV_k + 1$.
        Else, $DC_k = DC_k + q_{share}$, $STFV_k = STFV_k + (q_{share}/q_k)$ , $q_{share} = 0$
        If $DC_k >$ length($P_k^{top}$) , $DC_k = DC_k$ - length($P_k^{top}$), and remove $P_k^{top}$ from queue of flow $_k$.
        If $q_{share} > 0$, return Step 2-1. Else, go to Step 3.
    Step 2-2. If next_group of gid is in SQL, and its group id is k,
        $q_{gid's\ share} = q_{gid's\ share} + q_{k's\ share}$, adds all the flow in k's flow list into gid's flow list. Removes node k from SQL.
    Step 2-3. get next_group and return to Step 2-1 until all the next_group are considered. If $q_{gid's\ share} > 0$, group id=pid, set_associate_node(pid)and insert the node into that tail of SQL.
Step 3: Selects next node a from SQL until SQL becomes empty
```

Fig. 2. HDRR algorithm

3 Simulation Results

We implemented a simulator to evaluate HDRR performance based on the simulation results. We conducted simulations to compare the differences between the DRR, WFQ, WF2Q+, and HDRR. The simulation environment is shown in Figure 3. The system bandwidth is 10 Mbps, and the packet length is 1KB. The bandwidth allocated ratio is 1:1:1. In our simulation, at time 1 to 50 second, the packet arrival ration for each flow is 1:1:1. Between 51 to 100 second, flow q1 blocked, and it means that the resource of q1 will be distributed. Between 101 to 150 second, the packet arrival ration for each flow is 2:0:1. Between 151 to 200 second, the packet arrival ration for each flow is 1:0:2. Between 201 to 250 second, the packet arrival ration for each flow is 2:0:2. In this study we are interested in two performance metrics: the average waiting time of each flow, and the average throughput of each flow.

Figures 4 show the average waiting time of each flow of WFQ, DRR, WF2Q+, and HDRR. It's obviously that the results of WFQ and DRR are much simulated. It is because that in WFQ and DRR, all the unused bandwidth will share to all backlog flows. However, in WF2Q+ and HDRR, the unused resource of q1, will only be

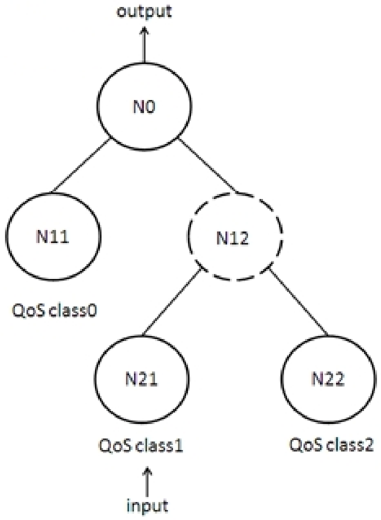
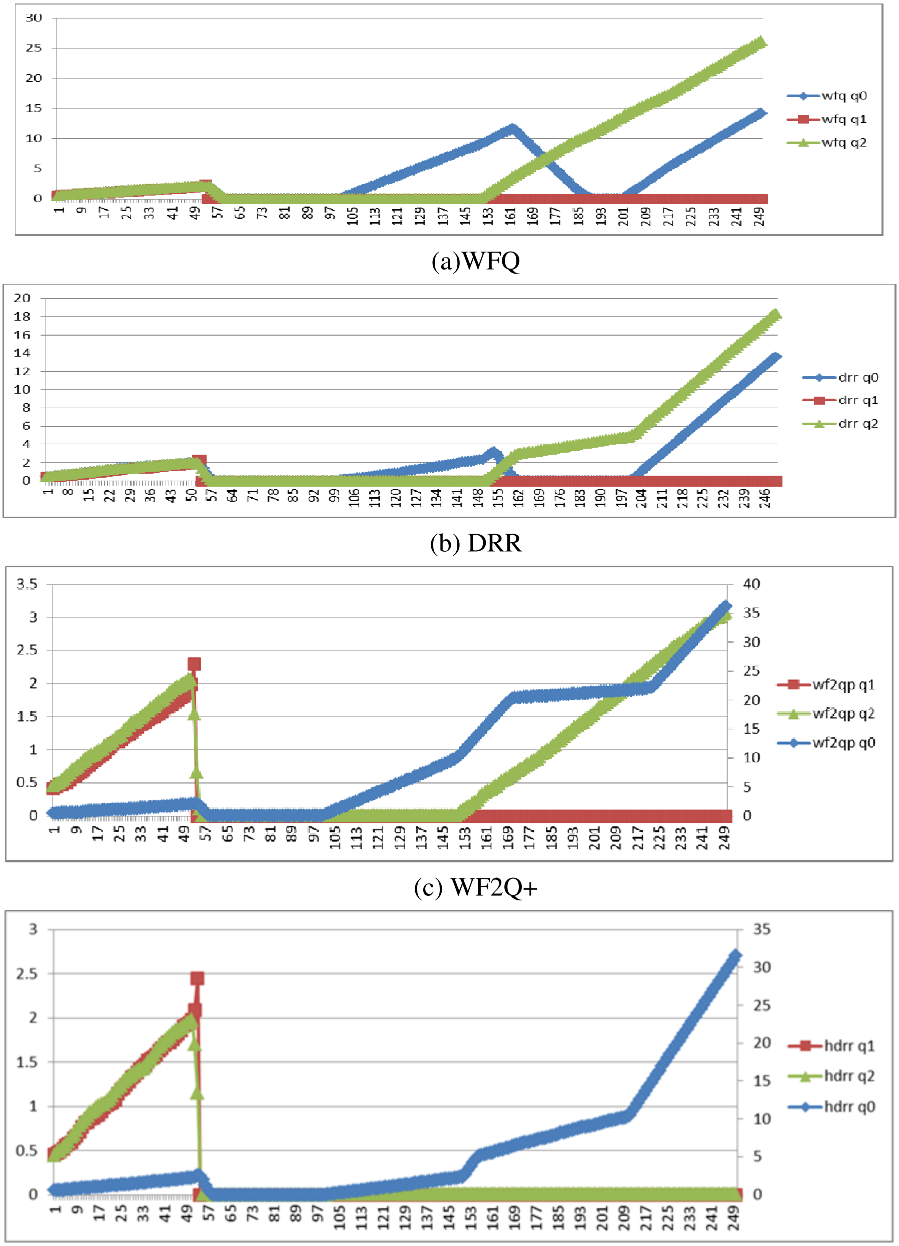


Fig. 3. Simulation Environment

shared to q2. Thus, when the traffic increased, in WFQ and DRR, all the flows' waiting time increased. In contrast with WFQ and DRR, in WF2Q+ and HDRR, the bandwidth of q0 will never be shared to q1 or q2 when q0 is backlog, and the bandwidth of q1 will never be shared to q0 when q2 is backlog. Thus, in WF2Q+ and HDRR, the waiting time of q0 is better than that of q1 or q2 from 0 to 50 second.

Moreover, when the traffic of q1 blocked, from 50 to 150 second, the waiting time of q2 is very low. After 150 second, the waiting time of q2 in WFQ, DRR, and WF2Q+, are increased, but that in HDRR is still very low. The waiting time of q0 is decreased is due to the traffic ratio of q0 is decreased. After 200 second, the traffic ratio of q0 and q2 are increased, but in HDRR, the waiting time of q2 is still very low.

Figures 5 show the average throughput of each flow of WFQ, DRR, WF2Q+, and HDRR. It's obviously that the results of WF2Q+ and HDRR are much simulated. It is because that in WF2Q+ and HDRR, all the unused bandwidth will only be share to q2, when q2 is backlog. However, in WFQ and DRR, the unused resource of q1, will be shared to q0 and q2. Thus, when the traffic increased, in WF2Q+ and HDRR, q2 will get most of the unused bandwidth of q1. In contrast with WF2Q+ and HDRR, in WFQ and DRR, the bandwidth of q0 will be shared to q1 or q2 even when q0 is backlog, and bandwidth of q1 will be shared to q0 even when q2 is backlog. Thus, as shown, in WFQ and DRR, after 150 second, the traffic of q2 is increased, but the throughput of q2 is increased slowly, and in WF2Q+ and HDRR, the throughput of q2 is increased immediately. Moreover, from these simulation results, we can found that the HDRR can provide better performance.



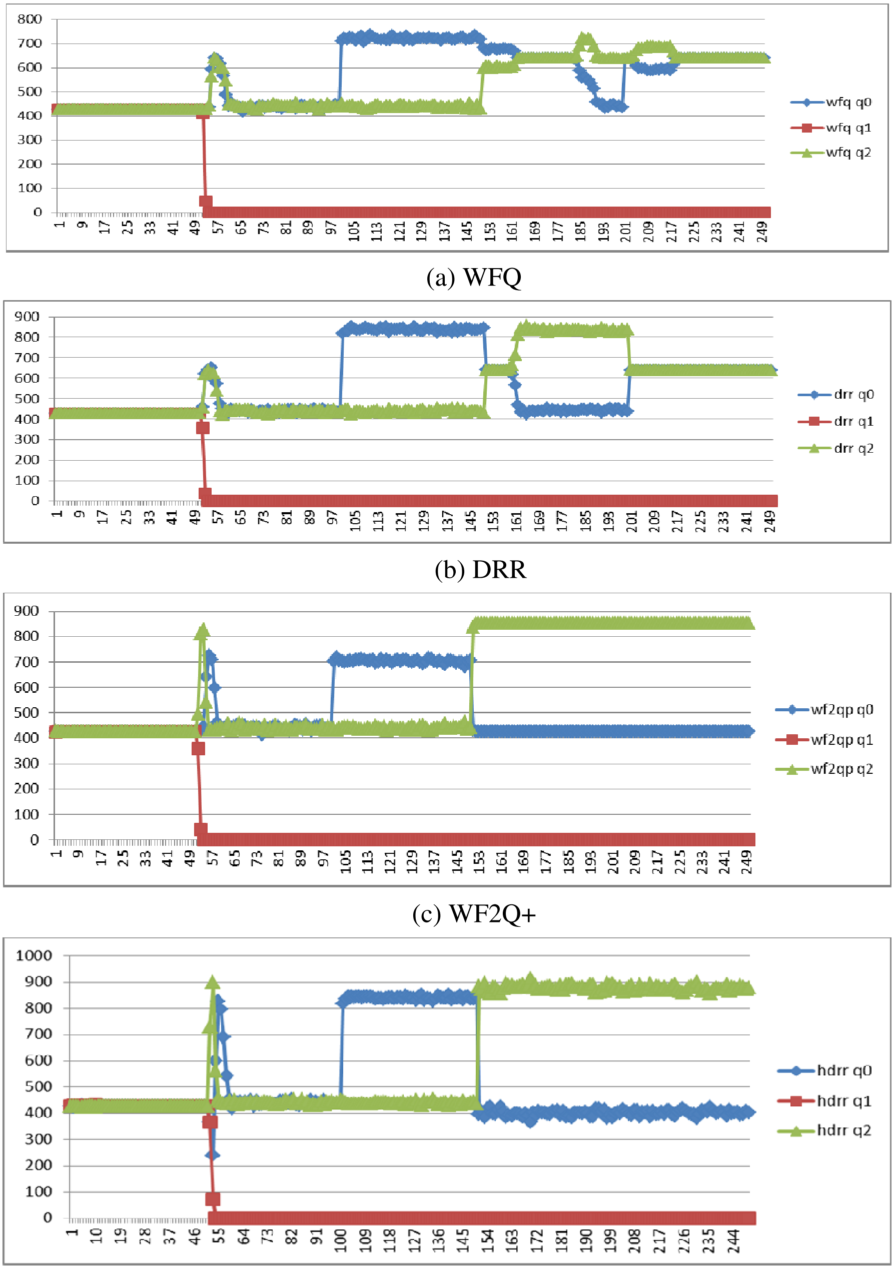
(a)WFQ

(b) DRR

(c) WF2Q+

(d) HDRR

Fig. 4. Average Waiting Time



(a) WFQ

(b) DRR

(c) WF2Q+

(d) HDRR

Fig. 5. Average Throughput

4 Conclusions and Future Work

How to design a packet scheduling strategy to support multiple classes of service with quality of service (QoS) guarantee for each user is a great challenge of computer network. In the paper, we propose the Hierarchical Deficit Round-Robin (HDRR) packet scheduling algorithm based on the DRR packet scheduling algorithm. From the simulation results, we can found that in HDRR, the unused bandwidth can be distributed based on the hierarchical structure, and can provide better performance.

Acknowledgement. This work was supported by the National Science Council of Taiwan, R.O.C. under Grant NSC100-2221-E-259-029, and NSC101-2628-E-259-004-MY3.

References

1. Braden, R., Clark, D., Shenker, S.: Integrated Services in the Internet Architecture: An Overview, RFC 1633 (July 1994)
2. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services, RFC 2475 (December 1998)
3. Parekh, A.K., Gallager, R.G.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. IEEE/ACM Transactions on Networking 1(3), 344–357 (1993)
4. Golestani, S.J.: A self-clocked fair queuing scheme for broadband applications. In: IEEE INFOCOM 1994, pp. 636–646 (April 1994)
5. Bennett, R., Zhang, H.: WF2Q: Worst-case fair weighted fair queueing. In: Proc. IEEE INFOCOM 1996, pp. 120–128 (March 1996)
6. Goyal, P., Vin, H.M., Chen, H.: Start-time fair queuing: A scheduling algorithm for integrated service. In: Proc. ACM-SIGCOMM 1996, pp. 157–168 (August 1996)
7. Bennett, J.C.R., Zhang, H.: Hierarchical Packet Fair Queueing Algorithms. IEEE/ACM Transactions on Networking 5(5), 675–689 (1997)
8. Shreedhar, M., Varghese, G.: Efficient fair queueing using deficit round-robin. IEEE/ACM Transaction on Networking 4(3), 231–243 (1996)
9. Parulkar, G., Adiseshu, H., Varghese, G.: A reliable and scalable striping protocol. In: ACM SIGCOMM, pp. 131–141 (August 1996)
10. Wu, C.-C., Moh, C., Wu, H.-M., Tsaur, D.-J., Lin, W.: Efficient and Fair Hierarchical Packet Scheduling Using Dynamic Deficit Round Robin. GESTS International Transactions on Computer Science and Engineering 21(1), 85–96 (2005) ISSN:1738-6438
11. Back, D.-S., Pyun, K., Lee, S.-M., Cho, J., Kim, N.: A Hierarchical Deficit Round-Robin Scheduling Algorithm for a High Level of Fair Service. In: International Symposium on Information Technology Convergence, pp. 115–119 (November 2007)
12. Yu, H., Van, Y., Berger, M.S.: Topology-based Hierarchical Scheduling Using Deficit Round Robin: Flow Protection and Isolation for Triple Play Service. In: First International Conference on Future Information Networks, pp. 269–274 (October 2009)

A System of Publishing and Managing Contents in Multiple Web2.0 Websites

Jyun-Kai Huang, Shyan-Ming Yuan, Hung-Yu Chen, and Yuan Chang

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
jkhunag.cs98g@nctu.edu.tw, smyuan@cs.nctu.edu.tw,
soulinlove541@gmail.com,xxxxxyuan@gmail.com

Abstract. The content that websites provide were static in Web1.0, the clients could only acquire information from the website. This changed when Web2.0 was released, websites had the capability to allow clients to comment, share songs/videos and interact with other clients. As time goes by, more and more people draw attention from the public by sharing interesting content. Attention can be easily transformed to income by adding advertisement to the content, so now there are a lot of talents that make a living out of the content they provide on various platforms over the internet. Because of policy or high page views, the content provider often uploads contents to multiple platforms, which is very time consuming and a waste of bandwidth. This paper proposes a content managing system that not only will help talents publish their work and status to multiple platforms in an efficient fashion, but also provides backup and maintenance mechanisms.

Keywords: Web1.0, Web2.0, policy.

1 Introduction

In the Web2.0, Web pages was dynamically generated. The content was not only created by webmasters but also by general users. The website usually only provide a platform to let users share their knowledge, diary, comment, music and video. Therefore all users on the website can see, read, and interact with the content such as leaving a comment[1][2].

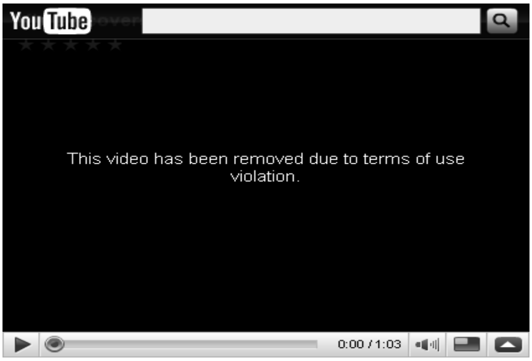
Since Youtube was created in 2005, the video sharing website has been getting more and more popular. Similar websites are springing up on the Internet such as Dailymotion, Vimeo, Tudou, etc. The host provides a platform to allow content providers to upload their work and allow audience to leave comments and interact. Because of video sharing hosts, professional content providers became a viable job[3]. They make a living on generating content and earn income by adding advertisement to the content[10]. The video content includes comedy, music, gaming record, talk shows and criticism.

Many people have commercial activity with content providers. And the content providers usually use multiple social network websites[4] and own their own website to promote their content. Sometimes they even sell their own merchandise.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 429–438.
DOI: 10.1007/978-3-642-35452-6_44

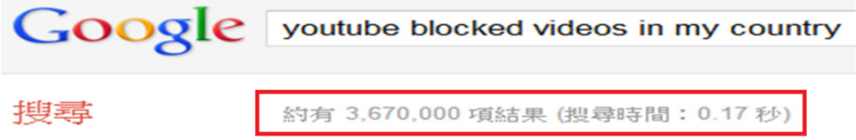
So comes the problems with it. We can see in Fig. 1, although most content providers usually upload their work to Youtube, sometimes Youtube declines to provide service for certain content. The content provider could only choose to publish their work to another platform that will accept his/her content. In many cases, the content provider uploads to multiple platforms, which is very time consuming and a waste of bandwidth.

Sometimes, the content providers also encounter the video is blocked in some countries[11]. We can see the Fig. 2. The count of result of searching "Youtube blocked video in my country" is up to 3670000 and some content provider makes the guide video to redirect users to other platforms when a certain content is blocked due to policies on the current platform[12]. The video snap is shown in Fig. 3.



This video has been removed due to terms of use violation.

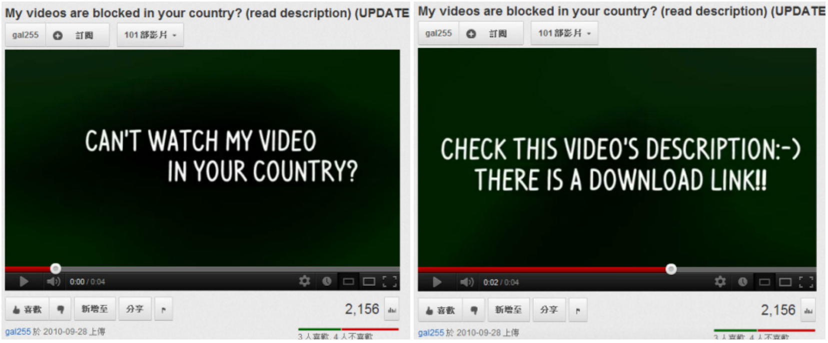
Fig. 1. Deleted video on Youtube



youtube blocked videos in my country

約有 3,670,000 項結果 (搜尋時間：0.17 秒)

Fig. 2. Result of Google Search



My videos are blocked in your country? (read description) (UPDATE

CAN'T WATCH MY VIDEO IN YOUR COUNTRY?

My videos are blocked in your country? (read description) (UPDATE

CHECK THIS VIDEO'S DESCRIPTION:-) THERE IS A DOWNLOAD LINK!!

Fig. 3. Some content provider makes the guide video

In this paper, we develop a system (web application) which let content providers easily and conveniently upload their content to multiple platforms. And provides some additional functionalities such as backing up their contents, comments and video, automatically re-upload (recovery) if the video was deleted by service providers and automatic guide video generation to redirect users to other platforms when a certain content is blocked due to policies on the current platform

The rest of this paper is organized as following: In chapter 2, we discuss the background and related works. In chapter 3, we present the system design and the details of implementation. In chapter 4, we demonstrate our system. Next, in chapter 5, we discuss some system evaluation. Finally chapter 6 concludes this paper and presents our future work.

2 Background and Related Works

2.1 Background

Communication with Web2.0 Websites. If any applications (web, desktop or Mobile) would like communicate with Web2.0 websites, there are two approaches. The first one is to call the API (Application Programming Interface) that the websites provide which usually follows RESTful web API design[5], meaning that we can use standard HTTP methods to retrieve and manipulate resources. For example, to get the profile of a user on Facebook, you might send an HTTP request like:

```
GET https://graph.facebook.com/{username}
```

Then, we will receive a response whose format is XML or JSON like:

```
{ “id”:”xxxxxx”, “name”:”xxxx” }
```

The second approach is to parse the HTML of each page of user action and observe what parameters is sent by HTTP request and what method is used by HTTP request then think of ways to reproduce the same request by some tools. The advantage of this approach is that we can do the things which the API do not provide. However, the drawback is that it is difficult. The websites usually obfuscate HTML to make us hard to parse it in order to prohibit us from doing illegal things or attacking them. In this paper, we adopt the official API which the websites provide.

2.2 Related Works

In our survey, we discover some applications similar to our system but we have some features they do not. We will introduce them in detail in the following sections.

HTC Friend Stream. HTC friend stream is Android App which is embedded in the HTC sense [21]. It integrated many social network sites including Facebook, Twitter, Plurk and Flickr. It allows users to view all feeds sent by those social network sites in one window. So that Users don’t have to always change the application in order to see all feeds from different platforms. And it can post contents (status) to different platforms simultaneously [6].

Yoono. Yoono is a web browser plug-in that acts as a social aggregator by gathering social networking information and keeping it in one place. Beyond just displaying status information, Yoono will allow the user to instantly chat with any friend across multiple networks, quickly and easily maintain their social networks, receive live updates from friends and discover new websites. And Yoono has also released Desktop, iPhone App versions[7].

Digsby. Digsby is a desktop application, which integrated Instant Messenger, Email and Social Networks. Including a multiprotocol IM client that lets you chat with all your friends on AIM, MSN, Yahoo, ICQ, Google Talk, and Jabber with one easy-to-manage buddy list. And it integrated many E-mail services including Gmail, Hotmail, Yahoo Mail, AOL/AIM Mail, etc. You can check all your inboxes of different platforms without needing to check them one by one. In social networks, you can update your status everywhere at the same time and get their feedback[8].

Meebo. Meebo is an advertising-supported embedded social media platform. It began in 2005 as a browser based instant messaging program which supported multiple IM services, including Yahoo! Messenger, Windows Live Messenger, AIM, ICQ, MySpaceIM, Facebook Chat, Google Talk, CafeMom and others. Meebo is generally seen as a nuisance on third party websites. Meebo also developed a version of Meebo for use on the iPhone and a native application that runs on Android mobile phones[9].

Comparison. After the above introduction, we can understand the services they provide and we can see that they're mostly integrated social network services, emails or instant messenger without video sharing websites. We focus on video sharing websites and provide functionalities which are backing up, recovery mechanism and automatic guide video generation to allow the talents to concentrate on their work and worry less about the policies and uploading. The tables below shows the difference between them

Table 1. The difference in platform

App. Name	Desktop based	Web based	Mobile App.
Our System	N	Y	N
Yoono	Y	Y	Y
Digsby	Y	N	N
Meebo	N	Y	Y
Friend Stream	N	N	Y

Table 2. The difference in provided services

App. Name	Social Networks	Video Sharing	E-mail	Instant Message
Our System	Y	Y	N	N
Yoono	Y	N	N	Y
Digsby	Y	N	Y	Y
Meebo	N	N	N	Y
Friend Stream	Y	N	N	N

3 System Design and Implementation Details

Our System is a web application. At present, we have integrated three social networks and two video sharing websites, which are Facebook[13], Plurk[17], Twitter[14], Youtube[16] and Dailymotion[15]. The system architecture is shown in Fig. 4. Our application connects five platforms.

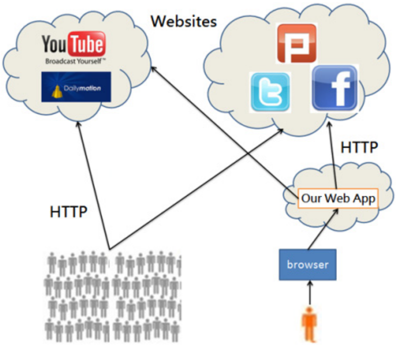


Fig. 4. System architecture

3.1 System Functionalities

Considering the problem which content providers encounter, we provide the following functionalities: First, User can post statuses and upload videos to multiple platforms via our application by browser. Then People can see their work and status in my application which reside on different platforms and see the comments that the people leave. Third, we provide backing up functionality which lets users pack their content which they posted before along with comments which the people left, and download it to their computer. Fourth, we can automatically re-upload the video which was deleted by somebody or service providers and send a message to notify the content provider and remind them that they have deleted videos. Last, in the fourth functionality, we can re-upload the video three times. This means if your content was deleted for no reason, and we can recovery it, and if the one was deleted again then we recovery twice. In our design, if one of your videos was deleted three times, we will no longer upload this video to this platform. We think the service providers don't permit us to place this video on their platform. They don't delete videos mistakenly, at least not three times in a row. They deleted the video for a legit reason. So we provide a special functionality which is a redirecting mechanism. If we detected the video was deleted three times and we can generate a video which include some URLs which we uploaded to other platforms and tell people where they can watch the deleted video.

3.2 The Detail Workflow of the Functionalities

In this section, we present the details of our system with five functionalities.

First, we must get Access token so that we make our system work properly. We follow the OAuth1.0a and OAuth2.0 protocol to get Access Token by HTTP request which will return JSON or XML format data[18]. Eventually, we will get it and store into database. To allow some functionalities of our system to work properly, we need to store the access token into the database.

In posting status and video to Web2.0 websites, we will send the posting status and video API to websites and the posted data will be stored in a MySQL database. We do the same thing when posting a comment to a video or status. And we have many background programs designed to collect comments which people leave in websites. For each platform, we create a program to fetch comments individually. The programs always send List comment API to websites we integrated and store those comments we received to database. Once users refresh our application and then they will see the comments. We can see in Fig. 5, the blue circle labeled "Fetch Comment Platform x" sends an HTTP request to website and receives XML or JSON format data, then we parse only the data that we want to store in the database. The reason why collecting comments program must be executed in background is response of API is very slow. If it is executed in foreground, then the users have to spend a lot of time waiting and blocking if we send too many API requests and usually collecting comments require a lot of API requests. More precisely, the number of requests sent is as many as the count of the statuses in database. The reason why we create programs individually to each platform is distributing waiting time. Therefore if we only have one program to collect comments of many websites, the users need to wait a long time to see those comments. It would take more than ten seconds. If we integrate more websites, then the time will be even longer. That's why we distribute it so it would only spend a few seconds.

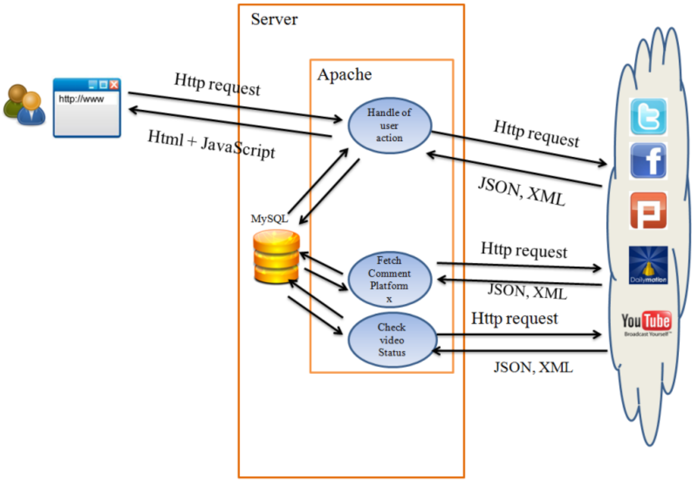


Fig. 5. The details of the system architecture

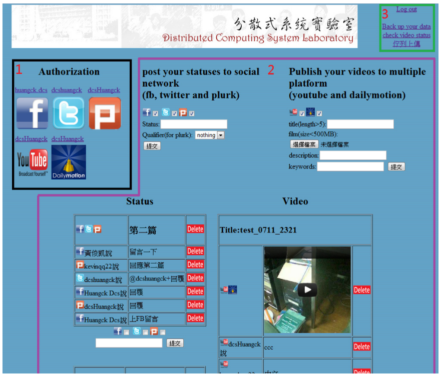
The back up functionality, as previously mentioned, the statuses, comments and are stored in the database. Therefore we read data from the database to let user download.

For the recovery functionality, we have a background program similar to the comment collecting program. We always send check video API request to website to check if the video was deleted. If the video was deleted then we will send post API to re-upload the video with access token which we stored in database and record it into database.

For the redirecting mechanism, we use PHP GD Library[19] to generate a picture and call an external program, which is FFmpeg[20], to generate a guide video with the picture generated by GD Library. For the recovery functionality, if we detect the video was deleted more than three times then we will upload the guide video in order to guide people to where they can watch it.

4 System Demonstration

In the Fig. 6(a), we can see the system snap. The OAuth authorization panel and contents management panel are labeled "1" and labeled "2" respectively. The upper right corner is for backing up and checking video status functionalities, etc. In OAuth authorization panel, Users can authorize us to access the private data on the Web2.0 sites. Once we get the user's authorization, we can publish the content to the multiple Web2.0 sites we integrated. The details will be shown in contents management panel. There are publishing content panel, the posted contents and the comments we have fetched from Web2.0 sites. Fig. 6(b) shows the guide video, if we detected the video was deleted three times and we can generate a video which include some URLs which we uploaded to other platforms and tell people where they can watch the deleted video.



Status:
Qualifier(for plurk): nothing
提交

title(length>5):
film(size<500MB):
選擇檔案 未選擇檔案
description:
keywords: 提交

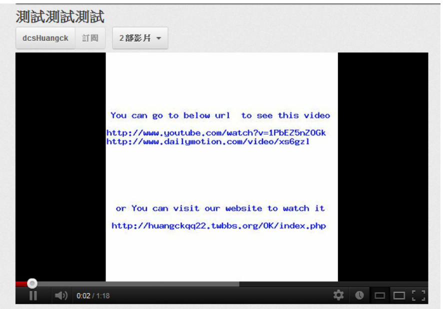
	第二篇	Delete
黃俊凱說	留言一下	Delete
kevinqq22說	回應第二篇	Delete
dcshuangck說	@dcshuangck+回覆	Delete
Huangck Dcs說	回覆	Delete
dcsHuangck說	回覆	Delete
Huangck Dcs說	上FB留言	Delete

提交

Title:test_0711_2321

		Delete
dcsHuangck說	ccc	Delete

Fig. 6(a). System snap



You can go to below url to see this video

http://www.youtube.com/watch?v=1PbEZ5nZOGk
http://www.dailymotion.com/video/xs6gzl

or You can visit our website to watch it

http://huangckqq22.twbbs.org/OK/index.php

Fig. 6(b). Guide video

5 System Evaluation

As mentioned before, The response time of web API(web service) is very slow and we measured the response time of APIs of each platform. We only measured some parts of the APIs which are list comment API, check video API because we always send list comment API and check video to the websites in order to synchronize the comments of status and monitor video status if the video is deleted. The results are following tables and figures:

In evaluating list comment API, We send two rounds of APIs to the service provider, each round is composed of six successive list comment APIs.

We can see in Table 3. the individual average response time of each API is 0.5596s, 1.4869s, 0.7511s, and 0.3925. Although the time of Youtube is only 0.0559s, comparing to a local program, the time is much slower. Especially Dailymotion, the response time is too long. Other platforms' response time is between 0.4s and about 0.8s and it's also much slower than a local program. If we have a lot of requests, then the users need to wait for a long time.

In the Table 4, we can see that there is no big difference between Dailymotion check video API average response time and the list comment API response time. For Youtube, the average, minimum and maximum response times are a little higher than the list comment API.

Table 3. List comment API query time

Platform	Minimum (sec)	Maximum (sec)	Average (sec)
Youtube	0.01530	0.24407	0.055966
Dailymotion	1.47230	1.53308	1.486921
Facebook	0.71738	0.78276	0.751166
Plurk	0.37570	0.43107	0.392543

Table 4. Check video API query time

Platform	Minimum (sec)	Maximum (sec)	Average (sec)
Youtube	0.02576	0.26906	0.08827
Dailymotion	1.46151	1.60067	1.498117

The Solution for Long Response Time. For the comment aggregation in our present design, we distribute the API calls. Our system forks an exclusive process for each platform, so that they can handle the comment collection and store it into our local database in a parallel fashion. All the contents users see are read from our local database, by using local databases we can enhance efficiency. There is no problem when only a few people are on our system. The synchronization time is not too long. It's about a few seconds. If there are many people using our system or a few people processing a lot of data, then the time might be too long for practical use. A possible solution is to distribute the requests again, as a number of process or thread as users or some unit. Obviously, this is not a very good solution. This solution generates heavy loading and wastes a lot of CPU time, even when there is no user online.

The following might be a practical solution. We can mix AJAX and polling. We might have to wait half an hour or one hour to synchronize once and when the user just logged into the system, we present the data which we loaded from our local database first, and use AJAX to background check and fetch new comments according to the user's focus point then dynamically update the data and store it into our database. However, the requests still need to be distributed to reduce waiting time.

The user still see the data which is loaded from our local database, which prevents executing all the requests at the same time and let the user feel that our system is too slow, and use the user's focal point to check if it has new data. The solution solves the long response time of APIs, reduces the server loading and Balances the user waiting time.

For the video recovery and guide video generation, the original method is to check the status of the videos in background repeatedly.. If we execute it at run-time , then the deleted video would not be able to be uploaded immediately. Our system considers the server loading and plans to polling and distribute to handle the recovery/guide video generation.

The solutions of this section will be implemented in the future.

6 Conclusion and Future Works

Nowadays, more and more Web2.0 website spring up. Many people become content providers and usually use more than one Web2.0 sites.

In this paper, we propose a system which integrates multiple Web2.0 websites including not only social network websites but also video sharing websites and provide functionalities including publishing content, comment aggregation and backing up content and comment. Especially for video sharing websites, we also provide video recovery and automatic guide video generation.

These functionalities help content providers publish and manage their content. The content provider no longer needs to go every website to post their content, just only post once and comment aggregation will enable them see all the comments which reside on different platforms. Also, content providers do not to read the comments and reply on each platform, they can simply do it on our system and we will publish it to the corresponding platforms for them. Backing up mechanism allows the user to never worry about the Web2.0 websites crashes and lose data and video recovery and automatic guide video generation make them worry less about platform policy.

In the future, we will integrate more platforms like Todou, Vimeo, and Google+ and use AJAX to enhance user experience on the web. And we will combine our system with PHPmotion Media Sharing CMS. Every professional content provider usually own a website by themselves, if we integrate our system with PHPmotion and release to them for use, then users can own a video sharing website to place their works with our providing functionalities.

References

1. Krishnamurthy, B., Cormode, G.: Key differences between Web 1.0 and Web 2.0. First Monday 13(6) (2008)
2. O'Reilly, T.: What is web2.0: Design Patterns and Business Models for the Next Generation of Software. International Journal of Digital Economics (65), 17–37 (2007)

3. Cheng, X., Dale, C., Liu, J.: Understanding the Characteristics of Internet Short Video Sharing, YouTube as a Case Study. Technical report, Cornell University (2007)
4. Boyd, D.M., Ellison, N.B.: Social Network Sites: Definition, History, and Scholarship. Journal of Computer-Mediated Communication 13(1), 210–230 (2007)
5. Richardson, L., Rub, S.: RESTful Web Services: Web services for the real world. O'Reilly Media (2007)
6. How to use Friend Stream, http://www.htc.com/help/tw/howto_iframe.aspx?id=3285&type=1&p_id=312
7. Yoono, http://www.yoono.com/
8. Digsby, http://www.digsby.com/
9. Meebo, https://www.meebo.com/
10. Become a Partner, http://www.youtube.com/yt/creators/partner.html
11. YouTube Blocked in China Google Says, http://msl1.mit.edu/furdlog/docs/nytimes/2009-03-24_nytimes_youtube_blocked.pdf
12. My videos are blocked in your country (read description) (UPDATED 12/15), http://www.youtube.com/watch?v=jiJZSHucXa0
13. Facebook, https://www.facebook.com/
14. Twitter, http://twitter.com/
15. Dailymotion, http://www.dailymotion.com/
16. Youtube, http://www.youtube.com/
17. Plurk, http://www.plurk.com/
18. OAuth, http://oauth.net/
19. PHP GD Library, http://php.net/manual/en/book.image.php
20. FFmpeg, http://ffmpeg.org/index.html
21. Wikipedia: HTC Sense, http://zh.wikipedia.org/wiki/HTC_Sense

A Web Service Selection Mechanism Based on User Ratings and Collaborative Filtering

Chin-Chih Chang and Chu-Yen Kuo

Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu City, Taiwan
{changc,m09602044}@chu.edu.tw

Abstract. As cloud computing is gaining its popularity, more and more services are deployed through Web service technology. How to quickly select a suitable service is an important issue. In recent studies on the Web service selection, the most approaches are based on service requirements and quality of services designated by the user but less take the user's own evaluation into account.

In this paper, we propose a Web service selection mechanism based on user ratings and collaborative filtering. In this mechanism the quality of service of Web services, the feedback from the users and similarity among the users are taken into consideration for selecting Web services. The proposed method is verified by a case study of a travel information system and then the Mean Average Precision (MAP) is evaluated by the experiments.

Keywords: Web services, Web service selection, Feedback mechanism, Personalization.

1 Introduction

Recently, cloud computing has not only become a catchword but also changed the way we think about technology. It is not uncommon to hear people talk about using data in the cloud. A cloud is a computing model delivering software, platform, and infrastructure as services over the Internet. The key to cloud computing is the service provision. Service-Oriented Architecture (SOA) is an architectural approach for cloud computing while Web services are the de facto technology for SOA. With the fast development of Web services more and more Web services are available. There is a need for Web services discovery and selection. To search and select a suitable Web service that can meet user's requirements from the numerous Web services has become an important research topic.

The existing service discovery technologies are mainly keyword-based search which cannot fit user's need well [1]. In a lot of scenarios, users' preferences are an important factor. In order to offer Web services that satisfy users' need we present a Web service selection mechanism based on user ratings and collaborative filtering. We use the existing Web service registry and integrate with QoS (quality of service) filtering, similarity calculating, ranking, and feedback mechanisms to construct a

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 439–449.
DOI: 10.1007/978-3-642-35452-6_45

system which facilitates Web service selection. When a user wants to search and select a web service, she or he can access the system interface to accomplish the task.

The first step of the proposed service selection is QoS filtering. The service availability is used to filter which services will be taken into consideration. From this collection of services the services are then ranked based on similarity to the user's ratings from the collected feedback database from the users. The services on the top of the list will be recommended to the user. Finally, the user will evaluate the recommended services and this evaluation will be sent back to the feedback database for the further usage.

We use the travel information system as an example to validate the feasibility of our system. We will show how a user utilizes our system to find a tourist spot. Finally, we will analyze the algorithms that we present, and compare their advantages and disadvantages. And we also analyze our system and compare with other research work.

2 Personalized Web Service Selection

2.1 Web Service Selection

Web service discovery and selection is a complex process. There is still no clear distinction between discovery and selection. Service selection begins with discovery. In this paper we adopt the view that discovery is referred as the activities related to identifying the functional properties of user's requests and selection as nonfunctional properties [2]. Crasso et al. further enumerate four functional and four nonfunctional criteria [3]. The discovery process is mainly keyword-based search which returns a list of candidate services and the selection process is a refinement of the discovery process [4]. For the recent survey on service discovery and selection interested readers can refer to [3].

Functional criteria are mainly used to match against the specifications offered by the service provider. QoS is commonly used to describe nonfunctional criteria [5]. In this paper, we focus on nonfunctional QoS because they are often the decision factor. Various QoS for Web services have been identified and can be classified into the following categories: runtime-related, transaction support related, configuration management and cost-related QoS, security-related QoS, and user-related QoS [6, 7]. From our point of view, user-related QoS is the most important factor in these nonfunctional characteristics because users are always the ones who make the final decision. Hence, we focus on Web service selection in terms of user's ratings.

The goal of Web service discovery is to find appropriate Web services that match user's functional requirements. Web service discovery through UDDI (Universal Description, Discovery, and Integration) is the most basic discovery method. UDDI provides only a category-based browsing and keyword-based matching discovery service [7]. Discovery by UDDI is simply a process of matching the WSDL (Web Service Description Language) based on keyword and category. After matching keyword the possible services are listed but the most appropriate service is not retrieved. Therefore, quite a number of approaches have been proposed to enhance UDDI. Four main categories of these approaches have been specified: information retrieval-based, QoS-aware, semantic-based, and distributed [3]. The discovery process usually finds a list of candidate services. To further refine the discovery results the selection process is conducted.

Web service selection is a process to select the most suitable service from a list of candidate services after service discovery or composition mainly based on nonfunctional properties. Singh and Huhns pointed out three categories of Web service selection strategies: semantic service selection, social service selection, and economic service selection [8]. Semantic service selection finds a match based on the semantic description. Social service selection uses social rating such as reputation, recommendation, referrals, etc. to select a service. Economic service selection uses cost related information to choose a service. The main activities of service selection contain (1) matching nonfunctional service requests, (2) evaluation of service offerings, and (3) result aggregation [7].

Though quite amount of work has been dedicated to Web service selection, a solution that is more practical, less costly, and more universal is still not there. Web service selection is still a research topic for the time being. For a recent development interested reader can refer to [9].

Most selection practice is a composition of semantic, social, and economical strategies. From our point of views social service selection is the key factor. In this paper we will focus on user's ratings in social selection because the user is the final one who makes the decision.

2.2 Personalized Recommendation Techniques

The personalized recommendation technique is the recommendation technique that is focused more on user's preferences [10]. Here the preferences can be collected from registration form, survey, feedback, online behaviors, etc. There are mainly four approaches [11, 12]:

1. Content-based filtering: The method recommends items that are similar to the ones that the user liked in the past.
2. Collaborative filtering: The method recommends the items that are likely used by those who have the similar interest to the user.
3. Knowledge-based approaches: One possible approach is to ask the user directly about her or his requirements. Based on the criteria provided by the user the items are recommended.
4. Hybrid approaches: The method is a hybrid of above methods.

A comparison of these four techniques is shown in Table 1. In general, these techniques heavily depend on users' interaction. In collaborative filtering Pearson correlation coefficient or similarity matrix is commonly used to compute the similarity between two users [5].

Table 1. Comparison of recommender techniques

Recommendation techniques	Advantages	Drawbacks
Content-based filtering	Effective in locating items that are relevant to the topic	Capturing only certain aspects of the content; over-specialization
Collaborative filtering	The items are recommended based on user's rating.	The coverage of rating could be very sparse; the new items would not be recommended; algorithm is not so efficient.
Knowledge-based approaches	It does not rely on the existence of a purchase history.	Detailed knowledge about items might be required.
Hybrid approach	Efficient and more accurate	Not so simple.

3 System Architecture

3.1 System Architecture

The system architecture contains four modules: broker server, monitor server, store server, and UDDI server as shown in Fig. 1.

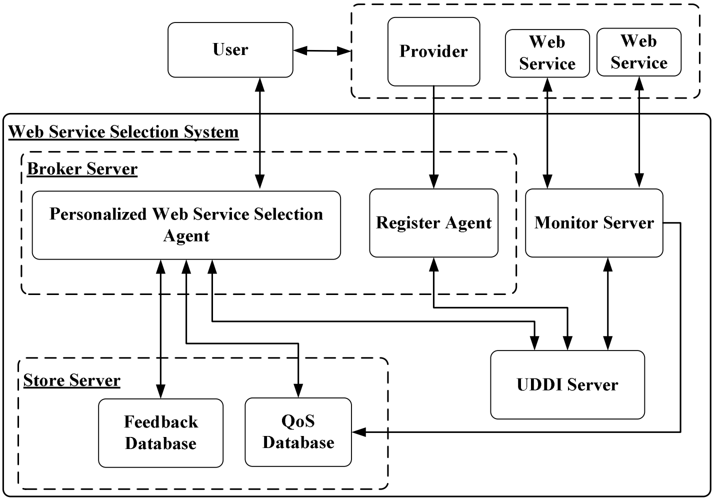


Fig. 1. System Architecture

- Broker server which provides the user with the service selection agent and for the provider with the register agent.
- Monitor server which monitors QoS of Web services in UDDI server and stores them into QoS database.
- Store server which contains QoS database of Web services and feedback database for storing the user ratings.
- UDDI server which contains a UDDI registry and a database as the service repository. The registry is implemented using jUDDI which is an open source Java UDDI implementation.

3.2 Methodology of Personalized Web Service Selection

The core part of the proposed system is the personalized Web service selection agent which mainly adopts collaborative filtering enhanced with user's feedback. The user first enters the keyword which is converted to the SOAP request sent to UDDI server. The services which satisfy the request will be filtered based on the availability. Those services whose availability is larger than 0.5 are collected. These services are then ranked based on similarity to the user's ratings from the collected feedback database from the users. The services on the top of the list will be recommended to the user. The user rates the recommended services and the rating is stored in the feedback database.

In our mechanism the similarity is calculated based on Pearson correlation coefficient shown as the equation (1):

$$Sim(x,y) = \frac{\sum_{i \in I_x \cap I_y} (R_{xi} - \overline{R_x})(R_{yi} - \overline{R_y})}{\sqrt{\sum_{i \in I_x \cap I_y} (R_{xi} - \overline{R_x})^2} \sqrt{\sum_{i \in I_x \cap I_y} (R_{yi} - \overline{R_y})^2}} \tag{1}$$

where each symbol is defined as follows:

- *Sim(x, y)* denotes the similarity between the x and y user.
- I_x and I_y denote the set of services which the x and y user have evaluated respectively.
- $i \in I_x \cap I_y$ denotes the intersection of services that the x and y user have both evaluated
- R_{xi} and R_{yi} denote the ratings of the x and y user on the service i respectively.
- $\overline{R_x}$ and $\overline{R_y}$ denote the mean ratings of services which the x and y user have evaluated respectively.

After calculating the similarity, we use the equation (2) to rank the services.

$$WS_{rank}(i,x) = \overline{R_\imath} + \sum_{i \in I_x \cap I_y,\ u \in U(i)} R_{ui} \times sim(x,u) \tag{2}$$

where the symbol is defined as follows:

- $WS_{rank}(i, x)$ denotes the predicted score for the x user on the service i.
- $\overline{R_\imath}$ denotes the mean ratings of all users on the service i.
- $u \in I_x \cap I_y$ denotes u belongs to $I_x \cap I_y$.
- R_{ui} denotes the ratings of the u user on the service i.
- *sim(x, u)* denotes the similarity between the x and u user.

The services with high scores are listed on the top and recommended to the user. The algorithm of the service selection is shown in Fig. 2 and composed of three major parts:

1. Collect the ratings from the user.
2. Calculate the similarity among user ratings.
3. Calculate the score of Web services.

```
Algorithm of personalized Web service selection
$WS = \{WS_1, WS_2WS_3, \dots, WS_n\}$ // select the services whose availability > 0.5
Variables：
n：number of services in WS
m：number of users who rate the services
current_user：current user
i, j ：counter
ratings[m][n]：ratings of services in WS from the user
WS_rank[m][n]：the evaluation of Web services from the user
Step 1：Data collecting   // Collect the evaluations of Web services from the user
Begin
    Collect rating[m][n] from all users by WS;
End;
Step 2：Similarity calculating        // Calculate Person's similarity
Begin
    For i = 1 to m do
      For j = 1 to m do
        For k = 1 to n do
           If ratings[i][k] $\neq$ empty or ratings[j][k] $\neq$ empty
             Compute sim(i, j) using Pearson correlation coefficient;
        End for
      End for
    End for
End;
Step 3：Recommendation computing
Begin
    For i = 1 to n do
      Compute WS_rank[current_user][i]; // Compute the scores of Web services
    End for
    Select top three scores to current_user;
End;
```

Fig. 2. Algorithm of personalized Web service selection

The objective of the feedback mechanism is to dynamically adapt to the user's preferences and requirements. After a user uses a Web service, she or he will be asked to give this service one of ratings including very satisfied, satisfied, neutral, dissatisfied, and very dissatisfied corresponding from 5 to 1 point. The survey result is then stored in the feedback database.

4 System Implementation and Experiments

This section describes the system development, service provision, and experimental results.

4.1 System Implementation

This system is built on a Microsoft Windows 7 machine equipped with Intel Core Quad Q6600 2.4GHz CPU, and 4GB DDR RAM. Java and JSP is used for developing applications. The Web server is Apache Tomcat 5.5 which is used as a container for broker server, UDDI server, and storage server. The DBMS is MySQL 5.0 for feedback, QoS, and travel spot databases. The UDDI server is jUDDI-0.9rc4. The UDDI Client API is uddi4j-2.0.5 for the broker server. The Web Service API is Apache Axis2-1.4.1 for the service deployment.

To verify the system we develop a travel information system which contains information of 100 sightseeing spots in the database. 30 travel information services are registered in the UDDI server and classified into two categories: attractions and spots services. When the user requests the travel information, the request is sent through Web services and then the travel information is replied to the user as shown in Fig. 3.

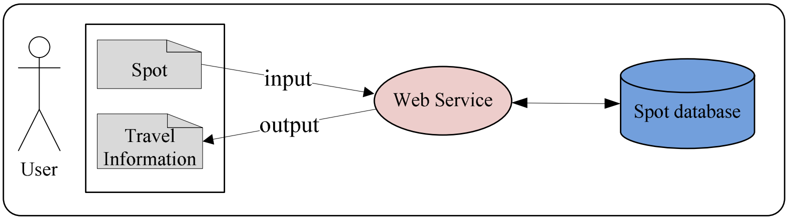
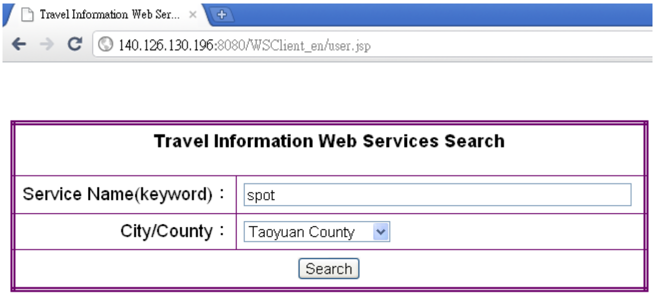


Fig. 3. Operation of travel information requests

The system includes four main functions:

- UDDI register and service publication which is based on jUDDI and with the interface for the user to input services.
- Service QoS monitor which is used to monitor the service availability. If the service is available, the service status is set to 1, otherwise, set to 0.
- Service discovery and selection which is functioned through an interface as shown in Fig. 4.
- User's feedback which is a page for the user to input the feedback from very satisfied to very dissatisfied.

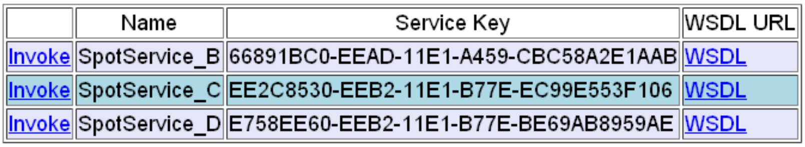
After the request is sent, the result is replied and the services of the highest three are listed as shown in Fig. 5. The user can further clicks the invoke button and retrieve the information provided by services as shown in Fig. 6.


Travel Information Web Ser...
140.126.130.196:8080/WSClient_en/user.jsp

Travel Information Web Services Search

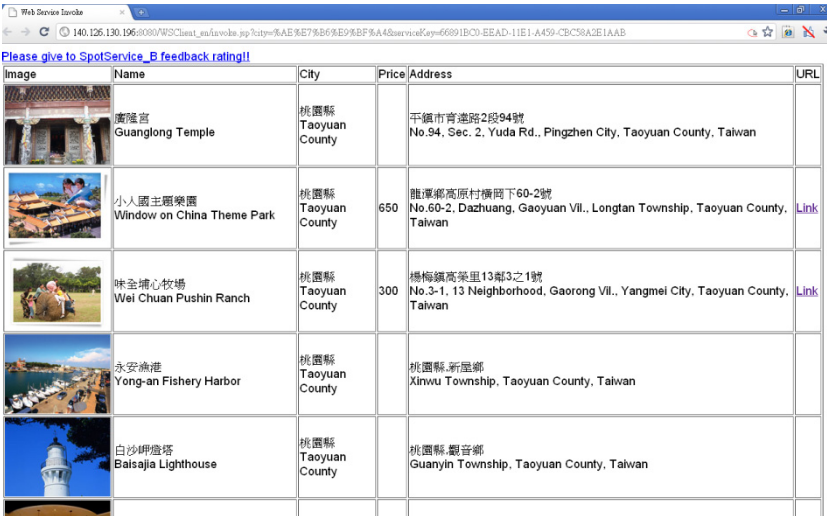
Service Name(keyword)：	spot
City/County：	Taoyuan County
Search	

Fig. 4. Travel information service interface



	Name	Service Key	WSDL URL
Invoke	SpotService_B	66891BC0-EEAD-11E1-A459-CBC58A2E1AAB	WSDL
Invoke	SpotService_C	EE2C8530-EEB2-11E1-B77E-EC99E553F106	WSDL
Invoke	SpotService_D	E758EE60-EEB2-11E1-B77E-BE69AB8959AE	WSDL

Fig. 5. Selection result


Web Service Invoke
140.126.130.196:8080/WSClient_en/invoke.jsp?city=%AE%E7%B6%E9%BF%A4&serviceKey=66891BC0-EEAD-11E1-A459-CBC58A2E1AAB

Please give to SpotService_B feedback rating!!

Image	Name	City	Price	Address	URL
	廣隆宮 Guanglong Temple	桃園縣 Taoyuan County		平鎮市育達路2段94號 No.94, Sec. 2, Yuda Rd., Pingzhen City, Taoyuan County, Taiwan	
	小人國主題樂園 Window on China Theme Park	桃園縣 Taoyuan County	650	龍潭鄉高原村横岡下60-2號 No.60-2, Dazhuang, Gaoyuan Vil., Longtan Township, Taoyuan County, Taiwan	Link
	味全埔心牧場 Wei Chuan Pushin Ranch	桃園縣 Taoyuan County	300	楊梅鎮高榮里13鄰3之1號 No.3-1, 13 Neighborhood, Gaorong Vil., Yangmei City, Taoyuan County, Taiwan	Link
	永安漁港 Yong-an Fishery Harbor	桃園縣 Taoyuan County		桃園縣,新屋鄉 Xinwu Township, Taoyuan County, Taiwan	
	白沙岬燈塔 Baisajia Lighthouse	桃園縣 Taoyuan County		桃園縣,觀音鄉 Guanyin Township, Taoyuan County, Taiwan	

Fig. 6. Service invocation result

4.2 Experiment Analysis and Discussion

There are usually two approaches to analyze the system performance. One is to collect the data from the user's usage and the other is to use the simulation. Usually, the simulation will be applied before the prototype of the system is being tested by the users. In addition, in the time being of paper writing, the travel information system is still evolving. Hence, we adopt Jester Datasets [13] for the simulation analysis. In Jester Dataset, Dataset 1 contains over 4.1 million continuous ratings (-10.00 to

+10.00) of 100 jokes from 73,421 users collected between April 1999 and May 2003. In our experiment these ratings are adopted as user ratings for predicting the preference of the next user through the proposed Web service selection mechanism. In this simulation the joke provision is regarded as Web services.

In this experiment we use MAP (mean average precision) as a measure to evaluate the effectiveness of the system. The basic idea is that a service with a higher rank will have a higher MAP value. The equation is shown as follows:

$$\text{MAP} = \frac{\sum_{\text{i}=1}^{\text{N}} \frac{\text{i}}{\text{Rank i}}}{\text{N}} \tag{3}$$

where N is the total number of selected services and Rank_i is the rank of the ith service. For example, if a user requests three services and the system finds three services located in the rank first, fourth, and fifth. The MAP value is 0.7 where $(\frac{1}{1}+\frac{2}{4}+\frac{3}{5})/3 = 0.7$.

We randomly select 100 users and 30 jokes to conduct the three experiments:

1. Use the equation (1) as the measure to recommend top 3, top 5, and top 10 services.
2. Use $\sum_{i \in I_x \cap I_y,\ u \in U(i)} R_{ui} \times sim(x,u)$ as the measure to recommend top 3, top 5, and top 10 services.
3. Use the proposed evaluation equation (2) as the measure to recommend top 3, top 5, and top 10 services.

The results are shown in Fig. 7. From the results we can find the proposed evaluation method produced the best MAP and the effect of the mean rating of the service i ($\overline{R_\iota}$) is not so obvious.

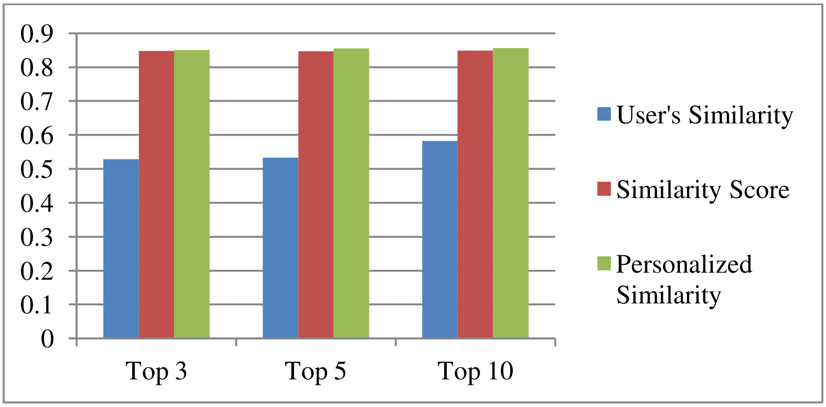


Fig. 7. Comparisons of different score calculation methods

5 Conclusions and Future Work

In this paper, we present a personalized Web service selection which collects the ratings from users to recommend the services for a user. The main objective is to enhance collaborative filtering with the user's feedback. The mechanism proves to be functional and effective and is applied to a travel information system to help a user recommend the spots they like.

Currently, we are updating the UDDI server to UDDI 3, increasing the services of the travel information system, and enhancing the first stage of filtering by taking more QoS into consideration. In addition, we plan to use the user's profile and social networks to more precisely target at the user's need. We will also conduct a comparison with other similar systems.

Acknowledgement. This work is partly supported by the National Science Council of Taiwan under the grant number NSC100-2218-E-216-001.

References

1. Yu, Q., Bouguettaya, A.: Guest Editorial: Special Section on Query Models and Efficient Selection of Web Services. IEEE Transactions on Services Computing 3(1), 161–162 (2010)
2. Carenini, A., Cerizza, D., Comerio, M., Valle, E.D., De Paoli, F., Maurino, A., Palmonari, M., Sassi, M., Turati, A.: Semantic Web Service Discovery and Selection: a Test Bed Scenario. In: Proceedings of the 6th International Workshop on Evaluation of Ontology-based Tools and the Semantic Web Service Challenge (EON-SWSC-2008), Tenerife, Spain, June 1-2 (2008)
3. Crasso, M., Zunino, A., Campo, M.: A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. Journal of Database Management 22(1), 102–132 (2011)
4. Balke, W.-T., Wagner, M.: Towards Personalized Selection of Web Services. In: WWW (Alternate Paper Tracks) 2003, Budapest, Hungary, May 20-24 (2003)
5. Wang, H.-C., Lee, C.-S., Ho, T.-H.: Combining Subjective and Objective QoS Factors for Personalized Web Service Selection. Expert Systems with Applications 32(2), 571–584 (2007)
6. Ran, S.P.: A Model for Web Services Discovery with QoS. ACM SIGecom Exchanges 4(1), 1–10 (2003)
7. Yu, H.Q., Reiff-Marganiec, S.: Non-functional Property based service selection: A survey and classification of approaches. In: Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop Co-located with the 6th IEEE European Conference on Web Services, Ireland, Dublin, November 12 (2008)
8. Singh, M.P., Huhns, M.N.: Service-Oriented Computing: Semantics, Processes, Agents. Wiley (2005)
9. Yu, Q., Bouguettaya, A.: Foundations for Efficient Web Service Selection. Springer (2010)
10. Yu, P.S.: Data mining and personalization technologies. In: Proceedings of the Sixth International Conference on Database Systems for Advanced Applications (1999)

11. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press (2011)
12. Resnick, P., Varian, H.R.: Recommender Systems. Communications of the ACM 40(3), 56–89 (1997)
13. Jester Datasets, University of California Berkeley, `http://eigentaste.berkeley.edu/dataset/`

Cross-Layer-Based Adaptive TCP Algorithm in 4G Packet Service LTE-Advanced Relaying Communications

Ben-Jye Chang and Yi-Hsuan Li

Department of Computer Science and Information Engineering,
National Yunlin University of Science and Technology, Taiwan, ROC
{changb,g9817718}@yuntech.edu.tw

Abstract. Transmitting data packets under different wireless networks suffers from insufficient available bandwidth, long transmission delay and high wireless interference. The traditional TCP congestion control mechanism does not consider available bandwidth or buffer size and blindly halves the congestion window after receiving three duplicate ACKs. Thus, 1) we propose a bottleneck link identification approach to identify bottleneck link is on the shared link or access link, 2) to adjust the congestion window according to the bottleneck link location, 3) to differentiate whether packet loss is due to network congestion or wireless random loss, 4) to adjust the congestion window upon receiving adaptive modulation and coding (AMC) cross-layer messages. Numerical results demonstrate that the proposed approach significantly improves the goodput with accurate packet loss identification at different loss rates. Especially, in the case of 4% packet loss rate in wireless links, the proposed approach increases goodput up to 77.25% as compared with NewReno.

Keywords: Adaptive Modulation and Coding, ARQ, Congestion control, Cross-layer, Delay-based, Loss-based, TCP.

1 Introduction

The TCP congestion control protocols, such as Reno and NewReno, have been successfully used in wired network. The traditional TCP can achieve good performance under the low bandwidth-delay product (BDP) networks. However, coming with the high-speed and high-interference LTE-A relaying network, the conservative Additive Increase Multiplicative Decrease (AIMD) mechanism would lead to slow bandwidth exploitation and blindly reduces the congestion window (*cwnd*) to the half, which cause significant waste of network bandwidth. When the High Bit Error Rate (HBER) causes random loss in wireless networks, tradition TCP misinterprets the packet losses as congestion-related resulting in unnecessary reduction of network utilization.

In LTE-A relay networks, the AMC scheme changes the available bandwidth in accordance with the User Equipment (UE)'s high mobility. However, TCP congestion

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 451–460.
DOI: 10.1007/978-3-642-35452-6_46

control cannot correctly adjust *cwnd* instantaneously; it may suffer from packet losses or bandwidth under-utilization. If an evolved Node B (eNB) and a Relay Node (RN) try to retransmit lost packets with ARQ, the TCP End-to-End delay will be increased to cause a timeout as a result of slow *cwnd* response. The high interference attributed to relaying multihop keeps packet loss going, which makes TCP misinterprets the packet losses as congestion-related and reduces *cwnd* unnecessarily.

The wireless TCP congestion control related works can be divided into three main cases. First, the cross-layer based methods [4]-[6] send extra Layer-1, Layer-2 or Layer-3 information gained from wireless nodes on the receiver's side to Layer-4. Unfortunately, these TCP are not capable of having good congestion control with the cross-layer messages about the wireless conditions. Second, the ECN based methods [7]-[9], the router status is marked with ECN. Whenever network congestion happens, the sender can determine the reason why packet loss takes place, but the cost to get the information is unaffordable. Third, the End-to-End based methods [10]-[18], these TCP congestion control algorithms perform without direct participation of the intermediate nodes, and without the need for explicit feedback about the congestion state of the network. However, the cause of network congestion is judged with only the help of RTT and packet loss. It's not easy to adjust *cwnd* accurately.

To summarize, many approaches have been proposed to address the above-mentioned problems. For improving TCP congestion control over LTE-Advanced relaying network, we propose our method motivated from the 4 points below: 1) Multi-Hop Relay suffers from high packet error rate, multipath and high-interference, traditional TCP cannot distinguish whether the lost packets are due to wired-network congestion or wireless random loss. 2) In high-mobility environment, the wireless bandwidth is adaptive to the channel state information, and the available bandwidth is varying dramatically. TCP is not able to adjust *cwnd* instantaneously; hence, the TCP has poor goodput, packet loss and serious timeouts. 3) LTE-Advanced supports high speed transmission rate. Traditional TCP's linear increasing at congestion avoidance is too slow to cause inefficient network utilization. 4) ARQ mechanism in LTE-A increases TCP end-to-end delays, which leads to slow increase on *cwnd*.

2 Network Model

In this section, we first define the TCP network model including several performance metrics, such as goodput and congestion window size. In Fig. 1, N_s^k and N_c^k denotes a server node and a client node of *k*th connection respectively. A TCP connection connected between N_s^k and N_c^k is called $C^k_{N_s^k, N_c^k}$. For example, the path of the TCP connection may consist of several physical wired links, and a single wireless link or two wireless links, including one wireless backhaul link and one wireless access link, i.e., $C^2_{N_s^2, N_c^2}$ and $C^1_{N_s^1, N_c^1}$. With considering a cross layer mechanism in this paper, first gathers the wireless network L1 channel state information (CSI) and sends the information to L4 protocol through the CL message,

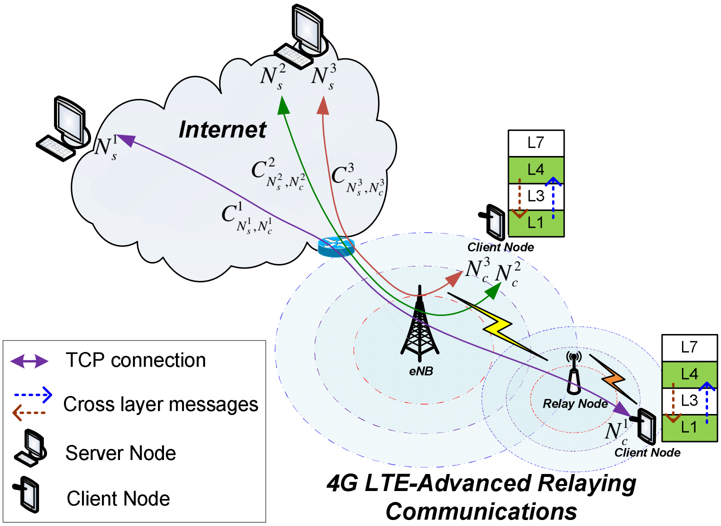


Fig. 1. An example of TCP connections in 4G LTE-A Relaying Communications

and then the L4 TCP receiver replies the TCP sender with a TCP ACK that includes the CL information.

A variety of performance metrics are used for the evaluation of congestion control mechanism: goodput and congestion window size. Since a sent packet may be dropped or lost due to network congestion or wireless loss, not all the transmitted packets will arrive at the receiver without retransmit. Therefore, the goodput is adopted to evaluate the congestion control mechanisms. The goodput is defined by

$$Goodput=\frac{Number\ of\ Transmitted\ bits\text{-}Number\ of\ \ Retransssmitted\ bits}{Transmission\ Time}\ , \qquad (1)$$

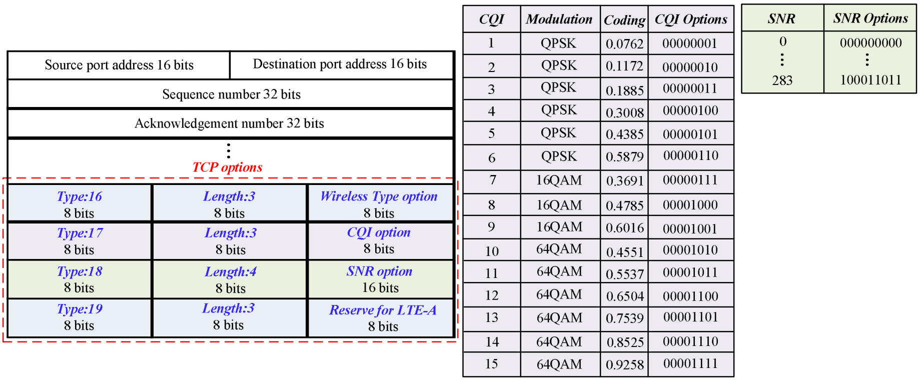
in which the numerator represents the total number of arriving bits. Higher goodput means better TCP performance.

3 The Proposed CL-Based Wireless TCP Approach

For improving the congestion control over LTE-A relay networks, we propose a cross-layer-based adaptive TCP that consists of four primary functionalities: Bottleneck Link Identification approach, Hybrid-Based Congestion Control approach, Loss Differentiate approach and Sudden Bandwidth Changes approach. **First**, Bottleneck Link Identification approach is to identify bottleneck link is on the shared link or access link. **Second**, the proposed approach TCP adjusts the congestion window according to the bottleneck link location. **Third**, loss differentiate approach is to differentiate whether packet loss is due to network congestion or wireless random loss. **Fourth**, the proposed TCP approach adjusts the congestion window upon the received cross-layer-based AMC messages. The detail description of the cross-layer-based approach is expressed as follows.

3.1 Bottleneck Link Identification Approach

According to the different modulations and code rates, it can be divided into 15 different sets of CQIs. The 15 different sets of CQIs and the corresponding SNR are shown in Fig. 2. When the TCP receives a cross-layer message from PHY, it replies to the sender with the SNR information added in the TCP ACK option fields, as indicated in Fig. 2. The CL ACK option has a unique code number and the CQI option ACK field for storing the CL value ranging from 1 to 15. When TCP sender receives ACK, we can use these information to adjust *cwnd* accurately.



Source port address 16 bits	Destination port address 16 bits	
Sequence number 32 bits		
Acknowledgement number 32 bits		
⋮ *TCP options*		
Type:16 8 bits	*Length:3* 8 bits	*Wireless Type option* 8 bits
Type:17 8 bits	*Length:3* 8 bits	*CQI option* 8 bits
Type:18 8 bits	*Length:4* 8 bits	*SNR option* 16 bits
Type:19 8 bits	*Length:3* 8 bits	*Reserve for LTE-A* 8 bits

CQI	*Modulation*	*Coding*	*CQI Options*
1	QPSK	0.0762	00000001
2	QPSK	0.1172	00000010
3	QPSK	0.1885	00000011
4	QPSK	0.3008	00000100
5	QPSK	0.4385	00000101
6	QPSK	0.5879	00000110
7	16QAM	0.3691	00000111
8	16QAM	0.4785	00001000
9	16QAM	0.6016	00001001
10	64QAM	0.4551	00001010
11	64QAM	0.5537	00001011
12	64QAM	0.6504	00001100
13	64QAM	0.7539	00001101
14	64QAM	0.8525	00001110
15	64QAM	0.9258	00001111

SNR	*SNR Options*
0	000000000
⋮	⋮
283	100011011

Fig. 2. TCP ACK Option Structure

We employ the bandwidth estimation scheme in TCP Westwood [10] as follows

$$b_k = d_k / \Delta t_k \quad , \tag{2}$$

where d_k is the amount of data in bytes that has been received by the TCP receiver in the kth InterACK, Δt_k is the difference between current and previous ACKs.

Then employ a low-pass filter to determine Bandwidth Estimation (BWE) as follows

$$BWE_{new} = \alpha \cdot BWE_{new} + (1-\alpha)\left(\frac{b_k + b_{k-1}}{2}\right) \quad . \tag{3}$$

Finally, according to BWE, we calculate the new target *cwnd* as

$$W_{new} = \frac{BWE_{new} \cdot RTT_{min}}{S} \quad , \tag{4}$$

where S is the length of a TCP segment in bytes, RTT_{min} is the minimum RTT observed.

According to BWE and cross-layer messages to determine bottleneck link is on the shared link or access link, when $BWE_{new} < BW_{CQI}$, bottleneck link is on the shared link, $BWE_{new} = BW_{CQI}$ is on the access link.

3.2 Hybrid-Based Congestion Control Approach

Upon receiving 3DupACKs, the sender calculates the new target *cwnd* according to (4). Then we use logarithm increase technique to increase the *cwnd* up to W_{max}. After the *cwnd* climbs back to W_{max} or logarithm growth function is less than linear growth function as in (5), TCP turns into hybrid-based congestion control.

$$\frac{W_{max} - W_{current}}{2 \cdot W_{current}} < \frac{1}{W_{current}} \quad . \tag{5}$$

In hybrid-based congestion control phase, considering ARQ mechanism will be activated when wireless random loss occurs. When wireless channel state becomes bad, TCP sender will receive CL-ARQ message, TCP sender will hold current *cwnd* till wireless channel state becomes better.

3.3 Loss Differentiate Approach

We use the delay variation measurement to estimate the number of buffered packets in a bottleneck queue, as follows

$$L = \frac{W_{current}}{RTT_{current}} \left(RTT_{current} - RTT_{min} \right) \quad , \tag{6}$$

where $RTT_{current}$ is the current RTT, and RTT_{min} is the minimum RTT.

We assume the loss is caused by congestion-related when L is larger than $L_{max} \cdot 0.7$, then reduce *cwnd* according to (4). Otherwise, the loss is due to wireless random loss. In this case, we will retransmit the lost segment, but should not reduce *ssthresh* or *cwnd* because the available buffer size is adequate.

3.4 Sudden Bandwidth Changes Approach

Upon receiving CL AMC messages, the available bandwidth may suffer from increasing or decreasing drastically, therefore, we divided into 3 cases, as shown in Fig. 3.

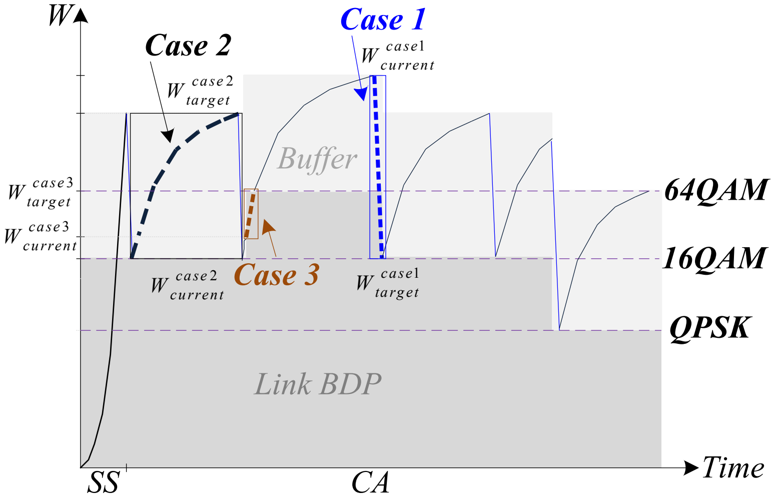


Fig. 3. The *cwnd* variations of sudden bandwidth changes

Case 1. Bandwidth sudden decrease

Although available bandwidth decreases, with considering $cwnd_{old}$ is larger than $cwnd_{new1}$, and bottleneck link queue, we have to estimate available buffer Size, l,

$$l = \left(RTT_{max} - RTT_{min}\right) \cdot BWE_{old} \quad . \tag{7}$$

When $\left(W_{new1} + l\right) < W_{old}$, it means the available bandwidth cannot handle current *cwnd*, immediately reduces *cwnd* to BDP size, $W^{case1}_{current} \leftarrow W^{case1}_{target}$, to avoid 3DupAcks or timeout.

When $\left(W_{new1} + l\right) > W_{old}$, although available bandwidth decreases, the bottleneck link buffer is still enough for flight packets, therefore, we keep *cwnd* remain.

Case 2. Bandwidth sudden increase when some segments are queued in buffer

In order to fully utilize the available buffer space meanwhile to avoid burst packet loss, we use logarithm increase technique as follows

$$W^{case2}_{current2} \leftarrow W^{case2}_{current2} + \frac{W^{case2}_{target2} - W^{case2}_{current2}}{2 \cdot W^{case2}_{current2}} \quad . \tag{8}$$

Case 3. Bandwidth sudden increase when there's no segments queued in the buffer

In order to fully utilize the available link bandwidth, increases *cwnd* to target *cwnd* in one RTT as follows

$$W^{case3}_{current} \leftarrow W^{case3}_{current} + \frac{W^{case3}_{target} - W^{case3}_{current}}{W^{case3}_{current}} \quad . \tag{9}$$

4 Numerical Results

This section evaluates the proposed cross-layer-based approach for high speed LTE-A network by comparing various performance metrics, including goodput and congestion window size. Moreover, the performance under different velocities of mobility, different wireless loss rate and sudden bandwidth changes are also evaluated.

The Network Simulator (NS-3) is adopted for all simulations with the network parameters as shown in Table 1.

Table 1. Simulation Model

Simulation parameters	Values
Number of MNs	1
Modulation Scheme	BPSK, QPSK, 16QAM, 64QAM
Buffer size	20 packets
Packet size	1500 bytes
Simulation Time	10 s

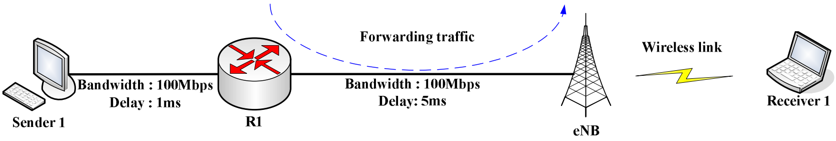


Fig. 4. Scenario 1: Various Wireless Packet Error Rates

First, Fig. 5 shows the results of goodput in scenario 1 with packet error rate ranging from 0% to 7.2%. All goodput decreases as the packet error rate increases. Our approach outperforms NewReno in goodput under various packet error rate. The main reason is that our loss differentiation mechanism can estimate occupied buffer size, that is, our approach do not blindly decrease *cwnd* even received three duplicate

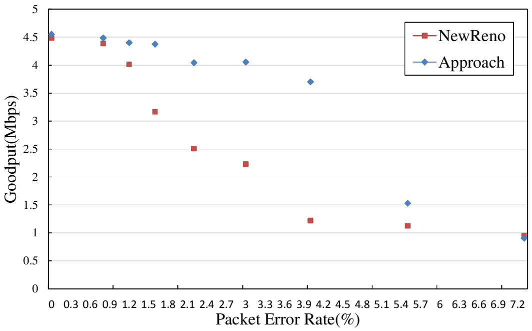


Fig. 5. Goodput of compared approach under different PER

ACKs when available buffer size is adequate. Consequently, our approach not only efficient for low packet error rate but also for high packet error rate.

Second, Fig. 6 indicates that the *cwnd* variations and goodput of our LDA approach and NewReno under 4% packet error rate. Although high packet error rate easily triggers three duplicate ACKs, our approach do not halves *cwnd* when available buffer size is adequate, hence yields the higher *cwnd* and maintains a higher goodput. However, NewReno suffers from high packet error rate and halves *cwnd* upon every three duplicate ACKs events, and thus leads to the lower *cwnd* and goodput.

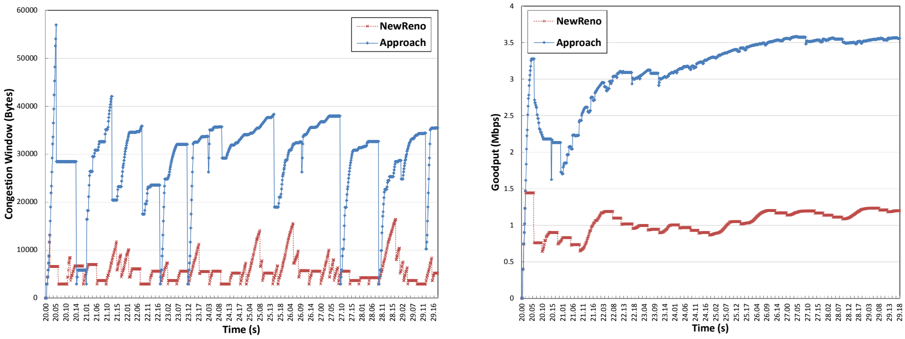


Fig. 6. The *cwnd* and goodput of LDA with NewReno under 4% packet error rate

The purpose of the scenario 2 is to evaluate whether a TCP congestion control approach can deal with sudden bandwidth changes while receiver suffers from high-mobility or high-interference and quickly determine an accurate *cwnd* for achieving high performance or not.

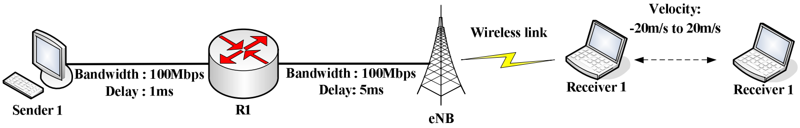


Fig. 7. Scenario 2: Sudden Bandwidth Changes

Fig. 8 shows the results of *cwnd* in scenario 2, it is observed that the proposed scheme adapts the CL-AMC information to the accurate *cwnd* after suffering from bandwidth increment at 25 seconds while NewReno take much time to probe *cwnd*. And the goodput of our proposed scheme (11.7623Mbps) is better than NewReno (11.4693Mbps).

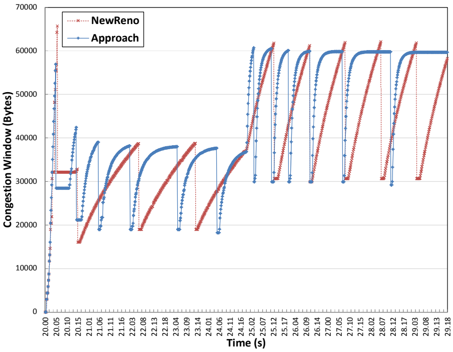


Fig. 8. The *cwnd* variations of sudden bandwidth increment

5 Conclusions

In LTE-A relaying communications, TCP suffers from high-mobility and high-interference that significantly degrades performance. Several studies adjust congestion window while receiving three duplicate ACKs or new ACKs with considering wireless conditions. Two significant issues exist. First, TCP cannot differentiate whether packet loss is due to wired-network congestion or wireless random loss and blindly halves the congestion window after receiving three duplicate ACKs, causing underutilization of the network bandwidth. Second, the AIMD mechanism leads to slow response and low throughput while suffering from sudden bandwidth changes. This paper thus proposed a cross-layer-based mechanism. **First**, a bottleneck link identification approach is to identify bottleneck link is on the shared link or access link. **Second**, hybrid-based congestion control approach is to adjust the congestion window while suffering from ARQ retransmission. **Third**, loss differentiate approach has the capability to determine whether packet loss is due to network congestion or wireless random loss, and then determines target congestion window for each case. Finally, the proposed approach adjusts the congestion window based on the received cross-layer AMC messages. Numerical results demonstrate that the proposed approach significantly improves the goodput with accurate packet loss identification at different loss rates. Especially, in the case of 4% packet loss rate in wireless links, the proposed approach increases goodput up to 77.25% as compared with NewReno.

Acknowledgements. This research was supported in part by the National Science Council of Taiwan, ROC, under contracts of NSC-101-2221-E-224-021-MY2, NSC-101-2221-E-224-022-MY3 and NSC-101-2221-E-252 -008.

References

1. TS 36.213 Version 10.5.0, Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures, 3GPP (March 2012)
2. TS 36.826 Version 10.13.0, Evolved Universal Terrestrial Radio Access (E-UTRA); Relay radio transmission and reception, 3GPP (March 2012)

3. Akyildiz, I.-F., Wang, X.: Cross-Layer Design in Wireless Mesh Networks. IEEE Transactions on Vehicular Technology 57(2), 1–6 (2008)
4. Shetty, S., Tang, Y., Collani, W.: TCP Venoplus - A Cross-Layer Approach to Improve TCP Performance in Wired-Cum-Wireless Networks Using Signal Strength. In: Conference on Networking, Sensing and Control, pp. 693–697 (April 2010)
5. Mishra, M., Sivalingam, K.M.: Enhancing TCP Performance in AMC Based Broadband Wireless Access Networks. In: IEEE International Conference on Communications, pp. 2984–2989 (May 2008)
6. Cheng, R.-S., Lin, H.-T.: A Cross-layer Design for TCP End-to-end Performance Improvement in Multi-hop Wireless Networks. Computer Communications 31(4), 3145–3152 (2008)
7. Qazi, I.A., Znati, T.: On the Design of Load Factor Based Congestion Control Protocols for Next-Generation Networks. Computer Networks 55(1), 45–60 (2011)
8. Park, E.-C., Kim, D.-Y., Lin, H., Choi, C.-H.: A Cross-Layer Approach for Per-Station Fairness in TCP over WLANs. IEEE Transactions on Mobile Computing 7(7), 898–911 (2008)
9. Lien, Y.-N., Yu, Y.-F.: Hop-by-Hop TCP over MANET. In: IEEE Asia-Pacific Services Computing Conference, pp. 1150–1155 (February 2009)
10. Kliazovich, D., Granelli, F., Miorandi, D.: Logarithmic Window Increase for TCP Westwood+ for Improvement in High Speed, Long Distance Networks. Computer Networks 52(12), 2395–2410 (2008)
11. Ha, S., Rhee, I., Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant. ACM SIGOPS Operating Systems Review 42(5), 64–74 (2008)
12. Ko, E., An, D., Yeom, I., Yoon, H.: Congestion Control for Sudden Bandwidth Changes in TCP. International Journal of Communication Systems (August 2011)
13. Zhang, K., Fu, C.-P.: An Enhancement of TCP Veno with Forward Acknowledgement. Computer Communications 31(15), 3683–3690 (2008)
14. El-Ocla, H.: TCP CERL: Congestion Control Enhancement over Wireless Networks. Wireless Networks 16(1), 183–198 (2008)
15. Chan, Y.-C., Lin, C.-L., Chan, C.-T., Ho, C.-Y.: CODE TCP: A Competitive Delay-Based TCP. Computer Communications 33(9), 1013–1029 (2010)
16. Hayes, D.-A., Armitage, G.: Improved Coexistence and Loss Tolerance for Delay based TCP Congestion Control. In: IEEE Conference on Local Computer Networks, pp. 24–31 (April 2010)
17. Liu, S., Basar, T., Srikant, R.: TCP-Illinois: A Loss- and Delay-Based Congestion Control Algorithm for High-Speed Networks. Performance Evaluation 65(6), 417–440 (2008)
18. Mehrotra, S., Li, J., Sengupta, S., Jain, M., Sen, S.: Hybrid Window and Rate Based Congestion Control for Delay Sensitive Applications. In: IEEE Global Telecommunications Conference, pp. 1–5 (December 2010)

Dynamic Load Balancing in Cloud-Based Multimedia System Using Genetic Algorithm

Chun-Cheng Lin[1] and Der-Jiunn Deng[2,*]

[1] Department of Industrial Engineering and Management,
National Chiao Tung University, Hsinchu 300, Taiwan
cclin321@nctu.edu.tw

[2] Department of Computer Science and Information Engineering,
National Changhua University of Education, Changhua 500, Taiwan
djdeng@cc.ncue.edu.tw

Abstract. This paper considers a centralized cloud-based multimedia system (CMS) consisting of a resource manager, cluster heads, and server clusters, where the resource manager assigns clients' requests for multimedia service tasks to server clusters, and then each cluster head distributes the assigned task to the servers of its server cluster. It has been a research challenge to design an effective load balancing algorithm for a CMS, which spreads the multimedia service task load on servers with the minimal cost for transmitting multimedia data between server clusters and clients under some constraints. Unlike previous works, this paper takes into account a dynamic multi-service scenario in which each server cluster only handles a specific type of multimedia tasks, and each client requests a different type of multimedia services at different time. Such a scenario can be modelled as an integer linear programming problem, which is computationally intractable in general. Hence, this paper further solves the problem by an efficient genetic algorithm. Simulation results demonstrate that the proposed genetic algorithm can efficiently cope with dynamic multi-service load balancing in CMS.

Keywords: Genetic algorithm, load balancing, cloud computing.

1 Introduction

Cloud-based multimedia system (CMS) [11] generally incorporates infrastructure, platforms, and software to support a huge number of clients simultaneously to store and process their multimedia application data in a distributed manner and meet different multimedia QoS requirements through the Internet. Most multimedia applications (e.g., aduio/video streaming services, etc.) require considerable computation, and are often performed on mobile devices with constrained power, so that the assistance of cloud computing is strongly required. Consider a centralized hierarchical CMS (as shown in Figure 1) composed of a

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 461–470.
DOI: 10.1007/978-3-642-35452-6_47

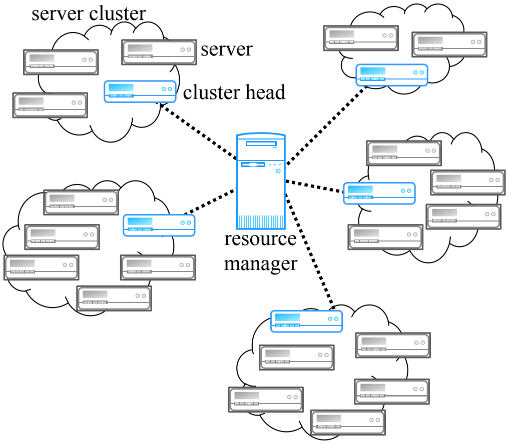


Fig. 1. Illustration of a centralized hierarchical cloud-based multimedia system

resource manager and a number of *server clusters*, each of which is coordinated by a *cluster head*. Every time when the CMS receives clients' task requests of multimedia services, the *resource manager* of the CMS assigns those requests to different *server clusters* according to the characteristics of the requested tasks. Subsequently, the *cluster head* of each *server cluster* distributes the assigned task to a server of the server cluster. It is not hard to observe that the load of each server cluster significantly affects the performance of the whole CMS. In general, the resource manager of the CMS is in pursuit of fairly distributing the task load across server clusters, and hence, it is of importance and interest to be able to cope with load balancing in the CMS.

There have existed a few previous works on load balancing for the CMS, e.g., see [3,6]. Among them, the load balancing problem for the CMS in [3] is concerned with spreading the multimedia service task load on servers with the minimal cost for transmitting multimedia data between server clusters and clients, while the maximal load limit of each server cluster is not violated. A simplified concern in their setting is to assume that all the multimedia service tasks are of only one type. In practice, however, the CMS offers multiple types of multimedia services, e.g., the QoS requirement of hypertext webpage services differs from that of video streaming services. In addition, the setting in [3] did not consider that load balancing should adapt to the time change. To respond to the practical requirements, we assume that in the CMS, each server cluster only can handle a specific type of multimedia service tasks, and each client requests a different type of multimedia services at different time. At each time step, we model such a problem as an integer linear programming formulation, which is computationally intractable in general [1]. Conventionally, intractable problems are usually solved by metaheuristic approaches, e.g., simulated annealing [5], genetic algorithm [2], particle swarm optimization [4,7], etc. In this paper, we propose a genetic algorithm (GA) for the concerned dynamic load balancing problem for the CMS. The experimental results show that to a certain extent, our approach is capable of dynamically spreading the multimedia task load equally.

2 Problem Description

2.1 System Overview

This paper considers the centralized CMS consisting of a *resource manager* and a number of *server clusters* each of which is coordinated by a *cluster head*, as illustrated in Figure 1. Once receiving the clients' requests for multimedia service tasks, the *resource manager* of the centralized CMS stores the global service load information collected from *server clusters*, and decides the amount of client's multimedia service requests assigned to each server cluster so that the load of each server cluster is as balanced as possible, in term of the cost of transmitting multimedia data between server clusters and clients. The decision of assignment is based upon the characteristics of different service requests and the information collected from server clusters.

2.2 Problem Formulation

To formulate that our CMS can adapt to the time dynamics, we assume time to be divided into different time steps. At the t-th time step, the CMS can be modelled as a complete weighted bipartite graph $G_t = (U, V, E, \phi, \psi^t, q, r^t, w^t)$ in which

- U is the set of vertices that represent the server clusters of the CMS;
- V is the set of vertices that represent clients;
- E is the set of edges between U and V, in which each edge $e_{ij} \in E$ represents the link between server cluster $i \in U$ and client $j \in V$;
- $\phi : U \to \mathbb{N}$ is a function used to represent that server cluster i only can cope with multimedia tasks of type ϕ_i;
- $\psi^t : V \to \mathbb{N}$ is a function used to represent that client j requests the multimedia service of type ψ_j^t at the t-th time step;
- $q : U \cup V \to \mathbb{N}$ is a function used to represent that server cluster i can provide multimedia service of QoS q_i;
- $r^t : U \cup V \to \mathbb{N}$ is a function used to represent that client j requests the multimedia service of QoS requirement r_j^t at the t-th time step;
- $w^t : E \to \mathbb{R}^+$ is the weight function associated with edges, in which w_{ij}^t denotes the w^t value that represents the cost for transmitting multimedia data between server cluster i and client j, which is defined as follows:

$$w_{ij}^t = \begin{cases} \infty, & \text{if } d_{ij}^t \to \infty \text{ or } \phi_i \neq \psi_j; \\ d_{ij}^t l_{ij}^t, & \text{otherwise.} \end{cases} \qquad (1)$$

where d_{ij}^t is the network proximity between server cluster i and client j; l_{ij}^t is the traffic load of the link between server cluster i and client j that is defined as follows:

$$l_{ij}^t = \sum_{k \in K_i} u_{ikj}^t S_{ik} \tag{2}$$

where K_i is the set of servers in server cluster i; u_{ikj}^t is the server utilization ratio of server k in server cluster i due to client j, and S_{ik} is its capacity;

Note that the proximity d_{ij}^t between server cluster i and client j in Equation (1) is required to be measured at every time step due to dynamic network topology change. This paper continues to use the setting of [3] based upon the *distributed binning scheme* in [9].

With the above notations, the mathematical model of our concerned problem at the t-th time step can be stated as the following integer linear programming formulation:

$$\begin{aligned}
\text{Minimize} \quad & \lambda \frac{\sum_{i \in U} \sum_{j \in V} x_{ij}^t w_{ij}^t}{\sum_{j \in V} (\max_{i \in U} \{w_{ij}^t\})} \\
& +(1-\lambda)(1 - \frac{\sum_{j \in V} \sum_{i \in U} x_{ij}^t}{|V|}) && (3) \\
\text{subject to} \quad & \sum_{i \in U} x_{ij}^t \le 1, \forall j \in V, && (4) \\
& \sum_{j \in V} x_{ij}^t l_{ij}^t \le \sum_{k \in K_i} S_{ik}, \forall i \in U && (5) \\
& x_{ij}^t \phi_i = x_{ij}^t \psi_j^t, \forall i \in U, j \in V && (6) \\
& x_{ij}^t q_i \ge x_{ij}^t r_j^t, \forall i \in U, j \in V && (7) \\
& x_{ij}^t \in \{0,1\}, \forall i \in U, j \in V && (8)
\end{aligned}$$

where x_{ij}^t is an indicator variable defined as follows:

$$x_{ij}^t = \begin{cases} 1, \text{ if client } j \text{ is assigned to server cluster } i \\ \quad \text{at } t\text{-th time step;} \\ 0, \text{ otherwise.} \end{cases} \tag{9}$$

In the above model, indicator variable x_{ij}^t (see Equation (8)) is used to determine whether to assign the link e_{ij} between server cluster i and client j in the complete bipartite graph $U \times V$. The objective (3) of the model is a weighted sum of two terms: the first is to minimize the total weighted values of the bipartite graph, i.e., to minimize the total cost of transmitting multimedia data at the t-th time step, while the second is to maximize the number of link assignments. Note that the denominators of the two terms of the objective are used for normalizing them to the range $[0,1]$, and $\lambda \in [0,1]$ is used to adjust the weights of the two terms, so that the objective value always falls into the range $[0,1]$. Constraint (4) guarantees that each client only allows at most one link to be assigned. For each client j in V, the constraint enforces that x_{ij}^t of at most one server cluster i is 1. Constraint (5) enforces that the utilized capacity of each server cluster cannot exceed its capacity at the t-th time step. Constraint (6) enforces that the multimedia service type requested by each client j is consistent with that provided by server cluster i. Constraint (7) enforces that each client j requests the multimedia server of the QoS no more than that offered by server cluster i.

Our concerned problem (Dynamic Multi-Service Load Balancing in Cloud Multimedia System (CMS-dynRMLB)) is to solve the above mathematical programming problem at each time step. Since the problem at each fixed time step can be modelled as an integer linear programming problem as mentioned above, it is computationally intractable in general [1], i.e., there does not exist an efficient polynomial time algorithm for the problem. Hence, this paper proposes a genetic algorithm (GA) [2] for solving the problem in the following section.

3 Our Genetic Algorithm with Immigrant Scheme

Our dynamic load balancing algorithm based on the GA is given in Algorithm 1, which is explained as follows. At each time step t, Algorithm 1 iterates on t to reallocate the network load assignments to adapt the time change. At first, Line 2 constructs a complete weighted bipartite graph G_t, described in the previous section. Subsequently, we remove infeasible cases and then apply the GA to finding a locally optimal load assignment solution. Note that feasible solutions are restricted to Constraints (4), (5), (6), and (7). In Algorithm 1, Line 3 removes the links in G_t violating Constraints (6) and (7), while the other two constraints will be considered in the GA. Before using the GA to calculate solutions, the information of $\{l_{ij}^t\}$ and $\{w_{ij}^t\}$ is required, so Line 4 of Algorithm 1 calls Algorithm 2 to obtain those information. After that, Line 5 of Algorithm 1 calls the algorithm detailed in Algorithm 3 to compute our final load assignment solution.

Algorithm 1. Dynamic Load Balancing Algorithm

```
1: for $t = 1, 2, \cdots$ do
2:     consider complete weighted bipartite graph $G_t$
3:     remove the links in $G_t$ violating Constraints (6) and (7)
4:     calculate $\{l_{ij}^t\}$ and $\{w_{ij}^t\}$ by calling Algorithm 2
5:     assign $\{x_{ij}^t\}$ by calling Algorithm 3
6: end for
```

3.1 Basic Definitions

To use GA to solve the CMS-dynRMLB problem, we first define the chromosome and fitness function of GA for the problem as follows. A solution for the CMS-dynRMLB problem consists of all the indicator variables $\{x_{ij}^t | \forall i \in U, j \in V\}$.

Algorithm 2. Calculate Weights

```
for each client $j \in V$ do
    measure the latency from client $j$ to each landmark
    compute the landmark order $\ell_j$ of client $j$
    obtain the set of available server clusters $U_j$
    for each $i \in U_j$ do
        measure the latency from server cluster $i$ to each landmark
        compute the landmark order $\ell_i$ of server cluster $i$
        if $\ell_i = \ell_j$ then
            measure the network proximity $d^t_{ij}$ between server cluster $i$ and client $j$
            measure server utilization ratios $u^t_{ikj}$ for all $k \in K_i$
            calculate $l^t_{ij}$ and $w^t_{ij}$ by Equations (2) and (1), respectively
        else
            $w^t_{ij} = l^t_{ij} = \infty$
        end if
    end for
end for
```

Algorithm 3. Genetic Algorithm (graph G_t, time step t)

```
if $t = 1$ then
    generate and evaluate the initial population $P^t_0$ of size $\eta$ in which each chromosome has to satisfy Constraints (4) and (5).
else
    $P^t_0 \leftarrow P^{t-1}_\tau$
end if
$i \leftarrow 0$
while $i < \tau$ or the convergent condition is not achieved do
    select the parental pool $Q^t_i$ from $P^t_i$
    reproduce a new population $P^t_i$ of size $\eta$ by performing crossover procedure on pairs of chromosomes in $Q^t_i$ with probability $p_c$
    perform mutation procedure on chromosome in $P^t_i$ with probability $p_m$
    repair each infeasible chromosome in $P^t_i$
    evaluate $P^t_i$
    if $r_e > 0$ then
        a number of the best chromosome $E^t_{i-1}$, called elite, are selected from the previous generation $P^t_{i-1}$
        generate and evaluate $r_e \cdot \eta$ elite immigrants by mutating $E^t_{i-1}$ with $p^e_m$
    end if
    if $r_r > 0$ then
        generate and evaluate $r_r \cdot \eta$ random immigrants
    end if
    replace the worst chromosomes in $P^t_i$ with the above elite and random immigrants
    $P^t_{i+1} \leftarrow P^t_i$
    $i \leftarrow i + 1$
end while
output the best found chromosome as the solution at the $t$ time step
```

The solution is represented by a chromosome in GA. Recalling that $|U| = m$ and $|V| = n$, the solution for indicator variables $\{x^t_{ij}\}$ is encoded as a sequence of decimal numbers of length n:

$$\langle \sigma_1, \cdots, \sigma_j, \cdots, \sigma_n \rangle \tag{10}$$

where $\sigma_j \in \{0, 1, 2, \cdots, m\}$ represents the link assignment of client j with the following two cases:

- if $\sigma_j = 0$, then each $x^t_{ij} = 0$ for any $i \in U$, i.e., client j is not linked at the t-th time step;
- otherwise, $\sigma_j = k \neq 0$, meaning that $x^t_{kj} = 1$ and $x^t_{ij} = 0$ for any $i \neq k$, i.e., client j is linked to server cluster k uniquely.

That it, all the x^t_{ij} values for a fixed j can be determined based on the value of σ_j. By doing so, it is guaranteed that $\sum_{i \in U} x^t_{ij} \leq 1, \forall j \in V$ (Constraint (4)).

For making the represented solution to be feasible, we require the chromosome to satisfy all the problem constraints. Since the cases violating Constraints (6) and (7) have been removed in Line 3 in Algorithm 1, the only remaining concern to guarantee the solution feasibility of the chromosome is to satisfy Constraint (5). Since computation of Constraint (5) requires the information of $\{l^t_{ij}\}$, the constraint is checked in Algorithm 3, after calling Algorithm 2 to calculate $\{l^t_{ij}\}$. For those chromosomes violating Constraint (5), we apply the *repairing* procedure to revise them to be feasible in Line 11 of Algorithm 3. The *repairing* procedure will be detailed in the next subsection.

As for fitness function, we directly let the objective (3) of our concerned problem as our fitness function.

3.2 Main Components of Our GA

For initialization, Lines 1–5 of Algorithm 3 consider two cases for producing the initial population at each time step. If $t \neq 1$, we just take the final population at the previous time step as the initial population at the current time step (see Lines 4 of Algorithm 3). For the other case (i.e., $t = 1$), we randomly produce the initial population by using Algorithm 4.

For the evolving operators in our GA, we continue to use the conventional roulette wheel selection, one-point crossover operation, and mutation. Interested readers can be referred to [2] for the details on roulette wheel selection and one-point crossover. After executing those evolving operators, the modified chromosomes may become infeasible. Hence, we need to repair them to be feasible. Since the truth of Constraint (4) can always be guaranteed in our design of chromosomes, and the infeasibility of Constraint (6) and Constraint (7) has been filtered in preprocessing stage of the GA, thus we only need to repair those chromosomes that violate Constraint (5). In fact, the repairing operation has been used in initializing the chromosomes in Algorithm 4 (see Lines 6–21).

Algorithm 4. Initialize

```
for $j = 1$ to $n$ do
    $arg(U_j) \leftarrow$ the set of integers be the indices of the available cluster servers $U_j$ that are linked with client $j$ in the reduced bipartite graph $G_t$ (where the links violating Constraints (6) and (7) have been removed)
end for
for $p = 1$ to $\eta$ do
    construct the $p$-th chromosome $c_p = \langle \sigma_1, \cdots, \sigma_j, \cdots, \sigma_n \rangle$, in which $\sigma_j$ is a number chosen arbitrarily from $arg(U_j)$ for each $j \in \{1, \cdots, n\}$
    let $o$ be the set of the positions of 0's in $c_p$ that do not violate our problem constraints
    for $i = 1$ to $m$ do
        scan chromosome $c_p$ to compute the $\sum_{i \in U} x_{ij}^t l_{ij}^t$ value for server cluster $i$
        $g_i \leftarrow \sum_{i \in U} x_{ij}^t l_{ij}^t - \sum_{k \in K_i} S_{ik}$
        let $o_i$ be the set of $o$ associated with cluster server $i$
        while $g_i > 0$ do
            find the index $x$ such that $\sigma_x = i$ in $c_p$, $l_{ix} > g_i$, and the gap between $l_{ix}$ and $g_i$ is the smallest
            if there is a position $o_y$ in $o$ that can accommodate $l_{ix}$ and satisfy all the problem constraints then
                swap the values of $o_y$ and $\sigma_x$
                remove $o_y$ in $o$
                $g_i \leftarrow g_i - l_{ix}$
            else
                $\sigma_x \leftarrow 0$
            end if
        end while
    end for
end for
```

4 Implementation and Experimental Results

We consider an instance where there are 20 server clusters ($m = 20$) and 100 clients ($n = 100$). The weight of each link is bounded in range $[0, 5]$ in general. That is, the normalizing factor of the first term in Objective (3) is $5 \cdot 100 = 500$, while that of the second term is 100. If the link is infeasible, its weight is set 1000, which is viewed as infinity in our experiments.

In our experiments, unless otherwise described in the rest of this paper, our GA algorithm applies the following parameter settings: $\lambda = 0.7$, $\tau = 500$, $\eta = 50$ (i.e., there are 500 generations at most, and there are 50 chromosomes in a generation); the time period between two time steps is the time taken by 50 generations of the main loop of the GA algorithm. That is, clients move at each time step, and their corresponding criteria are measured at every 50 generations. The reason why we use $\lambda = 0.7$ is from a lot of tests.

Our simulation was tested on an Intel Core i7-3770 CPU @ 3.40 GHz with 16 GB memory. The average running time for determining a placement of an

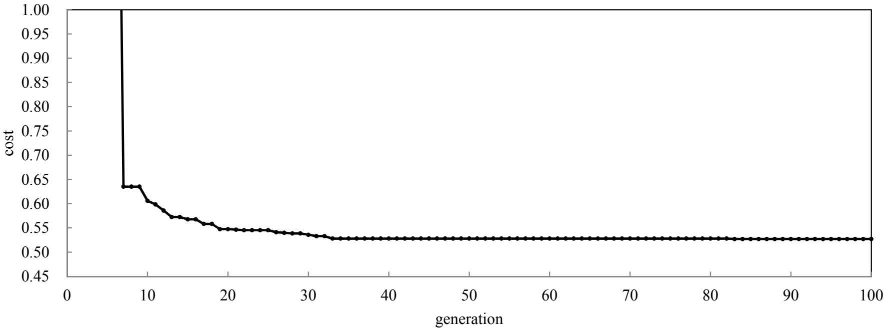


Fig. 2. Static scenario

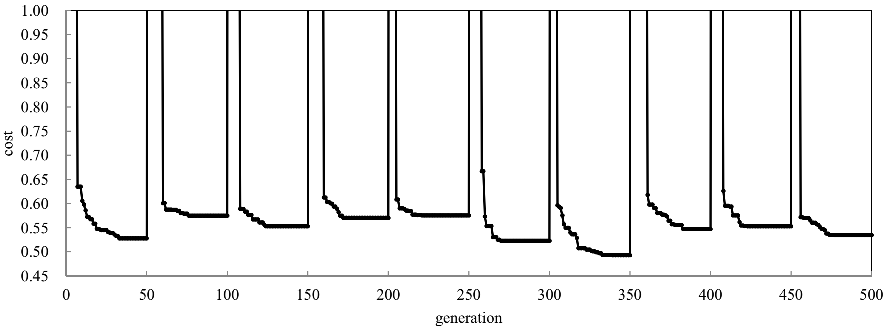


Fig. 3. Dynamic scenario

instance (i.e., 50 generations) is about 0.00125 seconds. It implies that our GA has the ability to efficiently cope with the CMS-dynRMLB problem.

In order to observe the convergence of the best cost values in our GA method, we plot the best cost values versus the number of generations of our GA for the test instance in Figure 2, from which we can observe that the best cost value converges at the 33-th generation, and there is no drastic modification after the 33-th generation. Hence, we choose 50 generations to be executed within a time step. In addition, we observe that the best cost values of the first six generations are out of the range $[0, 1]$, which tend to be convergent later. It implies that our GA has the ability to make the solutions to be convergent.

In order to demonstrate the ability of our approach to adapt the time changes (where we suppose that the topology graph changes in each 50 generations), we run 500 generations of our GA algorithm on the test instance in a dynamic scenario, and its plots of best cost versus the iteration number are given in Figure 3. The dynamic scenario assumes that all of the clients change their locations in each 50 generations in Figure 3, from which we observe that every when clients change their location in each 50 generations, the cost value goes to infinity, and later converges by our GA scheme.

5 Conclusion

A GA approach for the CMS-dynRMLB problem has been proposed and implemented. The main difference of our model from previous models is that we consider a practical multi-service dynamic scenario in which at different time step, clients can change their locations, and each server cluster only handles a specific type of multimedia tasks. Our main contribution in this paper is to propose a mathematical formulation of the CMS-dynRMLB problem, so that interested readers are able to investigate the problem precisely in the future. The performance of our GA approach is evaluated by detailed simulation. In the future, we intend to to propose some variants of GA or the GA mixed with other techniques to improve the computational performance, e.g., the scheme of parameters in GA [8], fuzzy GA [10], among others.

References

1. Garey, M., Johnson, D.: Computers and Intractability - A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
2. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
3. Hui, W., Zhao, H., Lin, C., Yang, Y.: Effective load balancing for cloud-based multimedia system. In: Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, pp. 165–168. IEEE Press (2011)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press (1995)
5. Kirkpatrik, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220, 671–680 (1983)
6. Nan, X., He, Y., Guan, L.: Optimal resource allocation for multimedia cloud based on queuing model. In: Proceedings of 2011 IEEE 13th International Workshop on Multimedia Signal Processing (MMSP 2011), pp. 1–6. IEEE Press (2011)
7. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73. IEEE Press (1998)
8. Xiong, Y., Golden, B., Wasil, E.: A one-parameter genetic algorithm for the minimum labeling spanning tree problem. IEEE Transactions on Evolutionary Computation 9(1), 55–60 (2005)
9. Yang, L., Guo, M.: High-performance Computing: Paradigm and Infrastructure. John Wiley and Sons (2006)
10. Zhang, X., Hu, S., Chen, D., Li, X.: Fast covariance matching with fuzzy genetic algorithm. IEEE Transactions on Industrial Engineering 8(1), 148–157 (2012)
11. Zhu, W., Luo, C., Wang, J., Li, S.: Multimedia cloud computing: An emerging technology for providing multimedia services and applications. IEEE Signal Processing Magazine 28(3), 59–69 (2011)

An InfO-Based Content Sharing System for Small Communities

Y.L. Tai[1,*], S.R. Tsai[2], C.K. Wen[3], K.F. Ssu[1], and Y.S. Huang[1]

[1] Institute of Computer and Communication Engineering,
National Cheng Kung University, Taiwan, R.O.C.
{n2892139,q3895113}@mail.ncku.edu.tw, ssu@ee.ncku.edu.tw

[2] Department of Information Management Chang Jung Christian University, Taiwan, R.O.C.
srtsai@mail.cjcu.edu.tw

[3] Department of Information Management, Tainan University of Technology, Taiwan, R.O.C.
ckwen@venus.tut.edu.tw

Abstract. There is no doubt that the Web is a very successful system on the Internet for content sharing. In the Web system a Web page is a basic unit for content providers to organize their contents. However, in an application, using a single Web page is often not adequate to represent the complete information. How many information entities must be included in a unit to represent adequate information basically depends on the requirements of an application. From another point of view, in current information applications on the Web system, the data and the relevant operations are not bound together. Consequently, if the content units are to be modified, moved, or extended in functionalities, the designers or the users would encounter some inconvenience or problems. We apply the information object model to construct a community object system to share content for virtual communities on the Internet. A community object is basically an information object that is portable and extensible in functionality. The information object model we propose in this paper is simple but powerful for presenting both contents and users. The simple and uniform mechanisms of method managements in the information object model can be used to fulfill many various requirements in developing applications for the Internet services.

Keywords: Information Object, Social Network, Information Sharing, Content Portability.

1 Introduction

There is no doubt that the Web is a very successful system on the Internet for content sharing. In the Web system a Web page is a basic unit for content providers to organize their contents. However, in an application, using a single Web page is often not adequate to represent the complete information. How many information entities must be included in a unit to represent adequate information basically depends on the requirements of an application. It also depends on the subjective perception of the application designers. From another point of view, in current information applications

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 471–479.
DOI: 10.1007/978-3-642-35452-6_48

on the Web system, the data and the relevant operations are not bound together. Consequently, if the content units are to be modified, moved, or extended in functionalities, the designers or the users would encounter some inconvenience or problems. Related researches propose object-oriented methodologies as possible solutions for those issues [1,2,3,4]. Let's look at one common example, the photo sharing Web site. There are many photo sharing Web sites on the Internet. Most photos sharing Web sites purely emphasize on the sharing of photos. They provide albums as the containers to save users' photos. Users can create an album and save photos in an album. They can play the photos with the slide show function supported by the Web sites.

To cope with the various requirements in applications, we thus propose the information object, i.e. InfO, as the basic unit for constructing the information system. An InfO is an entity that encapsulates the required data and methods in a unit. The methods of an InfO can be flexibly installed or changed as needed. The application designers can organize any data that are required within an InfO. Meanwhile, the designers can develop the object methods that fulfill the required operations to be applied on the object's data or the required functions or services relevant to the object. The data and the methods are bound as a single unit managed by the system. In the design of the InfO system, we carefully investigate into aspects in core system support and various application requirements. The basic applications we target at are around the information systems that provide long-term useful information sharing and information aggregation. We have presented some basic ideas and important features of the information object system[5,6]. We also apply the information object model to construct a community object to organize a small and personalized virtual community[7].

In this paper, we apply the InfO model to build the InfO-based Content Sharing System for communities, i.e. the ICSS. The contents to be shared are organized as information objects. We provide three basic information objects including GoodyInfO, MemberInfO, and CommunityInfO. The members in the community can exchange and share information by using the three basic objects which we think adequate in expressing various types of content sharing for a small community in practice. We will present the InfO model in Section 2. The design and implementation of the ICSS will be discussed in Section 3 and Section 4 separately. Section 5 comes the conclusions.

2 The Information Object Model

We will investigate on the design and its applications of customer methods in an information object. The design of the InfO system involves two aspects. One is the core system design; the other is on application specific design. An extensible core system design can apply to all applications that follow the concept of information objects. In this section, we will discuss the design of the information object model as well as the issues on the design of methods.

2.1 The Design Considerations of an Information Object

An information object has some important characteristics: self-containedment, extensibility, portability. In the design of an information object system, how to reveal these characteristics is a major issue. In the following we discuss the design considerations for achieving the goals.

1. Keeping data items in their original format

   In our design, the data items included in an information object are kept in their original format. Such design will be helpful in managing and preserving information. We can use tools in operating systems to manipulate the data items in information objects. It also facilitates the reuse of data items in information objects.

2. Encapsulating data and operations in the same unit

   We propose the information object model for preserving long term valid information. For this purpose, it would be proper and even mandatory to encapsulate the related and required data items in a single structured unit. In addition to data, we also encapsulate the related operations in the same unit in order to process data items correctly.

3. A consistent and well defined format for data unit

   Imposing too many data formats would restrict the applicable applications. However, some basic formats or structures should be defined in the core system design for unifying the information objects. Besides, it would be helpful for the InfO to extend in data and operations with regular structure design.

4. Using the Web as the platform

   Information objects have a variety of media packaging capabilities and can compose most of the information formats. The Web is a wide range of applications system which is cross-platform. Using the Web as the platform would make it easy for more different devices to share the InfO information. Therefore, we use web technology to construct the InfO and use a web browser to read the InfO as simple as operating the files in the system.

Figure 1 shows the model of the InfO information unit. An information unit is consisted of four components, i.e. Main, Data, Metadata, and Method. Users and Managers utilize and supervise the InfO contents through the Main component that is the entry point of an InfO. The following will explain those components in details.

1. Main

The Main component mainly provides the entry point for users and managers to access and administrate the information contents. Typically, when an InfO is opened, the Main component will build a list of a set of Methods to guide the user for further operations.

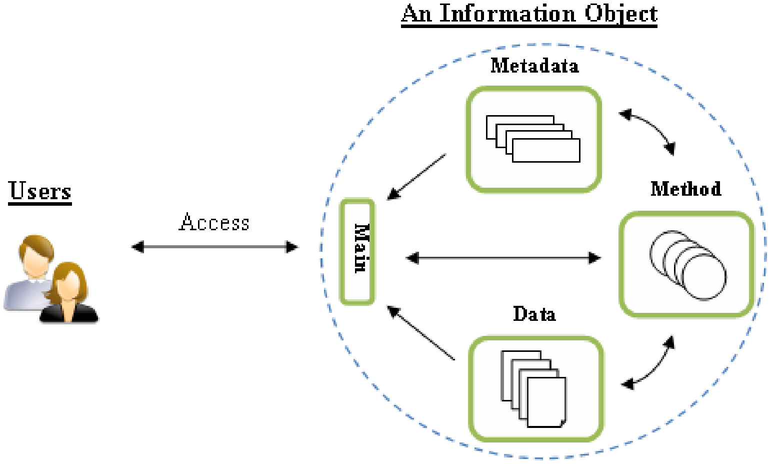


Fig. 1. The Information Object Model

2. Method

The Method is one of the key features in our design. The Method has the capabilities to process InfO data and presents output to the user. The Manager Method is one of the System Methods that can insert, remove, and update and other Methods. We apply Web technologies to implement many useful Methods. Some Methods that are System Methods involve server-side techniques such as PHP, Python, Perl, and CGI, while the other Methods that are designed for users to brows information contents implement JavaScript, CSS, and HTML on user browsers. We carefully organize the InfO structures with regularity in order to increase the degree of code reentrance of the Methods.

3. Metadata

The component of Metadata is a container for the metadata that describes the operations of an InfO. A typical Metadata component keeps two types of Metadata in it. One is InfO Metadata that describes the InfO itself attributes, while the other is Method Metadata that describes all the methods in an InfO. An InfO Metadata has standard and Extended Metadata. The items of Standard Metadata are necessary for an InfO, while the Extended Metadata is used to add more items for application needs. The main concern of designing both InfO Metadata and Method Metadata is to enhance the extensibility of Metadata in order that the InfO would be easily applied to more situations. To achieve this purpose the formats and structures of Metadata should be simple and easy handling with programming tools in the Web environment.

4. Data

The Data component is the container for user and application information. Typically a Data container would have many data items that could be any one of text files, photo files, video/audio files, or web pages. The Method processes those data items and returns the results to the InfO user.

2.2 The Design Issues of Methods

In some cases, we can simply install a customer method to an information object to fulfill the requirements in applications. The function of the customer method is an access operation to the information object to be shared. In this section, we will discuss the design concepts of the InfO information unit and the structure model for information preservation. We will investigate related design issues of Methods in the InfO.

A method of an InfO is triggered to execute either when the InfO is opened or when a user clicks on an item from the Method list. A Method is loaded by the Method Loader in the Main component fro execution when the InfO is opened. The former users the Method to provide the features of default information presentation and data processing, while the latter lets users make decisions themselves on how to use the information. Form the purpose of triggering a Method on the right time, the Method Loader of the Main component is used to handle the loading and execution of a Method. The features of the Method Loader will be stated as follows.

- Method Loading

Only those Methods with matched ClassID values of Method Metadata attribute to the InfO can be loaded and executed in the InfO. The Method Loader checks ClassID values to ensure that the matching rule can be complied. The Method Loader also checks current execution environment for the InfO and disables those Methods that could not execute correctly.

- Method Triggering

There are two execution modes for a Method. The Trigger attribute of Method Metadata is used as the key value to distinguish between these two modes. Mode 1 that
the Trigger is enabled that the Method will be loaded and executed right away when an InfO is opened. Mode 2 that the Trigger value is disabled causes the Method Loader to load but not execute immediately the Method and generate a Method list for users.

3 The Design of the ICSS

We will describe the basic concept of the member object and the community object as well as the design of the prototype system, i.e. the ICSS, developed on the Web platform. The ICSS is a content sharing system based on the InfO. In this Section, we will discuss the ICSS system architecture, content sharing model, and components design.

3.1 The System Architecture of the ICSS

The relationship of the ICSS is shown in Figure 2. A CommunityInfO is a group that content would be shared among all members. A MemberInfO represents a member in the same group. A GoodyInfO is a basic information unit for sharing in the community. The three components stated above are all designed based on the InfO model. The CommunityInfO offers features of member management and public methods for a group. The MemberInfO maintains personal profile and keeps those GoodyInfO contents.

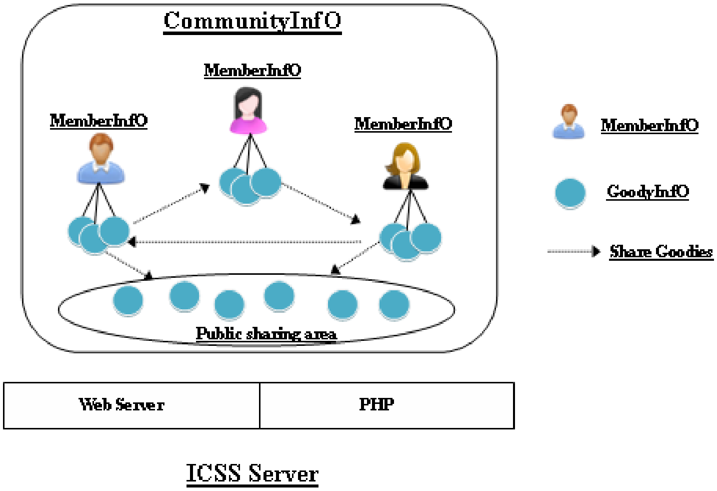


Fig. 2. The ICSS system architecture

The GoodyInfO can simply be played in content and be extended in functionalities by using Method mechanism in the GoogyInfO. For example, A CommunityInfO represents a class of graduated students. The MemberInfOs then would be those students in the class. When the class celebrates an anniversary with an elaborate party, the GoodyInfo can be used to accommodate many activities photos as its content and then to playback or slideshow those photos with background music for memories.

3.2 The Three Application InfOs in the ICSS

The GoodyInfO is an InfO that designed to encapsulate sharing content. A GoodyInfO can be propagated in a community under monitoring by the ICSS. The content of a GoodyInfO can be retrieved by the Methods come with it. Currently, there are three basic application types for the GoodyInfO, i.e. Album GoodyInfO, Discussion GoodyInfO, and Vote GoodyInfO. The Album GoodyInfO provides sharing of favorite photos in the community. The Discussion GoodyInfO offers a bulletin board to all members while the Vote GoodyInfO may issue topics for members to participate voting activities.

A member of the community is represented as a MemberInfO. A MemberInfO has a profile for member to keep personal description information. The management of GoodyInfOs that no matter is private or shared from the others is maintained in the MemberInfO.

A CommunityInfO is the administrator of a community. The CommunityInfO manages members and provides interchange of GoodyInfOs between members in a community. A public storage that is managed by the CommunityInfO accommodates those contents pushed by members for sharing. For example, in addition to the GoodyInfO, members may also push Discussion GoodyInfO or Vote GoodyInfO to the sharing area to invite members for discussions or voting on some interesting topics.

4 The Implementation and Demonstration of the ICSS

In this Section, we will describe the implementation of the ICSS with GoodyInfO, MemberInfO, and CommunityInfO.

4.1 The GoodyInfO

Figure 3 shows the organization of the Album GoodyInfO which is an information unit based on the InfO. The content of an Album GoodyInfO are photos and background music in the Data folder, an album.json that keeps all the metadata the AlbumInfO need in Metadata folder. The metadata includes description, Name, Owner and the list of all photos and background music. When the AlbumInfO being playing, the PlayBack Method will retrieve those photos and music from the album.json file.

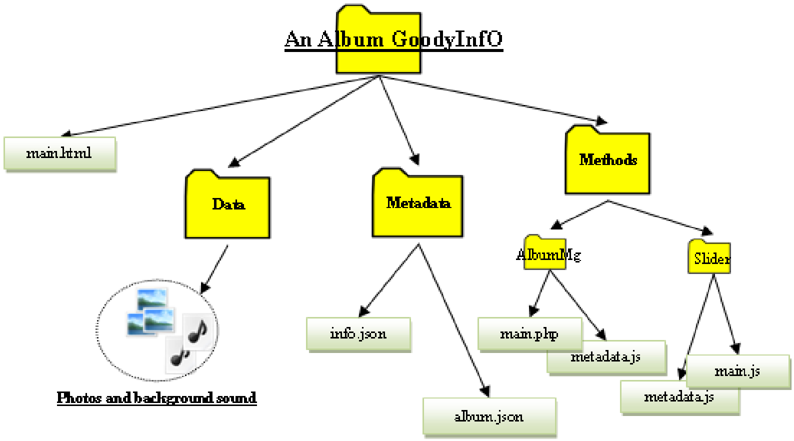


Fig. 3. The organization of the Album Goody InfO

There are two main Methods in the Album GoodyInfO. One is AlbumMgr that is a Server-side Method and provides the owner to manage photos and music, while the other one is Slider that is a browser Method and is used to play photos and background music.

4.2 The MemberInfO

A MemberInfO keeps the information and related operations of the member profiles of a community, the Goody InfOs that member owns, and the Goody InfO shared from other members. Figure 4 is the organization of the Member InfO. Personal and shared GoodyInfOs are stored in the Data folder. There are two Server-side Methods. One is ProfileMgr that provides those community members to edit personal profiles. The other Method is GoodyMgr that used by community members to operate on those GoodyInfOs owned by themselves or shared from other members.

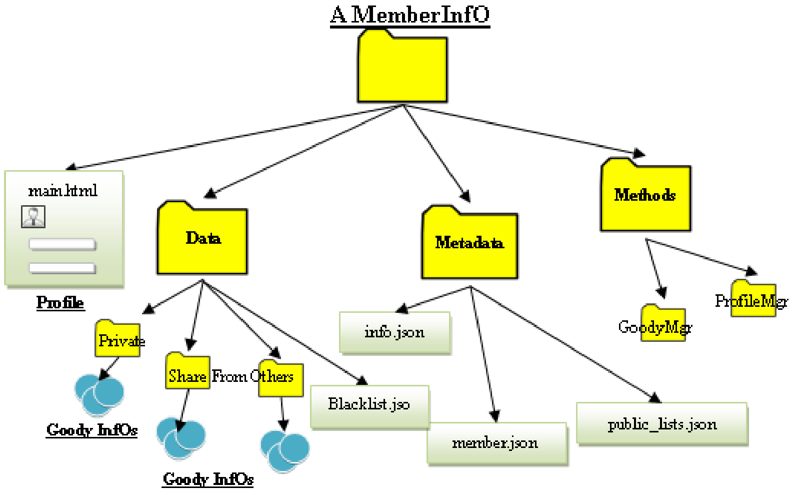


Fig. 4. The organization of the Member InfO

4.3 The Community InfO

To exchange contents among members in a community is an important feature in the ICSS. The GoodyInfO exchange features are implemented as two operations, URL sharing and full clone, in the CommunityInfO. To achieve URL sharing, the CommunityInfO encapsulates the URL of the target GoodyInfO for sharing in a RedirectInfO and copies the RedirectInfO to the target MemberInfO. The RedirectInfO will cause URL redirection occurred on the target Member's browser.

5 Conclusions

In this paper we began by introducing the varieties of challenges regarding intricate information management for social network and how the information object can be applied to help achieving the goals. We first introduced the information object model to represent information content and operations, as well as ways to migrate and reuse the content across various social web sites and personal applications. Then we demonstrated how the information model can be applied to construct the ICSS for

Internet communities, as well as to port and share diverse contents that contributed by users in the community. Consequently, the ICSS prototype implemented in this paper shows that the information object model is simple but powerful for presenting both contents and users.

References

1. Janée, G., Mathena, J., Frew, J.: A data model and architecture for long-term preservation. In: International Conference on Digital Libraries, pp. 134–144 (June 2008)
2. Lin, J.: An object-oriented development method for Customer Knowledge Management Information Systems. Knowledge-Based Systems 20, 17–36 (2007)
3. Peter, W., Jeremy Leighton, J., Ian, R.: The personal curation of digital objects: A lifecycle approach. Aslib Proceedings 61(4), 340–363 (2009)
4. Bernstein, M.S., Marcus, A., Karger, D.R., Miller, R.C.: Enhancing directed content sharing on the web. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, Atlanta, pp. 971–980. ACM, Georgia (2010)
5. Lee, C.-M., Tsai, S.-R., Huang, Y.-S., Tai, Y.-L., Huang, G.-H., Jiau, H.-J.: Information Objects Encapsulating Multimedia and Intelligence. In: 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP, pp. 993–997 (2009)
6. Tsai, S.R., Huang, Y.S., Dai, U.L., Huang, G.H., Lee, C.M., Fang, J.Y., Chen, Y.T., Lee, J.N.: An Object-Based Web System for Building a Virtual Community. In: 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 414–417 (2010)
7. Huang, Y.S., Tsai, S.R., Lee, C.M., Huang, G.H., Dai, U.L., Jiau, H.J.C.: A Portable and Extensible Community Object. In: 2011 International Conference on Computational Aspects of Social Networks (CASoN), pp. 88–93 (2011)

Construct Independent Spanning Trees on Chordal Rings with Multiple Chords[*]

Shyue-Ming Tang

Fu Hsing Kang School, National Defense University, Taipei, Taiwan, ROC

Abstract. Two spanning trees of a given graph $G = (V, E)$ are said to be *independent* if they are rooted at the same vertex, say r, and for each vertex $v \in V \setminus \{r\}$ two paths from r to v, one path in each tree, are internally disjoint. A set of spanning trees of G is said to be independent if they are pairwise independent. In 1989, Zehavi and Itai have conjectured that any k-connected graph has k independent spanning trees rooted at an arbitrary vertex. This conjecture is still open for $k > 4$.

A chordal ring is a ring network with extra links (chords) at each vertex. Y. Iwasaki et al. have proposed an algorithm to construct independent spanning trees on a chordal ring of degree four. (See [Y. Iwasaki et al., Independent panning trees of chordal rings, Information Processing Letters 69 (1999) 155-160].) In this paper, we shall propose an algorithm to solve the independent spanning trees problem on chordal rings with multiple chords.

Keywords: independent spanning trees, chordal rings, circulant graphs, vertex-symmetric graphs, internally disjoint paths, fault-tolerant broadcasting.

1 Introduction

Two paths connecting two vertices in a simple graph is said to be *internally disjoint* if they have no common vertex except two end vertices. Given a graph $G = (V, E)$, a tree T is called a *spanning tree* of G if T is a subgraph of G and T contains all vertices in V. Two spanning trees of G are called *independent* if they are rooted at the same vertex, say r, and for each vertex $v \in V \setminus \{r\}$, the two paths from r to v, one path in each tree, are internally disjoint. A set of spanning trees of a graph is said to be independent if they are pairwise independent [19].

Broadcasting in a computer network becomes fault-tolerant when it is equipped with a protocol based on the independent spanning trees (IST for short). It is also an ideal one-to-many broadcasting model [10]. We can disseminate message from a source node to all other nodes in the network by sending k copies of the message along k independent spanning trees rooted at the source node, and thus achieve the fault-tolerant broadcasting.

In 1989, Zehavi and Itai conjecture that, for any vertex r in a k-connected graph G, there exist k IST of G rooted at r. Although the conjecture has been

[*] This research is supported by the National Science Council of Taiwan under the Grants NSC101–2410–H–606–005.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 481–490.
DOI: 10.1007/978-3-642-35452-6_49

proved for k-connected graphs with $k \leqslant 4$ ([7] for $k = 2$, [3,19] for $k = 3$, and [4] for $k = 4$), it is still open for $k > 4$. Since the IST problem has drawn much attention, lots of research results are presented for special graph classes, such as planar graphs [6], chordal rings [8,14], hypercubes [11,12,15], recursive circulant graphs [16,17], multidimensional tori [13], and folded hypercubes [18].

Chordal rings are a variation of ring networks. By adding extra links (or chords) at each vertex in a ring, the fault-tolerance of the network is enhanced [1,2]. A chordal ring CR(N;$c_1, c_2, \ldots, c_{m-1}$) has $N(\geqslant 3)$ vertices labeled by $0, 1, \ldots, N-1$ in which (u, v) is an edge if and only if $v - u \equiv \pm 1$ or $\pm c_i$ (mod N), where $1 < c_1 < c_2 < \ldots < c_{m-1}$, and $c_i \leqslant \lfloor N/2 \rfloor$ for $1 \leqslant i \leqslant m-1$. One parameter c_i (called the i-th *hop*) can determine two chords for a vertex. Thus the degree of a chordal ring is $2m$ or $2m-1$ (in case that N is even and $c_{m-1} = N/2$). An example of chordal ring CR(30;3,7,15) is shown in Figure 1.

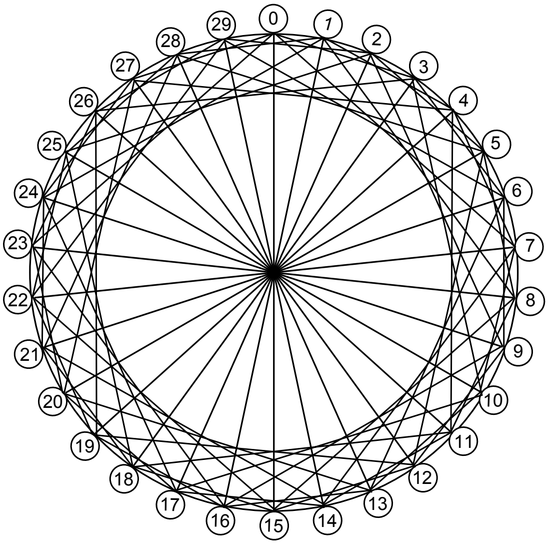


Fig. 1. Chordal ring $CR(30; 3, 7, 15)$

In the past, chordal rings have been widely studied, especially in the routing and the fault-tolerant routing problem [5,9]. In [8], Iwasaki et al. proposed a linear time algorithm to solve the IST problem in chordal rings of degree four. However, their IST construction algorithm can not be adapted to chordal rings with degree more than four. In this paper, we would like to propose a new algorithm to construct IST rooted at any vertex of a chordal ring whose degree may range from 3 (a wheel) to $N-1$ (a complete graph). The proposed algorithm is based on the computation of vertex label and can be implemented easily.

The remaining part of this paper is organized as follows. In Section 2, we discuss some essential properties for the IST problem in chordal rings. In Section 3, we present the algorithm. In Section 4, we prove the correctness of the algorithm. The last section contains our concluding remarks.

2 Preliminaries

To explicitly represent the adjacency of vertices in CR(N;$c_1, c_2, \ldots, c_{m-1}$), we say that vertex x takes *hop* ($h-$) (respectively, ($h+$)) to reach vertex y if $x - y = h$ (respectively, $-h$) for h= 1 or c_i ($1 \leqslant i \leqslant m-1$). Thus, the neighbors of x can be represented as a hop set $\{(1-), (1+), (c_1-), (c_1+), \ldots, (c_{m-1}-), (c_{m-1}+)\}$. The hop set representation is helpful to write the construction algorithm of IST.

Similar to the complement number of a number system, every vertex in a chordal ring has its complement vertex. For simplicity, every vertex computation hereinafter is taken modulo N.

Definition 1. In CR(N;$c_1, c_2, \ldots, c_{m-1}$), the *complement vertex* of vertex x is denoted by x' and $x' = N - x$ where N is the number of vertices in the graph. In case x=0 or $N/2$ (where N is even), x'=x.

Proposition 1. *If y is a neighbor of x, then y' is also a neighbor of x'.*

Proof. By the definition of complement vertex, $|y' - x'| = |(N - y) - (N - x)|$ $= |y - x|$. □

According to Propositions 1, we have the following lemma.

Lemma 1. *In CR(N;$c_1, c_2, \ldots, c_{m-1}$), vertex x takes hop ($h+$) (respectively, ($h-$)) to reach a neighbor y if and only if x' takes hop ($h-$) (respectively, ($h+$)) to reach vertex y' for $h \in \{1, c_1, c_2, \ldots, c_{m-1}\}$.*

Proof. When x takes hop ($h+$) to reach y, $x - y = -h$. That is, $(N - x) - (N - y) = h$ or $x' - y' = h$. On the Contrary, when x takes hop ($h+$) to reach y, $x' - y' = -h$. □

Since chordal rings are vertex-symmetric, without loss of generality, we can only consider the vertex 0 as the root of IST on a chordal ring. Due to Lemma 1, as soon as we have determined a number of internally disjoint paths from vertex x to vertex 0, the same number of internally disjoint paths from vertex x' to vertex 0 (self-complement) can be obtained. Thus, only a half number of vertices need to be computed when solving the IST problem in a chordal ring.

Based on the proposed algorithm, seven IST rooted at vertex 0 in $CR(30; 3; 7, 15)$ are constructed and shown in Figure 2. Following the definition of IST, for every vertex $v \neq 0$, seven paths from 0 to v in different trees are internally disjoint. Since chordal rings are vertex-symmetric, it turns out that every vertex in a chordal ring can be the root of seven IST.

Due to the internally disjoint requirement, it is obvious that the root has only one child in every independent spanning tree. If ($h\pm$) is the hop taken by the only child for reaching the root, the independent spanning tree can be denoted by $T_{(h\pm)}$. In Figure 2, each of the IST rooted at vertex 0 in $CR(30; 3, 7, 15)$ is named after the unique hop used to reach vertex 0.

Corollary 1 directly derive from Lemma 1.

Corollary 1. *Let $T_{(h-)}$ and $T_{(h+)}$ are two IST of CR(N;$c_1, c_2, \ldots, c_{m-1}$) and $h \in \{1, c_1, c_2, \ldots, c_{m-1}\}$. Then, $T_{(h+)}$ can be obtained from $T_{(h-)}$ by replacing every vertex x with $x'(= N - x)$.*

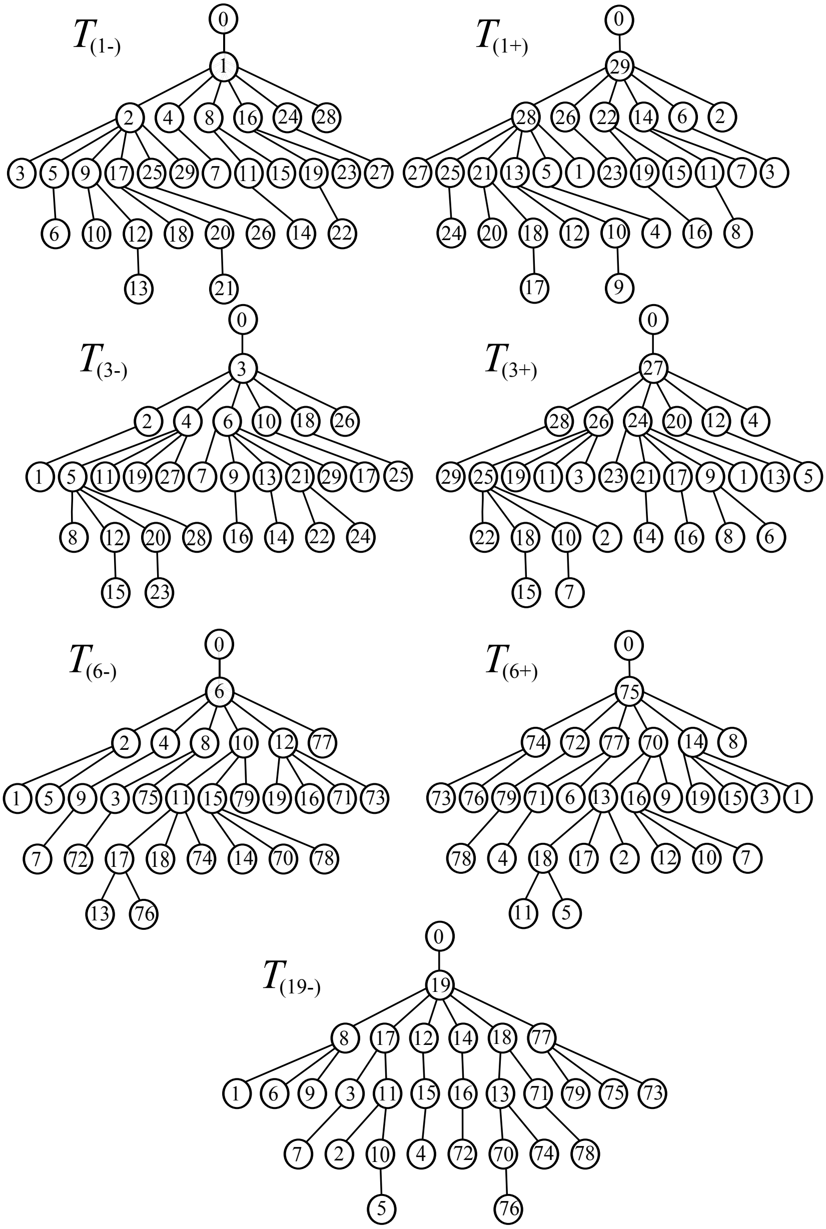


Fig. 2. Seven independent spanning trees on $CR(30; 3, 7, 15)$

In Figure 2, we can find that $T_{(i-)}$ and $T_{(i+)}$ are mutually complement for $i = 1, 3, 7$, and $T_{(15-)}$ is self-complement.

For the IST of a chordal ring, the hops for reaching the root in all trees must be distinct. In addition, for a single non-root vertex, the hops to reach its parents in different trees are also distinct. Since the hop sets of root-reaching and parent-reaching are equivalent, the IST construction algorithm is simply to determine a bijection from the parent-reaching hop set to the root-reaching hop set for every vertex. For example, in Figure 2, vertex 5 reaches its parent by taking hops (3−), (7−), (1−), (7+), (1+), (15−) and (3+) in trees which are labeled by hops (1−), (1+), (3−), (3+), (7−), (7+) and (15−), respectively.

Any path between two vertices in CR(N;$c_1, c_2, \ldots, c_{m-1}$) can be represented as a hop sequence. When a hop ($h\pm$) consecutively occurs k times ($k \geqslant 1$), we use $(h\pm)^k$ to denote the hop sequence. A hop sequence $[(h_1\pm)^{k_1}, (h_2\pm)^{k_2}, \cdots, (h_t\pm)^{k_t}]$ is *regular* if (i) $h_i \in \{1, c_1, \ldots, c_{m-1}\}$ for $1 \leqslant i \leqslant t$, (ii) $h_i < h_{i+1}$ for $1 \leqslant i \leqslant t-1$, (iii) $k_i \cdot h_i < c_{j+1}$ if $h_i = c_j$ for $1 \leqslant i \leqslant t-1$ and $1 \leqslant j \leqslant m-2$, and (iv) $k_1 < c_1$ if $h_1 = 1$.

A hop sequence $[(h_s\pm)^{k_s}, \cdots, (h_t\pm)^{k_t}, (h_1\pm)^{k_1}, (h_2\pm)^{k_2}, \cdots, (h_{s-1}\pm)^{k_{s-1}}]$ is called a *rotation* of the hop sequence $[(h_1\pm)^{k_1}, (h_2\pm)^{k_2}, \cdots, (h_{s-1}\pm)^{k_{s-1}}, (h_s\pm)^{k_s}, \cdots (h_t\pm)^{k_t}]$ when $1 \leqslant s \leqslant t$.

Next, we depict two special types of paths from vertex x to vertex 0 in CR(N;$c_1, c_2, \ldots, c_{m-1}$) by means of rotations of a regular hop sequence. Note that we only consider half of the vertices, i.e., for vertex $x \leqslant \lfloor N/2 \rfloor$.

Definition 2. An *anticlockwise-path* (or *a-path* for short) from vertex x to vertex 0 is composed of a rotation of the regular hop sequence $[(h_1-)^{k_1}, (h_2-)^{k_2}, \cdots, (h_t-)^{k_t}]$. That is, $x = h_1 \cdot k_1 + h_2 \cdot k_2 + \cdots + h_t \cdot k_t$.

For example, see Figure 1. Two rotations of the regular hop sequence $[(1-)^2, (3-)^1]$ form two a-paths from vertex 5 to vertex 0 in CR(30; 3, 7, 15). One a-path is the regular hop sequence itself, and the other is $[(3-)^1, (1-)^2]$.

Definition 3. A *clockwise-path* (or *c-path* for short) from vertex x to vertex 0 is either a rotation of the regular hop sequence $[(h_1+)^{k_1}, (h_2+)^{k_2}, \cdots, (h_t+)^{k_t}]$ (for $x \geqslant c_{m-1}$), or a rotation of the regular hop sequence $[(h_1+)^{k_1}, (h_2+)^{k_2}, \cdots, (h_t+)^{k_t}, (c_j-)]$ for $x < c_{m-1}$ with the smallest $c_j \leqslant c_{m-1}$. That is, for $x \geqslant c_{m-1}$, $N = x + h_1 \cdot k_1 + h_2 \cdot k_2 + \cdots + h_t \cdot k_t$. For $x < c_{m-1}$, there must be a $c_j \leqslant c_{m-1}$ such that $c_j = x + h_1 \cdot k_1 + h_2 \cdot k_2 + \cdots + h_t \cdot k_t$.

For example, see Figure 1 again. two rotations of the regular hop sequence $[(1+)^2, (7-)^1]$ form two c-paths from vertex 5 to vertex 0 in CR(30; 3, 7, 15). One c-path is the regular hop sequence itself, and the other is $[(7-)^1, (1+)^2]$.

Accordingly, we have the following three lemmas:

Lemma 2. *In a chordal ring, any two a-paths from a vertex are internally disjoint.*

Proof. Suppose A and B are two different a-paths from vertex x to vertex 0. Let v_1 and v_2 be internal vertices in A and B, respectively. If $v_1 = v_2$, then the path from x to v_1 and the path from x to v_2 have the same hops in their hop sequences. Since the hop sequences of A and B are rotations of a regular hop sequence, this condition never happens. □

Lemma 3. *In a chordal ring, any two c-paths from a vertex are internally disjoint.*

Proof. The proof is similar to Lemma 2 and is omitted. □

Lemma 4. *In a chordal ring, an a-path and a c-path from a vertex are internally disjoint.*

Proof. Suppose A is an a-path and B is a c-path from vertex x to vertex 0. Since the directions of A and B are anticlockwise and clockwise, respectively, so they must be internally disjoint. □

3 Constructing Independent Spanning Trees

In this section, an algorithm is presented to construct IST rooted at vertex 0 in a chordal ring. The algorithm determines a bijection from the parent-reaching hop set to the root-reaching hop set for every non-root vertex. There are two phases in the proposed algorithm. In the first phase, for every non-root vertex, the hop set is partitioned into four parts. In the second phase, different rules are applied in each of the four subsets to determine the bijection.

For each non-root vertex x in CR(N;$c_1, c_2, \ldots, c_{m-1}$), the following procedure is performed to partition the entire hop set into four hop sets, i.e., P_x, Q_x, R_x and S_x. Note that the procedure is designed for computing the first half of N vertices, i.e., $1 \leqslant x \leqslant \lfloor n/2 \rfloor$. By Lemma 1, the second half of the vertices can be computed from the result of their complement vertices.

```
Procedure. HOP_SET_PARTITION($x$)
begin
Step 1. Initialize hop sets.
        $P_x = \emptyset$; $Q_x = \emptyset$; $R_x = \emptyset$; $S_x = \emptyset$;
Step 2. Initialize $div_a$, $div_c$ and update $Q_x$.
        $div_a = x$;
        if $x \geqslant c_{m-1}$ then $div_c = N - x$;
        else if $c_{j-1} \leqslant x < c_j$ then $div_c = c_j - x$; $Q_x = Q_x \cup \{(c_j-)\}$;
        else $div_c = c_1 - x$; $Q_x = Q_x \cup \{(c_1-)\}$;
Step 3. Determine $P_x$ and $Q_x$.
        for $i = m-1$ downto $1$ do
            Substep 3.1. $quo = \lfloor div_a/c_i \rfloor$;
            Substep 3.2. if ($quo > 0$) then $P_x = P_x \cup \{(c_i-)\}$; $div_a = div_a - c_i \cdot quo$;
            Substep 3.3. $quo = \lfloor div_c/c_i \rfloor$;
            Substep 3.4. if ($quo > 0$) then $P_x = Q_x \cup \{(c_i+)\}$; $div_c = div_c - c_i \cdot quo$;
        enddo
Step 4. Determine $R_x$ and $S_x$.
        for $i = 1$ to $m-1$ do
            Suppose $(c_i\pm) \notin P_x \cup Q_x$.
            Substep 4.1. if $P_{x-c_i} \cap (P_x \cup Q_x) = \emptyset$ then $R_x = R_x \cup (\{(c_i-)\} \cup P_{x-c_i})$;
            Substep 4.2. if $Q_{x+c_i} \cap (P_x \cup Q_x) = \emptyset$ then $R_x = R_x \cup (\{(c_i+)\} \cup Q_{x+c_i})$;
            Suppose $(c_i\pm), (c_j\pm) \notin P_x \cup Q_x \cup R_x$ with the smallest $j > i$.
            Substep 4.3. if $(c_j-) \in P_{x-c_i}$ then $R_x = R_x \cup (\{(c_i-), (c_j-)\})$;
            Substep 4.4. if $(c_j\pm) \in Q_{x+c_i}$ then $R_x = R_x \cup (\{(c_i+), (c_j\pm)\})$;
            Substep 4.5. $S_x = S_x \cup \{(c_i\pm)\}$;
        enddo
end HOP_SET_PARTITION
```

Table 1. The four hop sets for vertex x $(1 \leqslant x \leqslant 15)$ in CR(30;3,7,15)

x	P_x	Q_x	R_x	S_x
1	$\{(1-)\}$	$\{(1+),(3-)\}$	$\{[(3+),(7-)],[(7+),(15-)]\}$	$\emptyset$
2	$\{(1-)\}$	$\{(1+),(3-)\}$	$\{[(3+),(7-)],[(7+),(15-)]\}$	$\emptyset$
3	$\{(3-)\}$	$\{(1+),(3+),(7-)\}$	$\{[(7+),(15-)]\}$	$\{(1-)\}$
4	$\{(1-),(3-)\}$	$\{(3+),(7-)\}$	$\{[(7+),(15-)]\}$	$\{(1+)\}$
5	$\{(1-),(3-)\}$	$\{(1+),(7-)\}$	$\{[(3+),(7+),(15-)]\}$	$\emptyset$
6	$\{(3-)\}$	$\{(1+),(7-)\}$	$\{[(7+),(15-)]\}$	$\{(1-),(3+)\}$
7	$\{(7-)\}$	$\{(1+),(7+),(15-)\}$	$\{[(1-),(3-)]\}$	$\{(3+)\}$
8	$\{(1-),(7-)\}$	$\{(7+),(15-)\}$	$\{[(1+),(3+)]\}$	$\{(3-)\}$
9	$\{(1-),(7-)\}$	$\{(3+),(15-)\}$	$\emptyset$	$\{(1+),(3-),(7+)\}$
10	$\{(3-),(7-)\}$	$\{(1+),(3+),(15-)\}$	$\emptyset$	$\{(1-),(7+)\}$
11	$\{(1-),(3-),(7-)\}$	$\{(1+),(3+),(15-)\}$	$\emptyset$	$\{(7+)\}$
12	$\{(1-),(3-),(7-)\}$	$\{(3+),(15-)\}$	$\emptyset$	$\{(1+),(7+)\}$
13	$\{(3-),(7-)\}$	$\{(1+),(15-)\}$	$\{[(3+),(7+)]\}$	$\{(1-)\}$
14	$\{(7-)\}$	$\{(1+),(15-)\}$	$\{[(1-),(3-)],[(3+),(7+)]\}$	$\emptyset$
15	$\{(15-)\}$	$\emptyset$	$\{[(1-),(7-)],[(1+),(7+)]\}$	$\{(3-),(3+)\}$

We use DR(30;3,7,15) as an example. Table 1 computes the four hop sets for the first half of vertices by using Procedure HOP_SET_PARTITION.

Suppose all hops in hop sets P_x and Q_x are arranged in an increasing order of hops. We define the *predecessor hop* of a hop as its previous hop. For the first hop of the set, its predecessor hop is the last one of the set. In hop set R_x, hops are added in a batch. Thus, each hop in one batch is the *partner hop* of the other hops.

When the hop set of vertex x is partitioned as mentioned above, the parents of x in every independent spanning tree can be determined by the following procedure.

```
Algorithm PARENT_HOPS_DETERMINE($x$)
begin
Step 1.  Determine $P_x$, $Q_x$, $R_x$ and $S_x$ by using Procedure HOP_SET_PARTITION($x$);
Step 2.  for all $(h-) \in P_x$ do
             Let $(p-)$ be the predecessor hop of $(h-)$ in $P_x$;
             Vertex $x$ takes $(h-)$ to reach its parent in tree $T_{(p-)}$;
         enddo
Step 3.  for all $(h\pm) \in Q_x$ do
             Let $(p\pm)$ be the predecessor hop of $h\pm$ in $Q_x$;
             Vertex $x$ takes $(h\pm)$ to reach its parent in tree $T_{(p\pm)}$;
         enddo
Step 4.  for all $(h\pm) \in R_x$ do
             Let $(p\pm)$ be the predecessor hop of $(h\pm)$ in its partner hops in $R_x$;
             Vertex $x$ takes $(h\pm)$ to reach its parent in tree $T_{(p\pm)}$;
         enddo
Step 5.  for all $(h\pm) \in S_x$ do
             Vertex $x$ takes $(h\pm)$ to reach its parent in tree $T_{(h\pm)}$;
         enddo
end PARENT_HOPS_DETERMINE
```

We use CR(30;3,7,15) as an example. Table 2 shows the parent-reaching hops of the first half of vertices by using Algorithm Parent_Hops_Determine. According to the rules in the algorithm, a bijection from parent-reaching hop set to root-reaching hop set is established.

Table 2. The parent-reaching hops of vertex x ($1 \leqslant x \leqslant 15$) in different IST on CR(30;3,7,15)

x	$T_{(1-)}$	$T_{(1+)}$	$T_{(3-)}$	$T_{(3+)}$	$T_{(7-)}$	$T_{(7+)}$	$T_{(15-)}$
1	(1−)	(3−)	(1+)	(7−)	(3+)	(15-)	(7+)
2	(1−)	(3−)	(1+)	(7−)	(3+)	(15-)	(7+)
3	(1−)	(3+)	(3−)	(7−)	(1+)	(15-)	(7+)
4	(3−)	(1+)	(1−)	(7−)	(3+)	(15-)	(7+)
5	(3−)	(7−)	(1−)	(7+)	(1+)	(15-)	(3+)
6	(1−)	(7−)	(3−)	(3+)	(1+)	(15-)	(7+)
7	(3−)	(7+)	(1−)	(3+)	(7−)	(15-)	(1+)
8	(7−)	(3+)	(3−)	(1+)	(1−)	(15-)	(7+)
9	(7−)	(1+)	(3−)	(15−)	(1−)	(7+)	(3+)
10	(1−)	(3+)	(7−)	(15−)	(3−)	(7+)	(1+)
11	(3−)	(3+)	(7−)	(15−)	(1−)	(7+)	(1+)
12	(3−)	(1+)	(7−)	(15−)	(1−)	(7+)	(3+)
13	(1−)	(15−)	(7−)	(7+)	(3−)	(3+)	(1+)
14	(3−)	(15−)	(1−)	(7+)	(7−)	(3+)	(1+)
15	(7−)	(7+)	(3−)	(3+)	(1−)	(1+)	(15−)

For constructing the IST shown in Figure 2, we have to transform the parent-reaching hops in Table 2 to vertex labels. The first half of the vertices ($1 \leqslant x \leqslant 15$) are computed from Table 2, while the second half of the vertices ($16 \leqslant x \leqslant 29$) are obtained directly from the first half by using Lemma 1. The time and space complexity of the construction is $O(mN)$, where m is with the degree of a vertex and N is the number of vertices.

4 Correctness Proof

To show the correctness of Algorithm Parent_Hops_Determine, we have to prove that the output of the algorithm are spanning trees of the input chordal ring. Then, we should prove that for every non-root vertex v, any two paths from v to the root in different spanning trees must be internally disjoint.

In Procedure Parent_Hops_Determine, it turns out that every non-root vertex x takes different hops to reach its parent in different spanning subgraphs (not yet proved to be spanning trees). To complete this proof, we need to show that there exists a unique path from every vertex x to 0 in each spanning subgraph. The following lemma depicts that the output of Algorithm Parent_Hops_Determine are spanning trees.

Lemma 5. *Procedure* Parent_Hops_Determine *can generate a set of δ spanning trees rooted at vertex 0 in a chordal ring, where δ is the degree or connectivity of the graph.*

Proof. According to a hop $(h\pm)$ from a non-root vertex x, we consider the following three cases:

Case 1: $(h-) \in P_x$. x takes an a-path to the root.

Case 2: $(h\pm) \in Q_x$. x takes a c-path to the root.

Case 3: $(h\pm) \in R_x$ or S_x. x either takes $(h-)$ and then followed by an a-path to arrive at the root, or takes $(h+)$ and then followed by a c-path to reach the root.

Since all of the paths from x to the root is unique, the resulting δ spanning subgraphs of Algorithm Parent_Hops_Determine are spanning trees of the input chordal ring. □

We now show the independency of the output IST of Algorithm Parent_Hops_Determine.

Lemma 6. *All paths generated by hop sets P_x and Q_x are internally disjoint.*

Proof. According to Lemmas 2.4, 2.5 and 2.6, all a-paths and c-paths are internally disjoint. □

Lemma 7. *All paths generated by hop sets R_x and S_x are internally disjoint.*

Proof. Suppose $(h\pm) \in R_x$ or S_x. x either takes $(h-)$ and then followed by an a-path to arrive at the root, or takes $(h+)$ and then followed by a c-path to reach the root. All of these paths must be internally disjoint. □

Lemma 8. *Let A and B are two paths generated by hop sets $P_x \cup Q_x$ and $R_x \cup S_x$, respectively. Then A and B are internally disjoint.*

Proof. A is either an a-path or a c-path, and B is either a negative hop followed by an a-path or a positive hop followed by a c-path. We can prove A and B are internally disjoint by unequal sums of hops in two paths. □

Combining Lemmas 4.1, 4.2, 4.3 and 4.4 gives the following theorem.

Theorem 1. *Algorithm* Parent_Hops_Determine *can correctly construct* $2m$ *or* $2m-1$ *IST on CR(N;*$c_1, c_2, \ldots, c_{m-1}$*) in* $O(mN)$ *time. (*$2m-1$ *holds in case that* N *is even and* $c_{m-1} = N/2$*.)*

5 Concluding Remarks

In this paper, an algorithm is proposed to solve the IST problem on chordal rings with multiple chords. Based on the algorithm, for each non-root vertex, a bijection from the parent-reaching hop set to root-reaching hop set is determined individually. Thus, the algorithm can be applied to a parallel or distributed computing environment.

References

1. Arden, B.W., Lee, H.: Analysis of chordal ring network. IEEE Transactions on Computers 30, 291–295 (1981)
2. Bermond, J.-C., Comellas, F., Hsu, D.F.: Distributed loop computer networks: A survey. Journal of Parallel and Distributed Computing 24, 2–10 (1995)
3. Cheriyan, J., Maheshwari, S.N.: Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs. Journal of Algorithms 9, 507–537 (1988)
4. Curran, S., Lee, O., Yu, X.: Finding four independent trees. SIAM Journal on Computing 35, 1023–1058 (2006)
5. Zimmerman, G.W., Esfahanian, A.-H.: Chordal rings as fault-tolerant loops. Discrete Applied Mathematics 37-38, 563–573 (1992)
6. Huck, A.: Independent trees in planar graphs. Graphs and Combinatorics 15, 29–77 (1999)
7. Itai, A., Rodeh, M.: The multi-tree approach to reliability in distributed networks. Information and Computation 79, 43–59 (1988)
8. Iwasaki, Y., Kajiwara, Y., Obokata, K., Igarashi, Y.: Independent spanning trees of chordal rings. Information Processing Letters 69, 155–160 (1999)
9. Mukhopadhyaya, K., Sinha, B.P.: Fault-tolerant routing in distributed loop networks. IEEE Transactions on Computers 44, 1452–1456 (1995)
10. Rabin, M.O.: Efficient dispersal of information for secursity, load balencing, and fault tolerance. Journal of the ACM 36, 335–348 (1989)
11. Ramanathan, P., Shin, K.G.: Reliable broadcast in hypercube multicomputers. IEEE Transactions on Computers 37, 1654–1657 (1988)
12. Tang, S.-M., Wang, Y.-L., Leu, Y.-H.: Optimal independent spanning trees on hypercubes. Journal of Information Science and Engineering 20, 605–617 (2004)
13. Tang, S.-M., Yang, J.-S., Wang, Y.-L., Chang, J.-M.: independent spanning trees on multidimensional torus networks. IEEE Transactions on Computers 59, 93–102 (2010)
14. Yang, J.-S., Chang, J.-M., Tang, S.-M., Wang, Y.-L.: Reducing the height of independent spanning trees in chordal rings. IEEE Transactions on Parallel and Distributed Systems 18, 644–657 (2007)
15. Yang, J.-S., Tang, S.-M., Chang, J.-M., Wang, Y.-L.: Parallel construction of independent spanning trees on hypercubes. Parallel Computing 33, 73–79 (2007)
16. Yang, J.-S., Chang, J.-M., Tang, S.-M., Wang, Y.-L.: On the independent spanning trees of recursive circulant graphs $G(cd^m, d)$ with $d > 2$. Theoretical Computer Science 410, 2001–2010 (2009)
17. Yang, J.-S., Chang, J.-M., Tang, S.-M., Wang, Y.-L.: Constructing multiple independent spanning trees on recursive circulant graphs $G(2^m, 2)$. International Journal of Foundations of Computer Science 21, 73–90 (2010)
18. Yang, J.-S., Chang, J.-M., Chan, H.-C.: Broadcasting Secure Messages via Optimal Independent Spanning Trees in Folded Hypercubes. Discrete Applied Mathematics 159, 1254–1263 (2011)
19. Zehavi, A., Itai, A.: Three tree-paths. Journal of Graph Theory 13, 175–188 (1989)

A Grid-Based Clustering Routing Protocol for Wireless Sensor Networks

Ying-Hong Wang, Yu-Wei Lin, Yu-Yu Lin, and Hang-Ming Chang

Computer Science and Information Engineering
Tamkang University
Tamsui Taipei, Taiwan, R.O.C.
inhon@mail.tku.edu.tw, {harry040,jerry198926}@hotmail.com,
chmcice174723@gmail.com

Abstract. After sensors are deployed, it cannot be recharged in Wireless sensor networks. Therefore, the energy of sensors is limited. In this situation, how to design a routing algorithm is very important. An effective algorithm can reduce energy consumption and prolong network lifetime. In this proposal, we proposed a routing protocol. First, this algorithm divides networks into several units and those units would be regarded as clusters. Second, it uses conditions of method we proposed to generate some clusters by combining. According to remaining energy of nodes, it will execute cluster-head selection. And it can reduce the energy consumption of nodes and use energy effectively to sense by clustering. Finally, we compare the methods we proposed with the others by simulation, and the conclusion proves the method we proposed can save more power to make network lifetime longer than other ways.

Keywords: Cluster, Power saving, Routing protocol, WSNs.

1 Introduction

In recent years, wireless application and wireless communication markets have become more popular due to the rapid development of wireless communications technology. Moreover, the advance in micro technology has led to wide adoption of multiple wireless network technologies consisting mainly of wireless sensor networks [1]. In wireless sensor networks, micro-manufacturing technology continues to increase capabilities in environmental sensing, information processing, wireless communications, computing ability, and storage capacity. In order to take full advantage of these advances, reducing energy consumption to extend network lifetime of wireless sensors is critical, and thus an important topic for research.

The characteristics of wireless sensor design calls for small footprint, low cost, power saving, and accurate sensing ability. The current areas of research can be divided into the following several categories: routing protocol, target tracking, locating, data aggregation, fault tolerance, sensor node deployment and energy management. Each sensor node has data processing, communication, and data sensing responsibilities, all consume a limited energy resource. In this premise, a wireless

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 491–499.
DOI: 10.1007/978-3-642-35452-6_50

sensor node achieves the greatest benefit from an increase in energy consumption efficiency. Therefore, how to design an effective routing protocol is a very important topic. In the wireless sensor network applications using the environment as a static target and more under consideration, we hope that the node does not have too strong computing power and other additional equipment in order to achieve as much as possible to reduce energy consumption and cost effectiveness. To achieve this goal we propose a cluster-based routing protocol for wireless sensor network that is Merging Grid into Clustering-based Routing Protocol for Wireless Sensor Networks, MGCRP.

2 Related Work

There are numerous papers on using routing protocols to make wireless sensor networks stable, effective and power saving. [2, 3] introduces the concept of using routing protocols in wireless networks.

Recently, there are three leading ways of routing. They are chain-based, cluster-based and tree-based; our paper will focus on cluster-based. In cluster-based routing nodes are divided into clusters and the cluster head will send the data collected from normal nodes to sink.

Low Energy Adaptive Clustering Hierarchy (LEACH) [4] is proposed by Heinzelman. This routing protocol divides nodes into several clusters by their location, and the nodes can only communicate with in the same cluster.

Energy-Balanced Chain-cluster Routing Protocol (EBCRP) [5] is proposed by Xi-Rong Bao. It is a cluster-based distributed algorithm that builds a path of chains through the use of a ladder algorithm.

3 Merging Grid into Clustering-Based Routing Protocol

Our proposed routing protocol is divided into two phases: Clustering Phase and Routing Phase, and we will add Cluster-Head Rotation Mechanism to maintain routing persistence in the Routing Phase.

3.1 Network Environment and Assumption

We assume the wireless sensor network is composed of a sink and a large number of static sensor nodes randomly deployed in the target area.

a. System Environment

We assume n sensor nodes randomly distributed in the area to be monitored are continuouly sensing and reporting events. These sensor nodes are static. We use S_i to indicate the i-th node, sensor nodes set $S = \{S_1,S_2,...,S_n\}$, and the number of S is n. We make the following assumptions about the sensor nodes and the network module.

i) The sink is deployed in a region away from the sensors and we assume that the energy of the sink is infinite.

ii) Sensor nodes will be assigned a unique identifier before deployed in the sensing area.

iii) All nodes have the same computing, storage and energy capabilities.

iv) The sensor node's transmission power can be changed according to the distance from the receiver.

v) All sensor nodes are static. In addition, each sensor node knows its own location and the sink knows their location though the use of the location mechanism.

b. *Energy Consumption Module*

Our paper is uses [7] the communication energy consumption module, and follows the formula:

$$E\ _T(k,d) = E_{Tx}k + E_{amp}(d)k \tag{1}$$

$$E_R(k) = E_{Rx}k \tag{2}$$

$$E_{fuse}(k) = E_{fuse}k \tag{3}$$

Formula (1) means the sensor nodes have an energy cost when transmitting data, (2) means the sensor nodes have an energy cost when receiving data, (3) means the sensor nodes have an energy cost when the data fuses. In these three functions, where k is data packet size, E_{Tx} is energy cost for transmitting one unit data of the sensor node, E_{Rx} represents the energy cost when node receives one unit data, E_{fuse} is energy cost of the fusing the data. When the sensor nodes transmit amplification is required, so transmitting nodes have an additional $E_{amp}(d)k$ energy cost. The value of $E_{amp}(d)k$ can be determined by formula (4).

$$E_{amp}(d)k = \varepsilon_{FS}d^2 \tag{4}$$

Where d is the distance between two nodes, ε_{FS} represents the amplified electric power energy cost.

c. *Sensor Node and Cluster Information*

Table 1. is the sensor node information table, which is used to record information about itself. Next we will introduce each field of the table. Node_ID is the identification of the node. Res_Energy is the residual energy of the node. Head_ID is the identification of the cluster head in its own cluster, if the Head_ID and Node_ID are the same, the node itself is the cluster head. Cluster_ID is the cluster number the sensor node belongs to. Next_Hop is the next sensor node to forward data to. Table 2. is the cluster table, it records information of every cluster member and including the following fields Node_ID, Res_Energy, and Cost. Node_ID is the identification of member of the node in the cluster. Res_Energy is the residual energy of the node in the cluster. Cost is transmission cost between two nodes. Table 3. Is Head List, it is used to record information one neighboring cluster heads. The fields include Node_ID, Cluster_ID and Cost. Node_ID is the identification of the neighboring cluster head. Cluster_ID is the cluster number of the neighboring of cluster head. Cost is transmission cost between two cluster heads.

Table 1. Sensor node information table

Node_ID	Res_Energy	Head_ID	Cluster_ID	Next_Hop

Table 2. Cluster_Table

Node_ID	Res_Energy	Cost

Table 3. Head_List

Node_ID	Cluster_ID	Cost

3.2 Clustering Phase

a. Clustering

Before discussing the clustering step, first, we must are define the routing protocol parameters and variables.

Rectangle Unit Block (Block): This is a rectangular block. The user defined value of N divides the network into several blocks which are the same size and are not overlapping.

Center of Block (BC): After dividing the network into several grids, we will calculate the center coordinates of the grids resulting in a vector of coordinates. We assume Block_Center i (BC_i) is the i-th center coordinates of the block.

Cluster: Cluster can be regarded as a set C which is includes several sensor nodes. We can represent a set C as $C = \{S_j\}, S_j \in S, j=1,2,\ldots,n$. Where j is the number of sensor nodes. We assume the Cluster_ID is i which represent the cluster number of the grid. In any cluster ($\mathbf{C_i}$), if any member of the sensor nodes are not in a cluster, it is an invalid cluster, otherwise, it is valid cluster.

Distribution: We define a new parameter in a valid cluster Distribution it is used to evaluate the distribution of nodes in a valid cluster. The number of nodes within a valid cluster closer to BC are the best. The formula (5) is used to calculate the distribution of the cluster. Where $\boldsymbol{d(S_m, BC_i)}$ is the distance between the member of the sensor nodes in cluster and the BC of the cluster and where $\boldsymbol{N(C_i)}$is the number of sensor nodes in the cluster. After defining these parameters and variables, the following details the description of each step.

Step 1: Network Gridding
After the deployment of the sensor nodes, we will make a grid of the network. In this paper, we assume that the sensor nodes in the network can be arranged to an M*M area, and assume every length of block is N. The network will be divided into $(\frac{M}{N})^2$ same size blocks. Where the user defined the N value and M value is the length of the sensor network.

Step 2: Calculate Center of Grid
Formula (6) calculates the center of the grid. BC_i is the two-dimensional coordinate vector, where i=1,2,…, $(\frac{M}{N})^2$, this is used to indicate the number of the grid, and also is also the Cluster_ID. The numbering starts from the (0,0) position along the X axis towards the right, Sequenced 1, 2, …, until numbered to the right- border of the sensor network, then back to left-border of the sensor network. In this moment, shift the Y-axis direction one unit block down, then repeat the sequencing step until the grid is complete. Then use the number of grids and formula (6) to get each center of grid.

Step 3: Calculate Distribution (C_i) of Valid Cluster
In this step, we will calculate the Distribution of the valid cluster. First, we give a set VC that includes all valid clusters in the network. N(VC) expresses the number of valid clusters. We will only calculate the Distribution of clusters in the VC set. After each Distribution in each cluster has been calculated, we will start the cluster merging process.

$$Distribution(\mathrm{C_i}) = \frac{\left[\sum_{S_m \in S} d(S_m, BC_i)\right]}{N(\mathrm{C_i})} \tag{5}$$

$$BC_i = \left(\left[(i-1)\%\left(\frac{M}{N}\right) + \frac{1}{2}\right] * \frac{M}{N}, \left[\left\lfloor(i-1)/\left(\frac{M}{N}\right)\right\rfloor + \frac{1}{2}\right] * \frac{M}{N}\right) \tag{6}$$

Step 4 : Merge Valid Cluster
First, we choose the fewest number of nodes and the cluster with the largest Distribution value from the VC set. Assume a cluster from the VC set that meets the above conditions is C_A, where $C_A \in$ VC, A=1,2,…,$(\frac{M}{N})^2$, then we will start the merge. Let the distance between S_a and S_b be minimal, where $S_a \in C_A$, $S_b \in C_B$, B=1,2,…,$(\frac{M}{N})^2$, $C_B \in$ VC and $C_B \neq C_A$. Then we add all the sensor nodes to C_B fromC_A. In other words, let all the Cluster_IDs of the sensor nodes from C_A change to C_B, and remove from the VCC_A, resulting in one less N(VC) .

Step 5: Clustering Finish
Assume the variable K is the user set up number of clusters in the network. The value of K will affect the efficiency of network, so we must decide the variable K according to the network size and number of nodes. The operation of clustering in step 4 will be repeated until N(VC)=K. After clustering finishes, the sink will send related information to the sensor node for an update.

b. Cluster Head Selection
The main task of the cluster head is to fuse data that sensor nodes sensed within a cluster, receive other cluster heads' sensed data, and, send to sink, after clustering finishes and, cluster head must be selected from each cluster. To do so, the sink will broadcast a Head_Elect Message packet to every sensor node in each cluster in the network. When a sensor node gets this packet, it will generate a random variable P between 0 and 1, where *P* is used to differentiate between the same residual energy from other sensor nodes. After sensor node got a random variable *P*, then immediately to calculate itself residual energy. The residual energy is then calculated.

The member nodes of the same cluster compare each of their residual energies according to the transmission power to obtain cost between them. Sensor nodes with the most residual energy will be selected as the cluster head. If more than one sensor nodes have the same residual energy in the same cluster then the sensor node with the larger P value will be selected as cluster head. After each of the cluster heads of cluster has been selected, each cluster head will send a Head_Confirm packet to the sink. The packet format is shown in Table 4. Each Head_Confirm packet contains three fields, they are Header, Node_ID and Cluster_ID. The Header records the name of packet, Node_ID expresses the Node_ID of the sensor node that is the cluster head, Cluster_ID expresses the Cluster_ID of the cluster to where the cluster head belongs. After the sink received all the Head_Confirm packets, it will consolidate the information and forward it to each cluster head allowing them to, update their Head_List table.

Table 4. Head_Confirm packet

Header	Node_ID	Cluster_ID

3.3 Route

In our paper, the transmission route can be divided into two parts: inner-cluster transmission route and outer-cluster transmission route. Inner-cluster transmission route refers to the path between cluster head and sensor nodes within the same cluster. Outer-cluster transmission route refers to the path between cluster heads.

Our proposed path selection method is mainly based on transmission cost between the sensor nodes. So we use Bellman-Ford shortest path algorithm[6] for the route selection method. We arrange the network as a graph, and assume the sensor nodes in the network are the vertexes of graph and the transmission cost between nodes are edges of the graph. Through the Bellman-Ford algorithm we can calculate the lowest cost of each sensor node to the other.

a. Inner-Cluster Transmission Route Build

Inner-cluster transmission routes are the path between sensor nodes within the same cluster. The sink broadcasts to all the sensors nodes their minimum path cost to the cluster head in their cluster using the Bellman-Ford algorithm according to member cost in the Cluster_Table. In (7), $C(i,j)$ defines the cost between node i and node j, where $P_t(i,j)$ is the transmission power of node i to node j during transmission. After the sensor node receives the minimum cost between the node and the cluster head, it then records the next hop target in the Next_Hop field. When the node wants to transmit data, it sends data to the sensor node based on Next_Hop field. During the transmission, if the sensor node dies or cluster head changes, the Bellman-Ford algorithm is invoked to re-calculate the minimum cost path and the Next_Hop field is updated.

$$C(i,j) = P_t(i,j) \qquad (7)$$

b. Outer-Cluster Transmission Route Build

The outer-cluster transmission route and inner-cluster transmission route have the same algorithm, but in the outer-cluster, the send object changes to cluster head to cluster head. The cluster head receives the data that members sent in the cluster and, then it integrates the received data and forwards it to its neighboring cluster head. According to the Cost field of Head_List and through the use of the Bellman-ford algorithm, the cluster head selects the next hop. After the calculation, the cluster head will record the transmission object. If the cluster head has been replaced, then the new cluster head will request a member to re-calculate the minimum cost between cluster heads.

c. Cluster Head Rotation Mechanism

The cluster head not only senses the environment but integrates the data from members of the same cluster, and transmits data to other cluster heads. In order to reduce the early death of sensor nodes, we add the cluster head rotation mechanism to distribute the energy consumption. We assume time divided into continuous periods of *T*, in the beginning *T* the sink will send a Cluster_Head Rotation Message to the sensor network. After a normal node receives this message, they will immediately send their residual energy information to the cluster head. Then the cluster head will select the node of with the most residual energy to be the new cluster head. At the same time, the cluster head will broadcast to members within cluster the new identify of the cluster head and update Head_ID of the normal node and send Head_Confirm packet to the sink. The sink will gather all the new cluster head information, consolidate, and send the data to all the cluster heads so they can update Head_List table. During *T*, the sink will repeat the above action to replace the cluster head until the energy of members within cluster is less than the energy defined by the cluster head threshold.

4 Simulation and Analysis

This paper uses the Dev C++ simulation environment. The condition of the sensor network and its related values [8] is shown on Table 5. A round is defined by data that is transmitted to sink safely; a conclusion that is made from the average of 50 kinds of conditions. We can observe that MGCRP is better than LEACH and EBCRP via Fig.1 and Fig.2, The selection of cluster head principle of EBCRP is better than LEACH because it chooses the nodes which are closer to sink to be clusters. The design does not have the transmission distance limitation of clusters in LEACH, but the routing of EBCRP is a chain which is connected by nodes resulting in redundant data transmissions. In this paper, MGCRP combines nodes which are closer to others in a cluster and when the nodes are distributed unevenly, it shortens the distance between nodes and the cluster head to attain power savings. To not overload any one member, we add cluster head rotation mechanism, to equally to distribute energy consumption to the members in the cluster prolonging the network lifetime.

Table 5. Experimental Parameters

Parameter	Value
Sensing range (m^2)	(0,0)~(100,100)
Sink location	(50,150)
Sensor node numbers (n)	100
Sensor node initial energy (E0)	0.5 J
E_{Tx}, E_{Rx}	50 nJ/bit
ε_{FS}	10 pJ/(bit•m^2)
E_{fuse}	5 nJ/(bit•single)
Data packet size	4000 bits
Grid length (N)	10 m
Cluster number (K)	5

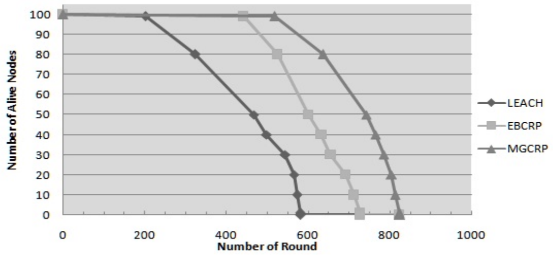


Fig. 1. Relation between the number of alive nodes and number of round with different routing protocol

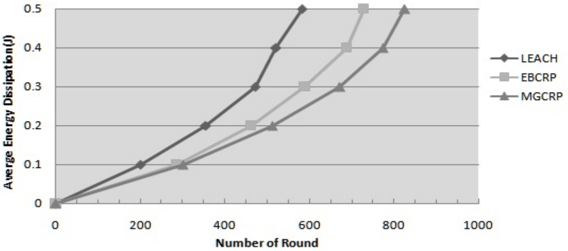


Fig. 2. Relation between consumption of average energy and number of round with different routing protocol

5 Conclusions

In this routing protocol, we grid the network and then combine these grids based on a user defined value, There are several advantages of this protocol show below. First, sensors will be allocated by high density in the same cluster no matter what the condition is. Second, we add the cluster head rotation mechanism, and it could allot workload equally to every node. Through effective clustering, the routing protocol which we proposed could save more energy and prolong the network lifetime.

References

1. Akyildiz, I.F., Weilian, S., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
2. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications 11(6), 6–28 (2004)
3. Qiangfeng, J., Manivannan, D.: Routing protocols for sensor networks. In: Proceedings of First IEEE Consumer Communications and Networking Conference, CCNC 2004, pp. 93–98 (January 2004)
4. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocolfor wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, vol. 2, p. 10 (January 2000)
5. Bao, X.-R., Zhang, S., Xue, D.-Y., Qie, Z.-T.: An Energy-Balanced Chain-Cluster Routing Protocol for Wireless Sensor Networks. In: Proceedings of the 2010 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC), pp. 79–84 (April 2010)
6. Bertsekas, D., Gallager, R.: Data Networks, 2nd edn. Prentice–Hall, Englewood Cliffs (1992)
7. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)
8. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)

CDTS: Coordinator Data Traffic Shunt Model for Zigbee Networks

Chinyang Henry Tseng[1,*], Shaiuhuey Wang[2], Bor-Shing Lin[1], Tong-Ying Juang[1], and Xiao-Ru Ji[1]

[1] National Taipei University, New Taipei City, Taiwan
{tsengcyt,bslin,juang}@mail.ntpu.edu.tw,
s710083105@webmail.ntpu.edu.tw
[2] Chung Yuan Christian University, Jungli, Taiwan
anglenew@cycu.edu.tw

Abstract. Zigbee, a wireless sensor network, is expected to have a explosive growth in the use of wireless control and monitoring applications because it has the advantages of low cost and easy deployment. As Zigbee is applied to more emerging applications, scalability becomes a critical issue: all sensor data is sent to the coordinator before forwarding to the sink node, and obviously the coordinator becomes the bottleneck. This paper proposes Coordinator Data Traffic Shunt (CDTS) performing data traffic shunt feature to reduce Coordinator's traffic. CDTS group consists of CDTS routes and forwards data directly to Sink instead of to the coordinator. To ease CDTS router implementation without modifying Zigbee standard, CDTS layer is added between Media Access Control (MAC) and network (NWK) layers in Zigbee stack. CDTS layer intercepts packets and redirect them to the Sink node without involving the coordinator. We implement CDTS routers in TI CC2530 Zigbee platform and NS2 simulation. Experiment results demonstrates CDTS provides better packet deliver ratio and lower coordinator loading than original Zigbee stack. Thus, CDTS successfully resolves Zigbee bottleneck problem and is fully compatible with current Zigbee stack design.

Keywords: Zigbee, Coordinator, Coordinator Data Traffic Shunt (CDTS), Personal Area Network (PAN).

1 Introduction

ZigBee is a wireless network standard and is expected to have an explosive growth in wireless control and monitoring applications because of its low cost and low power consumption. Popular applications include home and building automation, Industrial control, Embedded sensing, Medical data collection and warning systems. ZigBee standard becomes complete by adding four main components: network layer, application layer, ZigBee device objects (ZDOs) and manufacturer-defined application objects for customization and total integration. ZDOs, the most significant improvement,

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 501–510.
DOI: 10.1007/978-3-642-35452-6_51

are responsible for keeping device roles, management of requests to join a network, device discovery and security. Zigbee supports star, tree and mesh network topologies and, for all topologies, ZigBee must have one coordinator device responsible for a Personal Area Network (PAN) creation, parameter control and fundamental maintenance. Zigbee uses Ad-hoc On-demand Distance Vector (AODV) routing protocol to automatically construct an ad-hoc network as routes needed. Nodes not in use are in the sleep mode but can be woken up to the active mode around 3 seconds when they are needed for data collection and transmission. This on-demand characteristic results in low power consumption and long battery life.

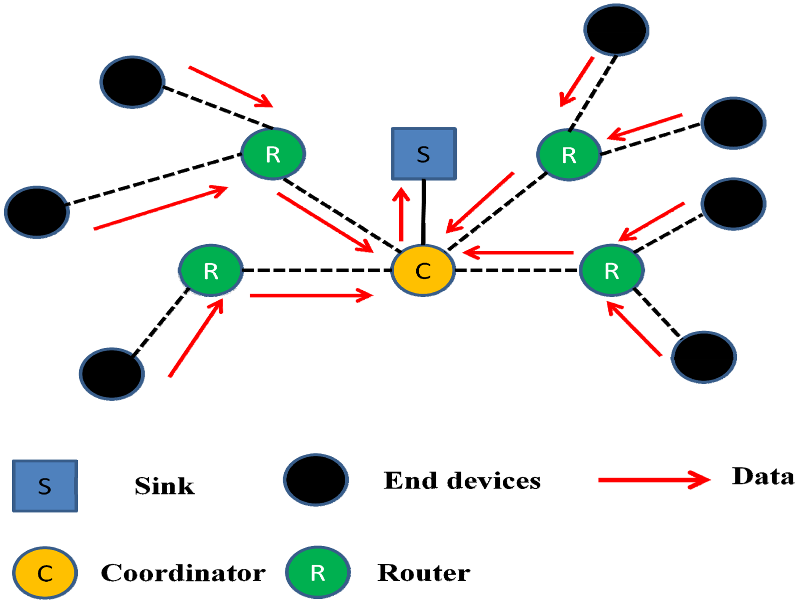


Fig. 1. Sensor data traffic in Zigbee network

In Zigbee, a Coordinator forms a PAN and is responsible for forwarding all the sensed data in that PAN to Sink, the backend server. As Zigbee is applied to more emerging applications requiring intensive data transmission, obviously, the Coordinator becomes the bottleneck and scalability issue becomes critical. With a larger scale of the Zigbee network, if the coordinator cannot handle the large amount of data, data loss and transmission will occur and even worse, the whole network becomes inoperable. The failure of network operation is unacceptable for time-sensitive applications, especially for real-time monitoring systems. For example in Fig. 1, Zigbee sensor nodes constantly send large amount of sensor data and therefore the Zigbee coordinator must have high data capacity. In addition, if the sensor data comes from critical biomedical systems, such as cardiovascular system, data transmission failure may cause emergency warning misses and life support delay. For these time critical applications, traditional Zigbee transmission mechanism needs to be improved to enhance the transmission reliability of coordinator as the Zigbee scale and the amount of sensed data becomes larger and larger.

To solve the Coordinator bottleneck problem, this paper proposes Coordinator Data Traffic Shunt (CDTS) which performs sensor data traffic shunt feature to reduce Coordinator's traffic. CDTS group consists of CDTS routers linked with the sink node through other Direct Communication Links (DCLs), such as RS232 or Bluetooth. CDTS group forwards data directly to the sink node instead of the coordinator. To ease the integration of CDTS feature in a Zigbee router without modifying Zigbee standard, CDTS layer is added between Media Access Control (MAC) and network layers in Zigbee stack. The CTD layer intercepts the packets from network layer and modifies the destination address to be the CDTS router itself instead of the coordinator. CTDS group redirects sensor traffic to the sink node without involving the coordinator. We implement CDTS routers in TI CC2530 Zigbee platform and NS2 simulation. Experiment results show CDTS provides better packet deliver ratio and scalability as well as lower coordinator loading than original Zigbee stack while increasing the throughput at the sink node. Thus, CDTS successfully resolves Zigbee bottleneck problem and is fully compatible with current Zigbee stack design.

This paper is organized as follows: Section 2 describes analysis, requirements, and related works. Section 3 presents the model design. Section 4 describes experiment results. Section 5 concludes the work.

2 Analysis and Requirements

2.1 Zigbee Network Analysis

In order to modify Zigbee data process procedures, it is required to analyze Zigbee stack processes of network and data initiation. First, in order to form a Zigbee Personal Area Network (PAN), this Zigbee PAN has a coordinator node, which is in charge of PAN initiation. The coordinator first selects a PAN ID and broadcasts beacon requests with this ID, with which allows other routers and end devices can join this PAN network. Then routers broadcast beacon requests for other remote Zigbee nodes such that all Zigbee nodes join this PAN network. Thus, all nodes are capable to reach and send their sensor data to the coordinator.

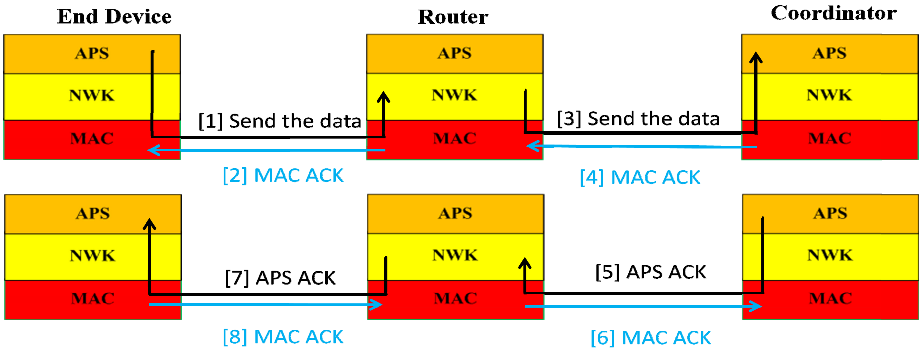


Fig. 2. Zigbee ACK processing in APS and MAC layers

When nodes begin to send sensor data to the coordinator, they usually require to establish clusters binding with the coordinator. Zigbee application support sublayer (APS) is responsible to establish the clusters, and the APS layer at the destination is required to send back APS acknowledge (ACK) messages to avoid data loss, as illustrated in Fig. 2, (step 1, 3, 5, 7). Because of high data loss rate in wireless environments, Zigbee MAC layer, IEEE 802.15.4, requires all Zigbee intermediates nodes sending MAC ACK frames to prevent data loss for APS data flows, as shown in Fig. 2 (step 2, 4, 6, 8). After receiving APS ACK, Zigbee nodes can complete cluster binding with the coordinator for sending sensor data . Thus, if Zigbee data process procedures are modified, cluster binding process and its APS ACK flow have to be maintained.

2.2 Requirement

In Zigbee PAN networks, Zigbee nodes sending sensor data with cluster bindings with the coordinator. If the cluster data traffic increases a lot, the coordinator will easily become the network bottleneck. In addition, the high traffic data load may cause the coordinator running out of battery or malfunction, which will paralyze the whole Zigbee PAN network. Thus, coordinator traffic loading becomes a major Zigbee network scalability issue, denoted as R_1.

To address this scalability issue, the desired solution should satisfy the following three requirements, denoted as R_{1A}, R_{1B}, R_{1C}, respectively. First, minimizing the coordinator traffic loading is required, denoted as R_{1A}, even though the sensor data traffic toward the sink node is increasing. Second, the flexibility of the desire solution deployment is required, denoted as R_{1B}, so the PAN network data traffic topology can be easily expanded. Third, since Zigbee data traffic delivering relies on cluster bindings with the coordinator, seamless maintenance of cluster binding is required, denoted as R_{1C}.

Because Zigbee embedded solutions usually do not provide source codes for the implementations of lower layers, such as MAC, Network (NWK), and APS layers, the desired solution is required to be compatible with these existing implementations without modifying them, denoted as R_2. For example, IEEE 802.15.4, AODV (Ad hoc On-demand Distance Vector) are Zigbee standard solutions for MAC and NWK layers respectively so they cannot be modified, denoted as R_{2A}. Besides, Zigbee sink applications are usually customized built according to application demands because they are usually black boxes for us. Thus, the desired solution is required to be fully compatible with all kinds of sink applications without modifying them, denoted as R_{2B}.

2.3 Related Works

Here we discuss related works that intend to enhance data delivering for Zigbee networks. Direct Diffusion [1] is a datacentric and query driven communication protocol. Its sink node must send queries to retrieve sensor data. This design may save periodic data sending but miss desirable data, and it requires the sink node cooperation and extra data queries. LEACH [2] is a two level data transfer protocol, and all its data must be sent through cluster heads. Xianghua Xu et al. [3] proposed an enhanced tree-based routing algorithm for reducing AODV route discovery overhead. Ruixia

Liu et al. [4] proposed a new cluster routing protocol, called AODV cluster, to reduce energy consumption in a large scale of AODV network. These two works require modify AODV design in order to enhance AODV performance in Zigbee networks.

Kartinah Zen et al. [5] proposed a load balancing scheme for child and parent coordinators in case some coordinators have connected with excessive neighbors. Zheng Sun et al. [6] proposed a new routing protocol to balance routing overhead in Zigbee networks based on AODV design. Kuei-Li Huang et al. [7] proposed a load balancing mechanism, called controller assisted distributed (CAD) for Zigbee networks to balance network loading between PAN networks. Most of these works require modify current Zigbee stack design. Although they can enhance Zigbee performance in certain scenarios, they cannot provide a generic solution for the bottleneck issue resulting from current Zigbee coordinator design.

3 CTDS Design

3.1 CDTS Group

To address R_1 in section 2.2, we introduce Coordinator Data Traffic Shunt (CDTS) group to solve this problem. CDTS group consists of CDTS routers, which have direct communication links (DCLs) linking with the sink node. While receiving data packets from end devices toward the coordinator, CDTS routers forward the packets to the sink node by DCLs instead of the coordinator, presented in Fig. 3.

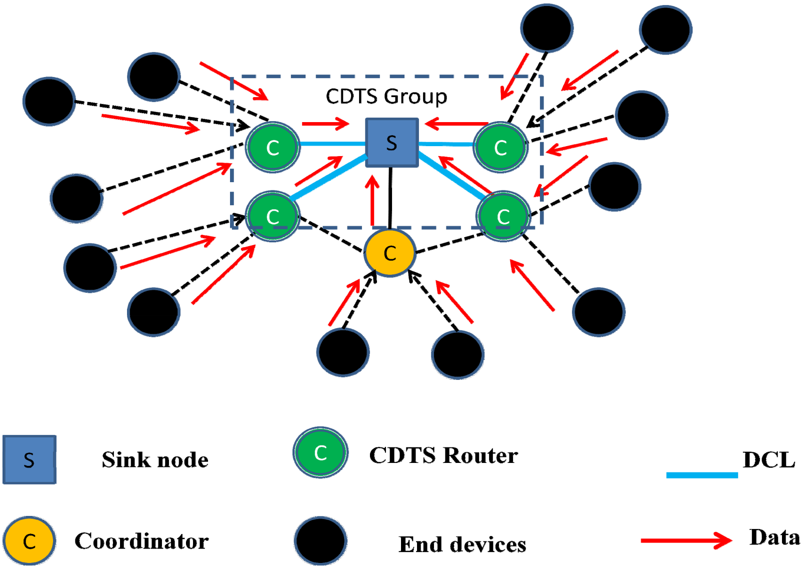


Fig. 3. CDTS Group

Because CDTS routers can handle lots of data flows in parallel without going through the coordinator, the CDTS group is able to mitigate network traffic burden of the coordinator, and thus R_{1A} is resolved. In addition, by having more CDTS routers, a CTDS group can handle more data traffic from Zigbee nodes, and thus R_{1B} is resolved.

3.2 CDTS Layer

To address R_2 in section 2.2, we introduce CDTS layer to solve this problem. CDTS layer is designed for CDTS routers in order to redirect the data packets to the sink without modifying Zigbee MAC and NWK layer. As illustrated in Fig. 4, CDTS layer intercepts data packets between MAC and NWK layers for perform data traffic shunt. It does not modify MAC and NWK layers in Zigbee stack so R_{2A} is satisfied.

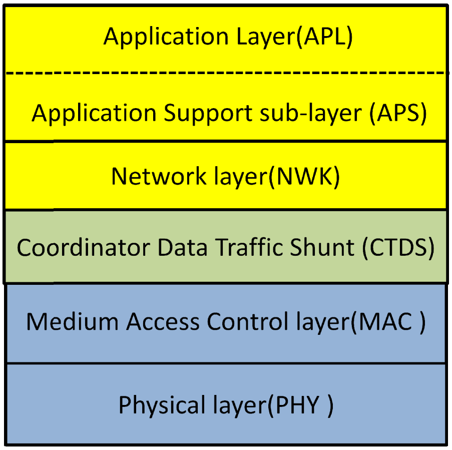

Application Layer(APL)
Application Support sub-layer (APS)
Network layer(NWK)
Coordinator Data Traffic Shunt (CTDS)
Medium Access Control layer(MAC)
Physical layer(PHY)

Fig. 4. CDTS Layer in Zigbee stack

While CTDS layer processes the data packets toward the sink, it modifies the PAN destination address to be the CDTS router itself instead of the coordinator, as illustrated in Fig. 5. Because the destination becomes itself, NWK layer forwards the data packets to its upper layer, APS layer, instead of sending them to the coordinator.

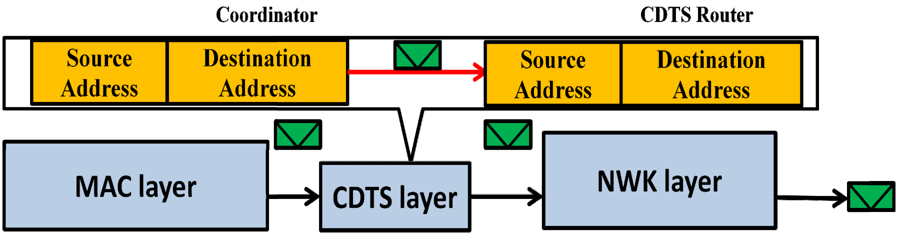


Fig. 5. CTDS Layer modifies PAN destination address

As illustrated in Fig. 6, the APS layer starts to build a cluster binding with the Zigbee node of source PAN address and sends back a ACK message without notifying the coordinator. While receiving sensor data by the binding cluster, the CDTS router then forwards the data to the sink by its DCL. Thus, the CDTS router seamlessly handles data traffic for the coordinator so R_{1C} is resolved. In addition, the sink applications receive data packets as usual without being modified so R_{2B} is resolved.

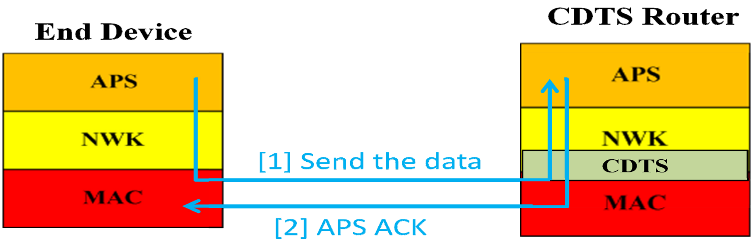


Fig. 6. Seamless cluster binding at the CDTS router

4 Evaluation

To evaluate our model, we implement the model in TI CC2530 Zigbee platform and Network Simulator 2 (NS2) simulation. The simulation is based on IEEE 802.15.4 and traffic sources are Constant Bit Rate (CBR). The evaluation of our work is based on the following performance metric:

- *Packet Deliver Ratio (%):* packets sent from Zigbee sensor nodes / packets received at the sink node.
- *Coordinator Loading (bps):* packets processed at the coordinator in bps.
- *Sink Throughput (bps):* packets received at the sink node in bps.

4.1 Embedded System

We use the CC2530 ZigBee Development Kit with ZigBee2007/PRO. The development kit comes with four ZBDC51 module and ZBDC51MB development boards. We use IAR Embedded Workbench software to compile the source code to create a hex file which downloads through the burner to RAM of the CC2530 chip. The communication between the host computer and the CC2530 Zigbee platform is RS232 serial port at baud rate of 115.2 K bps. The experiment network topology is 4 node square in 100*100 meter size. The experiment duration is set to 30 seconds. The number of data transmission rate from 7.5k to 47.3 kb/s according to the payload size.

As illustrated in Fig. 7, packet delivery ratio in the original Zigbee stack drops a lot while sink throughput increases from 50 K bps, but the same ratio in CTDS stays above 80%. This shows CDTS group can perform great data traffic shunt so packet deliver ratio does not drop as the throughput demand increases. However, since all data packets go to the coordinator in the original Zigbee stack, the ratio cannot sustain while the throughput demand increases. Thus, packet deliver ratio provides direct evidences that CDTS group does a great job for data traffic shunt for Zigbee networks.

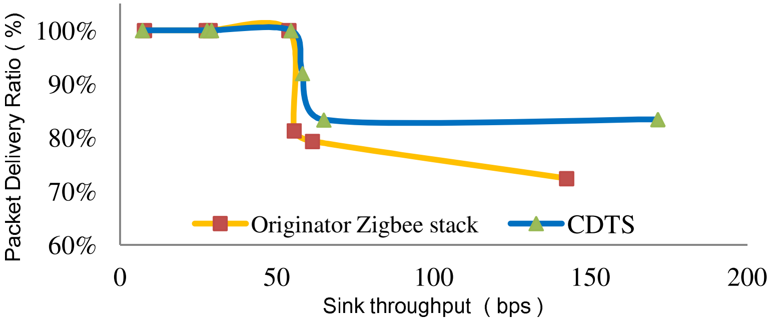


Fig. 7. Packets Delivery Ratio

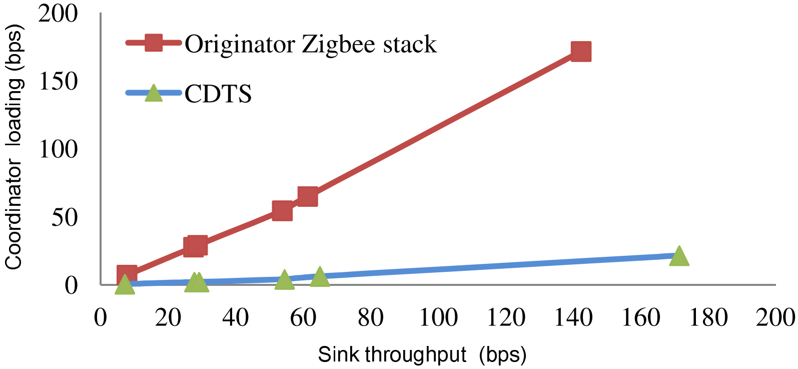


Fig. 8. Coordinator loading

As illustrated in Fig. 8, coordinator loading in the original Zigbee stack increases a lot while sink throughout increases from 60K bps, but the same loading value in CDTS stays below 20 bps. Obviously, increasing sink throughput results in extremely high network burden at the coordinator in the original Zigbee stack since all traffic must go through the coordinator. In CDTS, because of data shunt feature, increasing sink throughput does not result in the heavy network burden at the coordinator. This indicates that the coordinator is not the network bottleneck any more, and it can last much longer time, which means the whole PAN network can survive longer time.

4.2 NS2

We also perform similar experiments in NS2 simulation environment for scalability tests. The network size is 200*200m with transmission range as 30 meter. The duration of the simulation is set to 100 seconds. The CDTS group has 3 CDTS routers, and the CBR rate is 28 K bps at each Zigbee senor node. As illustrated in Fig. 9, packet delivery ratio in the original Zigbee stack starts to drop while the number of nodes increases from 4, but the same ratio in CDTS remains relatively high. This shows CDTS model is much more scalable than the original Zigbee stack because of CDTS data shunt feature.

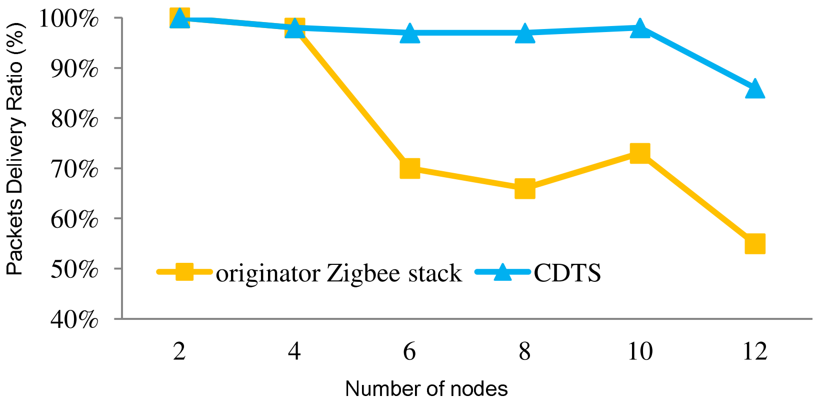


Fig. 9. Packets Delivery Ratio with 3 CDTS routers

5 Conclusion

Zigbee has lots of emerging sensor application because it has the advantages of low cost and easy deployment. Scalability becomes its major problem due to its coordinator design. First, we analyze Zigbee stack design and identify two groups of requirements for the desirable solution. Second, we propose Coordinator Data Traffic Shunt (CDTS) model and show CDTS successfully resolve the requirements. CDTS group performs data traffic shunt feature that forwards sensor data to the sink node without going through the coordinator. CDTS layer in the CDTS routers intercept packets and change the PAN destination address to be CDTS router itself such that APS layer can easily establish cluster binding from the sensor node toward the CDTS router itself instead of the coordinator. This clear and effective design of CDTS model thus provide a scalable solution for Zigbee sensor network and is fully compatible with existing Zigbee stack. At last, we implement CDTS model in TI CC2530 platform and NS2 simulation. Experiment results show CDTS provides better packet deliver ratio and scalability as well as lower coordinator loading than original Zigbee stack while increasing the throughput at the sink node. Thus, CDTS successfully resolves Zigbee bottleneck problem and is fully compatible with current Zigbee stack design.

References

1. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, pp. 56–67 (August 2000)
2. Shah, R.C., Rabaey, J.M.: Energy aware routing for low energy ad hoc sensor networks. In: Wireless Communications and Networking Conference, pp. 350–355 (March 2002)
3. Xu, X., Yuan, D., Wan, J.: An Enhanced Routing Protocol for Zigbee/IEEE 802.15.4 Wirless Networks. In: Second International Conference on Future Generation Communication and Networking, pp. 294–298 (December 2008)

4. Liu, R., Guo, Q., Fu, Y., Kong, X.: A New Cluster Routing Strategy Based on ZigBee. In: International Conference on Web Information Systems and Mining, pp. 271–274 (October 2010)
5. Zen, K., Lenando, H., Jambli, M.N.: Load balancing based on nodes distribution in mobile sensor network. In: International Conference on IT in Asia (CITA), pp. 1–6 (July 2011)
6. Sun, Z., Zhang, X.-G., Ruan, D., Li, H., Pang, X.: A Routing Protocol based on Flooding and AODV in the ZigBee Network. In: International Workshop on Intelligent Systems and Applications, pp. 1–4 (2009)
7. Huang, K.-L., Tseng, C.-C., Wang, J.-T., Yang, T.-H.: A Controller-Assisted Distributed (CAD) Load Balancing Scheme for ZigBee Networks. In: International Conference on Parallel Processing Workshops, pp. 1–5 (September 2011)

A Zone-Based Localization Scheme for Wireless Sensor Networks with Bilateration[*]

Chi-Chang Chen, Chi-Yu Chang, and Yan-Nong Li

Department of Information Engineering, I-Shou University, Kaohsiung City 84001, Taiwan
cchen@isu.edu.tw, {pop449,death-fierce}@hotmail.com

Abstract. This study presents an economical yet effective localization scheme for wireless sensor networks (WSNs). The proposed scheme uses only two anchor nodes and uses bilateration to estimate the coordinates of unknown nodes. Many localization algorithms for WSNs require the installation of extra components, such as GPS receivers, ultrasonic transceivers, and unidirectional antennas, to sensors. The proposed localization scheme is range-free (i.e., not demanding any extra devices for the sensors). Experimental results show that the range error of the proposed localization scheme is less than 0.3 when the communication range is greater than 40m and node density is greater than 0.0075(node number = 300 with 200*200 m^2 region). This study compares the proposed scheme with the well-known DV-Hop method. Simulation results show that the proposed scheme outperforms the DV-Hop method in localization accuracy, communication cost, and computational complexity.

Keywords: Bilateration, Localization Scheme, Range-Free, Wireless Sensor Networks, Zone-Based Method.

1 Introduction

Wireless sensor networks (WSNs) have gained worldwide attention in recent years. A WSN consists of spatially distributed autonomous sensors that cooperatively monitor a deployed region for physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, and pollutants.

The transmission power consumed by a wireless radio is proportional to the distance squared or even a higher order in the presence of obstacles. Thus, multi-hop routing is usually considered for sending collected data to the sink instead of direct communication. Most WSN routing algorithms require the position information of sensor nodes. However, for some hazardous sensing environments, it is difficult to deploy the sensor nodes to the required locations. Thus, for environments in which it is difficult to plan the location of sensors in advance, localization techniques can be used to estimate sensor positions. The simplest and most common localization

[*] This work was partially supported by the I-Shou University under Grant ISU100-01-06 and the Ministry of Education under the Interdisciplinary Training Program for Talented College Students in Science, 100-B4-01.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 511–520.
DOI: 10.1007/978-3-642-35452-6_52

technique is to install a GPS receiver on each sensor in the sensor networks. Although the cost of GPS receivers is falling, they are still too costly, in price and energy consumption, to install in a sensor network.

This study proposes an economical yet effective WSN localization scheme. This scheme needs only two anchor nodes and uses low-complexity operations to estimate the location of unknown nodes. This study also compares the performance of the proposed scheme with the DV-Hop method [1] to show its superiority.

The rest of this paper is organized as follows: Section 2 reviews related research on WSN localization algorithms. Section 3 presents the communication protocol used to divide the deployed region into zones and the preliminary localization method. Section 4 introduces the more accurate enhanced method to estimate the positions of the unknown sensor nodes. Section 5 provides a simulation of the proposed localization scheme and a comparison of its performance with the DV-Hop method. Finally, Section 6 offers a conclusion.

2 Related Work

Research interest in WSN localization has increased greatly in recent years. Traditional WSN localization technologies can be divided into two categories: range-based methods and range-free methods [2]. A range-based method positions the sensor nodes using additional devices, such as timers, signal strength receivers, directional antennas, and antenna arrays. In contrast, a range-free method requires no additional hardware, and instead uses the properties of the wireless sensor network and the appropriate algorithms to obtain location information. Range-free localization can be further divided into two categories: local techniques and hop-counting techniques [2].

In the local techniques, a node with unknown coordinates collects the position information of its neighbor beacon nodes with known coordinates to estimate its own coordinate. Examples of local techniques can be found in [2].

The hop-counting technique, called the DV-Hop method, was first proposed by Niculescu and Nath in [1]. In the DV-Hop method, each unknown node asks its neighboring beacon nodes to provide their estimated hop sizes and then attempts to obtain the smallest hop count to its neighbor beacon nodes using the designated routing protocol. Each unknown node estimates the distances to its neighbor beacon nodes by the hop counts to them and the hop size of the closest beacon node. The unknown nodes then apply trilateration to estimate their position based on the estimated distances to three suitable neighbor beacon nodes. There are many follow-up studies of the DV-Hop method, which can be found in [3-5].

Bilateration, which is derived from multilateration, is based on the distance differences from an unknown node to two beacons at known locations. Unlike multilateration, which usually uses an iterative method to estimate the location of an unknown node, bilateration uses a basic geometry property, the intersection of two circles, to calculate the location of an unknown node. The computation of bilateration is much simpler than that of multilateration, which usually applies more expensive computation such as the least-squares method. In [6], the authors presented a distributed and formula-based bilateration algorithm that can be used to provide an initial set of locations. A similar bilateration algorithm was also proposed by the authors in [7] independently.

3 Zone-Based Localization Scheme

This section presents the proposed localization scheme to estimate the location of sensors. The following section extends the scheme to obtain a more accurate estimation of sensor locations.

3.1 Localization Scheme

In the proposed localization scheme, called the Zone-Based Localization Method (ZBLM), two sink nodes are installed at the lower-left corner (Sink X) and the lower-right corner (Sink Y) of a square monitored region. This scheme assumes that 1) all the sensors are homogeneous, 2) they are uniformly deployed, and 3) the network is connective. The ZBLM consists of three major steps: compute minimum hop counts by flooding, divide the monitored region into zones, and assign the coordinate of sensors in each zone by bilateration.

Step 1: Compute minimum hop counts by flooding
First, we allow both Sink X and Sink Y to broadcast a Hop-Counting packet (HC packet in short) to their neighbor sensors. The HC packet contains two fields: 1) Min_hc (minimum hop count to the source node), initialized to 0, and 2) the source node ID (0 for Sink X and 1 for Sink Y).

Each sensor records two current minimum hop count values (say X_{hop} and Y_{hop}) to Sink X and Sink Y, respectively, which are both set to positive infinity. Once a sensor receives an HC packet, it checks the hop count field Min_hc in the HC packet. If Min_hc+1 is smaller than its corresponding current minimum hop count value X_{hop} (or Y_{hop}), then it increases Min_hc by one before forwarding the packet to its neighbors, and updates its X_{hop} (or Y_{hop}) to the new Min_hc value accordingly. Otherwise, the sensor discards incoming HC packets with larger hop count values.

Step 2: Divide the monitored region into zones
After finishing the flooding of HC packets by Step 1, each sensor should have two minimum hop-count values (X_{hop},Y_{hop}) for Sink X and Sink Y, respectively. Sensors with the same (X_{hop},Y_{hop}) pair are in the same zone (Note: the following subsection claims that the zone tends to be a geometry quadrilateral), and is denoted as zone(X_{hop},Y_{hop}). Fig. 1 shows a scenario of dividing the monitored region into zones, in which the color irregular arcs are added for ease of visualization. In this figure, each node has its own (X_{hop},Y_{hop}) pair. For example, X_{hop} of Node A is 3 and Y_{hop} is 8. Therefore, Node A is in zone(3,8). Similarly, Node B is in zone(6,5), and Node C is in zone(5,7).

Step 3: Assign the coordinate of sensors in each zone by bilateration
Suppose that the coordinates of Sink X and Sink Y are (0,0) and (w,0), respectively, where w is the length of the square monitored region. Denote the distances from an unknown sensor S to Sink X and to Sink Y as d_x and d_y (the estimation of d_x and d_y

will be given in the next subsection), respectively. The coordinate (x,y) of Sensor S is the intersection of two circles centered at (0,0) and (w,0), respectively. Therefore, (x,y) can be obtained using the following equations:

$$\begin{cases}(x-0)^2+(y-0)^2={d_x}^2\\(x-w)^2+(y-0)^2={d_y}^2\end{cases} \tag{1}$$

Thus, $x=\frac{{d_x}^2-{d_y}^2+w^2}{2w}$ and $y=\pm\sqrt{{d_x}^2-x^2}$

Because sinks are installed at the lower left and lower right corners of the monitored region, we can only take the positive solution of y. Therefore, the coordinate of the unknown sensor *S* is

$$\left(\frac{{d_x}^2-{d_y}^2+w^2}{2w}\ ,\ \sqrt{{d_x}^2-\left(\frac{{d_x}^2-{d_y}^2+w^2}{2w}\right)^2}\right) \tag{2}$$

For (1) to produce a real solution, the sum of d_x and d_y (the radii of two circles) must be greater than or equal to *w* (the distance between two centers). This constraint is considered when estimating the distances d_x and d_y.

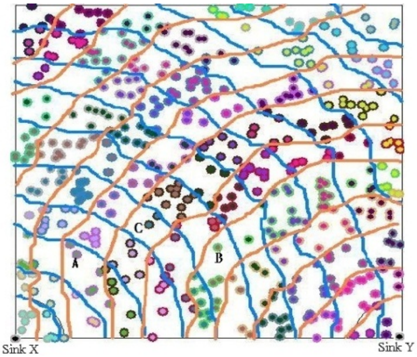


Fig. 1. A scenario of 300 sensors with a communication range of 20 m, with the monitored region (200 x 200 m^2) divided into zones. The colored irregular arcs are added to facilitate visualization of the zones.

3.2 Estimate the Hop Distances between Sensors and the Sinks

At first, we claim that sensors with the same (X_{hop}, Y_{hop}) pair tend to congregate in a quadrilateral.

Suppose the length of the square monitored region is m, the communication range of each sensor is CR, and the total number of nodes is n. The probability that a node v is within the communication range of another given node u is $\frac{\pi\times CR^2}{m^2}$. Since the total number of nodes is n, the expected number of neighbor nodes, say ρ, for u is

$(n-1)\frac{\pi\times CR^2}{m^2}$. For example, if m=200, CR=30, and n=300, then $\rho = (300-1)\times\frac{3.14\times 30^2}{200^2} \cong 21$. Previous research [8] provided a more precise estimation of node degree that considers the border effect. According to [8], $\rho = (n-1)(\pi r^2 - \frac{8}{3}r^3 + (\frac{11}{3}-\pi)r^4$, where $r = \frac{CR}{m}$. Therefore, $\rho \cong 19$ for the example.

According to [9], message forwarding between any two nodes through flooding occurs along the straight line path with a probability greater than 0.85 if the number of neighbor nodes is greater than or equal to 15. According to the previous paragraph, if CR=20, then ρ is greater than 15 as long as n>478. Alternatively, if CR=30, then ρ is greater than 15 as long as n>213. Thus, if both the forwarding paths from Sink X and Sink Y progress along straight lines, then the sensors with the same (X_{hop}, Y_{hop}) pair tend to congregate in a quadrilateral (called a "zone" in this paper). The experiment in this study shows that the "zone effect" still exists for the case CR=20 and n=300 ($\rho \cong 9$) (Fig. 1).

The following discussion presents two extreme cases in which the message is forwarded along the straight line. For high density, each sensor has a certain number of sensors within its communication range. Therefore, for Sink X (or Sink Y), it is highly possible that sensors are located at the edge of the communication range. For the extreme case shown in Fig. 2(a), sensor nodes always exist at the edge of the communication range of each hop from the sink. Therefore, assuming that the communication range is CR, the maximum distance of a sensor with hop count n from the sink is $n\times \text{CR}$.

The other extreme case occurs when the density of sensor nodes in the region is low and each sensor node has few neighbors, yet the network remains connective. As Fig. 2(b) shows, sensor nodes are two in a group located close to the edge of the communication range. The first node in each group is within the communication range of the second node of the previous group, but immediately beyond the communication range of the first node of the previous group. Meanwhile, the second node in each group is immediately beyond the communication range of the second node of the previous group.

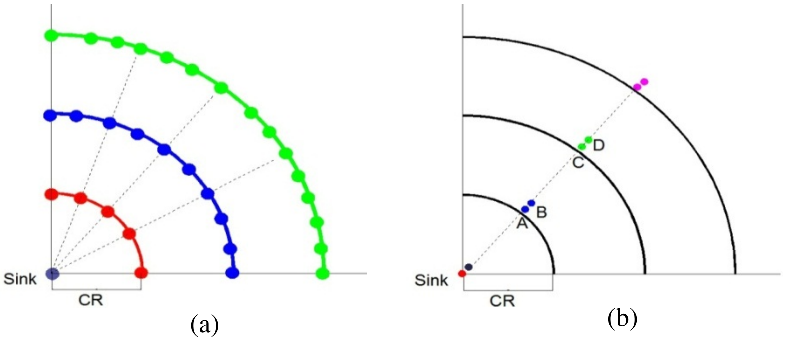
For example, in Fig. 2(b), Node C is within the communication range of Node B, but immediately beyond the communication range of Node A. Node D is immediately beyond the communication range of Node B. Thus, the minimum distance of sensors with hop count n is $\lfloor n/2 \rfloor \times \text{CR} + \epsilon$, where ϵ is the sum of distances between the two nodes in the same group. For example, the hop count of Node C is 4, and the distance to the Sink is $2\times\text{CR}+\epsilon$, and the hop count of Node D is 5, and the distance is $2\times\text{CR}+\epsilon'$, where ϵ' is a value larger than ϵ. If the two nodes of each group are very close to each other yet still satisfy these conditions, then we can ignore the small value ϵ and say that the minimum distance of sensors with hop count n is $\lfloor n/2 \rfloor \times \text{CR}$.

This analysis indicates that if the messages are forwarded along a straight line, the distance to the sink for any sensor with hop count n is between $\lfloor n/2 \rfloor \times \text{CR}$ and $n\times\text{CR}$ (ϵ is ignored). Therefore, if a sensor S is located in zone(m, n) (i.e., it has minimum hop count values (m, n) to Sink X and Sink Y), then we can use the following formula to approximate the distances, d_x and d_y, of sensor S to Sink X and Sink Y, respectively:

$$d_x = \{\lfloor {}^{m}/_{2} \rfloor + [(m - \lfloor {}^{m}/_{2} \rfloor) * \alpha_1]\} * \mathrm{CR} \tag{3}$$

$$d_y = \{\lfloor {}^{n}/_{2} \rfloor + [(n - \lfloor {}^{n}/_{2} \rfloor) * \alpha_2]\} * \mathrm{CR} \tag{4}$$

where α_1 and α_2 are parameters between 0 and 1. For simplicity, this study uses the same value α for both α1 and α2. To have a real solution for (1), it is necessary to rule out the α values that cause $d_x + d_y < w$. Section 5 shows that the value of α is related to the communication range and the density of the sensors, and identifies the best α value that minimizes the location error of ZBLM for each condition in a WSN.



(a) (b)

Fig. 2. (a). A scenario of maximum distance to the sink: sensor nodes are located at the edge of the communication range. Thus, the maximum distance of sensors from the sink with hop count n is n × CR, where CR is the communication range. (b). A scenario of minimum distance to the sink: sensor nodes are two in a group located close to the edge of the communication range.

4 Enhanced Zone-Based Localization Method

The last section presents a localization scheme to estimate the positions of unknown sensors and show that the sensors with the same hop-count pair tend to be clustered in the same zone. Sensors in the same zone have the same estimated coordinates. This can cause a certain amount of estimation error unless these sensors are located at the same place, and the error increases as the area of each zone increases.

For convenience, this study calls the coordinate of a sensor obtained using the ZBLM scheme the ZBLM coordinate. This section proposes an adjustment algorithm, called the Enhanced Zone-Based Localization Method (EZBLM), to reduce the estimation error. The basic concept of this algorithm is to determine the possible location of a sensor in the zone where it belongs, and adjust the coordinate of the given sensor by the ZBLM coordinates of its relevant neighbor zones. The following paragraphs detail how to determine which neighbor zones are closer to a given sensor in a zone, and how to adjust the coordinate.

Enhanced Zone-Based Localization Method

Step 1: Each sensor uses half the communication range to broadcast a message that contains its ID, its hop-count pair to Sink X and Sink Y, and its ZBLM coordinate.

(Note: According to our simulation, the outcome of broadcasting using a half communication range is better than that of the full communication range, because larger communication range may include too many neighbor nodes and therefore loss the position characteristic.)

Step 2: Each sensor that receives messages from neighbor nodes adjusts its coordinate according to the following step:

a. Extract the ZBLM coordinate from each received message, and consider the extracted coordinates (remove the duplicates) as a set of points S. Compute the centroid, say (x_c,y_c), of the points in S (i.e., take the arithmetic mean of all the points).
b. Suppose the ZBLM coordinate of the sensor to be adjusted is (x_u, y_u). The adjusted coordinate is set to the center of the two coordinates (i.e., $(\frac{x_c+x_u}{2}, \frac{y_c+y_u}{2})$).

The next section provides a comparison of the error rate of the coordinates obtained using both ZBLM and EZBLM, showing that EZBLM significantly improves the error rate of ZBLM.

5 Performance Analysis and Simulation Results

This section first identifies the values of α by experiments, and suggests the best choice of the α value for each condition. We then compares two performance metrics, communication overhead and computation overhead, for algorithms of the ZBLM, EZBLM, and DV-Hop. Finally, we simulate the three methods separately and compare their range errors. The range error is defined as follows:

$$Range\ Error = \frac{\sqrt{(X_{real} - X_{est})^2 + (Y_{real} - Y_{est})^2}}{CR}$$

where (X_{real}, Y_{real}) and (X_{est}, Y_{est}) are real coordinates and estimated coordinates, respectively, of a given unknown sensor. CR is the communication range.

5.1 Determine the Value of α

The term α is a parameter used to estimate the hop distance for each sensor to the sinks using (3) and (4). The value of α represents the ratio of the estimated hop distance to the communication range, and depends on the values of the communication range and the node density. However, as far as we know, no formula can calculate the exact value of α. Therefore, this study determines the value of α through experiment. The value of α is tested from 0.05 to 1.0 in 0.05 intervals. Each value of α is used to compute the location error of the ZBLM for each deployment. Table 1 lists the best α value that generates the least location error for each combination of communication range and node density over 500 different deployments.

Table 1. The best α value for different node densities and communication ranges

Number of Sensors (Node Density) Communication Range (meters)		300 (0.0075)	400 (0.01)	500 (0.0125)	500 (0.015)	600 (0.0175)	800 (0.02)	900 (0.0225)	1000 (0.025)
20	α values of least location error	0.45	0.5	0.55	0.6	0.65	0.65	0.7	0.7
30		0.6	0.65	0.7	0.7	0.7	0.7	0.75	0.75
40		0.65	0.7	0.7	0.7	0.7	0.7	0.75	0.75
50		0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
60		0.65	0.65	0.65	0.7	0.7	0.7	0.7	0.7

5.2 Performance Analysis

According to the algorithm proposed in Section 3, the ZBLM individually needs two flooding operations from Sink X and Sink Y. The coordinate estimation simply computes the intersection of two circles, which takes constant time and uses basic arithmetic operations such as addition, multiplication, and the square root.

In addition to the flooding operations needed for the ZBLM, the EZBLM requires one broadcasting operation for each node to determine the position of each unknown node in its zone. The adjustment of coordinate uses two average operations, which takes constant time.

In the DV-Hop method (described in Section 2), each node (both beacon nodes and unknown nodes) needs one flooding operation to calculate the minimum hop counts to all other nodes and the hop size for each beacon node. Each beacon node needs one extra flooding to broadcast the hop size to all the unknown nodes. Therefore, this method requires (number of all nodes + number of beacon nodes) flooding operations. Each unknown node uses trilateration to estimate its location. The trilateration needs a variable number of iterations (from two to hundreds in our experiments) to converge to a point.

Both the ZBLM and EZBLM use only two anchor nodes. The simulations in [1] show that the DV-Hop method requires at least 20% of the sensors to be anchor nodes to obtain better results. Table 2 presents a performance comparison of these methods.

Table 2. Performance comparison of the ZBLM, EZBLM, and DV-Hop (*n* is the total number of nodes)

	Communication cost	**Computation cost**	**Number of anchor nodes**
ZBLM	2 flooding operations	constant	2
EZBLM	2 flooding operations + *n* broadcast operations	constant	2
DV-Hop method	(1+20%)×*n* flooding operations	a variable number of iterations for trilateration	20%×*n*

5.3 Simulation Experiments

The simulation environments in this study were established as follows. The monitored region measured 200 m × 200 m. The number of sensors ranged from 300 to 1000, and the communication ranges are from 20 to 60 m. Sensors were uniformly deployed in the region. The α values were chosen from Table 1. Each simulation included 50 tests, with the mean serving as the final result.

Figs. 3 show the range errors of both the ZBLM and EZBLM, respectively, for different communication ranges and number of sensors. As expected, under the same communication range, the location error decreases as the sensor density increases for both the ZBLM and EZBLM. These simulation results show that the EZBLM improves the ZBLM scheme significantly.

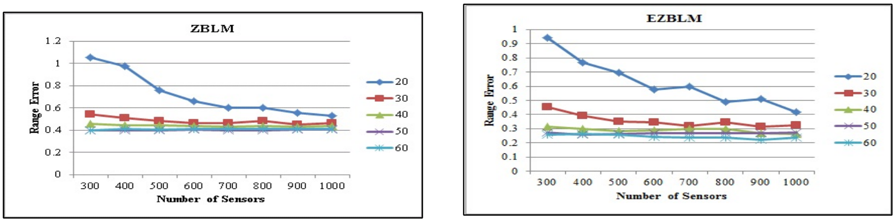


Fig. 3. Range errors under different communication ranges (20-60 m) and node densities for ZBLM and EZBLM

For the simulation of the DV-Hop method, the rate of anchor nodes was set to 20% because its performance drops significantly when using less than 20% of anchor nodes [1]. Fig. 4 shows that both the ZBLM and EZBLM outperform DV-Hop, except for the case of CR=20 and node number less than 500. These methods use only two

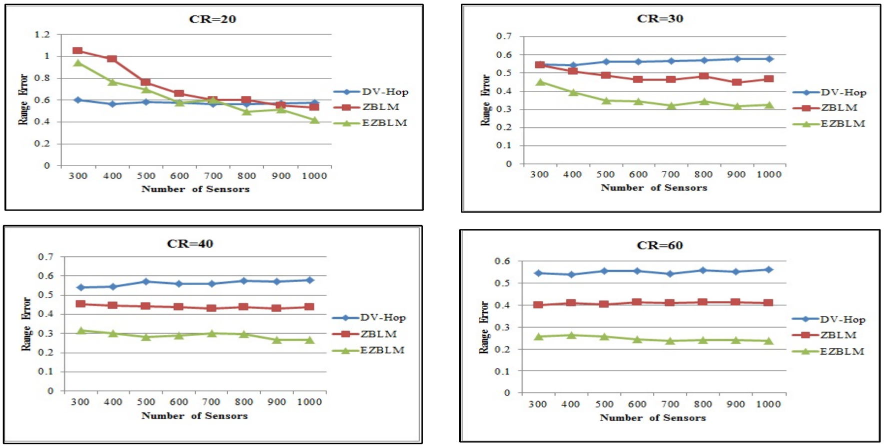


Fig. 4. Range errors of the proposed methods (ZBLM and EZBLM) vs. DV-Hop under different communication ranges

anchor nodes, whereas the DV-Hop method uses 20% of sensors as anchor nodes. These simulation results show that the proposed methods are more accurate than the well-known DV-Hop method.

6 Conclusions

Many studies have attempted to solve the range-free localization problems of WSNs. Most of them demand many anchor nodes and use the multilateration method, which requires complex computation and a variable number of iterations to estimate the location of sensors. This study proposes two range-free localization methods that use only two anchor nodes and apply the low-complexity bilateration method. Experimental results show that the range error of the EZBLM is less than 0.3 for all cases with node density greater than 0.0075 (node number = 300 with 200*200 m^2 region) when CR>=40m. Almost all the simulation results for the proposed method are better than those of the DV-Hop method, which requires many anchor nodes and more complex computations.

Although the proposed scheme uses a square monitoring region, the same algorithm can be applied to rectangular monitoring regions. In the future, we will work on other shapes of monitor regions.

References

1. Niculescu, D., Nath, B.: Ad Hoc Positioning System(APS). In: Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM 2001), San Antonio, USA, vol. 5, pp. 2926–2931 (December 2001)
2. Liu, F., Cheng, X., Hua, D., Chen, D.: TPSS: A Time-based Positioning Scheme for Sensor Networks with Short Range Beacons. In: Li, Y., Thai, M.T., Wu, W. (eds.) Wireless Sensor Networks and Applications, pp. 175–193. Springer, New York (2008)
3. Boukerche, A., Oliveira, H.A.B.F., Nakamura, E.F., Loureiro, A.A.F.: DV-Loc: A Scalable Localization Protocol Using Voronoi Diagrams for Wireless Sensor Network. IEEE Wireless Communications, 50–55 (2009)
4. Wang, Y., Wang, X., Wang, D., Agrawal, D.P.: Range-free Localization Algorithm usingExpected Hop Progress in Wireless Sensor Networks. IEEE Transactions on Parallel andDistributed Systems 20(10), 1540–1552 (2009)
5. Ma, D., Er, M.J., Wang, B., Lim, H.B.: Range-free wireless sensor networks localization based on hop-count quantization. Telecommunication Systems (to appear, 2012)
6. Cota-Ruiz, J., Rosiles, J.-G., Sifuentes, E., Rivas-Perea, P.A.: Low-Complexity Geometric Bilateration Method for Localization in Wireless Sensor Networks and Its Comparison with Least-Squares Methods. Sensors 12(1), 839–862 (2012)
7. Li, X., Hua, B., Shang, Y., Guo, Y., Yue, L.: Bilateration: An Attack-Resistant Localization Algorithm of Wireless Sensor Network. In: Kuo, T.-W., Sha, E., Guo, M., Yang, L.T., Shao, Z. (eds.) EUC 2007. LNCS, vol. 4808, pp. 321–332. Springer, Heidelberg (2007)
8. Li, K.: Topological Characteristics of Random Multihop Wireless Networks. In: Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops, Providence, Rhode Island, USA, pp. 685–690 (May 2003)
9. Bachrach, J., Nagpal, R., Salib, M., Shrobe, H.: Experimental Results for and Theoretical Analysis of a Self-Organizing Global Coordinate System for Ad Hoc Sensor Networks. Telecommunications Systems Journal 26(2-4), 213–233 (2004)

Improving Pairwise Key Predistribution in Wireless Sensor Networks

Neng-Chung Wang[1] and Hong-Li Chen[2]

[1] Department of Computer Science and Information Engineering,
National United University, Taiwan, R.O.C.
ncwang@nuu.edu.tw
[2] Department of Computer Science and Information Engineering,
Chaoyang University of Technology, Taiwan, R.O.C.
s9427615@mail.cyut.edu.tw

Abstract. In this paper, an efficient grid-based pairwise key predistribution scheme for wireless sensor networks (WSNs) is proposed. In the proposed scheme, multiple polynomials for each row, each column, and each diagonal in the grid are constructed. Then, each sensor node in each row, column, and diagonal in the grid establishes a pairwise key with the other node using the predistributed symmetric polynomial. Performance evaluation shows that the proposed scheme outperformed Sadi's scheme.

Keywords: Grid-based, key predistribution, pairwise key, symmetric cryptography, wireless sensor networks.

1 Introduction

Wireless sensor networks (WSNs) are gaining wide applications and booming research interest in recent years [6]. The current applications of WSNs cover a wide range, including military, environmental, and scientific applications. Sensor nodes can be easily compromised by an adversary because they are usually deployed in a hostile environment [1, 6].

A pairwise key increases security of communication between sensor nodes. Traditional techniques for pairwise keys, such as asymmetric key cryptography and key distribution center (KDC), can be used. In the asymmetric key approach, a private key is used for encryption and a public key is used for decryption. However, this is too complicated and expensive for WSNs. In KDC approach, a KDC assigns a signal master key to each node before it is deployed. After deployment, any two nodes can communicate with each other by a master key. If any one node is compromised by an adversary, it crashes the entire network's security.

Symmetric cryptography is preferred in WSNs because of its low computation and communication cost. In symmetric cryptography, two communication parties need to share a secret key before they communicate with each other. Due to unpredictable network topology such as sensor network, distributing secret keys to the sensor nodes

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 521–530.
DOI: 10.1007/978-3-642-35452-6_53

securely and efficiently is a big challenge. There are many key predistribution schemes proposed for WSNs [6, 7, 11, 13]. The basic idea of the above schemes is pre-loading a set of symmetric key into sensor nodes before they are deployed. If the deployed nodes share a common key, they can communicate securely with each other.

In this paper, an efficient grid-based pairwise predistribution scheme for WSNs is proposed. This scheme is more efficiently than previous proposals since any two nodes establish a pairwise key by path discovery. Furthermore, the most important property of this scheme is the significant improvement in the resilience compared to node capture.

The remainder of this paper is organized as follows. In Section 2, the related work is discussed. The proposed scheme is given in Section 3. Section 4 gives security analysis and evaluation of the proposed scheme. The paper is concluded in Section 5.

2 Related Work

Pairwise key establishment enables security services such as authentication and key management to achieve secure communication between sensors nodes [9, 10]. Due to the resource constrains of sensor nodes, it is not feasible to adopt traditional pairwise key establishment techniques. For security enhancement, Eschenauer and Gligor proposed a random key predistribution scheme [5]. The main idea of this proposed scheme is that each sensor node randomly picks up a set of keys from a key pool before it is deployed. Any two sensor nodes have a certain probability of sharing at least one common key.

Based on [5], Chen et al. proposed a "q-composite" key predistribution [3] that also uses a key pool but requires two sensor nodes to compute a pairwise key from at least q-predistributed keys that are common between them. The main difference between [3] and [5] is that the first scheme requires two neighboring nodes sharing at least q common keys to establish secure communication links. But both of them have poor resilience compared to node capturing as the number of compromised nodes increases, the fraction of affected pairwise keys increases quickly. In [8], Liu and Ning proposed two schemes, random subset assignment and grid based key predistribution. Liu' and Ning' scheme has a high probability to establish a pairwise key, but the performance deteriorates while a certain number of nodes are captured.

In [2], Blom proposes a key predistribution method. It enables any two sensor nodes in a group to communicate with each other by computing a common key. In Blom' scheme, a matrix *G* and a symmetric matrix *D* are constructed first. Matrix *G* is computed as $(\lambda+1)\times n$ and symmetric matrix *D* is computed as $(\lambda+1)\times(\lambda+1)$. Then *n* is denoted as the size of the group, and λ is denoted as the expected threshold to compromise the secret collusively. Each node in the group randomly stores a row vector from matrix *A* and the corresponding column vector from matrix *D*. Matrix *A* is computed by $(D\cdot G)^{\mathrm{T}}$. For example, node *u* gets the *i*th row from matrix *A* and *i*th column from *G* and node *v* gets the *j*th row from matrix *A* and *j*th column from *G*. Because node *u* and node *v* want to establish a common key, they exchange their column vector first. Then both nodes use their row vector to multiply their partner's column vector. Due to the same symmetric matrix, node u and node v share the common key after the calculation.

Based on [2, 3, 5], Du et al. proposed a pairwise key predistribution [4] for wireless sensor networks. In [4], if any two neighboring nodes share a common key, they can establish a pairwise key directly. Otherwise, they establish the pairwise keys using path discovery. This scheme improves the sensor network resilience but it does not guarantee any two neighboring nodes establishing a pairwise key directly. Although the path discovery procedure increases the network connectivity, additional communication and computational costs are also increased.

In order to overcome the shortcomings of these schemes, a grid based random key predistribution scheme (Sadi's scheme) for WSNs was proposed [12]. It combines a probabilistic random key predistribution scheme with a grid based scheme. This scheme has a number of attractive properties. First, it guarantees that any two sensors can establish a pairwise key when there are no compromised sensors, provided that the sensors can communicate with each other. Second, it is resilient to node compromise. Even if some sensors are compromised, there is still a high probability of establishing a pairwise key between non-compromised sensors. Third, a sensor can directly determine whether it can establish a pairwise key with another node, and if it can, which polynomial should be used.

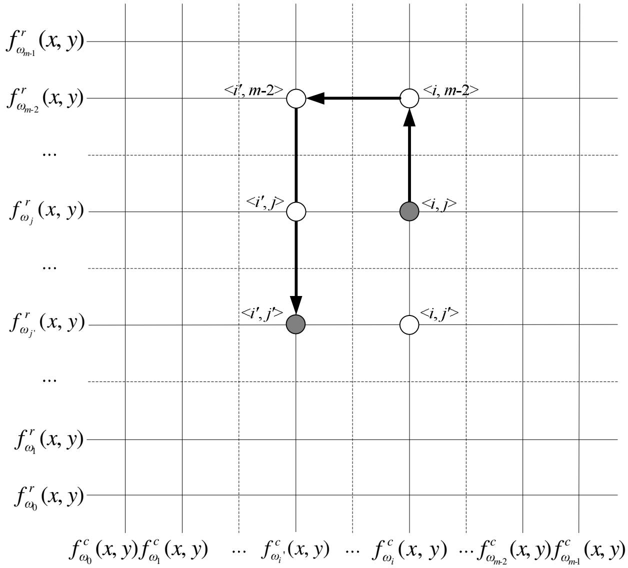

$f^r_{\omega_{m-1}}(x,y)$
$f^r_{\omega_{m-2}}(x,y)$
...
$f^r_{\omega_j}(x,y)$
...
$f^r_{\omega_{j'}}(x,y)$
...
$f^r_{\omega_1}(x,y)$
$f^r_{\omega_0}(x,y)$
<i', m-2>
<i, m-2>
<i',j>
<i,j>
<i',j'>
<i,j'>
$f^c_{\omega_0}(x,y)$ $f^c_{\omega_1}(x,y)$ ... $f^c_{\omega_{i'}}(x,y)$ ... $f^c_{\omega_i}(x,y)$ ... $f^c_{\omega_{m-2}}(x,y)$ $f^c_{\omega_{m-1}}(x,y)$

Fig. 1. GBR key predistribution scheme

As shown in Fig. 1, the scheme allocates ω polynomials for each row and column in the grid. Then the scheme randomly selects τ polynomials from each row and column respectively and assign them to the sensor node. If two nodes share a common polynomial they can establish a secure pairwise key. Suppose there are most N sensor nodest in a sensor network. Then the grid based random key predistribution scheme constructs a $m \times m$ grid with a set of $2\omega m$ polynomials $\{f_i^c(x, y), f_i^r(x, y)\}_{i=0,\ldots,m\omega\text{-}1}$, where $m = \lceil \sqrt{N} \rceil$. For each sensor, the setup server picks an unoccupied intersection (*i*, *j*) in the grid and assigns it to the node. Thus, the ID of this sensor is <*i*, *j*>. The setup server then distributes ID, and 2τ polynomials to the sensor node. Here τ polynomials are randomly chosen from row *i* and the remaining τ from column *j*. As a result, the sensor node can establish a pairwise key using share discovery and path discovery. In Fig. 1, besides node <*i'*, *j*> and node <*i*, *j'*>, node <*i*, *m* - 2> and node <*i'*, *m* - 2> can work together to help node <*i*, *j*> to setup a common key with node <*i'*, *j'*>. There are up to 2*m* - 2 pairs of such nodes in the grid.

3 Proposed Scheme

In this section, a grid-based pairwise key predistribute scheme for WSNs is proposed. First, the notations and their definitions are given in Table 1.

Table 1. Notation and definition

Notations	Definition
ω	The number of polynomials in each group
m	The number of row and the number of column in the grid
F_q	The finite field
τ	The number of polynomials stored in a sensor node
(*i*, *j*)	The column and the row coordinates of a sensor node
<*i*, *j*>	The identity of a sensor node
$f_{\omega_0}^r$	The polynomials corresponding to the row in the grid
$f_{\omega_0}^c$	The polynomials corresponding to the column in the grid
$f_{\omega_0}^d$	The polynomials corresponding to the diagonal in the grid

Suppose a sensor network consists of *N* sensor nodes, an $m \times m$ grid with a set of $4m\omega$ polynomials $\{f_i^c(x, y), f_i^r(x, y), f_i^d(x, y), f_i^{d'}(x, y)\}_{i=0,\ldots m\omega\text{-}1}$ is constructed. Besides, each node is located at a unique intersection in the grid and the off-line server randomly selects τ polynomials from each row, column, and diagonal. If any two nodes share a common polynomial, they establish a pairwise key. Otherwise, they establish a pairwise key using the path discovery.

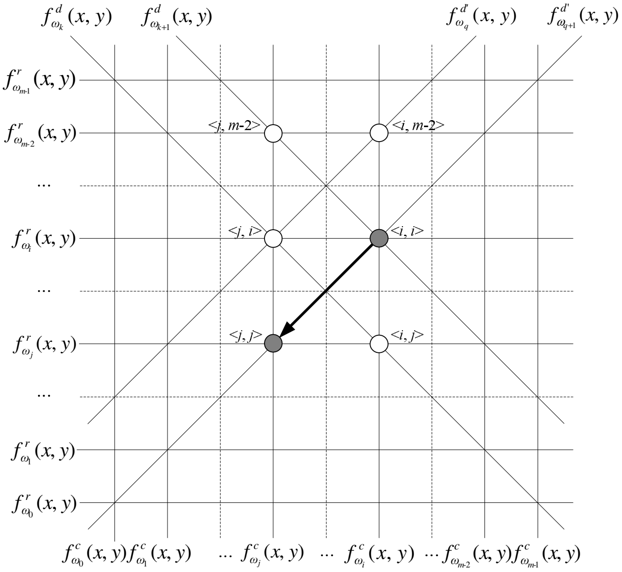
3.1 Subset Assignment

In the subset assignment phase, the setup server randomly generates $4m\omega$ symmetric polynomials of degree *t* over a finite field F_q. The m represents the number of groups which the setup server makes, and ω describes the number of polynomials each group has. Then each group is assigned to each row, column, and diagonal of the grid.

Each sensor node is assigned to a unique and unoccupied intersection (*i*, *j*) in the grid. The ID of a sensor is the concatenation of the binary representations of the column and the row coordinates. Syntactically, an ID is constructed from the coordinate (*i*, *j*) as <*i*, *j*>. Consequently the identity of the sensor node is <*i*, *j*>. Next, the setup server distributes identity and 4τ polynomials to the sensor node. Here is randomly selected from the *i*th row, *j*th column, and *k*th diagonal.

3.2 Polynomial Share Discovery

If both sensor nodes share the same polynomial, they can establish a pairwise key. For example, node *i* wants to establish a pairwise key with node *j*, and node *i* checks whether $c_i = c_j$, $r_i = r_j$, or $| c_i - c_j | = | r_i - r_j |$. If $(c_i, r_i) = (c_j + n, r_j + n)$, where *n* represents the real number, both node *i* and node *j* have a polynomial share of $f^{d'}_{\omega_{q+!}}(x, y)$ and establish a pairwise key by the $f^{d'}_{\omega_{q+!}}(x, y)$, as shown in Fig. 2.



$f^d_{\omega_k}(x, y)$
$f^d_{\omega_{k+1}}(x, y)$
$f^{d'}_{\omega_q}(x, y)$
$f^{d'}_{\omega_{q+1}}(x, y)$
$f^r_{\omega_{m-1}}(x, y)$
$f^r_{\omega_{m-2}}(x, y)$
...
$f^r_{\omega_i}(x, y)$
...
$f^r_{\omega_j}(x, y)$
...
$f^r_{\omega_1}(x, y)$
$f^r_{\omega_0}(x, y)$
<j, m-2>
<i, m-2>
<j, i>
<i, i>
<j,j>
<i,j>
$f^c_{\omega_0}(x, y) f^c_{\omega_1}(x, y)$ ... $f^c_{\omega_j}(x, y)$ ... $f^c_{\omega_i}(x, y)$ ... $f^c_{\omega_{m-2}}(x, y) f^c_{\omega_{m-1}}(x, y)$

Fig. 2. Polynomial share discovery

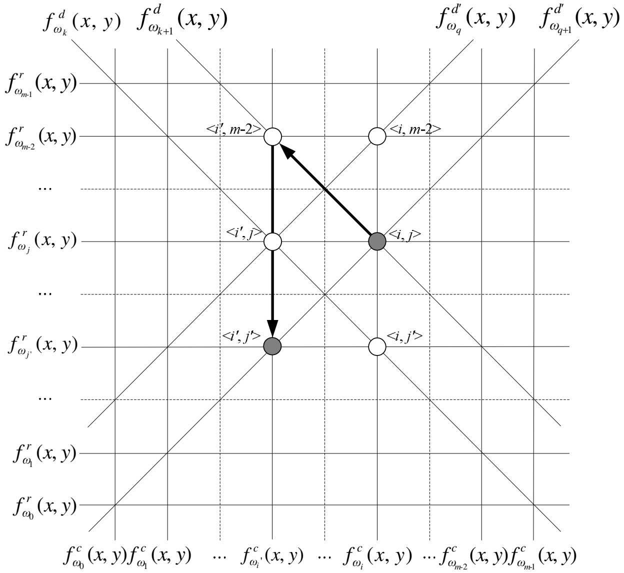
On the other hand, node *i* and node *j* don't share any polynomial with their τ polynomials stored in memory. Then node *i* and node *j* establish a pairwise key using path discovery.

3.3 Path Discovery

In the path discovery, node <*i*, *j*> and node <*i*', *j*'> need to use path discovery to establish a pairwise key if $c_i \neq c_j$, $r_i \neq r_j$, or $| c_i - c_j | \neq | r_i - r_j |$. However, either node <$c_i$', r_j> or node <c_i, r_j'> can share common polynomials to establish a pairwise key with both node <*i*, *j*> and node <*i*', *j*'>. Indeed, if there is no compromised node, it means that there is at least one node that can be used as an intermediate node.

For example, as shown in Fig. 3, node <*i*', *j*> and node <*i*, *j*'> share the common polynomials with the same column $f^c_{\omega_i}(x, y)$ and the same row $f^r_{\omega_{j'}}(x, y)$. Then node <*i*', *j*> and node <*i*, *j*'> can help node *i* and node *j* to establish a pairwise key. If both the intermediate nodes are compromised, node <*i*, *j*> will establish a pairwise key with node <*i*', *j*'> through node < *i*', *m*-2>.

Specifically, a sensor node <*i*, *j*>, the source node *S*, may use the following algorithm to establish a pairwise key with node <*i*', *j*'>, the destination node *D*, through the intermediate nodes. (Assume source node *S* knows the ID of the destination node *D*).



$f^d_{\omega_k}(x, y)$
$f^d_{\omega_{k+1}}(x, y)$
$f^{d'}_{\omega_q}(x, y)$
$f^{d'}_{\omega_{q+1}}(x, y)$
$f^r_{\omega_{m-1}}(x, y)$
$f^r_{\omega_{m-2}}(x, y)$
...
$f^r_{\omega_j}(x, y)$
...
$f^r_{\omega_{j'}}(x, y)$
...
$f^r_{\omega_1}(x, y)$
$f^r_{\omega_0}(x, y)$
<i', m-2>
<i, m-2>
<i', j>
<i, j>
<i', j'>
<i, j'>
$f^c_{\omega_0}(x, y)$ $f^c_{\omega_1}(x, y)$ ... $f^c_{\omega_{i'}}(x, y)$ ... $f^c_{\omega_i}(x, y)$ ... $f^c_{\omega_{m-2}}(x, y)$ $f^c_{\omega_{m-1}}(x, y)$

Fig. 3. Path discovery

4 Performance Evaluation

4.1 Compromising a Pair of Sensor Nodes

With the proposed scheme, it is difficult for adversaries to compromise and prevent both nodes from establishing a pairwise key without compromising the related sensor nodes. If the attacker wants to compromise nodes *u* and *v* to establish a pairwise key directly, at least 2(t+1) nodes will also have to be compromised. If an attacker succeeds in compromising the polynomials, the related nodes still could re-establish another pairwise key through path discovery.

Consider the case with nodes *u* and *v* establishing a pairwise key using path discovery. An adversary may compromise one of the nodes in the path used to establish the pairwise key. Consequently there are 4(m-1) paths between nodes *u* and *v* with two or four intermediate nodes. Besides the key paths with the compromised node, there are at least 4m-5 paths. To prevent pairwise key establishment, the attackers have to compromise at least one sensor in each path. In addition, each node has number of polynomials and if one of the polynomial is compromised then one of the remaining polynomial can be used for path key establishment. The attackers have to compromise one sensor involved in the path key establishment to compromise the pairwise key, and at least 4m-5 sensors to prevent *u* and *v* from establishing a pairwise key.

In summary, the proposed scheme is more efficient in preventing adversary attack and establishing a pairwise key between the nodes.

4.2 Attacks against the Network

In WSNs, if an adversary knows the subset assignment, the bivariate polynomial may be compromised, leading to an attack on the entire network. In [8], if a number of compromised polynomials is l, there are about ml nodes where at least one of the polynomial shares has been disclosed, where m is the number of row and the number of column in the grid. However, in the proposed scheme, each sensor randomly selects the polynomials from the group ω. In this way, there is a probability pc that ml nodes may be compromised. The adoption of randomness in Blom's scheme [2] enables a new scheme to be more resilient to node capture [6].

Otherwise, an adversary may disable the node to establish a pairwise key using path discovery by randomly compromising sensor nodes. It was assumed that an adversary randomly compromises N_c (N_c>t) sensor nodes. The probability of the polynomials being chosen for a sensor node is $\frac{\tau}{m\omega}$, and the probability of the polynomials being chosen k times among N_c is $p(k)=\frac{N_c!}{(N_c-k)!k!}\left(\frac{\tau}{m\omega}\right)^k\left(1-\frac{\tau}{m\omega}\right)^{N_c-k}$. Therefore, the probability of any polynomial being compromised is $p_c = 1-\sum_{t=0}^{k} p(k)$, where p_c is denoted as the probability of compromised links between non-compromised sensor nodes. Based on the analysis above, it can be determined how difficult it is to re-establish a pairwise key between non-compromised nodes while the sensor network is under attack. First, the probability of both nodes u and v not being able to directly establish a pairwise key is $1 - \frac{4}{m+1}$. Second, the probability of four nodes being able to help nodes u and v establish a pairwise key is p_c^4. In addition, d is the small number of intermediate nodes with which the nodes cannot communicate. Thus, the probability of the path discovery failing is p_c^{2d}. By combining the above three cases, the probability that nodes u and v are not able to establish a pairwise key is $(1 - \frac{2}{m+1}) p_c^{2d+4}$. Hence, the probability of the remaining nodes establishing a pairwise key is $1 - (1 - \frac{2}{m+1}) p_c^{2d+4} \approx 1 - p_c^{2d+4}$.

4.3 Simulation Results

Based on [5], the probability that both nodes share at least a polynomial is 1- Pr[the nodes do not share any polynomial]. To compute the probability that both nodes do not share any polynomial, it is noted that each polynomial of a sensor node is drawn out of a polynomial pool S without replacement. Thus, the number of possible chosen polynomial is $\frac{S!}{s'(S - s')!}$.

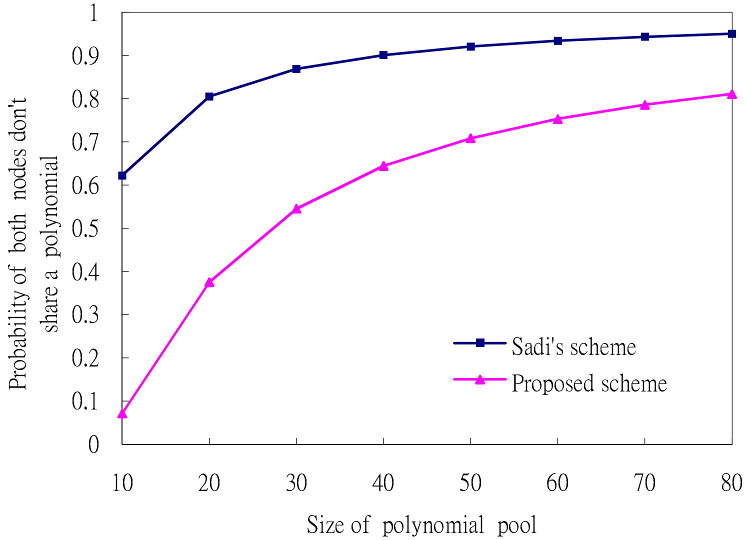


Fig. 4. Probability of two sensors not sharing a polynomial vs. size of polynomial pool

The first sensor node picks *s*' polynomials. The total number of possible polynomials that do not share a polynomial with another node is the number of polynomials that can be drawn out of the remaining *S* - *s*' unused polynomials in the polynomial pool, namely $\frac{(S - s')!}{s'(S - 2s')!}$.Therefore, the probability that no polynomial is shared between the two nodes is the ratio of the number of polynomials without a match by the total number of polynomials. Thus, the probability that there is at least a shared polynomial between two nodes is $1 - \frac{(S - s')!}{s'(S - 2s')!}$. In Fig. 5, the relationship between probability and the combinations of *s*' and *S* is shown. It is easy to see that the closer *s*' and *S* are, the more likely it is that two sensor nodes do not share a common polynomial.

5 Conclusions

In this paper, we propose a scheme to improve the pairwise key predistribution in WSNs. The proposed scheme consists of multiple polynomials for each row, each column, and each diagonal. Compared with Sadi's scheme, each sensor node at each row and column establishes a pairwise key using the predistributed polynomials with other nodes. Therefore, the proposed scheme has a higher probability of sensor nodes establishing a pairwise key using polynomial share discovery and path discovery. Performance evaluation shows that the proposed scheme outperformed Sadi's scheme.

Acknowledgments. This work was supported by the National Science Council of Republic of China under grants NSC-100-2221-E-239-023 and NSC-101-2221-E-239-032.

References

1. Agrawal, D.P., Zeng, Q.-A.: Introduction to Wireless and Mobile System. Brooks/Cole Publishing (August 2003)
2. Blom, R.: An Optimal Class of Symmetric Key Generation Systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)
3. Chan, H., Perrig, A., Song, D.: Random Key Predistribution Schemes for Sensor Networks. In: Proceedings of the 2003 IEEE Symposium on Research in Security and Privacy, pp. 197–213 (May 2003)
4. Du, W., Deng, J., Han, Y.S., Varsheny, P.K.: A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks. In: Proceedings of the 10th ACM Conference on Computer and Communication Security, pp. 42–51 (October 2003)

5. Eschenauer, L., Gligor, V.D.: A Key-Management Scheme for Distributed Sensor Networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41–47 (November 2002)
6. Jain, N., Agrawal, D.P.: Current Trends in Wireless Sensor Network Design. International Journal of Distributed Sensor Networks 1, 101–122 (2005)
7. Kwon, T., Lee, J., Song, J.: Location-Based PairwiseKey Predistribution for Wireless Sensor Networks. IEEE Transactions on Wireless Communications 8(11), 5436–5442 (2009)
8. Liu, D., Ning, P., Li, R.: Establishing Pairwise Keys in Distributed Sensor Networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 41–47 (October 2003)
9. Liu, D., Ning, P.: Location-Based Pairwise Key Establishment for Static Sensor Networks. In: Proceedings of the 2003 ACM Workshop on Security of Ad-Hoc and Sensor Networks, pp. 72–82 (November 2003)
10. Liu, D., Ning, P.: Improving Key Pre-Distribution with Deployment Knowledge in Static Sensor Networks. ACM Transactions on Sensor Networks 1, 204–239 (2005)
11. Rasheed, A., Mahapatra, R.: Key Predistribution Schemes for Establishing Pairwise Keys with a Mobile Sink in Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 22(1), 176–184 (2011)
12. Sadi, M.G., Kim, D.S., Park, J.S.: GBR: Grid Based Random Key Predistribution for Wireless Sensor Network. In: Proceedings of the 11th International Conference on Parallel and Distributed Systems, vol. 2, pp. 310–315 (July 2005)
13. Zhang, L.-P., Wang, Y.: An ID-Based Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In: Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing, pp. 1–4 (September 2010)

IARC – An Improved Coverage Based Sensor Network Topology Control Algorithm

Wei-Jun Lin, Tsen-Jui Lin, and Lei Wang

Department of Electrical Engineering, Feng Chia University, Taiwan
yume190@gmail.com, kyofire88@msn.com, leiwang@fcu.edu.tw

Abstract. As a wireless sensor network consists of a large number of randomly distributed nodes, one of the most challenging issues is to guarantee the required coverage and connectivity within the entire WSN's life cycle. This paper proposes a topology control algorithm which is improved from a previous study abbreviated as ARC for large-scale WSNs with randomly deployed nodes. According to the preliminary simulation results, the improvement has been proved that the proposed method can maintain the quality of service by coverage intensity much longer than original design.

Keywords: Wireless Sensor Network, Coverage Intensity, Routing Protocol.

1 Introduction

Researches in wireless communications have led to the development of Wireless Sensor Networks (WSNs), which are composed of low-power, low-cost, small-size and multifunctional sensor nodes. These nodes are densely deployed either inside or very closed to the phenomenon that is being monitored, every node has the ability to sense, process and transmit data to a base station (BS)[1]. Due to the limitations of size and cost, the nodes are equipped with small batteries with limited energy. Therefore, it is crucial to minimize the energy consumption to prolong network lifetime.

Based on the characteristics of WSNs, Al-Karaki and Kamal categorized routing protocols into flat routing protocol, location-based routing protocol and hierarchical routing protocol [2]. Flat routing protocol [3] is a data-centric routing protocol. Initially, the base station (BS) broadcasts query packets; once the query packets reach a sensor node, the sensor nodes return data to the BS if the data is available. The advantage of a flat routing protocol is that each sensor node does not need to store much route information; the disadvantage is that if the required data is returned by several sensor nodes simultaneously, it may cause network congestion or a broadcast storm. Consequently, it is not applicable to large-scale networks. In the location-based routing protocol [4], each sensor node is equipped with a Global Positioning System (GPS) to distinguish its own geographical position from others and figure out the best transmission path for itself. The location-based routing protocol reduces transmitting unnecessary packets (compared to the broadcast type) and is best for network topology that changes frequently. However, the cost is much higher than other protocols,

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 531–540.
DOI: 10.1007/978-3-642-35452-6_54

too. In the hierarchical routing protocol[5][6], the geographical region of the internet is divided into several clusters, where each cluster selects a cluster head (CH) responsible for collecting the data from cluster members and transferring data to the BS via hierarchical routing. Utilizing clusters in hierarchical routing protocol has its advantages because it allows less power consumption in each node and the CH is capable of processing data aggregation. However, it imposes a larger load on the CH, as a CH must manage not only data collection but also data relay.

As a wireless sensor network (WSN) consists of a large number of randomly distributed nodes, one of the most challenging issues is to detect events and send the corresponding data to a sink node successfully, to guarantee the required coverage and connectivity within the entire WSN's life cycle. The coverage rate reflects how well a sensor network is monitored or tracked by sensors. Besides the main viewpoint of the coverage issue that to increase coverage rate of a WSN to achieve most sensitive WSN, there is another viewpoint about coverage is proposed by recent studies: Can we limit the power consumption by guaranteeing an ideal coverage with less sensor nodes?

In this paper, we propose a topology control algorithm which is improved from another excellent research, Adaptive Random Clustering (ARC) [7], for large-scale WSNs with randomly deployed nodes. For the original ARC protocol: there are several novel features been exploited: First, instead of using location information, ARC forms a multi-hop cluster network with a required connectivity by using a novel cluster head competition scheme and proper transmit power settings. Second, required coverage is achieved by cluster heads and activated nodes, and thus less redundant nodes are activated than the existing algorithms which employ a coverage-first and connectivity-second activation procedure. Third, the lifetime of a WSN is prolonged through balancing energy consumption by updating cluster heads periodically, reducing redundancy of activated nodes by adaptively adjusting activation threshold, and reducing energy consumption by collision avoidance mechanism. Finally, ARC is suitable for practical applications of large-scale or high-density WSNs due to its distributed processing, scalable cluster topology, and easy management. It uses a very limited number of transmission channels to support a large number of clusters.

Although the features described above are achieved by ARC, there are some details about the technique are not well designed. In this paper, we proposed a new algorithm named Improved ARC (IARC) to achieve more efficient WSNs than ARC. In section 2, ARC is first introduced as the basis of this research. The idea and detail algorithm of IARC is then proposed in Section 3. By comparing the simulation results for both ARC and IARC described in Section 4, the improved performance shows the superiority of the proposed method. Finally, Section 5 is a conclusion of this research.

2 Adaptive Random Clustering Protocol

There are two metrics about the evaluation of coverage and connectivity should be defined before the introduction of ARC. The first is the coverage intensity C. If active nodes are independently and uniformly distributed in a deployment region, and each node can connect to the sink through a certain route, then network coverage intensity can be calculated from the number of active nodes n, the size of deployment region S and the sensing radius Rs as shown in following equation:

$$C = 1 - (1-q)^n, \qquad q = \frac{\pi R_S^2}{S} \tag{1}$$

In the equation, q is the probability that a point in the deployment region is covered by a single active node, and $(1 - q)^n$ is the probability that the point is not covered by any active nodes. The other is probability of connectedness that gives the probability for every active node in the network has at least one route to the sink. For a given network deployment, probability of connectedness is defined as the probability that the communication graph generated by the network is connected. In a communication graph, a sensor node is regarded as a vertex, and a link between a node and one of its one-hop neighboring nodes is regarded as an edge. A communication graph is connected if and only if there is a path consisting of one or multiple edges between each two vertexes.

ARC periodically resets the network in order to balance energy consumption of nodes and to achieve network robustness. Each round of operation consists of a setup phase and a steady phase. The setup phase is composed of three steps: CHN(Cluster Head Node) selection, route setup, and NCHN(Non-Cluster Head Node) activation. In the steady phase, every active NCHN sends data packets to its cluster head using specified band and time slot in a TDMA manner, and sleeps in other time slots. Every CHN sends data packets to its upstream CHN according to its route table established in the stage of route setup. CHNs operate in a CSMA/CA protocol for inter-cluster communication.

Similar to many other cluster based topology control algorithms, it is assumed in ARC that there are several channels (bands) available, and different clusters use different channels to prevent collisions of data packets. Besides, there is a primary band assigned for transmitting all kinds of data packets in setup phase and inter-cluster data packets in steady phase. Local synchronization within a cluster is assumed to apply the TDMA scheme. The transmit power for packet transmissions between a CHN and an NCHN is denoted as P_1, corresponding to the intra-cluster communication radius R_{C1} [8], and the transmit power for packet transmissions from a CHN to another CHN or to the sink is denoted as P_2, corresponding to the inter-cluster communication radius R_{C2}. P_1 is less than P_2 in order to reduce energy consumption and to inhibit mutual interference of data packets among different clusters.

The algorithm of setup phase is performed as the steps described below.

2.1 CHN Selection

CHNs should be the nodes with relatively more residual energy. They should not be distributed too closely in order to prevent data collisions caused by redundant CHNs. In the stage of CHN selection, all deployed nodes compete for being CHNs in CSMA/CA protocol. At first, a node has to wait a period of backoff time. If no broadcast packets are received from the other CHNs, it will declare itself as a CHN by broadcasting a CHN packet with transmit power P_1 to its one-hop neighboring nodes. The backoff time of Node *i* can be expressed as:

$$T_{ci} = T_1 \times \left[\frac{r_i}{R+1} \right] + T_2 \times Rand[0,1] \tag{2}$$

In the equation, the backoff time is mainly decided by number of rounds. R is the total number of rounds executed, and r_i is the number of rounds that Node i has already acted as a CHN in the past R rounds. A larger r_i makes a longer backoff time and then get a lower probability that Node i becomes a CHN in the current round. The second term in the equation is a random value introduced as an enhancement to the existing CSMA/CA algorithm. Therefore, the nodes have different random values and then different backoff times. This keeps the nodes close to each other from sending CHN packets at the same time, and avoids the collisions. T_1 and T_2 are weights of these two terms, respectively. T_2 should be set large enough for avoiding collisions.

If a node has received CHN packets from other CHNs before its timer expires, it will stop its timer and become an NCHN in the current round. At the end of CHN selection, a NCHN compares the signal strengths of all received CHN packets and then request to join the cluster of the CHN with the strongest signal strength. At last, CHN will choice its member randomly according to the number of NCHNs calculated in advance to achieve the required coverage. In case of no global synchronization, CHN selection can also be carried out.

2.2 Route Setup

The task of this step is to establish multi-hop route from every CHN to the sink so as to ensure network connectivity. Traditional cluster based protocols usually assume that CHNs could communicate with the sink directly. However, single-hop mode is not suitable for large-scale sensor networks due to packet collisions and high energy consumption. In ARC, multi-hop routes among CHNs are established based on the minimum hop count as figured in Fig. 1, and the band of each cluster for intra-cluster communication in steady phase is determined so that the bands of any neighboring clusters are different from each other. Here, neighboring clusters are defined as the clusters whose CHNs can communicate directly with transmit power P_2.

The process of route setup is described as follows: The sink broadcasts a HOP packet with transmit power P_2. A CHN waits a period of time T_b after it received the HOP packet [9], and rebroadcasts the packet with P_2. The HOP packet from a CHN consists of the serial number of the CHN, the minimum hop count to the sink, the total number of nodes in the cluster, and the band for intra-cluster communication in steady phase. The band chosen by a CHN is ensured to be different from the bands of its neighboring clusters, because neighboring CHNs broadcast their HOP packets in a CSMA/CA protocol and later a CHN is forbidden to choose the same bands as those chosen by the former CHNs. A CHN would drop the HOP packet pending in the sending queue and reselect its band if it receives another HOP packet at the backoff time of CSMA/CA, to deal with the situation that the selected band happens to be the same as that in the received HOP packet. The neighboring CHNs with the less minimum hop count are the candidates of the upstream CHN. The candidate from which the HOP packet arrives first is chosen as the upstream. Finally, every CHN will be able to determine its minimum hop count to the sink, its upstream CHN and its unique cluster band among its neighboring CHNs.

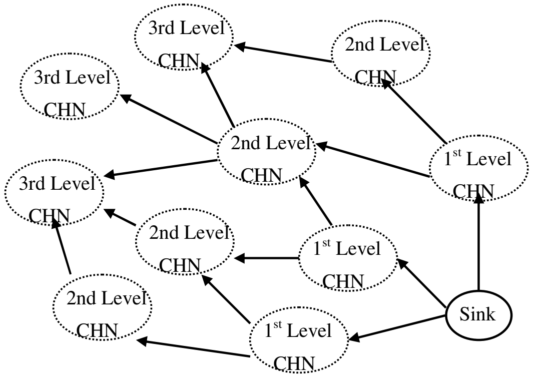


Fig. 1. Multi-hop routing

2.3 NCHN Activation

The number of nodes deployed in a deployment region is usually redundant in order to take duty in turn and prolong the network lifetime. For each cluster, only a part of NCHNs are activated for each round. For the sake of energy saving, the number of active nodes should be minimized while still satisfying the required coverage intensity. The CHN of the m-th cluster determines the activation threshold K_m according to the required coverage intensity C_0 and the information in the HOP packets from the neighboring clusters. Then, it broadcasts K_m to its NCHNs, where $K_m \in [0, 1]$.

Each NCHN generates a random number between 0 and 1, and becomes active in the steady phase of the current round if the random number falls in $[0, K_m]$. Therefore, K_m is the activation probability of NCHNs. Every activated NCHN sends a packet to its CHN and the CHN replies with a packet including the index of the time slot for the NCHN to send data in the steady phase. In such a way, the ratio of activated NCHNs in the m-th cluster is around K_m, and each active NCHN is assigned a dedicated time slot.

3 Improved ARC

Based on the method of ARC, some modifications are introduced in this section.

3.1 Modification of CHN Selection

For ARC, the nodes with more residual energy are selected as CHNs by carefully choosing the backoff time of all nodes as illustrated in section 2.1. The main factor in the step that decides the backoff time of a node is the number of rounds that the node has executed as a CHN. A larger number makes a longer backoff time and then get a lower probability to be a CHN. However, there are two conditions should be considered: First, the power of an executed node will be consumed no matter whether the node is a CHN or NCHN. Although a CHN will spend more energy, a node that stand as a NCHN many times should exhaust its energy, too. Second, two nodes may consume different power in a same round even they both act as CHN. Different CHNs may consume different energy in a same round since the different amounts of up-stream packets are passed through the nodes. Based on the considerations, the imprecise estimation of energy consumption of ARC will make a wrong choice for the determination of CHN, so IARC modifies the equation to include the considerations as list in equation (3).

$$\mathrm{T}_{\mathrm{ci}} = \mathrm{T}_1 \times \frac{\mathrm{R}_{execute\_i}}{\mathrm{R}_{current}} + \mathrm{T}_2 \times \frac{\sum_{k=1}^{Rcurrent} \left(\frac{\mathrm{N}_{\mathrm{pass\_throuth}(i,\,\mathrm{k}\,-\,1)}}{\mathrm{N}_{\mathrm{pass\_throuth}(i,\,\mathrm{k}\,-\,1)} + \mathrm{N}_{\mathrm{cluster}(\mathrm{k}\,-\,1)}} \right)}{R_{execute\_i} + 1} + \mathrm{T}_3 \times \mathrm{Rand}[0,1] \qquad (3)$$

The determination of the backoff time for node i will be more precisely by Equation (3). The symbols appeared in the equation are defined below:

$\mathrm{R}_{\mathrm{execute\_i}}$: The number of rounds that the node i had ever activated, no matter act as a CN or a NCHN.

$\mathrm{R}_{\mathrm{Current}}$: The total number of rounds executed.

$\mathrm{N}_{\mathrm{pass\_through(i,\,k)}}$: When node i acted as a CHN in the k_{th} round, the number means the total number of CHNs that transmit packets through the node i in the round; the number is zero otherwise.

$\mathrm{N}_{\mathrm{Cluster(k)}}$: The number of NCHNs required for a cluster in k_{th} round.

There are three terms to make up the value of backoff time. The first term is weighted by the times that the node has activated in the past rounds. The second term is weighted by the average amount of data that transmitted through the node. The third term is a random value introduced as an enhancement to the existing CSMA/CA algorithm. Based on the equation, the differences among nodes will be more precisely determined by the status of actual energy consumptions.

3.2 Modification of NCHN Activation

As described in section 2.3, if a node has received CHN packets from other CHNs before its timer expire, it will become a candidate of NCHN in the current round and try to join the cluster with strongest signal strength at the end of CHN selection step. At last, CHN selects its member randomly according to the number of NCHNs calculated in advance to achieve the required coverage. The selection of NCHN in ARC is made by means of a simple random function, thus lead the selection to choice some nodes with poor energy and abandon other stronger nodes.

The selection of NCHNs of a cluster is modified in the algorithm of IARC, too. For IARC, if a node decides to be a CHN, The node will calculate the amount of NCHNs needed to achieve the required coverage at first as defined in equation (4). When other nodes expired their backoff time, they will send JOIN message to a CHN with earliest claim to be their CHN to request for joining the cluster. Since every node is assigned different backoff time by their energy remained, so the CHN can receive the JOIN messages with the order of energy remained in nodes. The decision of whether a node can be a CHN's member is then very simple and efficient: When a CHN does not collect enough NCHN for the required coverage, the CHN will reply an ACCEPT message immediately when it receives a new JOIN message.

$$N_{cluster} = \frac{\log(1-C_0)}{\log(1-\frac{\pi R_s^2}{\pi R_c^2})} \tag{4}$$

When the required number of NCHNs is achieved, the CHN will stop the collection process and broadcast COMPLETION message to notify other nodes that still stuck in backoff time to abandon the CHN. On the other hand, a node receive a COMPLETION message does not means that the node can not be a NCHN in the current round. It can still request to join other cluster when it expires its backoff time. This method will guarantee that the members of a cluster are all recruited with the NCHNs with more energy that the range can provide.

4 Preliminary Simulation Results

For evaluating the performance achieved by IARC, a simulation based on the report of ARC is held to verify the modifications. For evaluating the improvement over the ARC, both the methods of ARC and IARC are simulated for comparison. All related simulation are designed by JAVA language with Eclipse development environment.

The related simulation parameters are list below:

There are N nodes uniformly and independently distributed in a deployment range. The sink node is located at the location of a corner with coordinate (0,0). The sensing range is set as a circle with radius of Rs. The communication ranges for intra-cluster and inter-cluster are circles with radius of Rc1 and Rc2, respectively. It is noted that the value of Rc2 is three times bigger than Rc1 in the simulation.

For the calculation of backoff time in simulation, the parameters of ARC are set as: the weighted ratio of T1:T2 is 7:3 that is the same as the original design of ARC. For IARC, the three weighting value, T1, T2, and T3 are 1:6:3 since according the observation of simulation tests, the ratio can offer good energy balance in most of simulations.

Energy consumption of a sensor node can be divided into two parts: energy consumed by the sensing module, and energy consumed by the transceiver. The former depends on the type of sensing module, while the latter consists of the energy costs for necessary data transmission, overhearing, and retransmissions due to collisions, respectively. Traffic load in our simulations is set very light such that there is almost no collision. As a result, the energy consumed by the transceiver is mainly for necessary data transmission and for overhearing. We adopt the energy model in [10] and assume that the energy cost of overhearing a packet equals to that of receiving a packet.

The simulation is the same as the ARC simulated in [7]. For a coverage emulated as function (1), and the nodes that can not connect to the sink are omitted for coverage. By setting the parameters as list below:

Deployment range : 100*100 m^2
Rs: 5m Rc1:20m, Rc2: 60m
Power initiated: 5J
Number of rounds: 2000
Working time per rounds: 1000 seconds
Transmitting rate: 5 seconds/packet
Packet size: 32 Bits

Fig. 2. is the average coverage intensity achieved in the first 2000 rounds. Readers can observe from the figure that the intensity of ARC begin to drop down around 1500 rounds. For the last round, the coverage intensity is dropped below 0.7. On the contrary, IARC can prolong the required sensing quality until the last 100 rounds. The dropped intensity achieved is about 0.75 in the last round that is still higher than ARC.

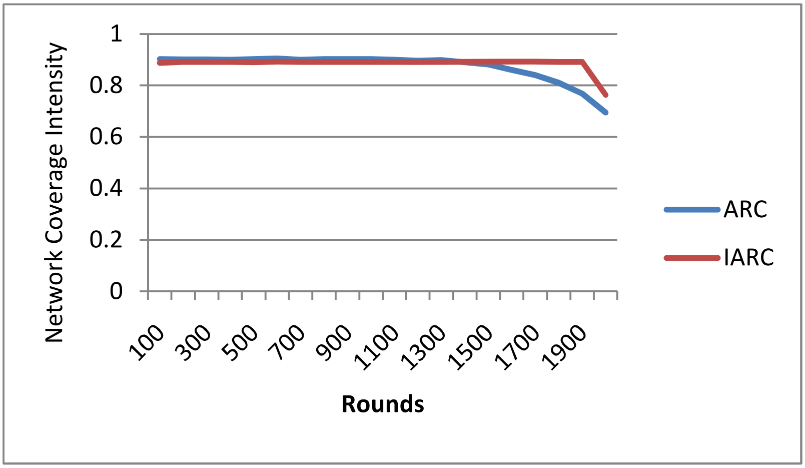


Fig. 2. Average coverage intensity for ARC and IARC

The average number of living nodes for the testing rounds are shown in Fig. 3. The curves shows that the number begin to drop down from around 1000 rounds. Because of the higher coverage intensity achieved by IARC, the energy been consumed is more than ARC. The truth leads the remaining living nodes of IARC to be less than ARC. It is interesting that by comparing the coverage intensity shown in Fig. 2 and the number of living nodes in Fig. 3, we can find that IARC can always achieve a higher intensity by means of fewer living nodes than ARC.

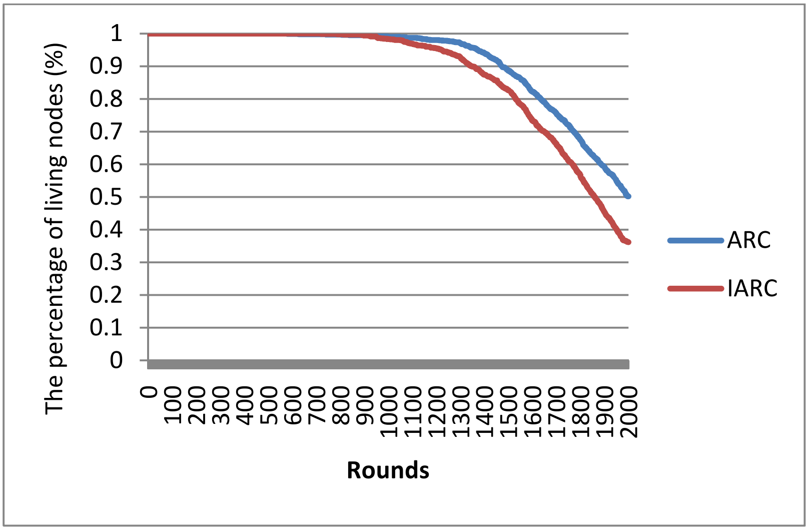


Fig. 3. The number of living nodes

5 Conclusions

This article proposes an improved topology control algorithm for large-scale WSNs with randomly deployed nodes. The study is done by carefully examine the detail of ARC method and then modify the rules and algorithm in setup phase to construct more reasonable cluster organization. According to the preliminary simulation results, the improvement is proved that IARC can maintain the quality of service by coverage intensity much longer than original ARC.

There are many issues worth to be investigated. For example, an interesting condition is found from the simulation that IARC WSNs always lose coverage dramatically since the death of the nodes that nearby the sink node. It is reasonable because these nodes must transfer the messages comes from outer range when they act as a CHN. On the contrary, when the WSN is dead, there are still about 30% of the nodes are usable. The condition prompts us if there are more sensing nodes be located in the innermost range, the life time of the scheme will be extended efficiently. As another example, the messages passed through CHNs to sink are transmitted by a data-centric routing paradigm, an energy-efficient routing protocol worthy to be proposed for multi-source transmission scenarios. An efficient routing protocol can significantly reduce network traffics, and thus promote energy efficiency.

References

1. Lewis, F.L.: Wireless Sensor Networks. In: Cook, D.J., Das, S.K. (eds.) To appear in Smart Environments Technologies, Protocols, and Applications. John Wiley, New York (2004)

2. Al-Karaki, J.N., Kamal, A.E.: Routing Techniques in Wireless Sensor Networks: A Survey. IEEE Wireless Communications, 6–28 (2004)
3. Kong Ling, P., Yang, Q.: The Comparison Study of Flatrouting and Hierarchical Routing in Ad Hoc Wireless Networks. In: 14th IEEE International Conference on Networks, ICON 2006, Singapore, pp. 1–6 (2006)
4. Zheng, K., Tong, L., Lu, W.: Location-Based Routing Algorithms for Wireless Sensor Network. ZTE Communications (1) (2009)
5. Joshi, A., Lakshmi Priya, M.: A Survey of Hierarchical Routing Protocols in Wireless Sensor Network. In: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), Chengdu, V3-650–V3-654 (2010)
6. Iwanicki, K., van Steen, M.: On Hierarchical Routing in Wireless Sensor Networks. In: IPSN 2009. International Conference on Information Processing in Sensor Networks, San Francisco, CA, pp. 133–144 (2009)
7. Xu, N., Huang, A., Hou, T.-W., Chen, H.-H.: Coverage and connectivity guaranteed topology control algorithm for cluster-based wireless sensor networks. Wiley Online Library (wileyonlinelibrary.com) (2010) (Published online January 8, 2010), doi: 10.1002/wcm.887
8. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless micro sensor networks. IEEE Transactions on Wireless Communications, 660–670 (2002)
9. Liu, C., Wu, K., Xiao, Y., Sun, B.: Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 562–575 (2006)
10. Kan, B.-Q., Cai, L., Zhu, H.-S., Xu, Y.-J.: Accurate Energy Model for WSN Node and Its Optimal Design. Journal of Systems Engineering and Electronics, 427–433 (2008)

Coverage Improvement for Directional Sensor Networks

Tien-Wen Sung[1,2] and Chu-Sing Yang[1]

[1] Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Taiwan

[2] Department of Network Multimedia Design, Hsing Kuo University, Taiwan

tienwen.sung@gmail.com

Abstract. This study used the characteristics of Voronoi diagram and direction-adjustable directional sensors to propose a distributed greedy algorithm, which can improve the effective field coverage of directional sensor networks. Considering the coverage contribution of convex polygonal cell of sensors and the coverage overlap of direction select between neighbor sensors, the working direction is adjusted and controlled, so as to improve the overall sensing field coverage ratio in the sensor network environment without global information. Finally, simulations were used to evaluate the efficiency of the proposed algorithm.

Keywords: Directional sensor networks, Voronoi diagram, Coverage.

1 Introduction

In recent years, the focus of development and research on WSN turned to the Directional Sensor Network (DSN). Differing from general WSN sensors, which have omni-directional (360°) effective sensing range, the effective sensing range of DSN sensors is limited to some direction and specific angular dimension, such as camera sensors (Kulkarni et al., 2005), infrared sensors, ultrasonic sensors, lidar sensors and radar sensors, so there are different research problems and considered characteristics in the sensing coverage subject of DSN (Charfi et al., 2009; Soro and Heinzelman, 2009). Although there are studies concerning DSN Coverage (Guvensan and Yavuz, 2011a), the problems and solutions are different. This study focused on the improvement of field coverage of DSN composed of rotatable directional sensors, used the Voronoi diagram characteristics to reduce the decision-making complexity of directional sensors in determining the sensing direction. Greedy algorithm was used to determine and control the sensing direction of sensor in the distributed computing environment without Global Information.

2 Related Works

The main purposes of studies related to DSN Coverage can be divided into two general classes (Pham et al., 2011): Known-Targets Coverage and Region Coverage.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 541–550.
DOI: 10.1007/978-3-642-35452-6_55

The Known-Targets Coverage determines the sensors subset to attain specific targets or position sensing coverage. Chen et al. (2008) proposed the Target Weight and Orientation Weight from the number of targets to be monitored and how many sensors cover each target. They then determined the direction of sensors and increased the targets coverage rate. Chow et al. (2009) assumed that one directional sensor only covers a part of a target, and proposed an algorithm using multiple sensors to provide 360° complete monitoring of a target. Wang et al. (2009) aimed at the targets with different priorities, and used genetic algorithm to obtain the minimum subset of directional sensors, which could cover all of targets. Hsu et al., 2012 proposed algorithms to solve Maximum Coverage with Rotatable Sensors (MCRS) problem in which the number of targets to be covered is maximized whereas the rotated angles of sensors are minimized. On the other hand, the Region Coverage makes the sensors attain the overall sensing coverage of DSN task region. Cheng et al. (2007) selected the minimum overlap direction of each directional sensor as the working direction, namely, selected the direction covering more additional area to increase the overall field coverage. In addition, Tezcan and Wang (2008) considered the DSN environment with obstacles, and proposed a rotatable sensor self-orienting algorithm which could reduce the effect of obstacles on the coverage. In order to approach to the real application simulation environment, Topcuoglu et al. (2012) built a 3D simulation environment to generate 3D landscape and analyzed their effect on the coverage. Our study proposed a different distributed greedy algorithm to determine the working direction of directional sensors based on Voronoi cell structure.

3 Preliminaries

3.1 Sensing Model

Fig. 1 shows the sensing model of the rotatable directional sensor used in the algorithm proposed in this paper. Each sensor has only one working direction. The effective sensing field F_s (Field of View) of sensor is in sector shape, i.e. the pie-shaped zone formed of effective viewing angle (Angle of View) with two sides in length of sensing radius r and corner dimension α.

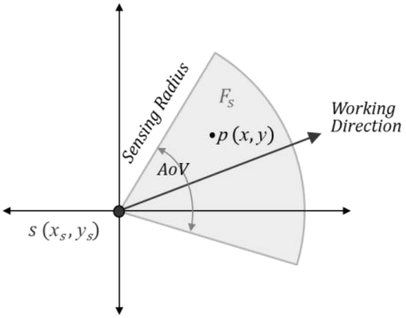


Fig. 1. Sensing model for rotatable directional sensors

3.2 Assumptions

The main assumptions of this paper are described below:

1. Each sensor can obtain its position by localization related technologies.
2. Sensor network is connected; therefore, the sensor has enough communication range or multihop transmission capacity to transmit related information to neighbor sensors.
3. Sensors are rotatable; they can rotate 360° to change working direction.

3.3 Problem Description

Based on the Voronoi cells structure, each sensor uses the vertices of its cell as the candidate targets of working direction. The working direction is selected and adjusted and the original direction of each sensor is changed on the principles: (1) to enlarge the area covered by the sensor inside the cell as much as possible; (2) to reduce the sensors overlapped coverage with neighboring cells as much as possible; so that the overall DSN field coverage is improved. Fig. 2 shows the schematic scenario brief examples of results before and after implementation of algorithm.

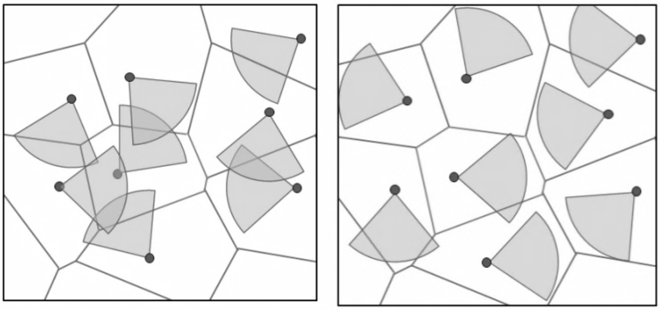


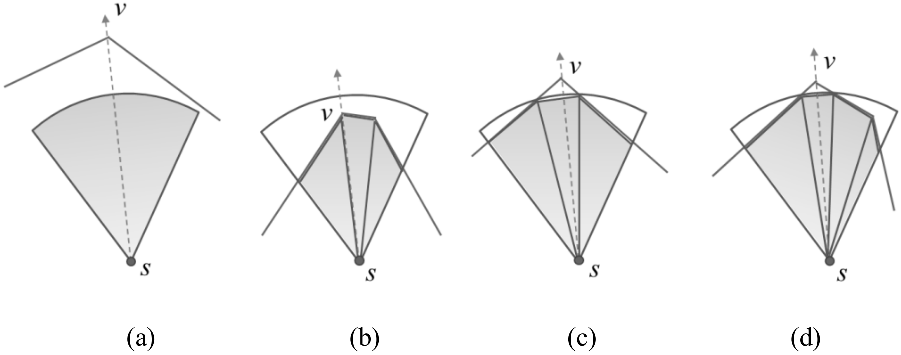
Fig. 2. Schematic brief examples of results before and after implementation of algorithm

4 Proposed Distributed Sensor Direction Control Algorithm

4.1 Intra-cell Working Direction Selection (IDS)

When each sensor completes the construction of its Voronoi cell using distributed/localized algorithm, the vertices of the cell are used as preliminary target working direction candidate. The selection strategy is that the vertex having the maximum coverage as preliminary working direction according to the obtainable intra-cell coverage size of sensor aligning with each vertex. In other words, the midsplit line of the sectorial effective sensing field of sensor points to the vertex in the position of sensor.

When the sensor aligns with each vertex to calculate the intra-cell coverage area, Fig. 3 shows several different cases. Except for Fig. 3 (a), which reduces the computation complexity of overlap area between the sectorial sensing field of sensor and Voronoi cell in irregular convex polygon, the intra-cell covered region is divided into multiple triangles according to the intersection between sector boundary and cell edges. The area can be estimated rapidly, and there is no significant error between the total area of all triangles and the sector computing mode.



(a) (b) (c) (d)

Fig. 3. Intra-cell coverage area estimation in different cases

As shown in Fig. 3 (b) (c) (d), as long as the vertices of various triangular regions are obtained, the side lengths of various triangles can be determined quickly. The area can thus be obtained. Besides the coordinates (x_s, y_s) of sensor, the vertices of triangles may be the Voronoi vertices in the sensing sector, and the intersections of sensing sector boundary and cell edges. The calculation of intersections of sensing sector boundary and cell edges contains sector side and arc edge.

If the three side lengths e_1, e_2 and e_3 of a triangle are given, the area T is calculated as follows:

$$d = \frac{e_1 + e_2 + e_1}{2} \tag{1}$$

$$T = \sqrt{d(d - e_1)(d - e_2)(d - e_3)} \tag{2}$$

If m is the number of vertices of Voronoi cell of the sensor s, and the vertex set is $V_s = \{ v_1, v_2, \ldots, v_m \}$ and e_{ij}^v is the j-th side length of the i-th triangle when s uses v as the assumed working direction to calculate intra-cell coverage area, k_v is the number of triangles of coverage area for corresponding vertex v, then the target vertex v_D selected by sensor s in the intra-cell working direction selection stage is shown below:

$\forall v \in V_s$

$$d_i^v = \frac{1}{2}\sum_{j=1}^{3} e_{ij}^v \ , \quad 1 \le i \le k_v \tag{3}$$

$$v_{\mathrm{D}} = \operatorname{argmax}_{v \in V_s} \sum_{i=1}^{k_v} \sqrt{d_i^v \cdot \prod_{j=1}^{3} \left(d_i^v - e_{ij}^v\right)} \tag{4}$$

Fig. 4 shows a sensor is in its Voronoi cell, and the intra-cell working direction selection target direction is determined from the six vertices from v_1 to v_6 according to the above method, the selected v_{D} is v_1, meaning that the intra-cell coverage is formed between sensing sector and various vertices, the maximum one is vertex v_1.

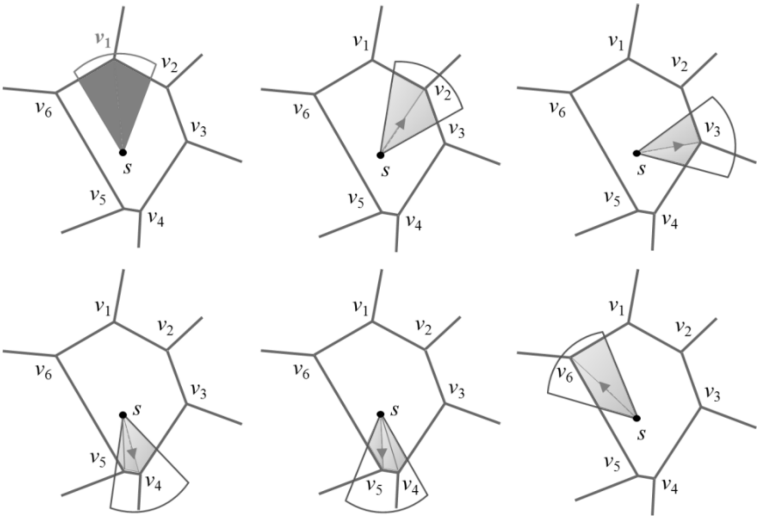


Fig. 4. Example of intra-cell working direction selection

4.2 Inter-cell Working Direction Adjustment (IDA)

When all the deployed sensors in the whole sensing field have finished preliminary working direction selection by using the aforesaid intra-cell working direction selection method, the overlapped coverage between adjacent cells should be considered. If more than two sensors of different Voronoi cells select the same Voronoi vertex as the working direction at preliminary stage, as these sensors face towards the same place (common vertex of the cells), the probability of coverage overlap is increased relatively, so that the coverage contribution of sensor is wasted in the application free from *k*-coverage. Fig. 5(a) shows the associated sensors s_i and s_j of two adjacent Voronoi cells, s_i and s_j select v_1 as working direction at initial stage, there is a large coverage overlap as a result. When this situation occurs, one sensor can be turned to avoid the common vertex. The further adjustment in direction for identifying any overlap and the implementation of adjustment in direction should be carried out at the inter-cell working direction adjustment stage.

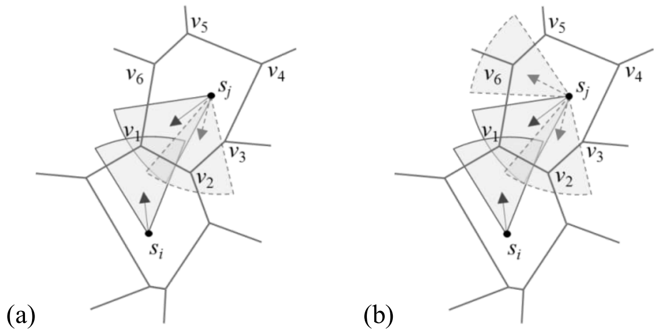


Fig. 5. Adjustment in direction

As for determining the sensor to be adjusted for direction and the change of direction, the direction of the sensor with a smaller intra-cell coverage can be adjusted. The direction is changed to the cell vertex with the second largest intra-cell coverage, the working direction faces towards different vertices, so that the coverage overlap probability is reduced. When the following conditions are tenable, sensor *s* should have adjustment in direction to change the direction selected at initial stage.

$\exists\ s_i \in N_s$ such that

$$v_{\mathrm{D}}^{s} = v_{\mathrm{D}}^{s_i}\ ,\ R_s^{v_{\mathrm{D}}} \cap R_{s_i}^{v_{\mathrm{D}}} \neq \emptyset\ ,\text{ and }\ A_s^{v_{\mathrm{D}}} < A_{s_i}^{v_{\mathrm{D}}} \tag{5}$$

where N_s is Voronoi cell neighbors set of s, v_{D}^{s} is the vertex selected as working direction of sensor *s* at initial stage, R_s^v represents the sector coverage area when *s* takes *v* as direction, A_s^v is the intra-cell coverage area corresponding to *v*, and the new working direction v_{D}' of *s* is

$$v_{\mathrm{D}}' = \mathrm{argmax}_{v \in U_s} A_s^v \tag{6}$$

$$\text{where }\ U_s = V_s - \{v_{\mathrm{D}}^{s}\} \tag{7}$$

Although different sensors selecting common vertex as common working direction has been avoided, they may have overlapped coverage (though it may be small) (see Fig. 5(a)), so this study further adjusts the direction control computation rule. The vertex with small overlapped region and maximum intra-cell coverage area is selected from all the cell vertices excluding common vertex as the new working direction, so as to obtain the optimum coverage contribution in the case of probable coverage overlap.

$$U_s' = \left\{ v \in U_s \,\middle|\, \left| R_s^v \cap \left(\bigcup_{s_i \in S_{v_{\mathrm{D}}}} R_{s_i}^{v_{\mathrm{D}}} \right) \right| < \left| R_s^{v_{\mathrm{D}}} \cap \left(\bigcup_{s_i \in S_{v_{\mathrm{D}}}} R_{s_i}^{v_{\mathrm{D}}} \right) \right| \right\} \tag{8}$$

$$v_{\mathrm{D}}' = \mathrm{argmax}_{v \in U_s'} A_s^v - \left| \left(\bigcup_{s_i \in S_{v_{\mathrm{D}}}} R_{s_i}^{v_{\mathrm{D}}} \right) \cap C_s^v \right| \tag{9}$$

where $S_{v_{\mathrm{D}}}$ is the set of all sensors taking v_{D} as working direction excluding s, C_s^v is the intra-cell region when s takes v as the direction. Take Fig. 5(b) as an example, s_j turns to other vertices with smaller overlapped region than v_1, and the vertex with the maximum intra-cell coverage excluding overlapped region is selected from these vertices, the result is v_6, better than v_2.

Under one condition, if all the vertices $V_s = \{ v_1, v_2, \ldots, v_m \}$ of Voronoi cell of the sensor s become the selected direction of different neighbor cells respectively after competition of direction selection, meaning that regardless of the direction selected by s, it is very likely to have coverage overlap with other neighbor sensors, so s can enter sleep mode to reduce energy consumption, and handled by scheduling process. However, this paper aims to maximize the field coverage, so the energy consumption, scheduling and DSN life time have not yet been considered. In this case, the algorithm selects a vertex with the presently optimal coverage contribution as the working direction of sensor s from all local cell vertices that have been selected as working direction by other neighbor cells.

4.3 Local Information Exchange and Direction Control Procedures

The algorithm proposed in this paper is distributed greedy algorithm. Each sensor has no global information, thus, only a little local information exchange between sensors is needed for selecting working direction. When each sensor has completed initial random deployment, it should inform neighboring coordinates; in the same way, the sensor will receive the positions of other neighbors, so that its Voronoi cell can be constructed. The cell vertices set V_s and Voronoi cell neighbors set N_s can thus be obtained. Afterwards, the intra-cell working direction selection is implemented to determine the initial working direction, and then the sensing model parameters should be given to neighbors N_s, so that the neighbors carry out inter-cell working direction adjustment. When the sensor has completed the procedure of inter-cell working direction adjustment, the direction may or may not be changed. If the working direction should be changed, the new direction must be given to the cell neighbors, and the neighbors will decide whether or not to change the direction again according to the notification. The step repeats, until the direction stops changing. In the worst case, the sensor s changes direction $|V_s|$ times at most.

5 Performance Evaluation

This study uses simulation for performance evaluation of sensing field coverage. The directional sensors are randomly deployed in the target sensing field. The related simulation environment parameters include sensing field size, number of sensors, angle of view and sensing radius. The coverage results of (1) randomly deployed direction (2) the proposed IDS (3) the proposed IDA are compared. The parameter of sensing field size is fixed at 500m500m, the number of sensors is 70~600, the sensing radius is 30m~120m (spacing 10m), and the sensing angle is 60°~360°. Each simulated condition is implemented 100 times, and the average is taken as the result data.

5.1 Coverage with Various Numbers of Directional Sensors

Fig. 6 shows the coverage results of simulation by changing the set value of number of sensor, while fixing the angle of view (α =120°) and sensing radius (r =50m) parameter values. Fig. 6 compares the coverage ratios of simulated IDA, IDS and random sensor direction deployment. For different numbers of sensors, the IDA effect is slightly better IDS, and IDS has better result than random mode. It should be additionally explained that as the sensors position is fixed. Regardless of the method used for determining the working direction, the overall field coverage improvement is still limited, and the maximum value will not increase.

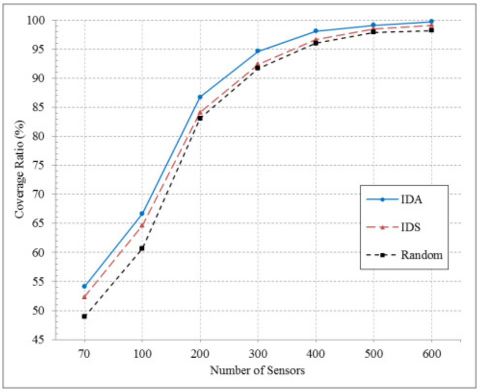


Fig. 6. Coverage ratio with various numbers of sensors

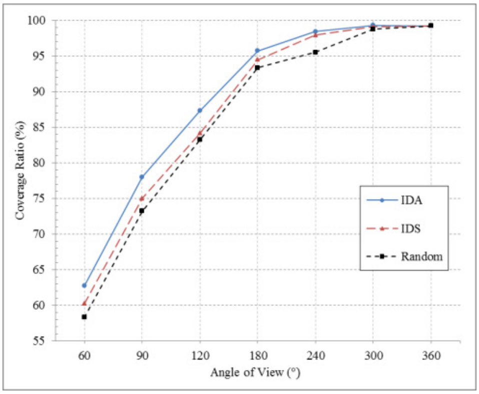


Fig. 7. Coverage ratio with various angles of view

5.2 Coverage with Various Angles of View

Fig. 7 shows the coverage results of simulation by changing the set value of angle of view while fixing the number of sensors (n =200) and sensing radius (r =50m) parameter values. According to Fig. 7 coverage ratio, the IDA is approximately better than other methods. When the α value is smaller, the obtained coverage improvement is better; on the contrary, if the α value is larger, as the sensing sector area is larger, the coverage overlap is increased, so the coverage improvement is limited. In addition, it is observed when the angle of view α =360°, the directional sensors are identical with omnidirectional sensors theoretically, so regardless of the working direction, the coverage may not be improved or increased. This is proven by the coverage simulation results in Fig. 7.

5.3 Coverage with Various Sensing Radiuses

When the angle of view (α =120°) and number of sensors (α =120°) parameter values are fixed and the set value of sensing radius is changed, the coverage result of simulation is shown in Fig. 8. According to the coverage ratio result, the IDA still has slightly better curvilinear trend approximately. When the sensing radius increases gradually, although there is still coverage improvement, the amplitude is small. It is because as the sensing radius increases, the degree of sensing sector crossing local Voronoi cell increases, meanwhile the degree of coverage overlap is increased.

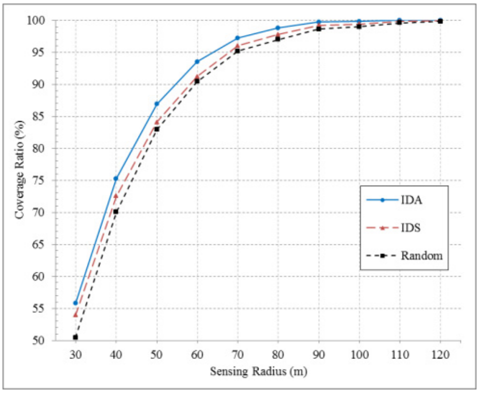


Fig. 8. Coverage ratio with various sensing radiuses

6 Conclusion and Future Works

The directional sensor networks have different characteristics and limiting conditions on the subject of sensing coverage compared with omnidirectional sensor networks, and the coverage is the elementary performance index and important issue of sensor network. This study combined the special geometrical features of Voronoi diagram with the advantages of distributed algorithm of without global information, low network bandwidth consumption and real-time response to dynamic environment change. It proposed the distributed/localized greedy algorithms of intra-cell working direction selection (IDS) and inter-cell working direction adjustment (IDA), aiming to improve the overall field coverage of directional sensor networks effectively. This study compared the simulation results of the proposed algorithms and random method, and analyzed the coverage performance improvement by changing and setting different sensing parameter values, so as to prove that the IDA method proposed in this paper has better performance.

Our future work will focus on prolonging network lifetime and work scheduling algorithm with fault tolerance/graceful degradation characteristics for directional sensors, and the advanced algorithm combining the mobile directional sensors with the proposed Voronoi-based algorithm, so as to obtain and analyze the optimized sensing field coverage.

References

1. Aurenhammer, F.: Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys 23(3), 345–405 (1991)
2. Charfi, Y., Wakamiya, N., Murata, M.: Challenging issues in visual sensor networks. IEEE Wireless Communications 16(2), 44–49 (2009)
3. Chen, U.R., Chiou, B.S., Chen, J.M., Lin, W.: An Adjustable Target Coverage Method in Directional Sensor Networks. In: Proceedings of the IEEE Asia-Pacific Services Computing Conference, pp. 174–180. IEEE Computer Society, Los Alamitos (2008)
4. Cheng, W., Li, S., Liao, X., Shen, C., Chen, H.: Maximal Coverage Scheduling in Randomly Deployed Directional Sensor Networks. In: Proceedings of International Conference Parallel Processing Workshops, p. 68. IEEE Computer Society, Xian (2007)
5. Chow, K.Y., Lui, K.S., Lam, E.: Wireless sensor networks scheduling for full angle coverage. Multidimensional Systems and Signal Processing 20(2), 101–119 (2009)
6. Guvensan, M.A., Yavuz, A.G.: On coverage issues in directional sensor networks: A survey. Ad Hoc Networks 9(7), 1238–1255 (2011a)
7. Hsu, Y.-C., Chen, Y.-T., Liang, C.-K.: Distributed Coverage-Enhancing Algorithms in Directional Sensor Networks with Rotatable Sensors. In: Bononi, L., Datta, A.K., Devismes, S., Misra, A. (eds.) ICDCN 2012. LNCS, vol. 7129, pp. 201–213. Springer, Heidelberg (2012)
8. Kulkarni, P., Ganesan, D., Shenoy, P.: The Case for Multi-tier Camera Sensor Networks. In: Proceedings of the 15th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 141–146. ACM, Skamania (2005)
9. Pham, C., Makhoul, A., Saadi, R.: Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications. Journal of Network and Computer Applications 34(2), 783–795 (2011)
10. Soro, S., Heinzelman, W.: A Survey of Visual Sensor Networks. Advances in Multimedia 2009, 1–22 (2009)
11. Tezcan, N., Wang, W.: Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints. Computer Networks 52(13), 2558–2567 (2008)
12. Topcuoglu, H., Ermis, M., Bekmezci, I., Sifyan, M.: A new three-dimensional wireless multimedia sensor network simulation environment for connected coverage problems. Simulation 88(1), 110–122 (2012)
13. Wang, J., Niu, C., Shen, R.: Priority-based target coverage in directional sensor networks using a genetic algorithm. Computers and Mathematics with Applications 57(11-12), 1915–1922 (2009)

Integration of EPC and Wireless Sensor Networks for Heterogeneous Networking in a U-Life Environment

Yao-Chung Chang

Department of Computer Science and Information Engineering,
National Taitung University, Taiwan
ycc@nttu.edu.tw

Abstract. One of the most popular technologies to implement the ubiquitous networks is RFID and wireless sensor networks (WSNs) which have been actively investigated in the literature. Before the RFID and WSN technologies can be applied to ubiquitous networks successfully, the understanding on the issues how to integrate WSNs with electronic product code (EPC) in the RFID systems plays a critical role. This work aims to propose an approach for integration of RFID and WSN architectures through interworking between WSN and electronic product code (EPC) to formulate the WSN/EPC network architecture. This combination leads to a flexible system framework that creates a convenient platform for users to select the modular they need in developing a digital home environment. The WSN/EPC framework can also be used to facilitate the implementation of heterogeneous networks in a U-life environment.

Keywords: Sensor network, RFID, heterogeneous network, EPC, OSGi service platform, U-life home.

1 Introduction

RFID technology has been widely used in wireless sensor networks. An RFID system [1-3] normally consists of a reader and an RFID tag. There are many daily-life applications for RFID, including EasyCard, electronic toll collection (ETC) intelligent card, etc. In an RFID system, the electronic product code (EPC) [4] is the most important component because it is used to uniquely identify a product or an item by a particular EPC code in its information recognition system. The EPC can be used in automatic identification technology without requiring human intervention. Hence, EPC can marks the small items with the traditional bar codes and it can also be used to identify a single large object, such as a cartoon box, a container, a truck, etc. The other more related applications for the EPC can be found in the literature [5]. Due to its various user-friendly features, a standard message exchange platform can be constructed by using EPC to access the global EPC networks and object information databases.

On the other hand, wireless sensor networking [6-8] is an emerging research topic with many diverse applications, including environment monitoring, smart space management, medical instrumentation, military, and robotic exploration missions. A

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 551–560.
DOI: 10.1007/978-3-642-35452-6_56

wireless sensor network usually requires numerous distributed mobile sensor nodes which are able to organize themselves [9-10] into a multi-hop wireless network. Each node has one or more sensors, an embedded processor, and a low-power radio, and they usually have limited battery power. Typically, WSN nodes are used to work together for a common purpose, such as to monitor some physical metrics continuously and transmit the sensed data through the network to a data sink.

It is also noted that the primary goal of the Open Service Gateway Initiative (OSGi) organization [11], which was established in 1999, is to construct an open standard interface to enable various wide area networking services to communicate with home networking equipment. The OSGi standard, which was developed based on Java technology, enables service operators, household appliances manufacturers, and the Internet service providers to link together with a standard networking interface. Therefore, the manufacturers can focus on developing various home network applications. The three core OSGi components have been defined including ”framework”, ”bundle”, and ”service”. Framework is constructed on Java virtual machine. Bundle is an application program implemented on the framework. Service is the interface service provided (export) and requested (import) by the Bundle. Bundle is remotely downloadable and can be installed and implemented automatically on the OSGi framework. For more information about the OSGi standard, the readers should refer to [11] and [12].

This work aims to propose an integrated RFID and WSN network architecture as an effort to combine the different features of WSN and EPC networks. With the help of the RFID technology, WSNs as well as the OSGi platform, a ubiquitous-life (U-life) home networking system can be implemented. The different elements in the U-life system can be easily controllable or actuated by a user through a mobile device, and the creation of such a U-life home network has a huge potential for its wide applications in the future, including e-health, smart-home, u-hospital, etc.

The remainder of this paper can be outlined as follows. Section 2 gives a comprehensive survey on the related works on wireless sensor networking techniques, EPC network, and OSGi standard. Section 3 describes the architecture and implementation of the U-life system based on the approach proposed in this paper. Three application scenarios are discussed based on the integrated WSN/EPC/OSGi network architecture in this section. Section 4 addresses the issues on numerical analysis of the WSN/EPC framework and presents the experimental results obtained from the proposed system. Finally, conclusions of this paper are drawn in Section 5.

2 Related Work

The applications of Pluggable RIFD components and lightweight RFID systems can be found in various home related service/infrastructure in the literature [13]. Based on the OSGi standard, RFID services or applications can be encapsulated into bundles of OSGi and downloadable to any OSGi environment. Therefore, this system architecture creates a common service environment that can be configured with various RFID readers, tag filters, and databases which can be encapsulated as pluggable components (as illustrated in Figure 1).

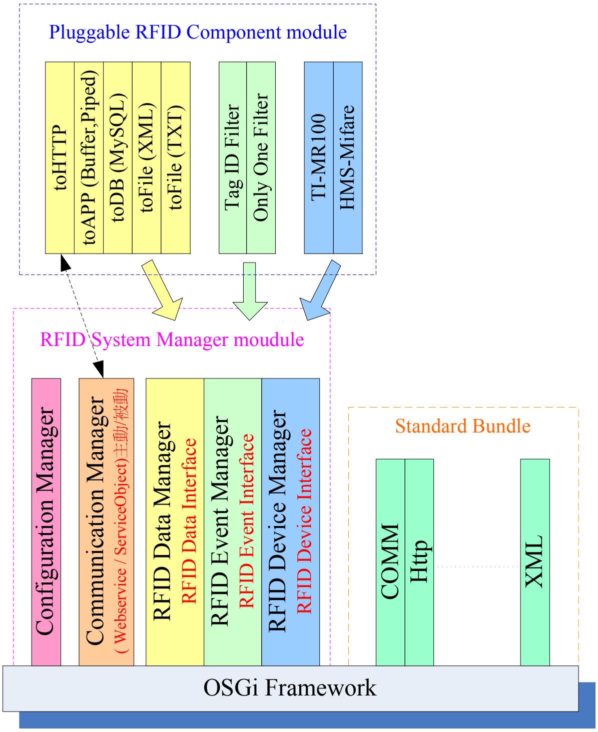


Fig. 1. Software architecture of pluggable RFID components based on OSGi

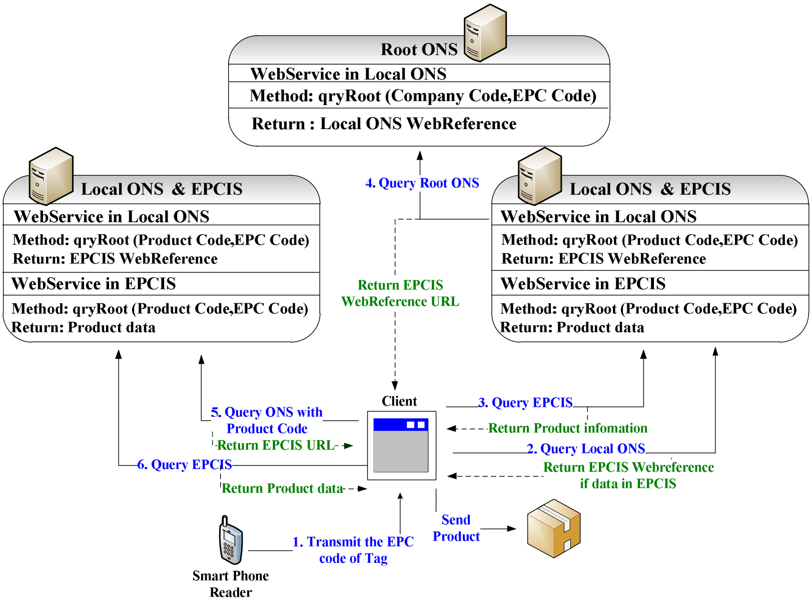

Root ONS
WebService in Local ONS
Method: qryRoot (Company Code,EPC Code)
Return : Local ONS WebReference
4. Query Root ONS
Return EPCIS WebReference URL
Local ONS & EPCIS
WebService in Local ONS
Method: qryRoot (Product Code,EPC Code)
Return: EPCIS WebReference
WebService in EPCIS
Method: qryRoot (Product Code,EPC Code)
Return: Product data
Local ONS & EPCIS
WebService in Local ONS
Method: qryRoot (Product Code,EPC Code)
Return: EPCIS WebReference
WebService in EPCIS
Method: qryRoot (Product Code,EPC Code)
Return: Product data
Client
5. Query ONS with Product Code
Return EPCIS URL
6. Query EPCIS
Return Product data
3. Query EPCIS
Return Product infomation
2. Query Local ONS
Return EPCIS Webreference if data in EPCIS
Send Product
1. Transmit the EPC code of Tag
Smart Phone Reader

Fig. 2. Architecture of a smart phone and EPC Network

Figure 2 shows the concept and architecture of a mobile purchasing system with smart phones and EPC networks [14]. This EPC network experimental scenario includes the following three servers: root object name service (ONS), local ONS, and EPC Information Services (EPCIS). The web service is used to implement the connections between the ONS and EPCIS servers. Clients use mobile RFID based smart phones to process the orders. However, only the RFID/EPC framework was discussed in this study. The applications of OSGi in an intelligent home environment were not discussed in this work.

The rapid advancement of communication and network technologies in recent years has facilitated the wide application of networking devices in homes and small/medium-sized enterprises (SMEs). The needs for data information exchanges over different networks are also expected to increase, further encouraging the use of networking devices in homes and SMEs [15].

3 Implementation of U-Life System

The wireless sensor networks play an important role in the implementation of a U-life home, in which all electrical and electronic appliances will be controlled and actuated based on the variable needs in the human life. Figure 3 shows a U-life home networking system architecture implemented with the help of the EPC network and OSGi framework. Three key components are illustrated, namely a) client with handheld devices, b) EPC framework with object tracing function and EPCIS servers, and c) intelligent home space with RFID based appliances. The OSGi platform works to connect the EPC based devices with the home environment, and the client uses mobile devices to control the appliances in the U-life home environment.

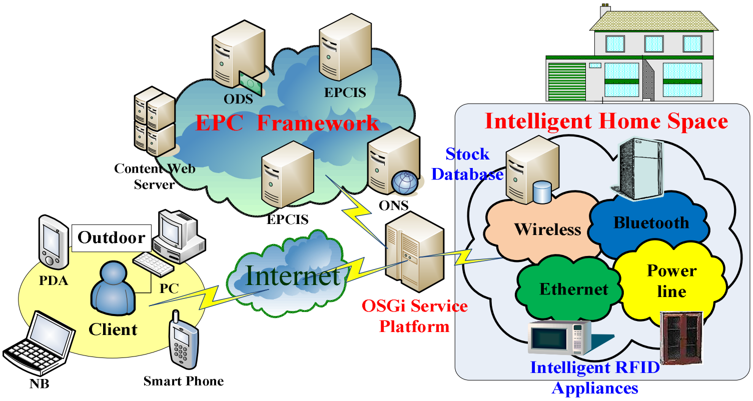


Fig. 3. A framework of U-Life home networking system

This study first discusses the issues on the implementation of the U-life home network system with RFID/EPC framework and OSGi platform. Here, the RFID reader is used to acquire the EPC code, and the EPC network and OSGi gateway are used jointly to implement the U-life home network. The system collects and tracks the product information of various home appliances with their EPC codes in this intelligent home environment. The intelligent RFID based appliances can also send data to the users actively if necessary. The following two scenarios are introduced to demonstrate the possible applications of the U-life home networking system concerned in this paper.

- Scenario 1: A user purchases some goods using a smart phone with RFID reader in a shopping mall. The content web server and the object directory service (ODS) server then record the EPC codes of the products and transport the product information with their object tracing functions through the OSGi gateway directly to the home of the user. The data can be used for various purposed, such as to update the users' spending profile, etc.
- Scenario 2: A user who wants to purchase some goods in a shopping mall can use a smart phone to access the OSGi gateway and check the status of goods, including their product numbers and the expiration dates, etc. This function can help the user to know much extra information about the goods, helping him/her to decide if the goods are suitable for purchasing.

In addition, this paper will also propose a WSN/EPC/OSGi network architecture, which formulates a novel home network architecture combining wireless sensor networks and EPC based heterogeneous networks. This study can make usages of the following farm or ecology observation and monitoring scenario to illustrate the advantages and services deliverable by the proposed integrated system.

- Scenario 3: Many wireless sensor nodes have been deployed in an area to make the observation for specific objects (e.g., animals, plants, or the other objects in the natural environment) with their unique EPC codes. As the initial record or observation, the planting sites, botanical names or the other information such as the climate conditions, etc., can be stored in the database. The observers can monitor the ecological status of a wide area in a real time basis by using PDAs with RFID readers. Moreover, users can obtain the environmental conditions for individual objects through the EPCIS system in order to identify the best environment with a specific time interval for a particular plant growth.

Figure 4 shows the proposed EPC/WSN/OSGi architecture. At the left-hand side of Figure 4, WSN and EPC framework are connected with a WSN/EPC gateway. The wireless network, which connects WSN and EPC, works based on WiMAX technology to transmit data. At the right-hand side of Figure 4, the OSGi service platform is deployed to integrate the EPC framework with an intelligent home environment.

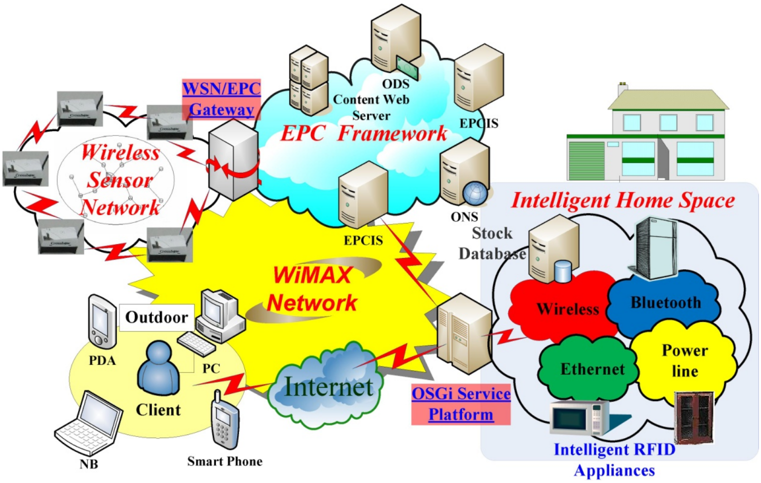
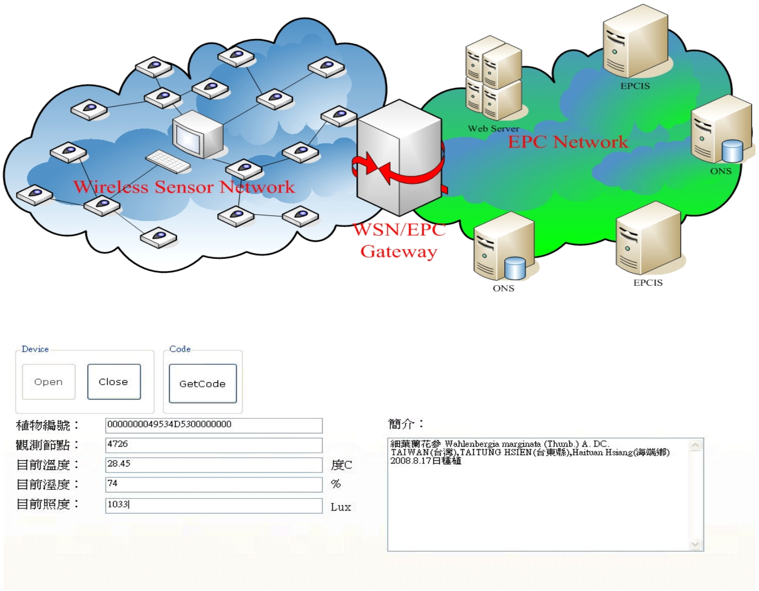


Fig. 4. Architecture of EPC/WSN/OSGi

Figure 5 shows the implementation of the WSN/EPC gateway. An ecological monitoring system interface for the EPC/WSN gateway may contain mobile device, EPC code, sensor node ID, temperature, humidity, illumination, and the other related information.



植物編號： 0000000049534D5300000000
觀測節點： 4726
目前溫度： 28.45 度C
目前溼度： 74 %
目前照度： 1033 Lux

簡介：

細葉蘭花參 Wahlenbergia marginata (Thunb.) A. DC.
TAIWAN(台灣),TAITUNG HSIEN(台東縣),Haituan Hsiang(海端鄉)
2008.8.17日種植

Fig. 5. Ecological monitoring system interface of EPC/WSN

4 Numerical Analysis of the U-life System

In order to map our proposed system (as shown in Figure 6) to a bipartite graph model, let S denote a set of sensor nodes that are used to sense the information of environments. Let CH denote the set of the cluster headers that transmit the query information of the sensor nodes to the WSN/EPC gateway.

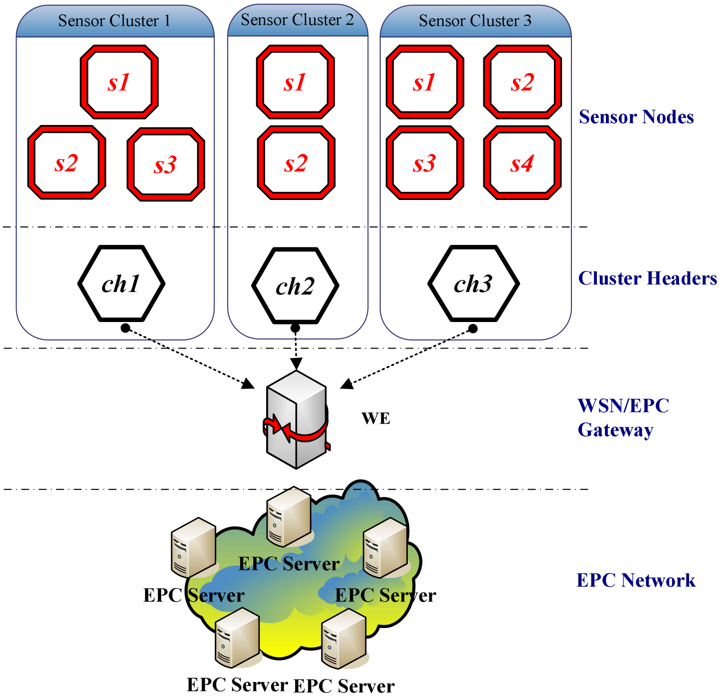


Fig. 6. Layers of proposed system

Let the bipartite graph be denoted by G = (S, CH, E, Q), where S (sensor nodes) and CH (cluster headers) are two disjoint vertex sets defined as follows:

$$E=\{(s, ch)|\text{where } s \in S\,; ch \in CH\}; \tag{1}$$

$$Q=\{(ch, we)|\text{ where } ch \in CH \text{ and } we \in WE\}. \tag{2}$$

The sense cost function SC(e) denotes the amount of cost for carrying out the sensing communications between the sensor node s and the cluster header ch.

$$\text{Sense Cost function= } SC(e), \text{ where } S(e) > 0 \text{ for each } e \in E. \tag{3}$$

The query cost function QC(q) denotes the amount of query cost that ch consumes, or

$$\text{Query Cost function= } QC(q), \text{ where } QC(q) > 0 \text{ for each } q \in Q. \tag{4}$$

An edge e = (s, ch), where $s \in S \, and \, ch \in CH$, denotes a linkage between a sensor node s sending data to a cluster header ch, or

$$\text{Edge } e{=}(s,\ ch) \text{ is a link connecting vertices } s \text{ and } ch \tag{5}$$

A query edge q=(ch, we), $ch \in CH$ and we ∈ WE denotes the cluster header ch can be used to transmit query to WSN/EPC gateway, or

$$\text{Query } q{=}(ch,\ we) \text{ is a link connecting the vertices } ch \text{ to } we. \tag{6}$$

In this paper, the main purpose of the proposed system is to minimize the communication costs between the sensor node, cluster node and the WSN/EPC gateway. Hence, the transmission matching function is defined as

$$\text{TM} = \min \sum\nolimits_{e \,\epsilon\, \mu\,(s,\ ch)} SC(e) \ + \ \min \sum\nolimits_{e \,\epsilon\, \psi\,(ch,we)} QC(q) \tag{7}$$

Where TM denotes a subset of edges of E and Q where each node S is linked with one cluster header in CH by at most one edge in TM, such that the system will have a cost equal to $\min \sum_{e \,\epsilon\, \mu\,(s,\ ch)} \mathrm{SC}(e) + \min \sum_{e \,\epsilon\, \psi\,(ch,\ we)} \mathrm{QC}(q)$, where μ(s, ch) is the set of edges in TM that are incident from s to ch and ψ(ch, we) is the set of edges in TM that are incident from ch to we. An edge (s, ch) in TM denotes the sensor node s is assigned to support the cluster header ch for transmitting the sensing data. An edge (ch, we) in TM means the cluster header ch is assigned to support the sensor node s for transmitting sensing data.

As a simple example shown in Figure 7,

S=$\{s_1, s_2, s_3\}$, CH=$\{ch_1, ch_2, ch_3\}$, E=$\{(s_1, ch_1), (s_2, ch_2), (s_3, ch_2), (s_3, ch_3)\}$.

Assume

$$\begin{cases} S(s_1, ch_1) = 50, \\ S(s_2, ch_2) = 40, \\ S(s_3, ch_2) = 20, \\ S(s_3, ch_3) = 30, \\ QC(ch_1, we) = 120, \\ QC(ch_2, we) = 100, \\ QC(ch_3, we) = 80. \end{cases}$$

The transmission matching (TM) can be calculated as

$$\text{TM} = \mathrm{S}(s_1, ch_1) + \mathrm{S}(s_2, ch_2) + \mathrm{S}(s_3, ch_2) + \mathrm{QC}(ch_1, we)) + \mathrm{QC}\ (ch_2, we),$$

as shown in Figure 7, where the system can finally obtain the value from

$$\min \sum\nolimits_{e \,\epsilon\, \mu(s,\ ch)} SC(e) \ + \text{mini} \ \sum\nolimits_{e \,\epsilon\, \psi\,(ch,\ we)} QC(q).$$

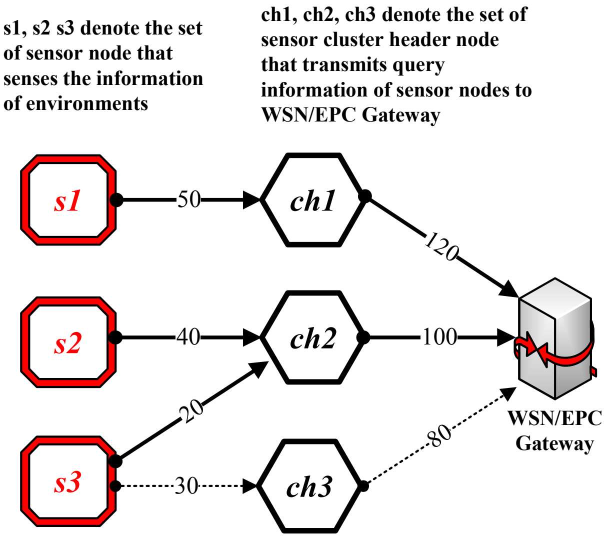

s1, s2 s3 denote the set of sensor node that senses the information of environments
ch1, ch2, ch3 denote the set of sensor cluster header node that transmits query information of sensor nodes to WSN/EPC Gateway
s1
s2
s3
ch1
ch2
ch3
50
40
20
30
120
100
80
WSN/EPC Gateway

Fig. 7. Transmission matching of WSN/EPC gateway

According to the definition: ”self-stabilizing system in spite of distributed control” as made in [16], a system is self-stabilizing if (1) after a finite number of steps, for any initial state it reaches to a legitimate state, (2) no node can execute any algorithm after the system reaches at a legitimate state, and (3) if the system is not in the legitimate state, at least one node can move to a new step. This study can prove that our proposed system has a minimum transmission matching (TM) and will become stable after a finite number of steps.

5 Conclusion

This paper proposed a WSN/EPC/OSGi network architecture which combines EPC and heterogeneous wireless sensor networks to form a home network environment. This work focused on the integration of various features of WSN, RFID, and EPC networks. In addition to discussing the issues on the design and implementation of U-life home networking system, this paper also examined several scenarios for U-life digital home applications. With regard to the security, this work constructed a PAS in the EPC network environment to expand the EPC network to mobile RFID/EPC network under an omni-directional security network framework. Furthermore, the proposed system can add services by adding bundles [11-12]. It was illustrated that the proposed WSN/EPC/OSGi network architecture has a great potential for its applications in futuristic U-life home environment.

Acknowledgements. The author would like to thank Mr. Chun-Han Cheng to assist this work, especially National Science Council of Taiwan under Contract No. NSC 101-2221-E-143 -005 and NSC 98-2221-E-143-002-MY2 for financially supporting this research.

References

1. Auto-ID Labs, http://www.autoidlabs.org/
2. EPC Global, http://www.epcglobalinc.org/home
3. Jalaly, I.: RFID Tagging Explained. IEE Communications Engineer., 20–23 (February 2003)
4. Leong, K.S., Ng, M.L., Engels, D.W.: EPC Network Architecture. In: Auto-ID Labs Research Workshop, Zurich, Switzerland (September 2004)
5. Kim, Y.I., Park, J.S., Cheong, T.S.: Study of RFID Middleware Framework for Ubiquitous Computing Environment. In: International Conference Advanced Communication Technology, vol. 2, pp. 825–830 (February 2005)
6. Akyildiz, I.F., Weilian, S., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor networks. IEEE Transactions on Communications Magazine 40, 102–114 (2002)
7. Tubaishat, M., Madria, S.: Sensor Networks: An Overview. IEEE Potentials 22(2), 20–22 (2003)
8. Tilak, S., Abu-Ghazaleh, N.B., Heinzelman, W.: A taxonomy of wireless microsensor network models. ACM SIGMOBILE Mobile Computing and Communications Review 6, 28–36 (2002)
9. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, pp. 56–67 (August 2000)
10. Zhao, S., Seskar, I., Raychaudhuri, D.: Performance and Scalability of Self-Organizing Hierarchical Ad Hoc Wireless Networks. In: Proceedings of IEEE Conference on Wireless Communications and Networking, vol. 1, pp. 132–137 (March 2004)
11. Open Service Gateway Initiative, OSGi Specification Release 3, http://www.osgi.org
12. Lee, C., Nordstedt, D., Helal, S.: Enabling Smart Spaces with OSGi. IEEE Pervasive Computing 2(3), 89–94 (2003)
13. Sun, K.H., Hou, T.W., Wu, Z.Y.: Pluggable RFID components and a Lightweight RFID System based on OSGi. In: International Conference on Consumer Electronics 2007 (ICCE 2007), pp. 1–2 (January 2007)
14. Chang, Y.C., Wang, H.W.: Mobile Business via Cross Layer Approach toward Intelligent RFID Purchasing System. Journal of Internet Technology 11(7), 965–973 (2010)
15. Ngo, C.: A Service-oriented Wireless Home Network. In: Proceedings of IEEE Consumer Communications and Network Conference, pp. 618–620 (January 2004)
16. Hsu, S.C., Huang, S.T.: A Self-Stabilizing Algorithm for Maximal Matching. Information Processing Letters 43, 77–81 (1992)

UDP-Based Reliable File Delivery Mechanisms for Video Streaming over Unstable Wireless Networks

Shih-Ying Chang[1], Xin-Yan Yeh[2], Hsin-Ta Chiao[1], and Hung-Min Sun[2]

[1] Information and Communications Research Laboratories
Industrial Technology Research Institute, Hsingchu, Taiwan, R.O.C.
[2] Department of Computer Science
National Tsing Hua University, Hsingchu, Taiwan, R.O.C.

Abstract. In recent years, wireless technologies have been widely deployed and the corresponding applications (e.g., video streaming services) have shown high economic potentials. Different from wired networks, wireless signals are more sensitive to the variations of deployed environments, resulting in variant Round Trip Time (RTT) and Packet Loss Rate (PLR). In this paper, we investigate the performance issues about a unicast video streaming service delivered as a file delivery service over unstable wireless networks. When deployed over unstable wireless networks, the throughput of UDP-based file transfer protocols (e.g., File Delivery over Unidirectional Transport (FLUTE) with Application-Layer Forward Error Correction (AL-FEC)) is comparable to that of TCP-based ones (e.g., FTP). However, the throughput of FLUTE may not be optimal on unicast channels due to the lack of feedback mechanisms. In order to improve the throughput, we proposed Reliable FLUTE (R-FLUTE) by introducing the feedback mechanisms into the FLUTE. We have fully implemented R-FLUTE and evaluate effective configurations of R-FLUTE. We also compare R-FLUTE with other conventional TCP-based and UDP-based file transfer protocols (e.g., FTP and TFTP) over unstable wireless networks. The experiment result shows that in comparison with FTP, R-FLUTE shows 10% to 80% throughput gain over unstable wireless networks.

Keywords: AL-FEC, File Transfer Protocol, FLUTE, UDP, Video Streaming, Wireless Networks.

1 Introduction

In recent years, video streaming services over wireless networks have received a lot of attention. In general, video contents are delivered as a streaming service or a file delivery service. For delivered by file delivery services, the streaming source is transfered into a series of streaming content files. In comparison with delivery by streaming services, delivery by file delivery services can provide longer protection period against burst errors over unstable networks when the file transmission is protected by Application-Layer Forward Error Correction (AL-FEC)

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 561–570.
DOI: 10.1007/978-3-642-35452-6_57

[1]. More specifically, in IP datacast (IPDC), File Delivery over Unidirectional Transport (FLUTE) [2] is mainly used to transfer streaming files with higher reliability. FLUTE is a pure UDP-based protocol for better multicast performance. In FLUTE, AL-FEC (e.g., Raptor codes [3]) is used to provide burst error protection. Besides applicable in multicast, Chiao et al. have shown that FLUTE can be used in unicast for transmitting streaming files over unstable wireless networks (e.g., WiMAX networks for stationary and vehicular reception [4][5]). The results show that in such a network condition, the throughput of FLUTE with systematic Raptor codes may be greater than the TCP-based file transfer protocols. Besides FLUTE, there are other TCP-based and UDP-based approaches. We take FTP and TFTP as the representations of TCP-based and UDP-based file transfer protocols, respectively. Both these two protocols possess feedback mechanisms for reliability. FTP contains a built-in ACK channel, and TFTP simulates an ACK channel with UDP packets.

However, due to the lack of a feedback channel, FLUTE uses data carousel (i.e., iterative transmitting a set of files) to provide reliable data transmission, resulting in transmission-inefficiency in the cases of unicast. In order to enhance the throughput of FEC-based reliable data unicast, Ahmad et al. [6] introduce feedback mechanisms to reduce transmission waste. The sender continuously sends the encoding symbols of a block, until receiving the block ACK message notifying that the receiver has received enough encoding symbol of the block. Afterwards, the sender continues the transmission of the next block.

In this paper, in order to solve the drawback of FLUTE, we extend Chiao's works [4][5]) by introducing feedback mechanisms into FLUTE with systematic Raptor codes. We call our proposed schemes Reliable FLUTE (R-FLUTE). R-FLUTE can outperform FLUTE in unicast over unstable networks. To demonstrate the feasibility of our schemes, we implement R-FLUTE and compare R-FLUTE with FTP and TFTP under different network conditions with different Round Trip Time (RTT) and maximum throughput. We establish desired mulit-hop experimental environments by using Wireless Distribution System (WDS) between WiFi APs. In four experimental cases, the result shows that the throughput of R-FLUTE is better than FTP when the average RTT value is more than 15ms and the maximum RTT value is more than 100ms. More specifically, R-FLUTE can outperform FTP by 10% to 80%.

2 Background

Application-Layer Forward Error Correction (AL-FEC). FEC enables the data reconstruction without re-transmission. Hence, the lost packets can be recovered by additionally generated packets thereby avoiding inefficiency caused by re-transmission. AL-FEC (e.g., Raptor codes or RaptorQ codes) supports much large block and symbol size than lower layer FEC and thus providing longer protection period. In current mobile TV standards, systematic Raptor code is the most famous AL-FEC coding algorithm. In Raptor codes, the encoder firstly departs a file into several source blocks. For each block, the encoder encodes the k source symbols of the block into the theoretically unlimited number of encoding symbols. The decoder can recover the original k source symbols from any the $k(1+\epsilon)$ encoding symbols, where ϵ is the decoder overhead (e.g., $\epsilon \leq 0.05$ in Raptor codes). In addition, when the block size grows, the overhead ϵ will be decreased whereas the encoding/decoding overhead will be increased at a rate faster than linear.

Network Model. When deploying wired networks is infeasible, Wireless Distribution System (WDS) becomes a better solution. WDS enables wireless interconnection of Access Points (APs) without a wired backbone. However, due to the properties of wireless connections, when the wireless signal is weak or the number of connected APs is increased, the network condition becomes more unstable (e.g., longer RTT or higher PLR). We use such property to simulate different network conditions in our experiments.

3 Related Works

The unstable networks with variant RTT and PLR will make the performance of TCP-based protocols decreased dramatically. TCP uses network congestion avoidance algorithms, such as additive increase/multiplicative decrease (AIMD). In AIMD, the sender extends the congestion window by a fixed amount for every round trip time but when congestion is detected, the sender diminishes the transmission rate by a multiplicative factor. Due to the property of slow start, when deployed over unstable networks, the performance of TCP-based protocols cannot be optimized. File Transfer Protocol (FTP) standardized in IETF RFC 959 [7] is a well-known file transfer protocol based on TCP. Since FTP is based on TCP, it suffers from the same limitation caused by TCP congestion avoidance algorithms, resulting in poor performance over unstable networks.

Compared to TCP-based protocols, UDP-based protocols have neither complex congestion controls nor reliable mechanism so UDP-based protocols can achieve higher throughput over such networks. Due to the lack of reliable mechanisms, it will be a challenge to implement file transfer protocols using UDP. However, considering its superior throughput over wireless networks, efficient UDP-based implementations will be comparable with those based on TCP. Although several other well-known UDP-based file transfer protocols (e.g., RBUDP [8], Tsunami [9], UDT [10] and PA-UDP [11]) have been proposed, these protocols are employed and evaluated in fixed broadband networks for bulk data transfer [12] (e.g., 1 Gbps networks for 100 MB files), much different from the requirements of real-time video streaming services (e.g., transferring streaming files of size 1 to 2 MB). Up to now, there are two well-known standardized implementations of UDP-based protocols for transferring small size data, TFTP and FLUTE.

Trivial File Transfer Protocol (TFTP). TFTP standardized in IETF RFC 1350 [13] uses UDP packets to simulate the ACK mechanism of TCP. Each packet contains one data block of size 512 bytes. Before sending the next block, the

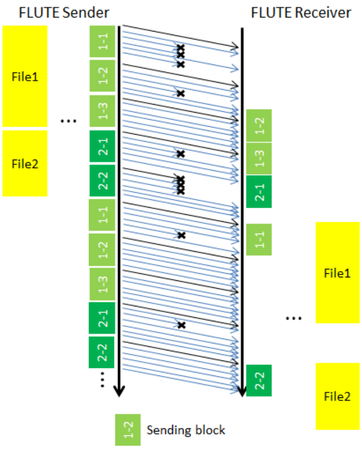


Fig. 1. The sending policy of data carousel in FLUTE

sender must receive the ACK packet from the receiver confirming the completion of present block. The sender only has to hold the last block for re-transmission, so TFTP only requires a small amount of memory, suitable for devices which have less storage or for delivering small size data.

File Delivery over Unidirectional Transport (FLUTE). FLUTE standardized in IETF RFC 3926 [2] is a UDP-based file transfer protocol. The protocol built on Asynchronous Layered Coding (ALC) is developed for massively scalable multicast distribution. FLUTE can signal and map files as well as file properties into ALC objects. An ALC object called File Description Table (FDT) instance is used to inform the receiver the extra file properties and is sent at the beginning of the file transmission. The sending policy in FLUTE is *data carousel* as shown in Fig. 1. FLUTE partitions a file into blocks and then it sequentially sends each block with the corresponding FEC redundancy. After all the blocks for the file are sent, the sender begins sending the blocks for the next file. After all the files are sent, the sender starts next round of the transmission from the first file to the last file. The sender continues sending until the process being terminated. The receivers may receive enough packets for different blocks in different rounds and finally recover all the files. In recent years, FLUTE has become a critical element in IP-based mobile TV standards for the streaming file delivery on multicast and broadcast channels.

FLUTE with Systematic Raptor Codes. Chiao et al. [4][5] introduce systematic Raptor code into FLUTE for better error resilience (e.g., longer protection period) over unstable wireless networks. Their results show that the throughput of FLUTE may be higher than that of HTTP-based approaches when transmitted over IEEE 802.16e mobile WiMAX unicast channel for stationary and vehicular reception. In such a scenario, the performance of TCP which is sensitive to packet loss will be decreased soon. In contrast, FLUTE

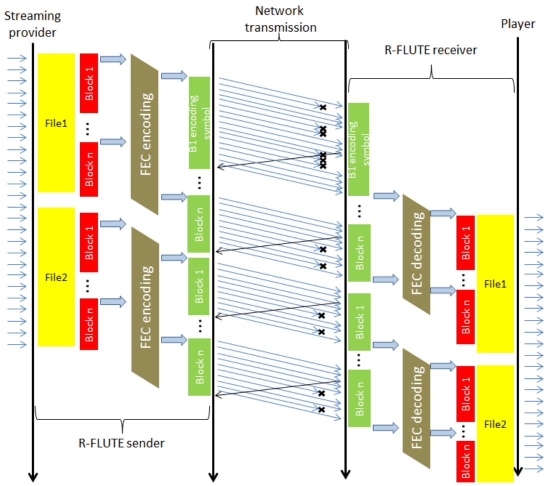


Fig. 2. The data flow of R-FLUTE

with large source block can effectively mitigate the impact of random burst errors. For better transmission-efficiency on unicast channels, we extend their works by introducing the feedback mechanisms into FLUTE with Raptor codes.

4 Reliable-FLUTE (R-FLUTE)

As mentioned in Section 3, for multicast, *data carousel* in FLUTE is very efficient but for unicast, it may not be an optimal solution especially when deployed for delivering a live video streaming service. To improve the throughput of FLUTE on unicast channels, we remove *data carousel* from FLUTE and we further introduce the feedback mechanism into FLUTE such that the sender can transmit blocks one by one reliably.

The data flow of R-FLUTE is shown in Fig. 2. There are four roles in our system. A streaming provider generates streaming files. An R-FLUTE sender (S) sequently sends these files to an R-FLUTE receiver (R). A player renders the streaming files received by the receiver. The details of the data flow are as follows.

1. Streaming provider: transforms a video source into a series of streaming files.
2. For each generated file
   (a) S → R: sends the corresponding FDT instance.
   (b) S: departs the file into source blocks.
   (c) For each source block of the file
      i. S: encodes the block into encoding symbols by the FEC encoder
      ii. S → R : continues sending encoding symbols of the block until receiving the block ACK message from the receiver R.

iii. R → S : sends the block ACK message to the sender when receiving enough encoding symbols of the block.
iv. R: recovers the block by the FEC decoder.

(d) R: merges the recovered blocks into the original streaming file.

3. Player: shows the video to the user through rendering the recovered streaming files sequently.

Here, we use block ACK based confirmation as our feedback mechanism in R-FLUTE. In this paper, we consider two different transport-layer implementations (i.e., TCP and UDP) to transmit the ACK messages. We compare two kinds of implementations as follows.

TCP-Based ACK. Since connection-orient TCP is reliable, carrying the ACK message by a TCP packet guarantees the reception of the ACK message. However, the TCP-based approach has performance drawbacks when deployed over unstable networks. TCP has to establish a connection before transmission. If the network condition is bad (e.g., long RTT or high PLR), the establishment of TCP connections will spend a lot of time. Similarly, the ACK message may get lost in the bad network condition. The behaviors of built-in timeout and re-transmission for the ACK message are also time-consuming. Recall that before receiving the block ACK message, the sender continues sending the encoding symbols of the block, also wasting the bandwidth and increasing the probability of further collision.

UDP-Based ACK. Compared to TCP-based ACK, UDP-based ACK is more efficient over unstable networks because connectionless UDP can sustain even very weak signal level. Moreover, without the built-in timeout mechanism, the ACK message can be re-transmitted as soon as possible. However, due to the lack of reliability control mechanisms, the receiving of the ACK message is not guaranteed. A trivial way to solve this problem is that the R-FLUTE receiver continues sending the ACK message before it receives the symbols of next block. When the network condition is bad, R-FLUTE with UDP-based ACK outperforms R-FLUTE with TCP-based ACK as shown in experiment 2 in Section 5.

For simplicity, we call R-FLUTE with TCP-based ACK and R-FLUTE with UDP-based ACK as R-FLUTE-TACK and R-FLUTE-UACK, respectively. We will compare them with FTP and TFTP in the next section.

5 Experimental Results

Configurations. We conduct our experiments in IEEE 802.11n WiFi networks in which the APs are connected by WDS. More specifically, the channel between the AP and the laptop is established in the 2.4 GHz radio spectrum, and the channel between two neighboring APs is established in the 5 GHz radio spectrum. The device of AP is D-link DAP-2690. The device of the sender is the laptop with Intel i5 2.30 GHz 4-CPU core and Broadcom 802.11n NIC, and the device of the receiver is the laptop with Intel i3 2.27 GHz 4-CPU core and Broadcom 802.11n

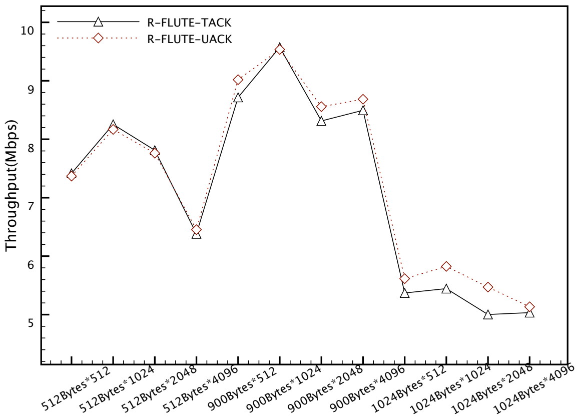


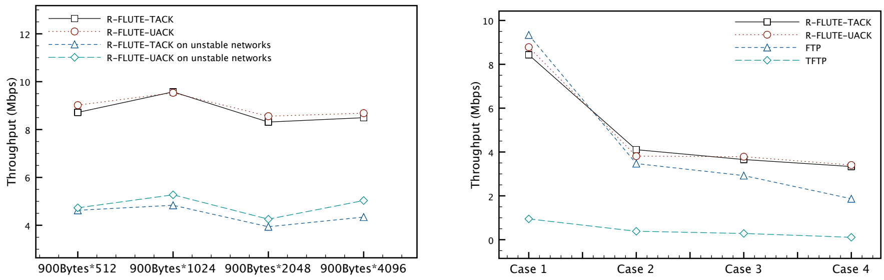
Fig. 3. The throughput of R-FLUTE-TACK and R-FLUTE-UACK for different block and symbol sizes over good network conditions

NIC. Both the programs of the R-FLUTE sender and receiver are running on the Windows 7. We connect two laptops to different pairs of APs in different physical locations in order to find the ideal network condition with desired RTT and maximum throughput between the sender and the receiver. For test files, we use H.264 video encoder to generate the tested streaming files. 200 pairs of MP2 (MPEG-1 Audio Layer II) and H.264 file clips are generated to store digital audio and digital video for five seconds in each file. To improve the transmission-efficiency, we encapsulate each pair of the MP2 file and the SVC file into a single tar file. These 200 test tar files is ranging from 840 KB to 1980 KB, amounting to 292 MB. In the following, we evaluate the optimal settings of R-FLUTE by the first two experiments and compare R-FLUTE with FTP and TFTP in the last experiment.

Experiment 1: Evaluating the Effective Symbol Size and Block Size In this experiment, we measure the throughput of R-FLUTE-TACK and R-FLUTE-UACK for different symbol sizes and block sizes as shown in Fig. 3. The symbol sizes are 512 Byte, 900 Byte and 1024 Byte, and the block sizes (i.e., the number of symbols in a block) are 512, 1024, 2048 and 4096.

Considering the same block size, when the symbol size is 900 Byte whose value is in the middle of three available symbol sizes, we get the highest throughput. This is because when the symbol size gets smaller, the PLR due to collision becomes lower whereas the data processing rate of Raptor encoding/decoding becomes slower. Therefore, the balance between encoding/decoding cost and PLR results in the highest throughput.

Considering the same symbol size, when the block size is 1024 whose value is the second smallest among the four available block sizes, we get the highest throughput. Although the setting with smaller block size requires more block ACK messages for a file (i.e., requiring bigger transmission cost for reliable transmission) and requires bigger decoder overhead in Raptor codes (i.e., bigger ϵ),



(a) The result of experiment 2

(b) The result of experiment 3

Fig. 4. (a) the throughput of R-FLUTE-TACK and R-FLUTE-UACK over different network conditions and (b) the throughput of R-FLUTE-TACK, R-FLUTE-UACK, TFTP and FTP over 4 cases of network conditions

it requires less computation overhead for Raptor encoding/decoding. Note that the computation overhead of Raptor encoding/decoding is at least linearly increased with the block size. Consequently, when the transmission cost is cheaper (e.g., running over good network conditions), we can choose a smaller block size. On the other hand, when the computation cost is cheaper (e.g., running on high-performance computers), we can choose a bigger block size. So, in the experiments with the better network conditions and the laptop devices, the settings with small blocks size will achieve better throughput.

As a result, the setting 900 Byte * 1024 achieves the highest throughput in both R-FLUTE-TACK and R-FLUTE-UACK. Another observation is that the difference between the throughput of R-FLUTE-TACK and R-FLUTE-UACK is not significant over a better network condition. We will show that the difference is significant over unstable networks in the next experiment.

Experiment 2: Evaluating the Impact over Different Network Conditions

In this experiment, we evaluate the throughput of two schemes over a unstable network condition whose average, minimum and maximum RTT value are 25ms, 1ms, and 186ms, respectively.

The result is shown in Fig 4(a). In the unstable network condition, the highest throughput is also happened in the settings with block size 1024. They are 4.836 Mbps and 5.273 Mbps in R-FLUTE-TACK and R-FLUTE-UACK, respectively. There is 9% throughput difference between them, whereas there is only 0.4% in the stable network condition. Due to the efficient ACK transmission, R-FLUTE-UACK is superior than R-FLUTE-TACK over unstable network conditions.

In summary, different network conditions and devices may require different effective settings of R-FLUTE, depending on the Raptor FEC encoding/decoding overhead, the number of block ACKs for a file, and the transmission cost of ACK messages (i.e., block sizes, symbol sizes, device capabilities, file sizes, and network conditions).

Table 1. Four cases with different RTT and maximum throughput

	Case 1	Case 2	Case 3	Case 4
RTT Avg. (ms)	4	15	17	35
RTT Min. (ms)	1	2	4	2
RTT Max. (ms)	44	264	132	154
Iperf UDP (Mbps)	12.9	6.83	6.42	7.64
Iperf TCP (Mbps)	15.3	5.99	4.21	4.57

Experiment 3: Comparisons with FTP and TFTP
To demonstrate the feasibility of our schemes, we compare R-FLUTE-TACK and R-FLUTE-UACK with the conventional file transfer protocols (i.e., FTP and TFTP) because we only compare with the standardized file transfer protocols. We set symbol size 900 Byte and block size 4096 in our schemes. We take FileZilla as our FTP client and server, and we take the TFTP implementation developed by Tandem Systems, Ltd as TFTP server and client. We measure the throughputs over four cases of network conditions built by WDS as shown in table 1. These four cases suffer from different RTTs and maximum throughputs. Here the maximum throughput of TCP and UDP are measured by iperf, a free testing tool for measuring network TCP and UDP traffic. In short, the first case denotes better and stable network condition and the last three cases denote worse and unstable network conditions.

As we can see, the results of iperf show that the throughput of UDP transmission is superior to that of TCP transmission in worse network conditions, confirming our assumption. Next, we compare their performance in Fig. 4(b). The performance of TFTP is much worse than other schemes in all network conditions because its ACK mechanism is only suitable for small size data (e.g., 512 Byte). In the first case denoting stable network conditions, FTP has higher throughput than our schemes but our schemes can still retain about 90% performance of FTP. This is because R-FLUTE requires extra transmission overhead (e.g., decoder overhead of FEC) and extra computation overhead (e.g., encoding/decoding overhead of FEC), making it slightly worse than FTP. However, in unstable network conditions (e.g., the remaining three cases), our schemes outperform FTP. R-FLUTE has about 10% improvement over FTP in case 2, 30% improvement over FTP in case 3, and 80% improvement over FTP in case 4. The results show that UDP-based file transfer protocols are superior to TCP-based file transfer protocols when deployed in unstable network conditions, and our proposed and implemented schemes are effective implementations of UDP-based file transfer protocols.

6 Conclusion

In this paper, we propose the R-FLUTE protocols (i.e., R-FLUTE-TACK and R-FLUTE-UACK) which are the integration of FLUTE, systematic Raptor codes and feedback mechanism. In the experiments, we evaluate the effective settings of our schemes and compare with other standardized file transfer protocols (i.e., TFTP and FTP). The result shows that our schemes have 10% $\sim$ 80% throughput gain better than FTP over unstable wireless networks. We want to improve the performance of R-FLUTE, making it applicable to more environments.

References

1. Gómez-Barquero, D., Poikonen, J.: Filecasting for Streaming Content Delivery in IP Datacast over DVB-H Systems. In: Proc. of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, pp. 1–6. IEEE (2008)
2. Paila, T., Roca, V., Walsh, R., Luby, M., Lehtonen, R.: FLUTE-File Delivery over Unidirectional Transport. Internet Engineering Task Force (IETF) RFC 3926 (2004)
3. Shokrollahi, A.: Raptor Codes. IEEE Transactions on Information Theory 52(6), 2551–2567 (2006)
4. Chiao, H., Li, K., Sun, H., Chang, S., Hou, H.: Application-Layer FEC for File Delivery over the WiMAX Unicast Networks. In: Proc. of 12th IEEE International Conference on Communication Technology (ICCT), pp. 685–688. IEEE (2010)
5. Chiao, H., Li, K.: Performance of the File Delivery Protocols over Mobile TV Interactive Channels. In: Proc. of IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2011)
6. Ahmad, S., Hamzaoui, R., Al-Akaidi, M.: Adaptive unicast video streaming with rateless codes and feedback. IEEE Transactions on Circuits and Systems for Video Technology 20(2), 275–285 (2010)
7. Postel, J., Reynolds, J.: File transfer protocol. Internet Engineering Task Force (IETF) RFC 959 (1985)
8. He, E., Leigh, J., Yu, O., DeFanti, T.: Reliable Blast UDP: Predictable High Performance Bulk Data Transfer. In: Proc. of IEEE International Conference on Cluster Computing, pp. 317–324. IEEE (2002)
9. Meiss, M., Tsunami: A High-Speed Rate-Controlled Protocol for File Transfer (2004)
10. Gu, Y., Grossman, R.: UDT: UDP-Based Data Transfer for High-Speed Wide Area Networks. Computer Networks 51(7), 1777–1799 (2007)
11. Eckart, B., He, X., Wu, Q.: Performance Adaptive UDP for High-Speed Bulk Data Transfer over Dedicated Links. In: Proc. of IEEE International Symposium on Parallel and Distributed Processing (IPDPS), pp. 1–10. IEEE (2008)
12. Yue, Z., Ren, Y., Li, J.: Performance Evaluation of UDP-Based High-Speed Transport Protocols. In: Proc. of IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), pp. 69–73. IEEE (2011)
13. Sollins, K.: The TFTP Protocol (Revision 2). Internet Engineering Task Force (IETF) RFC 1350 (1992)

Medium Access Control with SDMA for Wireless Ad Hoc Networks

Neng-Chung Wang[1] and Yu-Chun Huang[2]

[1] Department of Computer Science and Information Engineering,
National United University, Taiwan, R.O.C.
ncwang@nuu.edu.tw
[2] Department of Computer Science and Information Engineering,
Chaoyang University of Technology, Taiwan, R.O.C.
s9367607@mail.cyut.edu.tw

Abstract. In this paper, an SDMA-based MAC protocol (S-MAC) for wireless ad hoc networks with smart antennas based on IEEE 802.11 is proposed. The proposed protocol exploits the space division multiple access (SDMA) system to allow reception of more than one packet from spatially separated transmitters. Using SDMA technology provides collision-free access to the communication medium based on the location of a node. In wireless ad hoc networks, the SDMA system has several advantages over other wireless multiple access schemes. The proposed protocol solves the hidden terminal problem, the exposed terminal problem, and the hidden terminal problem due to asymmetry in gain. Performance evaluation shows that the proposed scheme outperforms the previous schemes.

Keywords: IEEE 802.11, medium access control, wireless ad hoc networks, SDMA, smart antennas.

1 Introduction

Nowadays, wireless ad hoc networks have played an increasingly important role in a wide range of applications. A wireless ad hoc network is a kind of wireless communication infrastructure that does not have base stations or routers. Its investment cost is less, and each node acts as both a host and a router [3, 7].

New ideas about increasing network throughput by using smart antennas, which allow the nodes to transmit packets in different spatial channels, have been proposed. Smart antennas [4, 6, 13] can enhance network throughput more efficiently than omni-directional antennas in wireless ad hoc networks [10, 12]. Smart antennas offer some advantages that enhance spatial reuse of the wireless channel and extension of the transmission range.

In recent studies, several MAC protocols have been proposed that suitably adapt IEEE 802.11 for directional antennas. However, directional transmissions in wireless ad hoc networks result in serious location information problems [11]. The transmitter must know the location of a receiver to turn the beam in a suitable direction before

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 571–580.
DOI: 10.1007/978-3-642-35452-6_58

transmitting RTS frames. The transmitter creates the appropriate beam patterns, depending on the location of a receiver. Some studies assume that the location of a node may be obtained using the global positioning system (GPS) [5, 10]. Other studies assume the transmitting of data through the routing layer to acquire information about the location of a receiver, or the location of a receiver. In [8], RTS frames are transmitted omni-directionally in order to find the location of a receiver. Korakis et al. [11] proposed the circular directional RTS/CTS, which scans the entire area around the transmitter to find the location of a receiver or a transmitter. Although this scheme attempts to extend the communication range, circular transmission increases the delay time and incurs a large control overhead. Ko et al. proposed a MAC protocol that utilizes the directional transmission capability of a directional antenna to improve bandwidth efficiency, called Directional MAC (D-MAC) [10]. Lal et al. proposed a MAC protocol that exploits the creation of spatial channels to enhance the throughput for ad hoc networks [12]. This scheme uses Space Division Multiple Access (SDMA) technology with smart antennas.

The rest of this paper is organized as follows. In Section 2, related work is presented. Section 3 describes the proposed SDMA-based MAC protocol in detail. In Section 4, the performance of the proposed protocol is compared with other MAC protocols through simulation. Finally, conclusions are given in Section 5.

2 Related Work

IEEE 802.11 [1] is a popular protocol that defines the functions of the MAC and PHY layers in wireless ad hoc networks and infrastructure networks. This popular protocol assumes the use of omni-directional antennas at the physical layer. IEEE 802.11 MAC protocol uses a handshake mechanism implemented by exchanging small control packets named Request-to-Send (RTS) and C1ear-ro-Send (CTS). The successful exchange of these two control packets reserves the channel for transmission to prevent collision occurrences. However, this mechanism wastes the network capacity because they reserve the wireless media over a large area. The IEEE 802.11 MAC protocol exploits the binary exponential backoff algorithm to resolve channel contention. When a frame exchanging sequence between two nodes is in progress, all the other nodes that are within the range of the transmitter or the receiver should defer their transmission to avoid interference with the ongoing sequence. In wireless ad hoc networks, RTS/CTS handshaking is normally used to deal with the common hidden terminal problem, which should reduce collision occurrence. All nodes share the medium when a frame is in the process of exchanging a sequence. This ensures that reservation of the wireless media prevents collision occurrence; however, that squanders spatial channel utilization. This problem is the so-called exposed terminal problem.

Ko et al. proposed a Directional MAC (D-MAC) using directional antennas to improve bandwidth efficiency [10]. D-MAC protocol is similar to IEEE 802.1 I in many ways. D-MAC send an ACK immediately after the DATA, as in 802.11. However, in D-MAC, the ACKs are sent using a directional antenna, instead of an omni-directional antenna. By using directional antennas, while one directional antenna at some node may be blocked, other directional antennas at the same node may not be blocked, allowing transmission using the unblocked antennas. In D-MAC, omni-directional transmission of a packet requires the use of all the directional antennas. Therefore, an omni-directional transmission can be performed if none of the directional antennas are blocked.

The most sophisticated utilization of smart antenna technology is in an SDMA, which employs advanced processing techniques to locate and track mobile terminals, and adaptive steering transmission signals toward users and away from interferers [2, 9, 12, 14]. The SDMA system allows multiple nodes to operate in the same frequency/time slot and uses the smart antenna to separate signals. Lal et al. proposed a MAC layer protocol using space division multiple access (SDMA) [12]. In this scheme, a node receives more than one packet from spatially separated transmitters. To achieve multiple receptions at a particular time requires the synchronization of the transmitters in this scheme. Before a receiver initiates reception, the node must send an omni-directional RTR (Ready-to-Receive) frame which contains a unique training sequence about the receiver to a transmitter. Each transmitter replies with a directional RTS message which contains a unique training sequence about the transmitter. This scheme transmits RTS/CTS/DATA/ACK using a directional transmission mode, which is used to reserve transmission. After reception of the RTS frame, the receiver simultaneously sends the directional CTS to all transmitters. Finally, data frames are transmitted directionally towards the receivers. After the data packets are received, the receiver replies with a simultaneous directional ACK.

3 Proposed SDMA-Based MAC Protocol

In wireless ad hoc networks, using an omni-directional transmission mode is effective for inquiring into the location of a destination node because nodes in wireless ad hoc networks are free to move anywhere independently. In the proposed S-MAC, the location of the intended receiver was obtained by using omni-directional RTS.

Suppose node *A* wants to transmit information to a destination node *B* in a wireless ad hoc network. Node *A* is the transmitter, node *B* is the receiver, D-ACK represents a reply with ACK using the directional transmission mode, D-DATA indicates that data is sent using the directional transmission mode, and omni-RTS/CTS/BDTS represents the transmission of RTS/CTS/BDTS using the omni-directional transmission mode. In the omni-directional transmission mode, each node uses four antenna elements having individual beams of 90 degree beamwidths covering 360 degrees, and simultaneously broadcasts a control frame using all the beams. Therefore, the coverage area in S-MAC is wider than the conventional MAC protocol using omni-directional antennas.

The modified RTS/CTS handshake of the proposed S-MAC is as follows.

1. Omni-directional RTS (Request-to-Send): Node *A* sends a RTS frame using the omni-directional mode. Initially, transmitter node *A* wants to send a packet to receiver node *B*, but node *A* does not know the location of node *B*. Therefore, node *A* broadcasts RTS to find the location of node *B* using the omni-directional transmission mode. Node *A* sends RTS to all the nodes in its coverage range. The neighbors

other than node *B* which receive RTS from transmitter node *A* realize that node *A* wants to communicate with node *B*. The neighbors in the coverage range of transmitter node *A* check their respective beam table so that they do not interfere with the communication between node *A* and node *B*.

2. Omni-directional CTS (Clear-to-Send): Node *B* sends a CTS frame using the omni-directional mode. Receiver node *B* broadcasts CTS in its coverage range by using omni-directional transmission after receiving RTS. The neighbors other than node *A* which receive the CTS from receiver node *B* realize that node *A* wants to communicate with node *B*. The neighbors of receiver node *B* check their respective beam table, and the nodes avoid interfering with this communication between node *A* and node *B*. Based on the direction in which CTS is received, the neighbors update the beam information on node *B*. The neighbors in the coverage range of receiver node *B* update their respective beam table according to the beam information in CTS. This beam information on the communicating nodes is attached to the CTS. Therefore, the neighbors do not interfere with the communication process.
3. Omni-directional BDTS (Beam-Direction-to-Send): Node *A* sends a BDTS frame using the omni-directional mode. The BDTS control frame must be modified in order to forward beam information about the communicating nodes to the neighbors. Transmitter node *A* broadcasts BDTS in its coverage range using omni-directional transmission after receiving CTS. The neighbors other than node *B* in the coverage range of transmitter node *A* update their respective beam table according to the beam information in BDTS. The beam information on the communicating nodes is attached to the BDTS. Therefore, the neighbors avoid collision and the hidden terminal problem due to asymmetry in gain.
4. Directional DATA: The transmitter sends data to the receiver using the directional transmission mode.
5. Directional ACK: The receiver replies by sending an ACK to the transmitter using the directional transmission mode.

The RTS/CTS handshaking attempts to reserve the wireless channel and exchange location information. CTS/BDTS handshaking attempts to forward the beam information to the neighbors. The modified CTS and BDTS frame formats are shown in Fig. 1 and Fig. 2, respectively. The CTS/BDTS frame has a few additional attributes: (a) the sender beam and (b) the receiver beam. The CTS/BDTS control frame must be modified in order to forward the beam information on the communicating nodes to the neighbors. The beam information of the communicating nodes is attached to the CTS/BDTS frame. Therefore, the neighbors in the coverage area of the transmitter and receiver can prevent collision during communication.

Bytes:	2	2	6	1	1	4
	Frame control	Duration	Receiver address	Sender beam	Receiver beam	FCS

Fig. 1. The modified CTS frame format

Bytes:	2	2	1	1	4
	Frame control	Duration	Sender beam	Receiver beam	FCS

Fig. 2. The BDTS frame format

3.1 Beam Table

The proposed S-MAC protocol employs a simple scheme to prevent collision and increase throughput. Using the SDMA system, medium access was assigned to the nodes based on their locations. In this scheme, each node must maintain a beam table. The transmitter must have the beam information of the intended receiver to point the beam in the appropriate direction.

Fig. 3 shows how these nodes set up their beam tables. It is assumed that node *A* transmits data to node *B* in this example. Initially, each beam table is empty and is updated upon every reception. Nodes *B* and *C* are neighbors that hear the omni-directional RTS from node *A*. The beam information of the transmitter is attached to the RTS frame. In the RTS frame, nodes *B* and *C* obtain the relative beam direction of node *A*. Tables 2 and 3 show the results after nodes *B* and *C* have updated their beam tables.

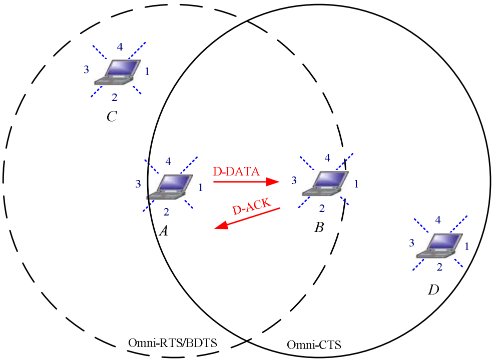


Fig. 3. Communication based on SDMA

Nodes *A* and *D* are both in the coverage area of node *B*, receiving the CTS frame. Nodes *A* and *D* obtain the relative beam direction of node *B*. Tables 1 and 4 show the results after nodes *A* and *D* have updated their beam tables. The beam information on the communicating nodes is attached to the CTS and BDTS frames. All nodes maintain the data in their beam tables when they receive a control frame, such as the omni-direction RTS/CTS/BTDS frame. The proposed scheme maintains the beam tables by broadcasting RTS/CTS/BDTS to prevent collision. The scheme is based on the SDMA mechanism to achieve multiple transmissions; thus, the performance greatly improves network capability.

Table 1. The beam table of node *A*

Me	Neighbor	My beam	Neighbor's beam
A	*B*	1	3

Table 2. The beam table of node *B*

Me	Neighbor	My beam	Neighbor's beam
B	*A*	3	1

Table 3. The beam table of node *C*

Me	Neighbor	My beam	Neighbor's beam
C	*A*	2	4

Table 4. The beam table of node *D*

Me	Neighbor	My beam	Neighbor's beam
D	*B*	3	1

3.2 Modified Network Allocation Vector (NAV)

The Virtual Carrier Sense (VCS) mechanism in IEEE 802.11 was essentially designed to calculate the duration value of a frame. The length of all the remaining frames in the sequence must be known a *priori*. In this mechanism, every station maintains a duration value, which is called the Network Allocation Vector (NAV). When nodes hear a RTS/CTS frame, they examine this duration value. Based on NAV, nodes can also recognize whether the transmission medium is busy or not. Then they can prevent collision accordingly.

The proposed scheme modifies VCS in IEEE 802.11, and is shown in Fig. 4. The omni-directional NAV contains the RTS/CTS/BDTS handshake. This scheme assigns

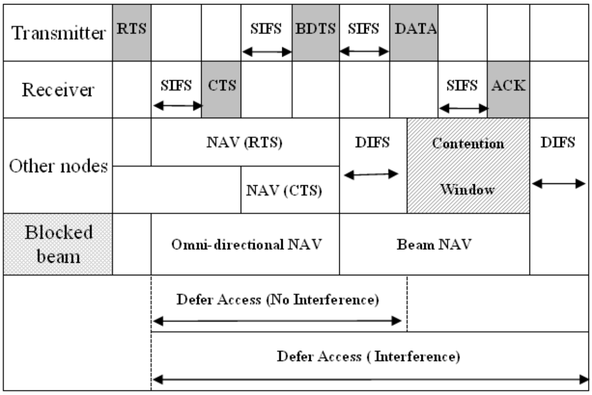


Fig. 4. The NAV of RTS/CTS/BDTS

an omni-directional NAV shorter than the conventional NAV. If other nodes do not interfere with this transmission and prevent collisions, these nodes start communication after a BDTS frame has been sent. The duration value would be attached to the RTS/CTS/BDTS frames. When the nodes hear a control frame (RTS/CTS/BDTS), they update their NAV. According to their beam tables, all the nodes can estimate whether interference with a communicating action will occur. Based on the beam information of the received CTS/BDTS, the neighbors decide to initiate a block if it is necessary to defer their transmission in a specific beam direction. The value of the beam NAV is updated from the duration field of the overheard CTS/BDTS.

3.3 Improvement in Wireless Ad Hoc Networks

In IEEE 802.11, RTS/CTS handshaking is always used to cope with the common hidden-terminal problem. However, the directional transmission of RTS/CTS that is usually used in directional MAC protocols generates the hidden terminal problem. The difference in antenna gain generate the hidden terminal problem due to asymmetry in gain. The proposed scheme transmits RTS/CTS/BDTS in all directions, solving the directional transmission problem and the exposed terminal problem.

Directional Hidden Terminal Problem Due to Unheard RTS/CTS

Directional transmission of RTS/CTS leads to a new hidden terminal problem [5, 11]. In the proposed S-MAC, each node unnecessarily wastes time waiting for the end of a communication. Because smart antennas have the SDMA mechanism, numerous nodes can communicate at the same time without influencing each other. Therefore, they can enhance network throughput and increase spatial reuse.

The example shown in Fig. 5 assumes that communication between nodes *A* and *B* is in progress, with node *A* transmitting to node *B*. Using the beam information that is contained in the CTS/BDTS, the neighbors know that node *A* is receiving data using beam 1 while node *B* is receiving data using beam 3. If node *E* wants to transmit data to node *B*, it realizes that it cannot interfere with the transmission that is in progress. If node *E* wants to transmit data to node *A*, it realizes that it can interfere with the reception of node *A*. Therefore, node *E* defers its transmission through beam 3, updates its beam NAV, and avoids collision.

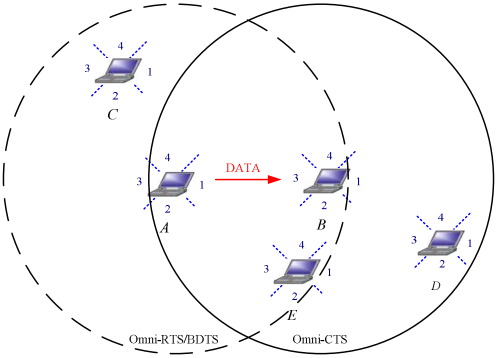


Fig. 5. An example of the hidden terminal problem due to unheard RTS/CTS

Exposed Terminal Problem

As is also shown in Fig. 5, according to the IEEE 802.11 MAC protocol, when a communication between node *A* and node *B* is in progress, the neighbors *C*, *D*, and *E* must defer their transmissions until the current communications is finished. In the proposed S-MAC, if node *E* wants to transmit data to node *D*, it realizes that it cannot interfere with the communication that is in progress. This solves the exposed terminal problem and increases the spatial reuse of a wireless channel.

4 Performance Evaluation

A simulator in C++ language was designed and implemented for performance evaluation. In the following, some assumptions on the parameters of the system architecture were made in the simulations. Each node was equipped with antenna arrays of 4 elements. The omni-directional transmission range was 150 m. The directional transmission range was 250 m. The beamwidth of the directional transmission was 90 degrees. The data packet size was 1024 bytes. Data transmission rate was set to 11 Mbps. The Short Interframe Spaces (SIFS) was 4 microseconds and the Distributed Coordination Function Interframe Spaces (DIFS) was assumed to be 10 microseconds. The sender and receiver were randomly chosen. Each node was randomly assigned an initial energy. At the start of every simulation the beam table of each node was empty and was gradually updated as the simulation progressed. Seventy nodes were arranged at random in a square area with a dimension of 500 m × 500 m. In this simulation, the network size was varied from 10 to 70 nodes to observe the effect.

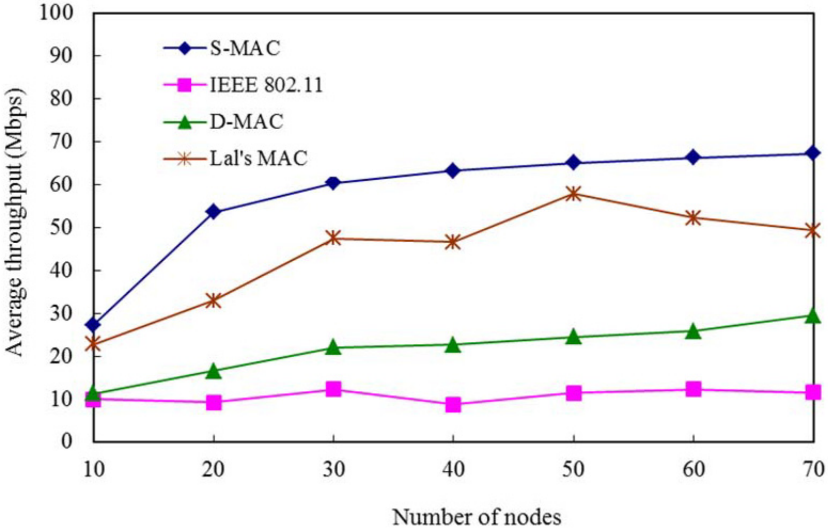


Fig. 6. Average throughput vs. number of nodes

Fig. 6 shows the average throughput of the four schemes with different numbers of nodes. In general, the throughput increased when the number of nodes increased. The throughput of S-MAC was roughly 720% of IEEE 802.11, 280% of D-MAC, and 135% of Lal's MAC. In this simulation, the proposed S-MAC significantly increased the throughput. IEEE 802.11 performed the worst because of the exposed terminal problem. D-MAC performed poorly as a result of the hidden terminal problem because of the unheard RTS/CTS and the hidden terminal problem due to asymmetry in gain. In Lal's MAC, the transmitter nodes deferred their communication until the received RTR led to increased transmission time and reduced throughput performance.

5 Conclusions

In this paper, we propose an MAC protocol for wireless ad hoc networks using smart antennas. The proposed scheme exploits the creation of spatial channels to effectively enhance network capacity. This protocol utilizes the broadcasting of RTS to find the location of a receiver, and it employs a directional beam to increase the spatial reuse of the wireless channel and to extend the transmission range. This protocol does not assume any knowledge of the location of neighboring nodes. Performance evaluation shows that the proposed scheme outperforms the previous schemes.

Acknowledgments. This work was supported by the National Science Council of Republic of China under grants NSC-100-2221-E-239-023 and NSC-101-2221-E-239-032.

References

1. ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (August 1999)
2. Bana, S.V., Varaiya, P.: Space Division Multiple Access (SDMA) for Robust Ad hoc Vehicle Communication Networks. In: Proceedings of the International IEEE Conference on Intelligent Transportation Systems, pp. 962–967 (August 2001)
3. Bellofiore, S., Foutz, J., Govindarajula, R., Bahceci, I., Balanis, C.A., Spanias, A.S., Capone, J.M., Duman, T.M.: Smart Antenna System Analysis, Integration and Performance for Mobile Ad-Hoc Networks (MANETs). IEEE Transactions on Antennas and Propagation 50, 571–581 (2002)
4. Bhobe, A.U., Perin, P.L.: An Overview of Smart Antenna Technology for Wireless Communication. In: Proceedings of the International IEEE Conference on Aerospace 2001, vol. 2, pp. 875–883 (March 2001)
5. Choudhury, R.R., Yang, X., Vaidya, N.H., Ramanathan, R.: Using Directional Antennas for Medium Access Control in Ad Hoc Networks. In: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, pp. 59–70 (September 2002)
6. Chryssomallis, M.: Smart Antennas. IEEE Antennas and Propagation Society 42(3), 129–138 (2000)

7. Corson, S., Macker, J.: Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations (Internet-Draft). Mobile Ad-hoc Network (MANET) Working Group, IETF (October 1998)
8. Fahmy, N.S., Todd, T.D., Kezys, V.: Ad Hoc Networks with Smart Antennas Using IEEE 802.11-Based Protocols. In: Proceedings of the International IEEE Conference on Communications, vol. 5, pp. 3144–3148 (May 2002)
9. Hui, X., Gao, T.: Random Channel Allocation Scheme for SDMA in a Smart Antenna Systems. In: Proceedings of the International Conference on Communications and Mobile Computing, vol. 2, pp. 130–134 (April 2010)
10. Ko, Y.B., Shankarkumar, V., Vaidya, N.H.: Medium Access Control Protocols Using Directional Antennas in Ad Hoc Networks. In: Proceedings of the 19th Annual Joint Conference on IEEE Computer and Communications Societies, vol. 1, pp. 26–30 (March 2000)
11. Korakis, T., Jakllari, G., Tassiulas, L.: A MAC Protocol for Full Exploitation of Directional Antennas in Ad Hoc Wireless Networks. In: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing Annapolis, pp. 98–107 (June 2003)
12. Lal, D., Toshniwal, R., Radhakrishnan, R., Agrawal, D.P., Caffery, J.J.: A Novel MAC Layer Protocol for Space Division Multiple Access in Wireless Ad Hoc Networks. In: Proceedings of the Computer Communications and Networks, pp. 614–619 (October 2002)
13. Lin, T.M., Dai, J.J.: A Collision Free MAC Protocol Using Smart Antenna in Ad Hoc Networks. In: Proceedings of the Consumer Communications and Networking Conference, pp. 301–306 (January 2004)
14. Zhou, S., Niu, Z.: An Uplink Medium Access Protocol with SDMA Support for Multiple-Antenna WLANs. In: Proceedings of the IEEE Wireless Communications and Networking Conference, pp. 1809–1814 (March 2008)

Game Theoretical Cooperative Resource Allocation for Broadcasting in Cognitive Radio Networks

Hsin-Chun Wu[1], Shuo-Cheng Hu[2], and Ai-Chun Pang[1]

[1] Graduate Institute of Networking and Multimedia, National Taiwan University, Taiwan
r99944031@csie.ntu.edu.tw
[2] Dept. of Information Management, Shih Hsin University, Taiwan
schu@cc.shu.edu.tw

Abstract. Cognitive radio (CR) network is proposed to solve the utilization of radio spectrum resource. Spectrum trading has been an interesting topic in CR networks. In this paper, we formulate the problem as a game between multiple primary users (PUs) and multiple secondary users (SUs). We model the dynamic behavior of SUs using the theory of evolutionary game and the competition among PUs using a non-cooperative game. An evolution algorithm is presented to achieve the evolutionary equilibrium for a SU. For a PU, an iterative algorithm for strategy adaption to achieve Nash equilibrium is presented. We also introduce the idea of giving discount to the SUs which can provide help for the PU in broadcasting information. The numerical results show that our proposed model can achieve the optimal equilibrium point and the payoffs of both PUs and SUs are higher than those in the ordinary pricing-based game.

Keywords: CR networks, Non-cooperative game, Evolutionary game, Nash equilibrium, Evolutionary equilibrium.

1 Introduction

Radio spectrum is a scarce resource for wireless communication. Cognitive radio network (CR network) is proposed to enhance the utilization of spectrum. In CR network, primary users (PUs) can sell unused spectrum to secondary users(SUs), and each secondary user can buy the opportunities provided by different PUs. Both primary users and secondary users are selfish, they hope to maximize their own payoff in each transaction. As each user dynamically interact with each other so that the outcome depends not only on its own action but also on other users'. Game theory is a good fit for solving the spectrum trading problem in CR networks.

In recent year, many researches have been working on the dynamic resource sharing problems in CR networks by using game theory model. In [1], authors modeled the dynamics of competitive spectrum sharing and pricing among multiple buyers and multiple sellers. Authors used game theory to solve the best pricing and sharing spectrum size of primary users and formulate the strategy adaption of secondary users. In [2], authors proposed the problem that primary user how to decide the spectrum price according to the secondary users' demand of bandwidth with other competitive

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 581–590.
DOI: 10.1007/978-3-642-35452-6_59

primary users, and authors used game theory to find the optimal spectrum price of primary users. In [4], authors found most of researches with game didn't describe secondary users how to make choice, so they proposed an evolutionary algorithm to describe secondary users how to make strategy and the meaning of equilibrium states in the dynamic pricing-based trading system. We observe those researches and find that those authors want to solve the best spectrum price and secondary users how to make a right decision. So, we want to make a comparison the payoff of our system with ordinary pricing-based model.

In this paper, we make an effort on improving the payoff of all users compared with ordinary pricing-based system. We introduce the idea of discount into the ordinary pricing-based system. In our system we think information has value, primary users can get extra revenue if the broadcasting range of primary users can be extended with the aid of secondary users. In return, the primary users will offer discount to those secondary users that provide help. If the parameters are properly set, our trading model can raise the payoff of both users.

The major contributions of this paper are as follows:

- Model the dynamics of a multiple-sellers, multiple-buyers spectrum trading market by using non-cooperative game and evolutionary game respectively.
- Define the utility function for each kind of users in the ordinary pricing-based system and our proposed system. We also define the algorithm of strategy adaption for each kind of users.
- Define the conditions that our system can bring higher payoff at the equilibrium point.

The rest of this paper is organized as follows: The system model is described in Section 2. In Section 3, we formulate all users' utility functions, algorithms of transactions and analysis of our system and ordinary pricing-based model, and present the results in Section 4. Finally the conclusion is stated in Section 5.

2 System Model

As shown in Fig.1, we consider a CR system with *M* primary users and *N* secondary users. The transmission time frame comprises four time periods, they are primary user's transmission period, sensing period, broadcasting period and sharing period. Every primary user *i* uses the spectrum F_i to transmit data during the "primary user's transmission period". After the transmission is finished, primary user *i* can start to trade sharing period with secondary users. During the sensing period, the SUs could probabilistically change their spectrum selection by observing the payoffs of different strategies adopted by other users buying different spectrum opportunities. Each PU will inform its buyers of the spectrum price and the discount for unit contribution. The contribution means the number of nodes which are out of the coverage of the PU can receive the broadcasting information with the help of its buyers. The contribution does not count in the interfered nodes which receive the same copy of information from different nodes during the same broadcasting period. We assume that all SUs can communicate with each other, e.g., to exchange the buying decision and the payoff. However, these SUs will make decision on buying spectrum opportunities independently. On the other hand, A PU has only local information (e.g., number of buyers, and contributions) and the size of spectrum opportunity (i.e., sharing period) to be sold to the SU is fixed. In our system, the primary user not only sells timeslots to the buyers but also wants the buyers' help in broadcasting information to his uncovered nodes. The SUs who buy spectrum opportunities from a PU have the obligation to broadcast for the PU during the broadcasting period. In return, the primary users will offer discount to those SUs that provide help. The amount of discount is proportional to the buyers' contribution. After the broadcasting period, all buyers can share the spectrum during the sharing period.

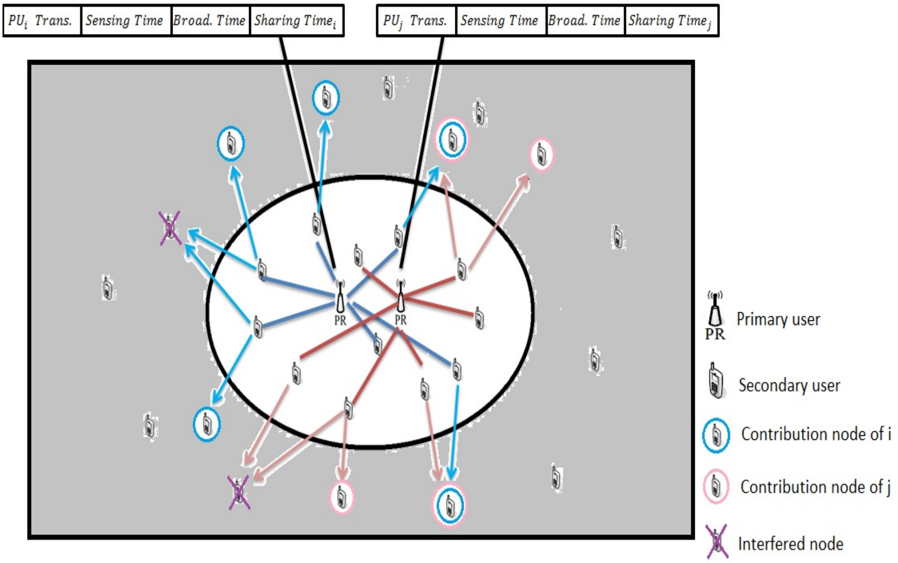


Fig. 1. Illustration of system model

We use two kinds of game to model our system, first is non-cooperative game, which is used to model primary users' competition. Because primary users only know his local information and each primary user only uses his local information to improve his payoff with time, in this game model we want to find *Nash equilibrium* of price finally. The *Nash equilibrium* of price means this price can bring primary user maximum payoff in the competitive game. Second game model is evolutionary game, which uses to model secondary users' behaviors and decision making. Because all secondary users know each other's payoff and decision, secondary user can use peers' information to evolve to evolutionary equilibrium. The evolutionary equilibrium means that all secondary users have same utility in our system. The objective of our system is to promote players' payoff at the equilibrium point which compares with ordinary pricing-based model.

3 Problem Formulation

3.1 Secondary Users' Game

Evolutionary game was initially used to model the ecological phenomena. In evolutionary game, players are categorized populations and use the populations' information to observe the evolutionary equilibrium. In our system, secondary users will process the evolutionary algorithm to make decision. The utility function of secondary users as follows.

$$U_i^k = \frac{SharingTime_i}{n_i(p)} \times profit_i^k - price_i + \frac{BN_i(n)}{n_i(p)} \times d_i - power\cos t_i \tag{1}$$

Where U_i^k means the utility of secondary user k who buy spectrum from primary user i. $profit_i^k$ means that secondary user k can get how much profit by using spectrum i each unit timeslot. We assume secondary users who buy spectrum i have same *profit*.

$price_i$ is the spectrum price. $BN_i(n)$ is the number of contributions made by buyers. d_i is discount for unit contribution. n is population vector.

$$n = [n_1(p) \quad n_2(p) \quad \cdots \quad n_M(p)] \tag{2}$$

$n_i(p)$ means the number of secondary users who buy spectrum i with price vector p

$$p = [price_1 \quad price_2 \quad \cdots \quad price_M] \tag{3}$$

$powercost_i$ is the transmission power cost which secondary user broadcast packet for primary user i.

In Evolutionary game, replicator dynamics is crucial, because it can provide the population information at particular time. Replicator dynamics is defined as follows[1][4][6].

$$\dot{x}_i(t) = \frac{\partial x_i(t)}{\partial t} = x_i(t)(U_i(t) - \overline{U}(t)) \tag{4}$$

where x_i(t) means the portion of SUs choose PU i, as shown in equation (5). And equation (6) represents the average utility in the system.

$$x_i(t) = \frac{n_i(p(t))}{N} \quad N = \sum_{i=1}^{M} n_i(p(t)) \tag{5}$$

$$\overline{U}(t) = \frac{\sum_{i=1}^{M} U_i(t) \times n_i}{N} \tag{6}$$

In this system, evolutionary equilibrium can be obtained from following equation. If (7) forms, it means none of players wants to change his strategy.

$$\dot{x}_i(t) = \frac{\partial x_i(t)}{\partial t} = 0 \quad \forall i \tag{7}$$

In this paragraph we present the algorithm of secondary users how to make choice and how to evolve to evolutionary equilibrium. The evolutionary algorithm can be describe as follows[1- 2][4]:

1. Secondary user randomly choose primary user *i*
2. Secondary user can calculate his utility and obtain average utility
3. If utility of secondary user is less than average utility, then that secondary user will be given a probability *Pb* to randomly choose higher utility strategy.

$$Pb = \frac{\overline{U} - U_i}{\overline{U}} \tag{8}$$

4. The steps from 2 to 3 are repeated until the following condition happening

$$U_i = \overline{U} \quad \forall i$$

After doing above algorithm, the evolutionary equilibrium has been evolved and none of secondary user has any argument about their choice.

3.2 Primary Users' Game

Because each secondary user has to broadcast information for the primary user before transmitting its own data, it will bring extra cost to secondary users. In order to get extra revenue, every primary user has to set the discount for unit contribution. The discount value can motivate the secondary user to broadcast information for primary users.

Now, we define the payoff function of primary user as follows.

$$Payoff_i(p) = price_i \times n_i(p) + BN_i(n) \times r_i - BN_i(n) \times d_i \tag{9}$$

where $Payoff_i(p)$ is the payoff of primary user *i* with price vector *p*, and r_i is the revenue for unit contribution. d_i is the discount for unit contribution. In non-cooperative game of primary users, we want to find *Nash equilibrium*. By definition, *Nash equilibrium* of game is a strategy profile with the property that no player can increase his payoff by choosing a different action, given other player's actions. *Nash equilibrium* can be obtained by best response. Best response function is the best strategy of one player given others' strategy defined as follows:

$$\beta_i(p_{-i}) = \arg\max_{price_i} Payoff_i(p_{-i} \cup price_i) \tag{10}$$

p_{-i} denotes the price set which consists of price of primary user *j* for $j \neq i$. The *Nash equilibrium* can be expressed by best response as follows.

$$price_i = \beta_i(p_{-i}) \quad \forall i \tag{11}$$

In order to obtain the *Nash equilibrium*, we have to solve the following equation: [3][5]

$$\frac{\partial Payoff_i(p)}{\partial price_i} = 0 \quad \forall i \tag{12}$$

We have assumed that primary users only know his local information previously. Therefore, primary users can use iterative algorithm to find *Nash equilibrium*. Let *price$_i$*(*t*) denote the price of primary user *i* at time *t*. The iterative algorithm of price as follows[1- 2][7]:

$$price_i(t+1) = price_i(t) + \alpha_i \frac{\partial Payoff_i(p)}{\partial price_i} \tag{13}$$

$$\frac{\partial Payoff_i(p)}{\partial price_i} \approx \frac{Payoff_i([\cdots price_i + \varepsilon \cdots]) - Payoff_i([\cdots price_i - \varepsilon \cdots])}{2\varepsilon} \tag{14}$$

where α_i is the speed of adjustment for spectrum price. The update of price depends on α largely, if α is too big, primary users can't match his *Nash equilibrium* probably. Iterative algorithm means that each primary user must wait an iteration to observe the number of secondary user and contribution made by buyers, and use local information to update his price for finding his maximum payoff. When all primary users get to *Nash equilibrium*, the following state forms.

$$price_i(t'+1) = price_i(t') \quad \forall i \tag{15}$$

4 System Simulation and Results Analysis

In this section, we present the simulation of this system and compare with ordinary pricing-based model. We consider a two cognitive radio network environment and each system with two primary users A and B. The coverage area of A and B are the same. There are 100 secondary users want to buy spectrum and there are 100 nodes that over the coverage area of A and B and could be broadcasted by secondary users in A or B. We set the initial $price_A$=2 , $price_B$=2, $SharingPeriod_A$=1000, $SharingPeriod_B$=900, $profit_A$=2, $profit_B$=2, α_A=0.0025, α_B=0.0025, ε=0.0001, broadcasting period=20, and iteration of each primary user is 5. Iteration value means primary user must wait all secondary users run the iteration value times evolutionary algorithm. Because we want to compare our system with ordinary pricing-based model, we have two kinds of revenue and discount value. First one is r_A=r_B=d_A=d_B=0, it means that there is no discount and revenue for players. Because primary users have no broadcasting period in ordinary pricing-based system, the broadcasting period must be shared by buyers in ordinary pricing-based system. So in ordinary pricing-based system, $SharingPeriod_A$=1020 and $SharingPeriod_B$=920. Second kind is r_A=r_B=5, d_A=d_B=2. We want to use this kind of parameter to show our system can bring higher payoff with appropriate *r* and *d*. We use above parameters setting to simulate the transactions of primary users and secondary users according to our proposed algorithm. If payoff of primary user wants be higher than ordinary pricing-based model, the parameters in our system must forms the conditions as follows.

$$BN_i(n) \times (r_i - d_i) \geq price_i^{'} \times n_i^{'} - price_i \times n_i \quad \forall i \tag{16}$$

where *price$_i$'* and *n_i'* are price and number of buyers at equilibrium point in the ordinary pricing-based system.

Fig2. describes the number of secondary users choosing primary user A or primary user B with two kinds of parameters. The number of choosing A or choosing B will be influenced by utility of secondary user. The utility of secondary users in our proposed system will be influenced by average discount, so number of secondary users in second kind parameters will have a lot of variations. We can see that population of each primary users with two kinds of parameter will be fixed when secondary users evolve to evolutionary equilibrium finally.

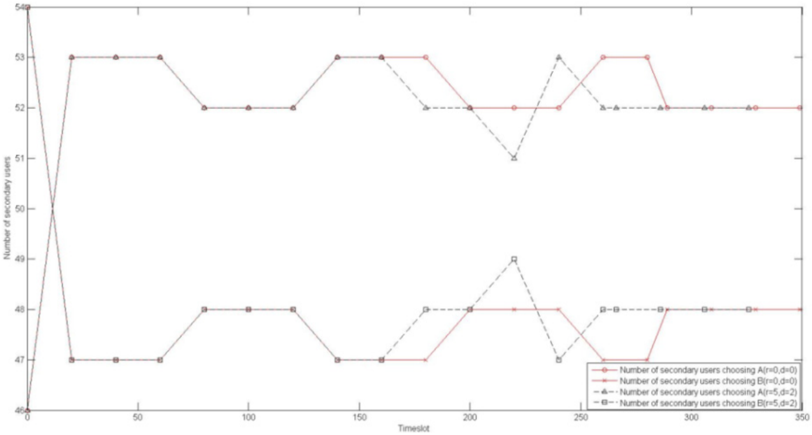


Fig. 2. The number of buyers with each primary users

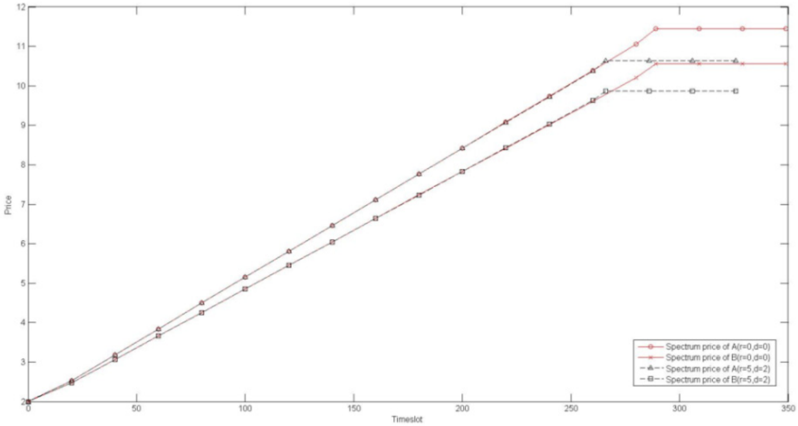


Fig. 3. The spectrum price of each primary user

Fig3. shows the price of each primary user changes with simulation time. Price will update at each iteration according to the marginal payoff of primary user. Primary user A and B will get to *Nash equilibrium* with time finally. At the equilibrium point, the price will bring primary user maximum payoff in the game. Though the price of our proposed system is lower than ordinary pricing-based model, the payoff of primary user will be higher than first kind of parameter by extra revenue.

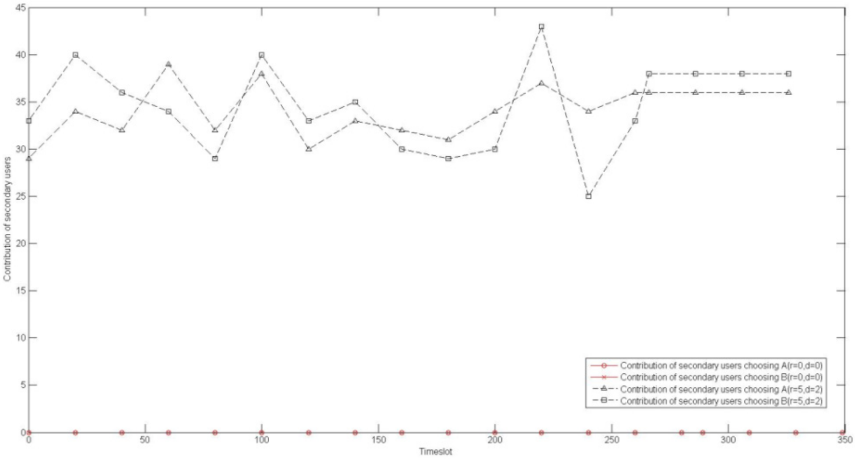


Fig. 4. The total contribution made by secondary users

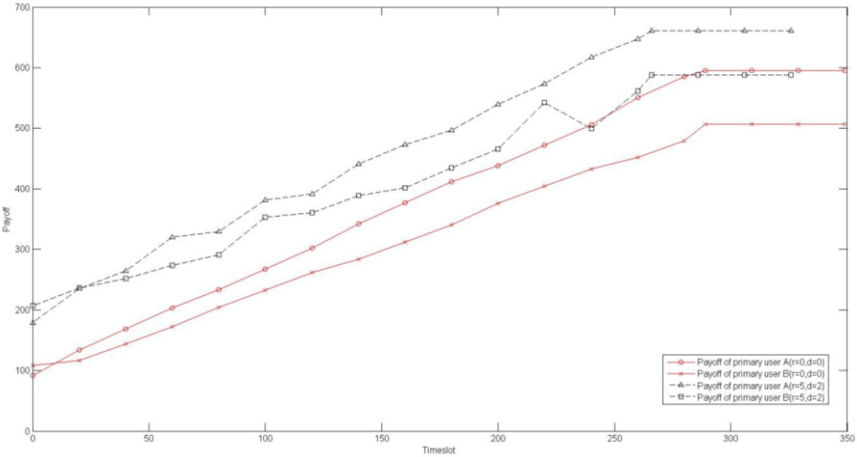


Fig. 5. The payoff of each primary user

Fig4. shows the important phenomena. The contributions(*BN*) are the essence of this paper. Because contribution means that secondary users broadcast information to how many nodes are uncovered by primary user without interference. Primary users may get higher payoff from contribution and secondary users may get higher utility from contribution too.

We can observe the payoff of each primary user in Fig5. with different kinds of parameters. In this figure, we want to show our system can give primary user higher payoff. But the revenue value and discount value must be set carefully. In second kind of parameter, the extra revenue are over the difference of spectrum selling between two kinds of parameters, so the payoff of primary users will be better than first kind.

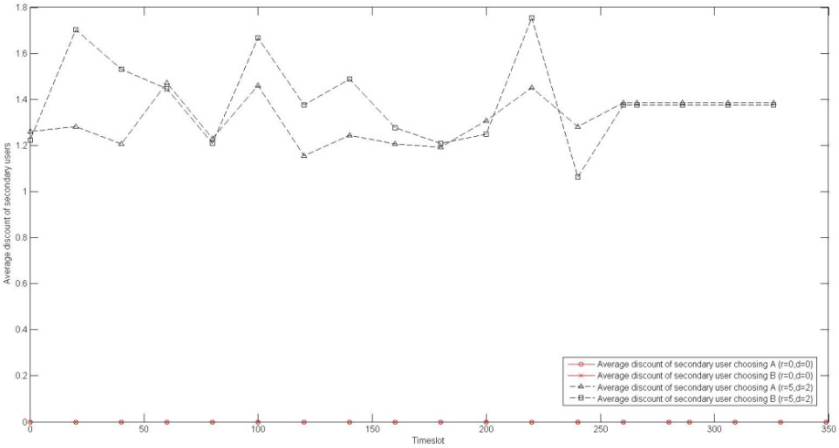


Fig. 6. The average discount for secondary users

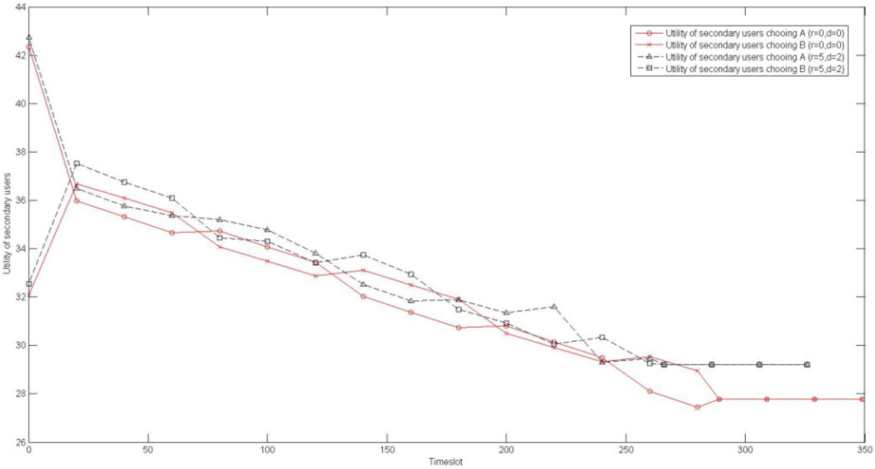


Fig. 7. The utility of secondary user

In Fig6.,we show the average discount of each secondary user can get in spectrum A and B. Secondary user has a will to buy the spectrum with broadcasting and discount, if he can get the average discount which is higher the sum of difference of price, transmission cost, and extra timeslots profits. The setting of value d is the work of primary user, if d is too small, none of secondary users has will to buy yours spectrum.

Evolutionary algorithm is the dominant in this paper. If secondary user wants to evolve, he must have the utility information. In Fig7., we present the utility of choosing primary user A and B with different kinds of parameters. With simulation time increasing, we can find that all secondary users' utility are close to each other until they have the same utility in each system. Improving utility of secondary users is one objective of this paper, and we see the utility of secondary users can be improved by appropriate *d* indeed.

5 Conclusion

In this paper, we propose a transaction system with concept of broadcasting for discount in TDMA based cognitive radio networks. The objective of our system is to improve the payoff of players from ordinary pricing-based model. We define the utility function of all users and elaborately describe the primary and secondary users' algorithm. We also depict the condition of evolutionary equilibrium and *Nash equilibrium*. Then, we describe why our system can give higher payoff than system only sells spectrum. At the end, we simulate the transaction of this system and compare with ordinary pricing-based model at the evolutionary equilibrium point. The simulation results show our system has better performance.

References

1. Niyato, D., Hossain, E., Han, Z.: Dynamics of Multiple-Seller and Multiple-Buyer Spectrum Trading in Cognitive Radio Networks: A Game-Theoretic Modeling Approach. IEEE Transaction on Mobile Computing 8(8), 1009–1022 (2009)
2. Niyato, D., Hossain, E.: Optimal Price Competion for Spectrum Sharing in Cognitive Radio: A Dynamic Game-Theoretic Approach. In: IEEE GLOBECOM, pp. 4625–4629 (2007)
3. Niyato, D., Hossain, E.: Competitve Spectrum Sharing in Cognitive Radio Networks: A Dynamic Game-Theoretic Approach. IEEE Transaction on Wireless Communications 7(7), 2651–2660 (2008)
4. Song, Q., Zhuang, J., Zhang, L.: Evolution Game Based Spectrum Allocation in Cognitive Radio Networks. In: EEE WiCOM, pp. 1–4 (2011)
5. Wang, H., Gao, L., Gan, X., Wang, X., Hossain, E.: Cooperative Spectrum Sharing in Cognitive Radio Networks: A Game-Theoretic Approach. In: IEEE ICC, pp. 1–5 (2010)
6. Wu, Z., Cheng, P., Wang, X., Gan, X., Yu, H., Wang, H.: Cooperative Spectrum Allocation for Cognitive radio Network: An Evolutionary Approach. In: IEEE ICC, pp. 1–5 (2011)
7. Jia, Y., Zhang, Z., Tan, X.: Spectrum Sharing Based on Duopoly Model in Cognitive Radio Networks. In: IEEE IMCCC, pp. 763–766 (2011)

A Neighbor-Aware Congestion Control Mechanism for Delay-Tolerant Networks

Shou-Chih Lo and Yu-Syuan Luo

Dept. of Computer Science and Information Engineering
National Dong Hwa University
Hualien 974, Taiwan ROC
sclo@mail.ndhu.edu.tw, m9921023@ems.ndhu.edu.tw

Abstract. Delay-Tolerant Networks (DTNs) are special types of network environments that are subject to delays and disruptions. Traditional end-to-end congestion control mechanisms are hard to be implemented in the DTNs due to frequent intermittent connections. In this paper, we propose a new hop-by-hop congestion control mechanism. This mechanism can be integrated into any quota-replication DTN routing protocols with a few modifications. Experimental results show the superiority of this kind of congestion control over existing types presently in use.

Keywords: DTN, Congestion Control, Routing Protocol, Quota Replication.

1 Introduction

Data communications using traditional Internet protocols cannot function normally in challenged networks such as interplanetary and deep-space networks in which communications are subject to delays and disruptions [1]. These challenged networks were generalized as Delay-Tolerant Networks (DTNs) by Kevin Fall in 2002 [2]. A DTN has representative properties such as high and variable latency, low data rate, frequent disconnection, and high error rate.

DTN related standards are studied by the Delay Tolerant Networking Research Group (DTNRG) [3]. DTN architecture was introduced in RFC 4838 [4], with a new bundle layer [5] added between the application layer and the transport layer. The bundle layer implements a store-and-forward mechanism, by which a node can store and carry data in its own buffer, and forward these data to other connected nodes when they are available. Data packets passing through the bundle layer are grouped into basic units called bundles or messages.

In DTNs, a stable end-to-end routing path between two nodes hardly exists, and this makes the design of a routing protocol more challenging. DTN routing protocols can be roughly classified into three categories based on the number of copies of the same message that are created: *forwarding*, *quota-replication*, and *flooding* [6]. In a forwarding scheme (such as MEED [7]), a single-copy message is forwarded through successive intermediate nodes to the destination. A quota-replication scheme (such as

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 591–600.
DOI: 10.1007/978-3-642-35452-6_60

Spray&Wait [8]) creates a specific number of copies (called *message quota*) for a message. A flooding scheme (such as Epidemic [9]) creates an extremely large number of copies and spreads them throughout the whole network. Generally, quota-replication schemes outperform the other two schemes in most cases.

However, the proper setting of the message quota becomes an important job for any quota-replication routing schemes. With a small quota, a message is hard to widely spread into a large network to reach the message destination. On the contrary, with a large quota, too many redundant message copies would occupy some resources like buffer space in each node. Moreover, a fixed setting scheme on quota values could not fit into any dynamic network with various traffic conditions.

In this paper, we improve any quota-replication routing protocols by introducing a mechanism to properly adjust the amount of message copies based on congestion conditions for a balance between high delivery rate and low drop rate. Our proposed mechanism can be integrated with any existing quota-replication routing with a few modifications.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of DTN congestion control. The proposed congestion control is presented in Section 3. Performance comparisons are presented in Section 4. Finally, we make some concluding remarks in Section 5.

2 Related Work

Traditional congestion control mechanisms such as flow control and sliding window in the TCP/IP protocol rely on the end-to-end conversation between the sender and the receiver. However, these mechanisms would work poor in the DTN environment which lacks for stable end-to-end communication. Nevertheless, some basic concepts of congestion control such as reducing traffic load, removing old or low-priority packets, and distributing traffic evenly into the network are still helpful. Due to the nature of using the store-and-forward mechanism in the DTN, each node has buffer space which would overflow when the network is congested. The control of messages in the buffer becomes a way to control network congestion. In the following, we introduce some these kinds of mechanisms.

Storage routing [10] uses message moving or pushing techniques to move messages from a node with full buffer space to another node with free buffer space. The moved messages can be selected based on the order that the message entered into the buffer (PushTail or PushHead policies) or the time the message was generated into the network (PushOldest policy). The PushOldest policy presented good performance after some simulation studies. If there are several neighboring nodes with free buffer space, the node with the lowest message moving cost and the largest available buffer space is selected. The messages are labeled with priority of low or high in [11], and high-priority messages have higher precedence to be pushed into one neighboring node. This action prevents high-priority messages being stuck in the buffer and hence being removed due to buffer overflow.

The proposed mechanism in [12] uses a local measurement to estimate the global congestion level. The local measurement is based on a ratio of the number of dropped messages to the number of message replications observed and accumulated by a node. The underlying routing is a quota-replication one, and then the quotas of messages in the buffer are decreased when the congestion level is growing and are increased otherwise.

A proactive mechanism is used in [13] to accept or reject messages to be entered into a buffer. Each message is associated with a price based on its priority and TTL (Time-To-Live), and this price implies the cost to route the message to the destination. A node can decide whether to accept or reject an incoming message from another node depending on whether this node has sufficient money to pay for the price of the message. The money hold by a node indicates the level of free buffer space.

Each node would identify itself into one of three states (normal, congestion adjacent, and congestion) in [14] according to its buffer usage. A node would inform its neighboring nodes with being enter into the last two states. Meanwhile, all neighboring nodes would avoid route any messages via this node. This action can prevent messages entering into a congested area.

3 Congestion Control

Here we introduce our proposed congestion control mechanism named NACC (Neighbor-Aware Congestion Control). NACC is modified from our previously proposed mechanism DCCR (Dynamic Congestion Control based Routing) [15]. NACC adopts a new control flow based on more information about surrounding nodes such as how many and what messages are stored in the buffer of these nodes. Some meta-data need to be exchanged when two nodes encounter. Then, a node would perform the congestion control according to the collected meta-data.

NACC is designed for the congestion control support for any quota-replication routing protocol. In quota-replication routing, each message generated from a source is associated with an initial quota. A node can conditionally duplicate a message in the buffer, and forward the duplicated message to another encountered node. Moreover, the duplicated message is allocated part of the quota from the original message based on a quota allocation function. Any message whose quota has been reached, and cannot be allocated any more, is therefore transmitted to the destination by direct contact. Our congestion control mechanism relies on the quota adjustment when a message is replicated hop-by-hop.

3.1 Meta-data Maintenance

There are several kinds of meta-data that need to be maintained in each node. We explain m-list, i-list, and BO first. Assume that each generated message in the network has a unique global ID (e.g., the combination form of the source node ID, the destination node ID, and a sequence number). The m-list summarizes the content of one buffer by listing message IDs within the buffer. By comparing the m-list records of two nodes, duplicated messages will not be transmitted. The i-list records the IDs of messages that are known to have reached their destinations. When a destination node successfully receives a message, this node adds a new record for this message into its i-list. Two contact nodes will exchange and merge their i-list records. The i-list is used to remove messages from the buffer that have become useless. The BO is the buffer occupancy in percentage of a node. A full buffer has a BO of 100%.

Moreover, each node should maintain three tables: MLT (Message List Table), NBOT (Neighbor Buffer Occupancy Table) and CPT (Contact Probability Table). MLT and NBOT record the m-list and the BO of each neighboring node in the communication range, respectively. When a neighboring node has gone beyond the communication range, the corresponding entries in MLT and NBOT are deleted. The CPT records the contact probability (between 0 and 1) of each node that has been encountered before and is updated each time any nodes are encountered.

Suppose that a node records a contact probability with node V by the notation CP_V. When this node encounters a node, CP_V is updated from an old value ($CP_{V,old}$) to a new one ($CP_{V,new}$) as in (1). That is, we increase the contact probability of the current encounter node and decrease the contact probability of each other non-encounter node (aging effect) in CPT.

$$CP_{V,new} = \begin{cases} CP_{V,old} \times (1-\alpha) + \alpha, & \text{If the current encounter node is node V.} \\ CP_{V,old} \times (1-\alpha), & \text{Otherwise.} \end{cases} \tag{1}$$

For an example shown in Fig. 1, suppose that we set $\alpha = 0.6$. At the beginning, node D contacts nodes A and B which have never been seen before. Each node (i.e., nodes A, B, and D) adds a new entry with an initial value of 0.6 for each newly contacted neighbor in its CPT. Next, nodes C and D encounter each other, and a new entry with an initial value of 0.6 is added correspondingly. Moreover, node D updates the two old entries (CP_A and CP_B) with a new value of 0.6×0.4. Then, nodes A and B encounter each other, and they follow a similar action with node D, as described above, to add one new entry and update one old entry in their CPTs.

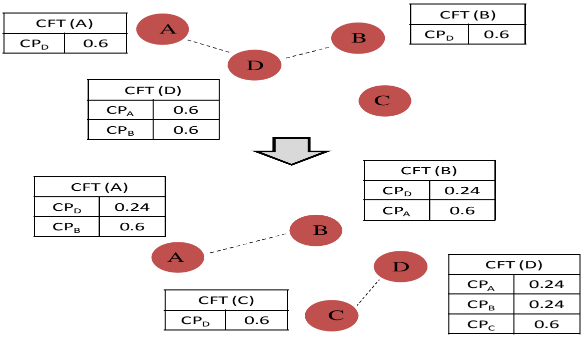


Fig. 1. Computing contact probability

Each time when two nodes contact, they exchange the following meta-data: m-list, i-list, BO, and CPT. After the exchange, these two nodes refresh their tables MLT and NBOT. Then, a subsequent congestion control is applied.

3.2 Quota Adjustment

As mentioned before, our proposed congestion control is based on the quota adjustment for each being transmitted message between two nodes. NACC refers to two network states observed by a node to adjust the quota: average buffer occupancy and idle state. The *Average Buffer Occupancy* (ABO) is the average of values in NBOT, which can imply the traffic condition in a local area. The idle state indicates that a node has no more routing jobs for messages in the buffer.

If one node is in *idle state*, one of the following two conditions should be satisfied:

(1) All messages in the buffer have a remaining quota of 1.
(2) All messages having a remaining quota greater than 1 in the buffer can be found in each record in MLT.

Condition (1) implies that all messages in the buffer cannot be replicated anymore and can only be directly forwarded to destination nodes. Condition (2) implies that all messages in the buffer that can be further replicated have been held by all neighboring nodes. Both conditions show that the idle-state node needs not perform routing for messages in the buffer.

The control flow of NACC is shown in Fig. 2. A node always waits for a contact event with another node at the beginning. Two contact nodes will exchange meta-data as introduced before and compute their own ABO values. Then, the idle state is examined. If a node is not in idle state, the control flow goes to the left part of the figure. Next, we check whether the current ABO is too high, which indicates the local network is congested, and is greater than a threshold (TH_A). If that is the case, the quota values of some messages might be decreased accordingly. Each message in the buffer is examined to find any deliverable message with quota great than 1. A message is called *deliverable* if this message is not currently held by the encounter node. Next, we decrease the quota of this selected message by Eq. (2). Denote the quota of message *m* by $Quota_m$. The current number of neighboring nodes is *NB*. Among these neighboring nodes, the number of nodes that have also held message *m* is NB_m. Eq. (2) implies that the quota of a message is decreased more if more neighboring nodes have held this message.

$$\mathrm{Q}uota_{\mathrm{m}} = MAX\{2, Quota_m * (1 - \frac{NB_m}{NB})\} \tag{2}$$

After the examination of each message in the buffer, the underlying routing protocol is then performed. If the local network is not congested, all quota values in the buffer remain unchanged and then the routing is performed directly.

If the node is in idle state, the control flow goes to the right part of the figure. First, we check whether the local network is congested and suitable for increasing the quota

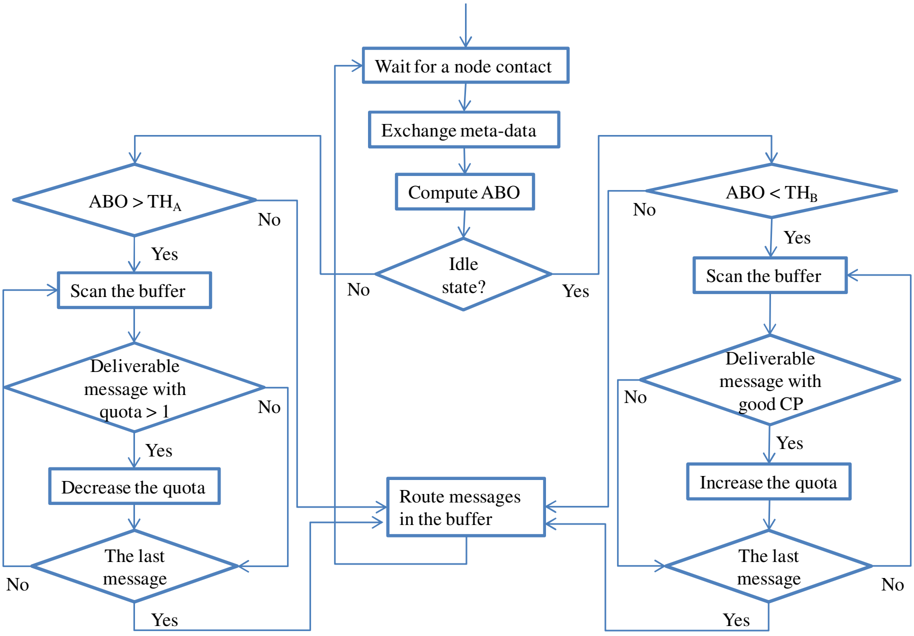


Fig. 2. Processing flow of DCCR

or not. Another threshold (TH$_B$) is used here to examine the current ABO. Note that we set TH$_A$ > TH$_B$, and let the constraint of increasing quota more strict. If the local network is not congested, we select deliverable messages in the buffer and try to increase their quotas. To make the message routing along with a good path, only the quota of the message for which the current encounter node has a higher contact probability (CP) with the message destination than the current message holding node is increased by one. After the examination of each message in the buffer, the underlying routing protocol is then performed. If the local network is congested, the routing is performed directly.

4 Performance Evaluation

We evaluate performances using the Opportunistic Network Environment (ONE) simulator [16]. ONE is a JAVA-based open-source software, which has provided implementation codes for some routing protocols. Two trace files which records contact events in 2005 and 2006 INFOCOM conferences are used [17]. To capture purely social contacts, we have removed all external nodes from the files and have left internal nodes only.

The transmission bandwidth of any communication link is 250 kBps. Messages of a fixed size of 500 kB are generated with a fixed time gap (generation interval). The source node and the destination node of a generated message are randomly selected from these internal nodes. The parameter settings are listed in Table 1.

We mainly compare NACC with four other congestion controls: SR[10], RR [12], PA [14], and DCCR [15]. The underlying routing protocol of these congestion controls is the same and is based on Spray&Wait (abbreviated as SnW) [8].

Table 1. Parameter Settings

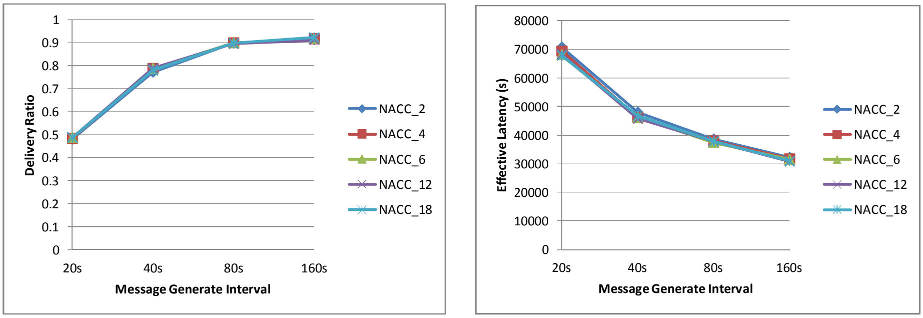
Parameter	Value
Simulation time	254150 s (05's), 337418 s (06's)
Number of nodes	41 (05's), 98 (06's)
Buffer size	16 MB
Transmission rate	250 kBps
Message size	500 kB
Message generate interval	20 s, 40 s, 80 s, 160 s
Message generate period	20000 s to 150000 s (05's), 10000 s to 200000 s (06's)
Initial quota	12
TH_A	50
TH_B	40
α	0.6

Three cost metrics are used in our evaluation, as described below:

Delivery ratio: ratio of the total number of messages received at destinations to the total number of messages sent from sources.

Average relay cost: ratio of the total message relay count to the total number of messages received at destinations. The relay count is increased by one as any message is forwarded or copied from one node to another node in the network.

Effective latency: ratio of the average end-to-end delay for a message from the source to the destination to the delivery ratio.



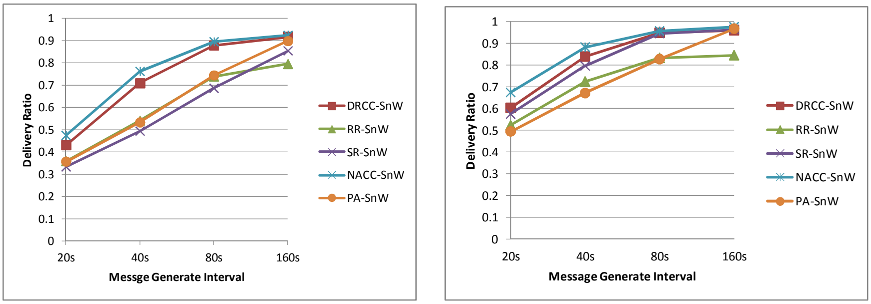
(a) Delivery ratio

(b) Effective latency

Fig. 3. Effect of initial message quota

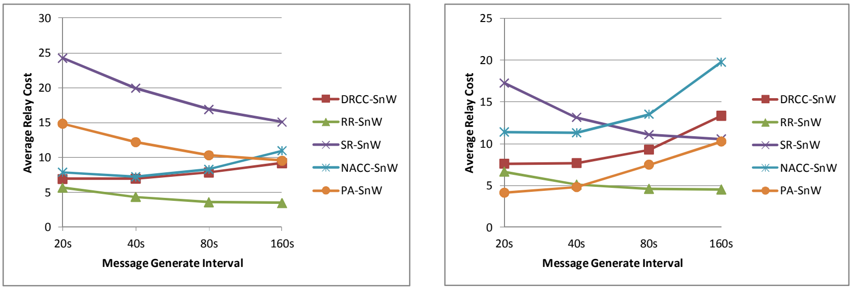
At first, we observe whether the setting of the initial message quota is still a concerned issue after adding our proposed dynamic quota adjustment. This experiment is performed using the trace file of INFOCOM05. As shown in Figs. 3(a) and 3(b), the delivery ratio and the effective latency change small if different initial quotas are used. This concludes that the quota setting becomes less sensitive in our method, and this also shows that our proposed method is more practical.

We then started comparing NACC with other congestion controls. In Figs. 4(a) and 4(b), NACC performs better than DCCR (our previous approach) and largely improves delivery ratio in comparison to other methods, particularly when there are many messages are generated. Although SR, RR, and PA can improve delivery ratio, it was found that they perform worse than NACC.



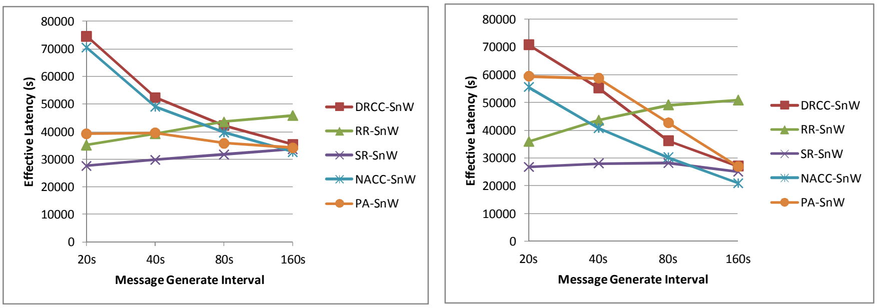
(a) INFOCOM05 (b) INFOCOM06

Fig. 4. Delivery ratio vs. Message interval



(a) INFOCOM05 (b) INFOCOM06

Fig. 5. Relay cost vs. Message interval



(a) INFOCOM05 (b) INFOCOM06

Fig. 6. Effective latency vs. Message interval

NACC incurs a moderate relay cost in heavy traffic load as shown in Figs. 5(a) and 5(b). When the message generation interval is large, the network is lightly loaded, and NACC generates many copies of messages, so the relay cost remains high but latency stays low as shown in Figs. 6(a) and 6(b).

5 Conclusions

Intermittent connections make it difficult for messages to reach their destinations in DTNs. Network congestion creates a situation whereby a DTN is even more hindered to deliver any messages. In this paper, we use quota-replication routing to generate many copies of messages to increase their probability of reaching their destinations successfully, and also control the amount of copies generated when the network becomes congested. A series of strategies are adopted to alleviate congestion of different degrees. Performance evaluation shows that our proposed mechanism outperforms other congestion controls.

References

1. McMahon, A., Farrell, S.: Delay- and Disruption-Tolerant Networking. IEEE Internet Computing 13(6), 82–87 (2009)
2. Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets. In: Proc. ACM SIGCOMM Conf., pp. 27–34 (August 2003)
3. Delay-Tolerant Networking Research Group (DTNRG), http://www.dtnrg.org
4. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Weiss, H.: Delay-Tolerant Networking Architecture. IETF RFC 4838 (April 2007)
5. Scott, K., Burleigh, S.: Bundle Protocol Specification. IETF RFC 5050 (November 2007)
6. Lo, S.C., Chiang, M.H., Liou, J.H., Gao, J.S.: Routing and Buffering Strategies in Delay-Tolerant Networks: Survey and Evaluation. In: Proc. IEEE ICPP Workshop (September 2011)
7. Jones, E.P.C., Li, L., Ward, P.A.S.: Practical Routing in Delay-Tolerant Networks. IEEE Trans. Mobile Computing 6(8), 943–959 (2007)
8. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In: Proc. ACM SIGCOMM Workshop, pp. 252–259 (August 2005)
9. Vahdat, A., Becker, D.: Epidemic Routing for Partially-Connected Ad Hoc Networks, Duke University. Technical Report CS-200006 (April 2000)
10. Seligman, M., et al.: Storage routing for DTN congestion control. Wireless Communications and Mobile Computing 7(1), 1183–1196 (2007)
11. Cao, Y., Cruickshank, H., Sun, Z.: Active Congestion Control Based Routing for Opportunistic Delay Tolerant Networks. In: Proc. Vehicular Technology Conference (2011)
12. Thompson, N., Nelson, S.C., Bakht, M., Abdelzaher, T., Kravets, R.: Retiring Replicants: Congestion Control for Intermittently-Connected Networks. In: Proc. IEEE INFOCOM, pp. 1–9 (2010)

13. Das, R., Prodhan, M.A.T., Kabir, M.H., Shoja, G.C.: A Novel Congestion Control Scheme for Delay Tolerant Networks. In: Proc. Intl. Conf. on Selected Topics in Mobile and Wireless Networking (2011)
14. Hua, D., Du, X., Cao, L., Xu, G., Qian, Y.: A DTN Congestion Avoidance Strategy Based on Path Avoidance. In: Proc. Intl. Conf. on Future Computer and Communication (2010)
15. Lo, S.C., Lu, C.L.: A Dynamic Congestion Control based Routing for Delay-Tolerant Networks. In: Proc. Intl. Conf. on Fuzzy Systems and Knowledge Discovery, pp. 2047–2051 (May 2012)
16. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE Simulator for DTN Protocol Evaluation. In: Proc. Intl. Conf. Simulation Tools and Techniques (March 2009)
17. The Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD), `http://uk.crawdad.org/index.php`

A Cluster-Based Link Recovery Mechanism for Spectrum Aware On-Demand Routing in Cognitive Radio Ad Hoc Networks

Chih-Chieh Tang, Kuo-Feng Ssu, and Chun-Hao Yang

Institute of Computer and Communication Engineering
National Cheng Kung University
Tainan, Taiwan
andy12125555@gmail.com, ssu@ee.ncku.edu.tw, jeffy@dcl.ee.ncku.edu.tw

Abstract. In cognitive radio ad hoc networks (CRAHNs), link failures among secondary users (SUs) primarily result from sudden appearance of primary users (PUs). In tradition, route recovery generates much packet loss before the routes are recovered completely. Recent researches on route recovery process in CRAHNs mainly adopt on-demand route maintenance protocols in wireless networks. This paper proposes a cluster-based link recovery mechanism (CLR) in CRAHNs. CLR recovers failed links locally and does not change the original routes. Since the routes have not been changed, the data packets are able to be cached in the buffer. Data packets can be sent after the links are recovered and hence decreases the amount of packet loss. If there is an intersection area of all PUs' radio coverage in the network with some SUs existence in the area, links between these SUs are apt to break no matter what channels these SUs choose. CLR recovers all link failures by using clustering method and effectively increases end-to-end throughput and successful packet delivery fraction. The simulation results indicate that CLR increases successful packet delivery ratio and end-to-end throughput. The performance is much superior particularly if there is an intersection area of all PUs' radio coverage in the network.

Keywords: cognitive radio networks, cluster, routing protocol.

1 Introduction

The traditional wireless ad hoc networks are regulated by a fixed spectrum assignment policy. In recent years, wireless devices and applications are well developed and the demand on spectrum resources has increased rapidly. The limited spectrum resources are not enough to share among more and more applications. Therefore, the efficiency of spectrum utilization becomes a huge issue. According to the report of FCC [1], the utilization of licensed spectrums ranges from 15% to 85%. It indicates that the efficiency of spectrum utilization is not well. To increase the efficiency of spectrum utilization, the cognitive radio network (CRN) is then proposed [2]. CRN consists of primary users (PUs) and secondary users

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 601–610.
DOI: 10.1007/978-3-642-35452-6_61

(SUs). PUs are the licensed users and have full authority to access their licensed spectrums. In contrast, SUs can access the spectrum only when PUs does not occupy the spectrum. SUs' activities can not interfere with PUs' transmission. Therefore, SUs equip the cognitive radio devices and have ability to sense spectrum utilization to find available free spectrums. Once PU is activated in a licensed spectrum, SUs have to relinquish the spectrum immediately and find other available spectrums.

Routing is one of the main issues in CRAHNs [3]. Due to distinct locations, each SU has different available channels at any given time. High variation of channel conditions results in frequent link failures. Researches in CRAHNs directly inherit from the processes of on-demand routing protocols such as AODV [16] and DSR [17]. These papers directly inherit from the process of AODV and result in high packet loss during the route discovery [4–7]. In CRAHNs, SU has to be able to detect link failure and reassign an available channel. The actual topology of CRNs is drastically affected by PUs' behavior since the sudden appearance of PUs may result in the unavailability of a given channel. In this scenario, frequent route reestablishment is required to recover failed routes with minimal effect on the perceived quality.

To overcome the severe environment where exists an intersection area of all PUs' radio coverage, a cluster-based recovery mechanism (CLR) has been proposed. When link failure occurs, link recovery is performed locally instead of reestablishing the whole route. SUs which is in the intersection area of all PUs' radio coverage form a cluster. The cluster head chooses an available channel and assigns it as communication channel to all cluster members. All links among cluster members are recovered identically.

2 Related Work

Many researches about routing protocol in CRAHNs are proposed in last few years. Some approaches are based on centralized algorithms to accomplish the optimal entire network performance [8, 9]. Those proactive approaches are based on previously-known global information, which is hard to obtain in CRAHNs. On-demand routing protocols are reactively make route decision and channel assignment simultaneously [11–13]. Those approaches use different metrics to measure the quality of routes. SER takes energy consumption into account when a SU attempt to find a route [10]. When destination node receives route request packet (RREQ) from different routes, SER can select the most energy-efficient route by considering maximal minimum residual nodal energy with lower hop count. SEARCH is designed for mobile CRAHNs based on the geographic forwarding principles [5]. It jointly undertakes route and channel selection to avoid regions of PU activity during the route construction. SEARCH discovers several routes from source node to destination node, and these routes are joined at the destination node to form low hop count routes.

Channel switching delay time is another significant factor of measuring the routes. Several papers provide formulation of channel switching delay and backoff

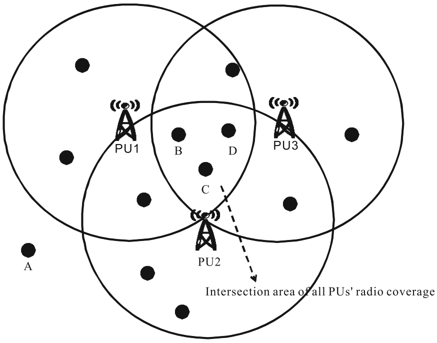


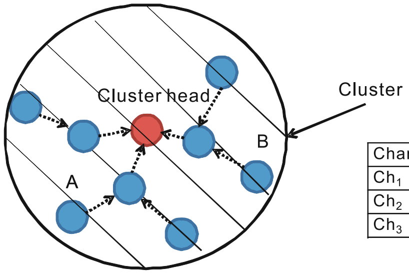
Fig. 1. Network model

delay [6, 7]. The channel switching delay occurs when a node switches from one frequency to another. When several nodes contend for the same channel, the backoff procedure begins and brings the backoff delay to them all. Relay-assisted Routing (RAR) defines relay aware link cost, which considers channel availability, channel condition, channel utilization and potential relays [13]. RAR chooses the route with minimum value of cumulative path cost.

The aforementioned routing protocols do not focus on the part of route recovery. A research gives attention to link restoration in CRNs [14]. The authors formulates the link restoration problem as an integer programming problem. A backup channel is prepared for each link. In their model, only one type of PUs shows up at a time, which means only one channel will be unavailable at a time. The assumption may not be reasonable in practice since many types of PUs may appear at the same time. Besides, the currently used channel and the backup channel may be unavailable simultaneously.

3 System Model

This paper considers a CRAHN consists of stationary PUs and SUs. The PU_i is the license holder for the $channel_i$. For each channel, there is exactly one PU. When PUs use their own licensed channels, they are in the active state. Otherwise, they are in the sleep state. SUs can only utilize the $channel_i$ when the PU_i is sleeping. SUs have ability to know whether they can use the channel or not by spectrum sensing capacity. Each SU has the same transmission and interference range. A SU has one radio for both data and control packets. There are N data channels and *one* common control channel (CCC). Furthermore, each SU has ability to estimate the presence probability of PUs via statistical utilization of channels or environment learning [15]. Due to sensing errors naturally caused by hardware devices, there are slight differences of the presence probability of PUs between each SU. By choosing the channel with the least interference from PUs, SU constructs the most robust links for data transmission.



Cluster head

Channel	$P_{presence}$	Available
Ch_1	P_{head1}	YES
Ch_2	P_{head2}	YES
Ch_3	P_{head3}	YES

A

Channel	$P_{presence}$	Available
Ch_1	P_{a1}	YES
Ch_2	P_{a2}	YES
Ch_3	P_{a3}	YES

B

Channel	$P_{presence}$	Available
Ch_1	P_{b1}	YES
Ch_2	P_{b2}	YES
Ch_3	P_{b3}	YES

Fig. 2. Channel lists of cluster members

In a CRAHN, there is probably an intersection area of all PUs' radio coverage. This intersection area is called the area $\mathcal{A}$ for simplicity. In $\mathcal{A}$, no matter what channels SUs choose, links between SUs are apt to break. As Figure 1 shows, for user B, C and D, the data transmission between them will be interrupted when the given PU is active. For other SUs, there exists some available channels without PU's interference. Since communication links in $\mathcal{A}$ are regarded as more fragile links than others, this paper focus on this area.

4 Routing Protocols

4.1 Initialization

In initialization phase, this protocol checks whether an intersection area $\mathcal{A}$ of all PUs exists in the network. If a SU is covered by all PUs, it is regarded itself as a cluster member. In this paper, the SUs which are covered by all PUs are called the *member SU*s. The member SU with lowest ID becomes cluster head. The neighboring member SUs rebroadcast the clustering packet and transmit a clustering ACK with their presence probability of PUs to the cluster head. The clustering packet are relayed until the packet reaches the non-member SUs. The cluster head received clustering ACKs with the presence probability of PUs information from all member SUs, as shown in Figure 2. This cluster head computes the average presence probability of PUs of all member SUs for the later use of channel assignment.

4.2 Route Discovery

Route discovery begins when a SU attempts to transmit data packets to another SU and no available route is found. A sender S broadcasts a RREQ via CCC to find a route to reach destination D, and looks forward to a RREP sent back from

Algorithm 1. Algorithm of communication channel assignment

```
Require:
  $X$: Sender. $Y$: Receiver.
  $Mem$: Member SUs.
  $CH_{cluster}$: Communication channel among the member SUs.
  $CH_{(X,Y)}$: Communication channel between sender and receiver.
  $AC_{(X,Y)}$: A set of all common available channels between $X$ and $Y$.
  $TC_{(X,Y)}$: A set of all common available channels of which utilization are under a
  given threshold between $X$ and $Y$. $TC_{(X,Y)} \subseteq AC_{(X,Y)}$.
Ensure:
  if $X$ is $Mem$ or $Y$ is $Mem$ then
    $CH_{(X,Y)} \leftarrow CH_{cluster}$.
  else
    Find a suitable common available channel between the sender and the receiver.
    for each $channel_i$ do
      if $channel_i$ is available for sender and receiver then
        $channel_i \cup AC_{(X,Y)}$.
        if utilization of $channel_i$ is lower than a given threshold then
          $channel_i \cup TC_{(X,Y)}$.
        end if
      end if
    end for
    if $TC_{(X,Y)} \notin \emptyset$ then
      $CH_{(X,Y)} \leftarrow c \in TC_{(X,Y)}$. c is a channel randomly selected from $TC_{(X,Y)}$.
    else if $AC_{(X,Y)} \notin \emptyset$ then
      $CH_{(X,Y)} \leftarrow c \in AC_{(X,Y)}$. c is a channel randomly selected from $AC_{(X,Y)}$.
    else
      $CH_{(X,Y)} \leftarrow 0$;
    end if
  end if
```

D. When an intermediate SU Y receives a RREQ from another intermediate SU X, Y updates the exiting reverse route table. Y computes the communication channel to communicate with X and stores it in the table. In this way, Y can find a channel to communicate with the previous SU X in the route from S to D. If there is no common available channel between X and Y, Y will drop the RREQ. Otherwise, Y rebroadcasts the RREQ. When the destination D receives the RREQ from S, D stores the channel information in the communication channel field of RREP packet. D unicasts the RREP to the SU which sends the corresponding RREQ. Suppose the previous SU in the route is Z. When Z receives the RREP, it creates a forward route table or updates the exiting forward route table. If Z is not the source, it replaces the communication channel field with the communication channel stored in the corresponding reverse route table. The RREP is forwarded until it is received by S. In short, each intermediate SU maintains a forward route table and a reverse route table. Thus, S has an available route to transmit data packets to D.

4.3 Channel Assignment

The algorithm of non-cluster-head channel assignment is given in Algorithm 1. If either the sender X or the receiver Y is a member SU, $CH_{(X,Y)}$ is assigned as $CH_{cluster}$. Otherwise, the receiver uses its channel list and the sender's channel list obtained from the field of RREQ to select a suitable common available channel. At first, a set $AC_{(X,Y)}$ of all common available channels is selected. Then in order to increase channel utilization, common available channels with low utilization have high priority to be used. Therefore, if the presence probability of PU_i is lower than a given threshold, the $channel_i$ will also be included in another set, $TC_{(X,Y)}$. The set $TC_{(X,Y)}$ is a subset of $AC_{(X,Y)}$. $TC_{(X,Y)}$ includes all common available channels that are under a quality threshold. If $TC_{(X,Y)}$ is not an empty set, a channel is randomly selected from $TC_{(X,Y)}$; otherwise, a channel is randomly selected from $AC_{(X,Y)}$. This channel is then assigned to be $CH_{(X,Y)}$. Unfortunately, if $AC_{(X,Y)}$ is an empty set, it means that there is no common available channel between X and Y. In this case, the receiver has to drop the RREQ and waits for a while.

Algorithm 2. Algorithm of CLR

```
Require:
  $X$: Sender. $Y$: Receiver.
  $Mem$: Member SUs.
  $CH_{cluster}$: Communication channel among the member SUs.
  $CH_{assign}$: Channel assignment packet from cluster head.
  A(): Channel assignment process.
Ensure:
  When PU is active, link failures happen.
  if $X$ is $Mem$ then
    Cache data packets in the buffer and waits for $CH_{assign}$.
    When it receives $CH_{assign}$, all outgoing links are recovered.
    $\forall Y$, $CH_{(X,Y)} \leftarrow CH_{cluster}$.
  else
    Send CHREQ to $Y$ and caches data packets in the buffer and .
    Receive CHREP from $Y$.
    $CH_{(X,Y)} \leftarrow$ return value of A().
    Send CHAssign to $Y$.
    Receive recovery ACK from $Y$.
    Link between $X$ and $Y$ is recovered.
  end if
```

5 Cluster-Based Link Recovery

CLR recovers broken links locally. When a PU_i is active, the process of CLR have chances to be invoked by the interfered SUs. If there is an area $\mathcal{A}$ in the network, all SUs within the area $\mathcal{A}$ constructs a cluster. The link recovery method of the member SUs is different from the one of the non-member SUs. The member SU link recovery is given in Algorithm 2. The cluster head is considered as a commander to make the channel decision for all member SUs. When PU_i is active and the communication channel among the member SUs is occupied, SUs cache data packets in the buffer and wait for a new channel assignment. The cluster head maintains the cluster channel table and uses the table to do channel assignment. The channel with the lowest average presence probability of PU is denoted as $CH_{cluster}$. The cluster head calculates and floods a channel assignment packets with $CH_{cluster}$ information to all member SUs. Thereafter, as long as $CH_{cluster}$ is occupies by PU, the cluster head performs the process again to compute a new $CH_{cluster}$. The average value of the presence probability can be calculated simply as $P_{i_{avg}} = \frac{\sum_{j=1}^{N} P_{ji}}{N}$, N is the number of the member SUs. . Therefore, all broken links in the area $\mathcal{A}$ will be recovered synchronously.

On the other hand, if the SU is a non-member SU, it caches data packets in the buffer and sends a channel request packet (CHREQ) to the receiver via CCC. The receiver receives CHREQ and sends a channel reply packet (CHREP) back to the sender. The CHREP contains the channel list of the receiver. The sender performs the channel assignment process to select a new available communication channel. The sender changes its channel field of forward route table and sends the channel assignment packet (CHAssign) to the receiver. The receiver changes its channel field of reverse route table and sends a recovery ACK back to the sender. At last, the sender sends all data packets to the receiver from its buffer.

6 Simulation Results

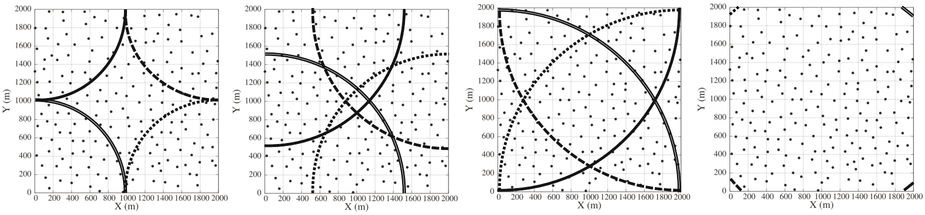
The performance of CLR is evaluated via NS2 [18], based on the CRCN simulator [19]. 200 SUs are randomly deployed. The number of data channels equals the number of PUs. The presence probability of PUs are randomly selected following normal distribution with mean value 70% and standard deviation 5. The active and idle time duration of PUs are randomly selected following normal distribution with mean value 5 and 2 seconds, and standard deviation 2.5 and 2, respectively. The parameter values in the simulations are listed in Table 1. The metric of successful packet delivery fraction, the aggregate end-to-end throughput, and the end-to-end packet delay are measured.

6.1 Coverage Range of PUs

There are 4 PUs respectively located at each corner of the network area. The coverage range of PUs varies from $1000m$ to $2800m$. As Figure 3 shows, there is no area $\mathcal{A}$ when the coverage range is $1000m$. The packet delivery fraction and the end-to-end throughput of CLR are slightly higher than that of AODV. When the coverage range is over $1500m$, an area $\mathcal{A}$ appears in the center of network. Figure 4 shows the successful packet delivery fraction of CLR is average 15% higher than AODV. The aggregate end-to-end throughput of CLR is average 68000 bps greater than AODV. As the coverage range is $2800m$, all SUs are

Table 1. Parameter Values in Simulations

Parameter	Value
Simulation area	$2000 \times 2000m^2$
Number of SUs	200
Transmission range of SU	$250m$
Interference range of SU	$550m$
Bit rate of CBR	50 $kbits/sec$
Size of data packet	4096 $bits$
Threshold of channel utilization	80%



(a) $1000m$. (b) $1500m$. (c) $2000m$. (d) $2800m$.

Fig. 3. Coverage range of PU in simulations

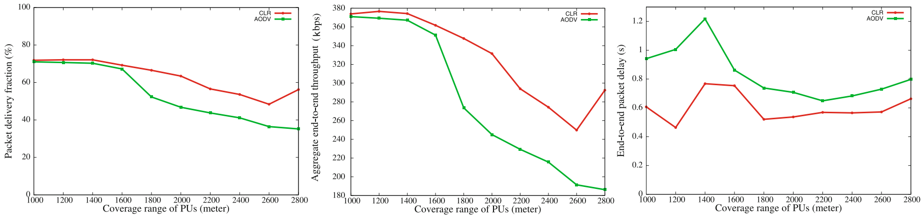
covered by all PUs. All failed links will be recovered by a channel assignment from cluster head. The delivery ratio and the throughput are better than the case where the range is set to 2600 or 2400. The end-to-end packet delay of CLR is average 0.23 seconds lower compared with AODV.

6.2 Number of PUs

The coverage range of PUs is 2000m. Each PU is randomly distributed. When there is an only PU in the network, once the PU occupies its own licensed channel, all links among SUs in the network will be failed. As Figure 5 shows, the successful packet delivery fraction of CLR is lower than 40%. The packet delivery ratio of AODV is 30% lower than CLR. More PUs means more licensed channels which can be used by SUs. Therefore, the packet delivery ratio of CLR is average 12% higher than AODV. The aggregate end-to-end throughput of CLR is average 60800 bps more than that of AODV. And the end-to-end packet delay of CLR is average 0.1 seconds lower than AODV.

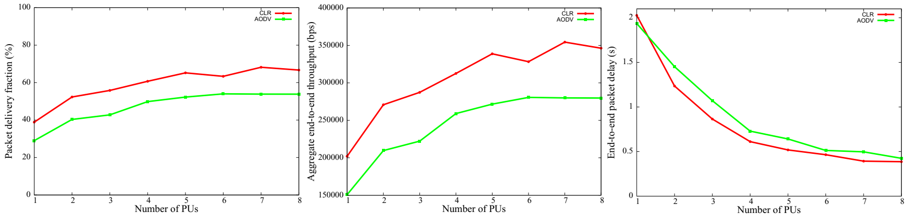
6.3 Number of Flows

The coverage range of PUs is 2000m as given in Figure 6. With more data flows, the difference of aggregate end-to-end throughput between AODV and CLR is larger. It indicates that CLR outperforms AODV in terms of throughput especially when the traffic is more congested.



(a) Delivery ratio. (b) End-to-end throughput. (c) End-to-end delay.

Fig. 4. Packet delivery fraction with different coverage range of PUs



(a) Delivery ratio. (b) End-to-end throughput. (c) End-to-end delay.

Fig. 5. Packet delivery fraction with different number of PUs

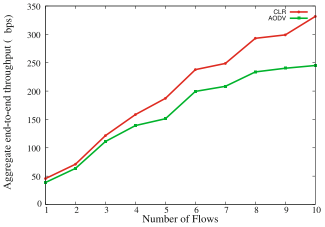


Fig. 6. Aggregate end-to-end throughput with different number of flows

7 Conclusion

The communication between SUs is interfered by PUs in CRAHNs, the route recovery mechanism has great influence in protocol efficiency. Link failures among SUs are caused by the presence of PUs, not by the movement of SUs. This paper proposes a cluster-based link recovery mechanism (CLR) to improve the method of route recovery. CLR recovers failed links locally and does not reestablish the original routes. Moreover, if there is an intersection area of all PUs' radio coverage in the network, CLR can recover link failures effectively by using clustering method. The successful packet delivery fraction and the aggregate end-to-end throughput effectively increases by using CLR. In a scenario where the whole network is covered by all PUs, the clustering method outperforms AODV. CLR provides higher throughput as traffic is more congested.

References

1. Spectrum Policy Task Force. Federal Communications Commission ET Docket (November 2002)
2. Akyildiz, I.F., Lee, W.Y., Vuran, M.C., Mohanty, S.: Next Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. Computer Networks 50(13), 2127–2159 (2006)
3. Akyildiz, I.F., Lee, W.Y., Chowdhury, K.R.: Ad Hoc Networks, vol. 7(5), pp. 810–836 (2009)
4. Sampath, A., Yang, L., Cao, L., Zheng, H., Zhao, B.Y.: High Throughput Spectrum Aware Routing for Cognitive Radio Networks. In: International Conference on Cognitive Radio Oriented Wireless Networks and Communications (May 2008)
5. Chowdhury, K.R., Felice, M.D.: SEARCH: A Routing Protocol for Mobile Cognitive Radio Ad Hoc Networks. In: IEEE Sarnoff Symposium Comm., pp. 1–6 (April 2009)
6. Cheng, G., Liu, W., Li, Y., Cheng, W.: Spectrum Aware On-demand Routing in Cognitive Radio Networks. In: IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, pp. 571–574 (April 2007)
7. Ma, H., Zheng, L., Ma, X., Luo, Y.: Spectrum Aware Routing for Multi-hop Cognitive Radio Networks with a Single Transceiver. In: International Conference on Cognitive Radio Oriented Wireless Networks and Communications, pp. 1–6 (May 2008)
8. Wang, Q., Zheng, H.: Route and Spectrum Selection in Dynamic Spectrum Networks. In: IEEE Consumer Communications and Networking Conference, pp. 625–629 (January2006)
9. Alicherry, M., Bhatia, R., Li, L.E.: Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks. IEEE Journal on Selected Areas in Communications 24(11), 1960–1971 (2006)
10. Kamruzzaman, S.M., Kim, E., Jeong, D.G.: Spectrum and Energy Aware Routing Protocol for Cognitive Radio Ad Hoc Networks. In: IEEE International Conference on Communications, pp. 5–9 (June 2011)
11. Salvo, P., Cuomo, F., Abbagnale, A.: Hidden Primary User Awareness in Cognitive Radio Routing: The SBBO Protocol. In: IEEE Global Telecommunications Conference, pp. 1–5 (December 2010)
12. Beltagy, I., Youssef, M., El-Derini, M.: A New Routing Metric and Protocol for Multipath Routing in Cognitive Networks. In: IEEE Wireless Communications and Networking Conference, pp. 974–979 (March 2011)
13. Jia, J., Zhang, J., Zhang, Q.: Relay-assisted Routing in Cognitive Radio Networks. In: IEEE International Conference on Communications, pp. 1–5 (June 2009)
14. Li, K.-F., Lau, W.-C., Yue, O.-C.: Link Restoration in Cognitive Radio Networks. In: IEEE International Conference on Communications, pp. 371–376 (May 2008)
15. Shih, C.F., Liao, W.: Exploiting Route Robustness in Joint Routing and Spectrum Allocation in Multi-hop Cognitive Radio Networks. In: IEEE Wireless Communications and Networking Conference, April 1–5, (2010)
16. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing (May 2003)
17. Johnson, D.B., Maltz, D.A., Broch, J.: DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In: Perkins, C.E. (ed.) Ad Hoc Networking ch. 5, pp. 139–172 (2001)
18. The Network Simulator, http://www.isi.edu/nsnam/ns/
19. Cognitive Radio Cognitive Network, http://stuweb.ee.mtu.edu/~ljialian/

A Flow-Aware Placement of Mobile Agent Control Network over Opportunistic Networks

Hsiao-Tzu You[1], Yao-Nan Lien[1], and Jyh-Shyan Huang[2]

[1] National Chengchi University, Taiwan, ROC
lien@cs.nccu.edu.tw
[2] Chunghwa Telecom, Taiwan, ROC
frank210@cht.com.tw

Abstract. Transmitting data on an opportunistic network is much more difficult than that on a conventional network. We propose to use mobile agent to enhance its message transfer efficiency. A mobile agent platform requires a search mechanism to control its agents. We investigated the application of mobile agent on the opportunistic network characterized by "CenWits Search and Rescue System" applied in YuShan National Park as well as proposed a control network technology to support rapid agent search. This paper enhances our previous control network placement models by taking into account the amount of agent flows. After proving it to be NP-hard, we designed a simple but efficient heuristic algorithms to solve the placement problem. In our simulation study, the new model outperforms our previous models easily.

Keywords: Opportunistic Network, Mobile Agent.

1 Introduction

Data transmission over opportunistic network is not only much slower but also more unpredictable than conventional networks. We proposed to apply mobile agent technology on opportunistic network to facilitate its control operation [5]. Applying mobile agent technology on opportunistic network will be able to enhance its message transfer efficiency as well as to enable more applications, as a consequence. However, a functional mobile agent system demands an agent tracking mechanism to facilitate the control of mobile agents such as termination, suspension, and resuming of a mobile agent. Our previously proposed control network technology will be able to fulfill this demand. This paper enhances our previous placement models by taking into account the amount of agent flows to achieve better services to mobile agents.

1.1 Opportunistic Network

Opportunistic Network, OppNet [1], is a special case of Delay-Tolerant Network (DTN), which may lack continuous network connectivity. In general, a DTN uses so called *store-carry-forward* mechanism to transfer messages. Typical examples [1,2,3]

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 611–623.
DOI: 10.1007/978-3-642-35452-6_62

are hiker tracking, wild-field animal tracking, battle field networking, etc. Since the message transfer on OppNet is slow and unpredictable, some network protocols such as TCP may suffer from severe performance degradation. For instance, a typical TCP protocol requires an acknowledgement message to be feedback to the sender within a predefined time period. Otherwise, the sender will take the network to be congested and will activate congestion control actions which may slow down or even stop the operation of TCP. Therefore, OppNet demands a more sophisticate message transfer mechanism other than store-carry-forward to prevent the network from interfered by inappropriate congestion control as well as other protocol operations.

1.2 Mobile Agent

A *mobile agent* is a composition of computer software and data which is able to migrate from one computer to another autonomously and continue its execution on the destination computer [4,6]. When an agent leaves its home node, it acts as a delegate of the originator and is able to make decision by itself based on the built-in logic and the information it collected from the network. On the completion of its assigned task, it returned to the originator to deliver the result. Due to the ability to make decision autonomously, mobile agent technology is a good candidate to provide message transfer service for OppNet. Nevertheless, a mobile agent platform demands a fast search mechanism to facilitate the control of mobile agents. Based on a special OppNet, *CenWits system* [3], this paper proposes a control network framework as well as an associated search mechanism to facilitate rapid agent search.

1.3 CenWits System

CenWits Search and Rescue System is a special Wireless Sensor Network (WSN) that uses loosely coupled connection and witnesses between wireless sensor nodes (i.e.. OppNet) to track locations of moving objects in the wilderness areas [3]. CenWits is comprised of GPS enabled mobile, in-situ sensors that are worn by subjects such as people and wild animals as well as access points (AP) that collect information from these sensors via wireless signals. CenWits records subjects' location/movement information as well as environment informaiton (e.g. weather) and conveys to the outside world. The system had been deployed in some places such as Rocky Mountain (US), Yosemite National Parks (US), and YuShan National Park (Taiwan). A typical application of this system is to narrow down the search area based on the recorded location and movement information once a rescue operation is launched for a lost hiker.

Conventional cellular networks may not be useful in the areas mentioned above due to the poor signal coverage. First, propagation of radio signals in mountain area is often interferenced by wavy terrain. Secondly, low user popularity and high deployment cost couldn't motivate cellular operators to increase cell tower densities. Therefore, in such a special OppNet, a special short range wireless communication mechanism such as ZigBee is used to support node-to-node communications.

In the system deployed in YuShan National Park, called *YushanNet*, every participating hiker carries a sensor node, called *mobile node* (*MN*) in this paper, which is designed to collect time, positions, and demanded environment information periodically. When two mobile nodes meet together in the park, they exchange the collected information to each other. When a mobile node meets an AP, it delivers all collected information to the AP, which then forwards the information to the headquarter through a high speed network. The entire system forms an OppNet. This paper takes this system as the reference system.

1.4 Challenges of Mobile Agent on OppNet

In YushanNet, each mobile node is carried by a hiker so that its moving behavior is actually the same as the moving behavior of its hosting hiker. Assuming hikers are walking in similar speeds, mobile agents are not only moving slowly, but also having difficulty to hop forward. As a consequence, neither moving nor searching a mobile agent is an easy task. Taking YushanNet as an example, we illustrate this difficulty using the example shown in Fig. 1. Assuming a search agent is launched to search another agent that left the starting point one day earlier, it is very difficult for the search agent to reach the target via forward hopping because all intermediate nodes are walking in a similar speed.

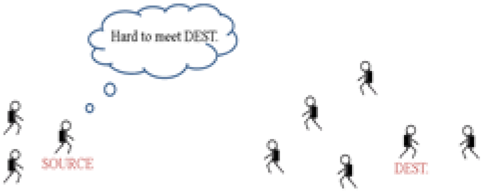


Fig. 1. Searching a mobile agent on OppNet

A real world system like YushanNet demands a fast agent search mechanism to support its control operations as well as urgent tasks, such as warning a hiking team member about a severe weather condition. However, most conventional agent search mechanisms are designed for conventional networks which are quite different from OppNet. Therefore, we propose to use a separated control network overlaid on top of a conventional high speed IP network as well as a search mechanism to facilitate rapid agent search on OppNet. A search agent can move to a control point near the target rapidly via the control network and then hop to a mobile node which then moves toward the target.

Section 2 will discuss current mobile agent search technology. Section 3 will discuss the concept of control network and its optimal deployment model. Section 4 will analyze problem complexity and present several heuristic algorithms. The evaluation will be presented in Section 5.

2 Related Work and Challenges

Mobile agent search technology has been studied for years [4]. They can be roughly classified into two categories: report-based and non-report-based. In a report-based system, every mobile agent has to report its locations periodically to the originator so that its current location can be easily determined. On the other hand, in a non-report-based system, a mobile agent doesn't report its locations such that a special search mechanism is needed to estimate its current location. All of them only consider the mobility of agents, but not the mobility of hosting devices. Therefore, they are not adequate for OppNet for the following reasons:

- Message transfer on OppNet requires longer delay time such that reported agent locations are usually outdated turning a search operation into a hide-and-seek game.
- The prior knowledge of the path and schedule a mobile agent took may not be useful in estimating the current location of a mobile agent because its hosts are moving too.

Therefore, it is difficult to calculate or to estimate the exact location of a mobile agent.

3 Control Network for Mobile Agent

3.1 Basic Concept of Control Network

We assume that there is a conventional high speed IP network (e.g. Internet) deployed in the concerned area. We can construct a control network by installing *control points* (CP) over the concerned area and connecting them together using the IP network. (Actually, control points can also be embedded in YushanNet's APs.) A search agent can move rapidly on this network to the control point nearest to the target, called *Egress Control Point,* then hop to a mobile node that is heading toward the target. The above procedure is illustrated in Fig. 2.

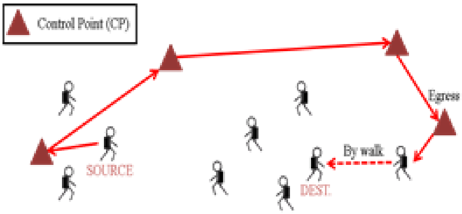


Fig. 2. Use Control Network to assist agent search

3.2 Agent Search Based on Control Network

The paper [5] uses a simple agent search algorithm, which is designed by modifying the Basic Binary Search (BBS) algorithm we designed for conventional networks [4]. To improve search efficiency, we are designing a more efficient search algorithm by estimating agent's current location based on the prior statistics of traveling time over each trail segment. The details will not be discussed here because it is out of the scope of this paper.

3.3 Deployment of Control Network

To deploy a control network under resource constraint is a typical combinatorial optimization problem. In our previous research, we propose several optimization models addressing different objectives based on the YushanNet.

3.3.1 Environment Assumptions

Based on YushanNet, we assume:

1. Each hiker moves in walking speed along a pre-determined trail.
2. All hikers move in approximately the same speed.
3. The deployment cost of a CP is location dependent.
4. Most candidate CPs are located at the end points and intersections of trails. They can also be deployed in the middle of a trail if the trail is too long.

3.3.2 Design Considerations and Objectives

Because resources are limited and deployment cost is location dependent, the selection of CPs to maximize the design objective becomes a combinatorial optimization problem, called *Control Point Selection Problem* (CPSP).

The best design objective might be "the average time to locate an agent". Thus, we define *MC_distance* (MCD) to be the distance (time) between a target agent and its Egress CP, which is the CP to be visited by the target agent in the nearest future.

Although MC_distance is a good performance index, it is very difficult to be formulated in mathematical format. Therefore, in our previous paper [5] we designed several optimization models addressing to different objectives. This paper proposes a more comprehensive model. It is up to the user to select the model that is the most appropriate w.r.t. his/her own need. Furthermore, all models have a common available resource constraint.

3.3.3 Existing Placement Models

We proposed three placement models in our previous paper [5]. In *Maximum Flow Model* (*CPSP-Flow*), the number of hikers entering a trail per time unit is a known constant. The objective is to maximize the total amount of hiker flows passing through the selected CPs. CPSP-Flow model is proven to be isomorphic to the famous 0-1 Knapsack problem. The advantage of CPSP-Flow model is its simplicity. However, it may lead to an uneven CP distribution. A good CSPS-Flow algorithm will inevitably choose the trails that have large hiker flows and neglect others. The mobile nodes that are hiking on less popular trails may never get a chance to meet any CP.

To solve the uneven distribution problem, CPSP model was modified as follows:

1. Each trail must have at least one CP if resource is sufficient.
2. CPs have to be evenly distributed to the geographic areas.

However, since graph is composed of nodes and edges carrying no geographic information, the objective of *Maximum Coverage Model* (*CPSP-Coverage*) is set to maximize the number of covered edges, instead. All edges have the same unit weight. In reality, trail segments may have different weights. For instance, hikers may be easier to get lost on some trail segments. It makes more sense to raise the priority of the nodes near such trail segments. Therefore, the model is refined by taking edge priority into account in next model. *Maximum Utility Model* (*CPSP-Utility*) assumes each edge has a weight representing the priority of a trail segment. All three models are proven to be NP-hard. Heuristic algorithms (0-1 knapsack, NPF, and NPF-U) were proposed to solve them. Although all three models are useful, each of them only addresses one or two issues. A more comprehensive model that can address several issues is yet to be studied.

3.3.4 Enhanced Heuristic Algorithms

For comparison purpose, we proposes two algorithms, NPF2 and NPF-U2 for CPSP-Coverage and CPSP-Utility modes to improve the performances of NPF and NPF-U, respectively. Both NPF and NPF-U are greedy algorithms to select control nodes one by one based on two potential indices, node potential and tail potential. Node potential gets higher priority than trail potential, which is taken into consideration only when there is a tie on the node potential. NPF2 and NPF-U2 change the node selection rule into the linear combination of both indices. Thus both indices can be taken into account simultaneously. Only NPF-U2 is shown in Fig. 3, while NPF2 can be easily figured out from NPF.

```
Candidate ListV ← all nodes
Iterate until Committed Cost >= Available Resources
{
      Delete the node from the candidate ListV with node_weight > Remaining Resources.
      Select the node from the candidate ListV with the largest value of
$\alpha$(total weights of unmarked edges /node cost)+
$\beta$(trail_potential[node]);
  If tie Then select the node with the smallest value of cost[node].
Mark the edges and trails that are connected to the selected node to "covered".

Update the value of total weights of unmarked edges[node] and trail_potential[node] of every node by marking covered trails.
}
# trail_potential[node] is the (no. of unmarked trail /cost) w.r.t. the node.
```

Fig. 3. Node Potential First for Utility (NPF-U2) Algorithm

3.4 Maximum Utility-Flow Model

This paper proposes a more comprehensive CPSP model, *Maximum Utility-Flow Model* (*CPSP-UF*), which takes not only weighted edge coverage but also the amount of agent flows into account.

We first define the total profit for a given set of nodes, *M,* as follows:

$$P(M) = \{\ \Sigma\ (z_{ij}\ \cdot\ f_k) \mid e_{ij}\ \in\ M',\ t_k\ \text{ passes } e_{ij}\ \}$$

In above equation, z_{ij} is the weight of edge e_{ij} and *M'* is the set of edges connected to the selected nodes in *M*. *P*(*M*) is then the total profit, the optimization objective. The model is defined as follows:

Given a graph $G = (V, E)$, where

- $V = \{v_1, \ldots, v_n\}$ is the set of possible nodes for building control points,
- $E = \{e_{ij} \,|v_i, v_j \in V\}$ is the set of edges (trail segments) between v_i and v_j,
- $W = \{w_i \,|v_i \in V\}$ is the set of the cost (required resources) of building a control point on v_i,
- C is the available resources,
- $T = \{t_1, \ldots, t_m\}$ is the set of hiking trails, each of which is a path, a sequence of nodes, used for hiking, and
-
- $G = \{g_{ki} \mid v_i \in V, t_k \in T\}$ is a passing_trail matrix,

  where $g_{ki} = \begin{cases} t_k & \text{if trail } \mathrm{t_k} \text{ passes node } \mathrm{v_i} \\ 0 & \text{otherwise,} \end{cases}$
- $Z = \{z_{ij} \,|v_i, v_j \in V\}$ is the weight of the edge e_{ij},
- $F = \{f_k \mid t_k \in T\}$ is the set of hiker flows, which is the number of hikers per time unit passing a trail,

the *CPSP- Utility-Flow Problem* is to find $S \subseteq V$, to

maximize $P(S)$
subject to
$\bigcup \{g_{kj}\ \cdot\ x_i\ \} = T$ and $\sum_{i \in V} w_i x_i \leq C$, where

$$x_i = \begin{cases} 1 & \text{if } v_i \in S\ , \\ 0 & \text{otherwise .} \end{cases}$$

4 Complexity Analysis and Solutions

Similar to our previous models, Maximum Utility-Flow Model can also be easily proven NP-Hard. To solve the problem, we propose a heuristic algorithm, called *Node Potential First for Utlity-Flow* (*NPF-UF*). As shown in Fig 4.

```
Candidate ListV ← all nodes
Iterate until Committed Cost >= Available Resources
{
    Delete the node from the candidate ListV with node_weight > Remaining
Resources.
    Select the node from the candidate ListV with the largest value of
(α (total weights of unmarked edges /node cost)+
β (trail_potential[node]))*γ (flow_potential[node]);

If tie Then select the node with the smallest value of cost[node].
Mark the edges and trails that are connected to the selected node to "covered"

Update the value of node_potential and trail_hit_count of every node by marking
covered trails.
}
# trail_potential[node] is the (no. of unmarked trail /cost) w.r.t. the node
# flow_potential [node] is the (no. of unmarked flow/ node_cost) w.r.t node
```

Fig. 4. Node Potential First for Utility-Flow (NPF-UF) Algorithm

5 Performance Evaluation

The performances of heuristic algorithms NPF-U, NPF-U2, NPF-UF were evaluated using simulative experiments on a regular personal computer.

5.1 Evaluation Metrics

Table 1 illustrates the evaluation metrics used in our experiments. Among them, MC_distance (MCD) are unified metrics to evaluate the adequacy of the models because their objectives are all different. MCD of a mobile node is the hop counts from the node to the nearest control point. And, the average MCD is the average distance (time) from a mobile node to the nearest CP over all mobile nodes.

Mean_MCD =

$$\frac{\sum_{\forall v_i \in V} MCD(v_i)}{|V|}$$

Note that minimization of mean_MCD is not the objective of all four models such that the mean_MCD calculated by a heuristic solution may be smaller than that is calculated by an optimal solution. Further, the granularity of MCD is an edge, which is quite large since it may require several hours to walk through an edge.

In each of following experiments, problem instances were randomly generated using the parameters listed in Table 2.

Table 1. Evaluation Metrics for CPSP

Name	Definition
flow coverage	total flows passing selected CPs
edge coverage	total no. of edges connected to the selected CPs
Weighted edge coverage	summation of the weight of the edges connected to the selected CPs
populated edge coverage	summation of the (no. of flow * weight) of the edges connected to the selected CPs
mean_MCD	average MCD over all mobile nodes
mean_weighted_ MCD	average weight_MCD over all mobile nodes

Table 2. Parameters for Problem Generation

Parameter	Small Scale	Large Scale Node Scale Sensitivity	Large Scale Trail Scale Sensitivity
Number of candidate CPs	20	10~100	100
Number of trails	8	8	4 ~ 40
Available resources	12	12	12
CP deployment cost	1 ~ 5	1 ~ 5	1 ~ 5
Edge weight	1 ~ 5	1 ~ 5	1 ~ 5
Flow size on trail	N.A.	1 ~ 10	1 ~ 10

5.2 Evaluation of Small Scale Problems

Because all CPSP models are NP-hard problems, we evaluate NPF-U, NPF-U2, and NPF-UF against optimal solutions on small scale problems. Five problem instances were randomly generated using the parameters listed in the second column of Table 2.

Two indices are defined: *Diff* is the difference between heuristic and optimal solutions and *Error* is the ratio of *Diff* over optimal solution. The results of NPF-U, NPF-U2, and NPF-UF experiments are showed in Table 3, 4, and 5.

5.3 Evaluation of Large Scale Problems

We randomly generated many large scale instances to evaluate NPF-U, NPF-U2 and NPF-UF. Because optimal solutions are difficult to compute, we only evaluated their sensitivities to the node scale and to the trail scale.

Table 3. NPF-U vs. Optimal solutions

Instance	1	2	3

edge coverage	0	Diff = 2
		Error = 1.01%
			0

mean_ weighted_MCD	Diff = 0.08
	Error = 2.48%
		Diff = 0.07
		Error = 2.22%
			0

max_ weighted_MCD	Diff = 2
	Error= 28.57%
		Diff = 1
		Error= 16.67%
			0

Instance	4	5	

edge coverage	Diff = 4
	Error = 1.34%
		Diff. = 20
		Err. =5.75%	

mean_ weighted_MCD	Diff = 0.01
	Error = 0.34%
		Diff = 0.01
		Error = 0.33%	

max_ weighted_MCD	0	0	

Table 4. NPF-U2 vs. Optimal solutions

Instance	1	2	3
edge coverage	0	0	0

mean_ weighted_MCD	Diff = 0.05
	Error = 1.55%
		0	0

max_ weighted_MCD	Diff = 2
	Error= 28.57%
		0	

Instance	4	5	

edge coverage	Diff = 2
	Error = 0.67%
		Diff = 15
		Error = 4.31%	

mean_ weighted_MCD	0	Diff = 0.01
		Error = 0.33%	

max_ weighted_MCD	0	0	

Table 5. NPF-UF vs. Optimal solutions

Instance	1	2	3

populated edge coverage	Diff. =3
	Err. = 0.4%
		Diff. =9
		Err. = 1.19%
			Diff. = 6
			Err. = 0.77%

mean_MCD	Diff. = 5.08
	Err. = 24.78%
		Diff. = 3.01
		Err. = 11.24%
			Diff. = 0.06
			Err. = 0.2%

Instance	4	5	

populated edge coverage	Diff. = 11
	Err. = 1.37%
		Diff. = 10
		Err. =1.23%	

mean_MCD	Diff. =4.34
	Err. = 11.79 %
		Diff. = 6.42
		Err. = 17.5%	

5.3.1 Experiments of Node Scale Sensitivity

In this experiment, 50 graphs were randomly generated with the number of nodes between 10 and 100. The parameters are listed in the third column of Table 2. Experiment results are showed in Fig 5, 6, 7, and 8.

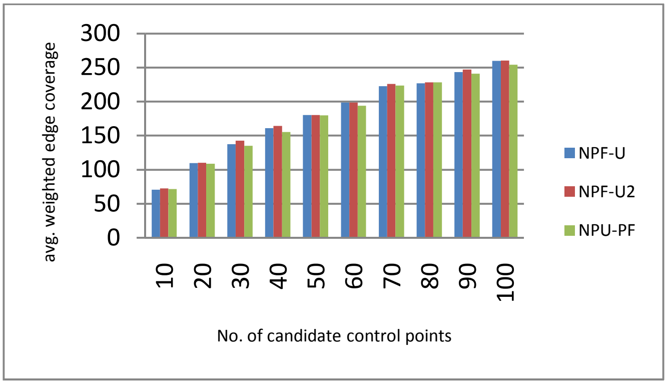


Fig. 5. Node Scale Sensitivity (avg. weighted edge coverage)

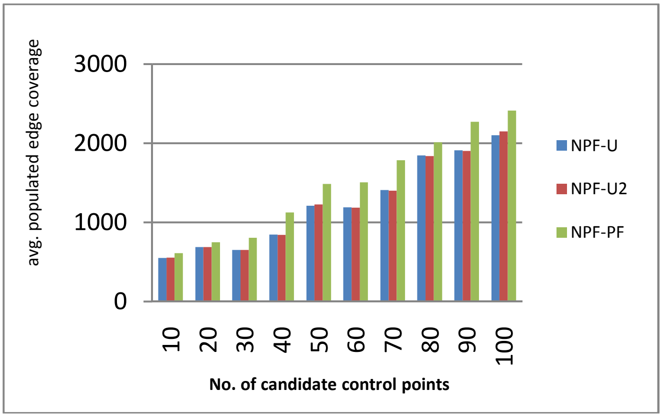


Fig. 6. Node Scale Sensitivity (avg. populated edge coverage)

From Fig. 5 to 8, we can see that NPF-U2 outperforms NPF-U when weighted coverage is taken as objective. NPF-UF performs worst since weighted coverage is not its optimization objective. When populated edge coverage is taken as the objective, NPF-PF performs the best and NPF-U2 performs next. Because NPU-UF takes both the weight of edges and flows into considerations, solutions provided by NPU-UF are better than NPU-U2 and NPF-U.

5.3.2 Experiments of Trail Scale Sensitivity

In this experiment, 50 graphs were randomly generated with the number of trails between 4 and 40. The parameters are listed in the fourth column of Table 2. The performance in terms of weighted edge coverage and populated edge coverage is similar to the experiments of node scale sensitivity. Therefore, only mean_weight_MCD and mean_MCD are shown in Fig. 7.

The results (see Fig. 8) show that the performance of three algorithms in terms of avg. mean_weighted MCD. NPF-PF performs the best in terms of avg. mean MCD but not avg. mean_weighted_MCD. Although NPF-PF performs the best in terms of avg. mean_weighted_MCD, the margin is nominal. Both indices increase proportional to the number of trails and number of candidate control points.

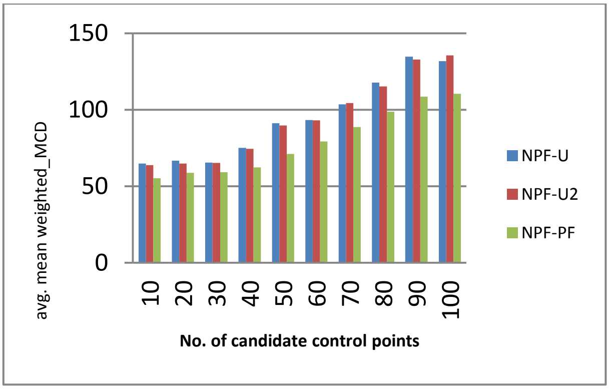


Fig. 7. Node Scale Sensitivity (avg. mean_weighted_MCD)

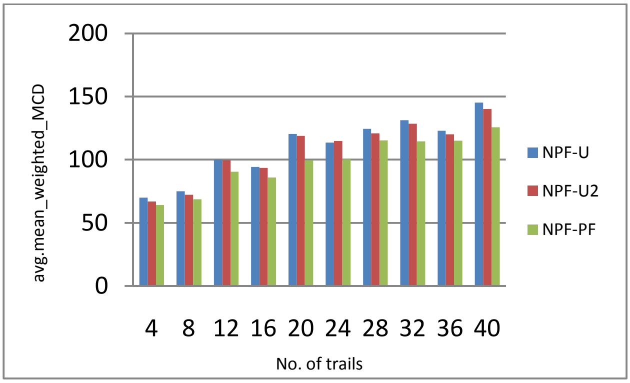


Fig. 8. Trail Scale Sensitivity (avg. mean_weighted_MCD)

6 Concluding Remarks

Transmitting data on an opportunistic network is much more difficult than that on a conventional network. We propose to use mobile agent to enhance its message transfer efficiency. We investigated the application of mobile agent on the opportunistic network characterized by "CenWits Search and Rescue System" applied in YuShan National Park. We propose to construct a control network using a high speed IP network for search agents to travel in high speed. Under different objectives and constraints, we propose several placement models for the placement of control network. This paper proposes a more comprehensive model, CPSP Utility-Flow, that taking not only weighted edge coverage but also the amount of agent flows into account. The simulative experiments show that CPSP Utility-Flow model is better than previous model. In our experimental environment, the average distance, or hiking time, from a mobile node to the nearest control point is no more than two trail segments.

References

1. Huang, C.-M., Lan, K.-C., Tsai, C.-Z.: A survey of opportunistic networks. In: Proc. of the 22nd International Conference on Advanced Information Networking and Applications, pp. 1672–1677 (2008)
2. Huang, Y.-T., Chen, Y.-C., Huang, J.-H., Chen, L.-J., Huang, P.: YushanNet: A Delay-Tolerant Wireless Sensor Network for Hiker Tracking in Yushan National Park. In: Proc. of Int'l Conf. on Mobile Data Management (MDM 2009), Taipei, Taiwan (2009)
3. Huang, J.H., Amjad, S., Mishra, S.: Cenwits: a sensor-based loosely coupled search and rescue system using witnesses. In: Proc. of the 3rd International Conference on Embedded Networked Sensor Systems, pp. 180–191 (2005)
4. Lien, Y.N., Leng, C.W.R.: On the search of mobile agents. In: Proc. of the 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 703–707 (1996)
5. Lien, Y.-N., Lin, Y.-S.: Placement of Control Network for Mobile Agents over Opportunistic Networks. In: Proc. of 8th International Workshop on Mobile Peer-to-Peer Computing, Lugano, Switzerland (March 2012)
6. Milojicic, D.S., Douglis, F., Wheeler, R.: Mobility: processes, computers, and agents. ACM Press/Addison-Wesley Publishing Co., New York (1999)

Achieving Weighted Fairness for High Performance Distributed Coordination Function with QoS Support in WLANs

Yeong-Sheng Chen[1], Fan-Chun Tseng[1], and Chih-Heng Ke[2]

[1] Department of Computer Science, National Taipei University of Education, Taipei, Taiwan
yschen@tea.ntue.edu.tw, fanchun.tseng@gmail.com
[2] Department of Computer Science and Information Engineering,
Quemoy University, Kinmen, Taiwan
smallko@gmail.com

Abstract. This study proposed a High performance Distributed Coordination Function with QoS support (QHDCF) protocol to enhance the throughput and achieve fair allocation of wireless medium access. QHDCF protocol has two working modes: contending mode and active mode. In the contending mode, all stations follow DCF scheme to contend for channel access. Once a station successfully accesses the channel, it switches to the active mode. In the active mode, the transmitting station determines the next transmitter according to a selection scheme from a probability formulation perspective. The selection scheme is based on relative weights of different traffic classes so that weighted-fairness can be achieved. Besides, since most stations do not contend for channel access, collisions are reduced a lot and therefore the throughput is enhanced. Simulation results show that QHDCF performs significantly better than EDCA in terms of both weighted fairness and throughput.

Keywords: EDCA, QoS, Collision, Weighted fairness.

1 Introduction

The IEEE 802.11 WLAN specification 1 standardizes physical (PHY) layer and medium access control (MAC) sub layer implementations. In this standard 23, there are two coordination functions of wireless MAC protocols: the Point Coordination Function (PCF) and the Distributed Coordination Function (DCF). Due to the growing demands of multimedia traffic communication, IEEE 802.11e standard 4 enhances the legacy DCF scheme to support QoS requirements by proposing Enhanced Distributed Channel Access (EDCA). In EDCA [5][6], the traffic flows are classified into four Access Categories (ACs). To supply service differentiation in channel access, each AC has different contention parameters including Arbitration Inter-Frame Space (AIFS) and Contention Window (CW). A lot of performance evaluations of the EDCA scheme have been reported through simulations in the literature [7]-9. In [8], the authors claim that under heavy loads of high-priority traffic, the EDCA scheme

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 625–634.
DOI: 10.1007/978-3-642-35452-6_63

suffers from high collision rate and starves low-priority traffic, bringing out of the problem of collisions and unfairness allocation of wireless bandwidth.

Many existing researches in the literature are devoted to the enhancement of network throughput and the fair allocation of wireless bandwidth for the contention-based MAC protocol. Qiao et al. [10] propose priority-based fair medium access control (P-MAC) protocol, by modifying the DCF scheme, to enhance the throughput and attain to fair allocation of bandwidth. In [11], HDCF is proposed to alleviate the problem of wasting time in contention resolution. Each station maintains its own Active List and determines the next transmitter from the Active List in accordance. Only new stations have to contend for the channel access. Since most stations do not contend for channel access, collisions are reduced and therefore the throughput is enhanced. However, P-MAC and HDCF do not support differentiated QoS in WLANs.

In this paper, we propose a High-performance Distributed Coordination Function with Quality of service support (QHDCF) protocol in WLANs. The contributions of this paper are two-fold: (a) minimizing collisions to enhance the throughput; and (b) achieving weighted fairness among stations. QHDCF has two working modes: contending mode and active mode. In the contending mode, all stations follow the legacy IEEE 802.11 DCF mechanism to contend for the channel access. The successful station will then switch to active mode to transmit data. In the active mode, the transmitting station can transmit a burst of data frames back-to-back within the transmission opportunity (TXOP) limit, and determine the next transmission flow from its Active List according to a selection scheme. The selection scheme is based on relative weights of four traffic classes so that the goal of weighted-fairness among stations can be achieved. Moreover, since most stations do not contend for the channel access, collisions are reduced a lot and therefore the throughput is significantly enhanced. The details of the proposed QHDCF protocol will be elaborated later in Section 3.

The remainder of this paper is organized as follows. Background and related work are summarized in Section 2. Section 3 introduces the high performance distributed coordination function with QoS support (QHDCF) protocol to achieve weighted fairness. The performance evaluations and simulation results are presented in Section 4. Finally, conclusions and future work are drawn in Section 5.

2 Background and Related Work

2.1 High Performance Distributed Coordination Function

High performance DCF (HDCF) [11] protocol provides a fair procedure among all stations by using Active List. Each active station maintains its own Active List and the current transmitting station selects the next station from its Active List in accordance with a uniform distribution. The selected station sends out a frame of its own after an interval of one PIFS following the previous transmission. Thus, all the stations are guaranteed to be contention free. In addition, new stations utilize an interrupt scheme to contend directly without delays. Thereafter, active stations would stop their transmissions and only new station would compete for the channel using DCF. As a result, HDCF achieves collision avoidance and fairness without backoff slots.

In the contending mode, the HDCF protocol is identical with the original IEEE 802.11 DCF protocol. However, in the active mode, once the current station has transmitted a frame and received an ACK frame, it also announces the Next-Station, which is selected by uniform distribution. Thus, all active stations will overhear the announcement know which station has the right to access the channel. The selected station waits for an interval of one PIFS following the previous transmission and sends out a frame of its own. The NAV is also used; stations will defer to end of the ongoing transmission. Besides, when the selected next transmission station detects a signal before the PIFS has elapsed, it stops its transmission to allow new stations to contend for the channel access and then turn into the active mode. For brevity's sake, descriptions of the mechanism are omitted. Interested readers may refer to [11] for details.

2.2 Related Work

Many existing researches were devoted to achieve fair allocation of bandwidth for the contention-based MAC protocol [13]-[15]. [13] and [14] attempt to provide equal shares of bandwidth to different stations, and the traffic weights are implicitly assumed to be the same. Banchs and Perez [15] extend the DCF function and proposed a method to provide weighted fairness by tuning the contention window (CW). However, the effect of exponential backoff does not consider in their analysis. Bianchi [16] proposes a Markov chain model to analyze the performance of the DCF scheme, which accounts for the exponential backoff and the backoff stage. Based on Bianchi's model, several schemes have been proposed aiming to achieve the weighted fairness issues for QoS differentiations [10][17]-[19]. In [10], Qiao and Shin propose a priority-based fair medium access control (P-MAC) protocol by modifying the DCF scheme. The basic idea is that the CW size for each wireless station is properly selected to reflect the relative weights among data traffic flows, so as to achieve the weighted fairness. [19] presents a new scheme which exploits differentiations of both inter frame space (IFS) and CW. The proposed scheme properly set the corresponding CWs such that the ratio of the two classes' successful transmission probabilities can attain a pre-defined weighted-fairness goal.

3 The Proposed Protocol

This study proposes a High-performance Distributed Coordination Function with QoS support (QHDCF) protocol to tackle two problems of IEEE 802.11e EDCA: (a) high collision rate under heavy load conditions; and (b) unfairness allocation of wireless bandwidth.

Like HDCF, a QHDCF station operates in one of two modes: contending mode and active mode. In the contending mode, stations follow legacy DCF mechanism to contend for the channel access. In the beginning, all the stations start transmission as soon as the DIFS and random backoff slots expire. Once a station successfully accesses the channel, it switches to active mode. That is, only new stations have to contend for the channel. In the active mode, the transmitting station determines the next transmission station according to a selection scheme. Thus, there will be no time waste caused by backoffs. In addition, since most stations do not contend for the channel, collisions are greatly reduced and therefore the throughput is significantly enhanced.

As for the unfairness problem, QHDCF modifies the selection scheme in HDCF [11] to achieve fairness. The key idea is that if the probability of the next transmission flow is determined according to its relative weight to other traffic flows, all traffic flows will fairly share the wireless medium [10]. The details will be elaborated later.

3.1 Traffic Class Priority and Active List

In HDCF protocol [11], since all traffics have the same priority, all stations have equal opportunity to be selected as the next transmitter by the current transmitting station from the Active List. That is, HDCF protocol is not suitable for multimedia environment, in which traffics are usually of different priorities. For supporting QoS differentiation, in the proposed protocol, the priorities of different traffic classes are taken in account.

To conform to the IEEE 802.11e standard 4, QHDCF protocol provides a QoS support through classifying the traffic flow into four classes, including Voice, Video, Best Effort, and Background. The four traffic classes are associated with different priorities. The first class is the Voice class with the highest priority and the fourth class is the Background class with lowest priority.

As mentioned above, the current transmitting station is responsible for selecting the next transmission station from its Active List. The Active List recodes all the active stations that are eligible to be the next transmitter. Each station maintains the Active List. An entry in the Active List consists of three fields: "Next-Station" is the station identifier, "Class" records the traffic class of this active station, and "MoreData" denotes that if this active station has more data to transmit after it transmits a data frame.

3.2 Weighted Fairness

Fair allocation of wireless bandwidth among different traffic classes has been identified as an important issue for wireless medium access control protocol [10][18][19]. In a WLAN supporting QoS, evidently, high priority traffics should get more transmission opportunity than low priority traffics. That is, considering fairness only may be improper; instead, weighted fairness should be considered. The ideal weighted fairness is defined as follows [10]. Assume that there are n (>1) different traffic classes and each station carries only one traffic flow. The relative weight associated with the traffic of class i ($1 \leq i \leq n$) is denoted as ϕ_i. The priority order among all the relative weights is $\phi_1 > \phi_2 \ldots > \phi_n > 0$. Let f_i denote the set of stations carrying class-i traffic. Consider the following equation

$$\forall i, j \in \{1, \ldots, n\}, \forall s \in f_i, \forall s' \in f_j,\ \frac{P_s}{\phi_i} = \frac{P_{s'}}{\phi_j} \tag{1}$$

where P_s is the probability that station *s* successfully transmits a data frame to the other station. If Equation (1) holds, all the traffic flows share the wireless medium fairly in a probabilistic sense, and we claim that the weighted fairness for data transmission is achieved. In other words, according to Equation (1), if the station carries a high-priority traffic flow, it should be assigned higher opportunity to access the wireless medium than the stations carries a low-priority traffic flow.

3.3 Selection Scheme and Transmission Opportunity (TXOP) Limit

Selection Scheme

From the above discussions, it is easy to see that the probability of the next transmission flow should be determined according to its relative weight to the other traffic classes so that the goal of weighted-fairness among traffics can be achieved. For this purpose, from a probability perspective, three dividing points of traffic classes are defined as follows.

$$d_i = \frac{\sum_{j=1}^{i}(\phi_j)}{\sum_{j=1}^{4}(\phi_j)}, i = 1, 2, 3 \tag{2}$$

where ϕ_i is the relative weight of traffic class *i*. (It is assumed that there are four traffic classes in the WLAN.) From Equation (2), it is easy to see $d_i \leq 1$, for i=1, 2, 3. To determine the next transmitter, the current transmitting station first generates a random value, say *p*. Let d_0=0 and d_4=1. Then, if $d_{i-1} < p \leq d_i$ (i=1, 2, 3, 4), the transmitting station randomly selects a station from the Active List whose traffic belongs to class *i* to transmit data frames.

Note that, in the WLAN supporting QoS, each traffic class is associated with a priority. Therefore, intuitively, the relative weight ϕ_i of four traffic classes can be defined as

$$\phi_i = W_i, \tag{3}$$

where *Wi* is the priority weight of traffic class *i* determined by its priority. In the proposed protocol, to conform to IEEE 802.11e standard, it is assumed that the priority weights of the four traffic classes, *Wi*, are 4, 3, 2, 1 for *i*=1,2,3,4, respectively.

However, from the point of view of station selection, it may still be unfair. For example, suppose that the number of stations in high-priority class is much larger than that of stations in low-priority class. Then, a station with traffic in high-priority class may still have lower transmission probability than another station with traffic in low-priority class. To alleviate this problem for achieving fair allocation of bandwidth among each active station, the number of stations in each class is taken into account and the relative weight of each traffic class is formulated as

$$\phi_i = N_i \times W_i, \tag{4}$$

where N_i is the number of stations whose traffic are in class *i*, and W_i is the priority weight of traffic class *i*.

Transmission Opportunity (TXOP) Limit

The TXOP modifies the standard transmission procedure by allowing multiple data frame transmission on a single channel access. Accordingly, a station is allowed to send a number of consecutive frames limited by the duration of allocated TXOP. Therefore, our proposed protocol considers the TXOP mechanism to satisfy complete QoS requirements and enhance the throughput of high-priority traffic class. The maximum duration of the transmission opportunity is determined by the TXOP limit parameter. During TXOP period, a station can transmit a burst of frames separated by SIFS interval within assigned time limit. The station ends its TXOP burst once it has no more frames to be transmitted, or there is not enough free space for the next frame exchange (Data + ACK).

The associated TXOP limit parameters of the four traffic classes in the proposed QHDCF protocol conform to the default EDCA parameter setting 4. A TXOP limit equal to zero indicates that the active station can transmit only one frame.

3.4 The Maximum Throughput in QHDCF

Assuming that the initial contending period is ignored, and the channel is error-free, the maximum throughput can be formulated as

$$S_{MAX\_QHDCF} = \frac{E[L]}{PIFS+E[T_{data}]+SIFS+T_{ack}+2\delta} \tag{5}$$

where δ is the propagation delay, $E[T_{data}]$ is the average time for sending a data frame, T_{ack} is the time for sending an ACK and $E[L]$ is the average frame size.

4 Simulations and Performance Evaluation

4.1 Simulation Environment

An ideal channel condition without hidden terminals and with error-free transmission is assumed. That is, all stations can directly receive data frames from other stations in the network.The values of PHY-related parameters with RTS/CTS mode are given from IEEE 802.11b standard 2. In particular, the normalized throughput is investigated. The normalized throughput is defined to be Throughput/Rate$_{\mathrm{MAC}}$, where Throughput is the total data transmitted successfully per the simulation period. In our simulations, the performance factors that are investigated include frame size, total number of stations, and number of stations of four classes.

4.2 Simulation Results

The normalized throughput of QHDCF and EDCA are shown in Figure 1. The frame size ranges from 50 bytes to 2250 bytes. The number of contending stations is 50. QHDCF_Max is the maximum throughput of QHDCF derived from Equation (5). QHDCF_W and QHDCF_NW are QHDCF with different relative weights derived by

Equations (3) and (4), respectively. Simulation results show that as the frame size increases, the normalized throughput increases for both QHDCF_W and QHDCF_NW. Besides, the normalized throughput of QHDCF is much better than EDCA and almost achieves the maximum throughput. This is because that the QHDCF protocol effectively minimizes the collision probability and therefore the throughput can be significantly enhanced.

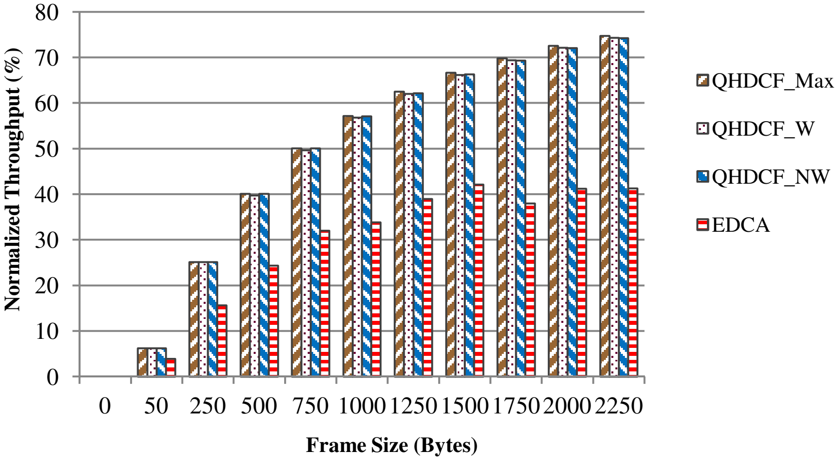


Fig. 1. Throughput vs. frame size

In Figure 2, the frame size is fixed at 1000 bytes and four scenarios of different number of stations with different traffic classes are considered. QHDCF_NW(4:3:2:1) denotes that the number of stations belonging to class 1, 2, 3 and 4 are in a ratio of 4:3:2:1. For example, assume that there are ten stations in the network. Then, for QHDCF_NW(4:3:2:1), there will be four, three, two, and one stations belonging to class 1, class 2, class 3, and class 4, respectively. Similarly, QHDCF_NW(1:1:1:1) and QHDCF_NW(1:2:3:4) denote that the number of stations belonging to class 1, 2, 3 and 4 are in a ratio of 1:1:1:1 and 1:2:3:4, respectively.

Figure 2 shows that the throughput of EDCA degrades as the number of stations increase since the collisions will increase and there will be more idle slots while the number of stations becomes large. However, both QHDCF_W and QHDCF_NW have better and stable performance than EDCA no matter how many stations join the network or how many stations have traffic belonging to a certain class.

In Figure 3, throughputs of different traffic classes in QHDCF_NW are compared with those of EDCA. Equation (4) are used to determine the relative weights while selecting the next transmission traffic class. Simulation results show that the throughput of each traffic class in QHDCF_NW is better than that of its corresponding AC in EDCA. Moreover, even though the number of stations in high-priority class is much more than that of stations in low-priority class, stations in lower priority of class always have transmission opportunity avoiding starvation.

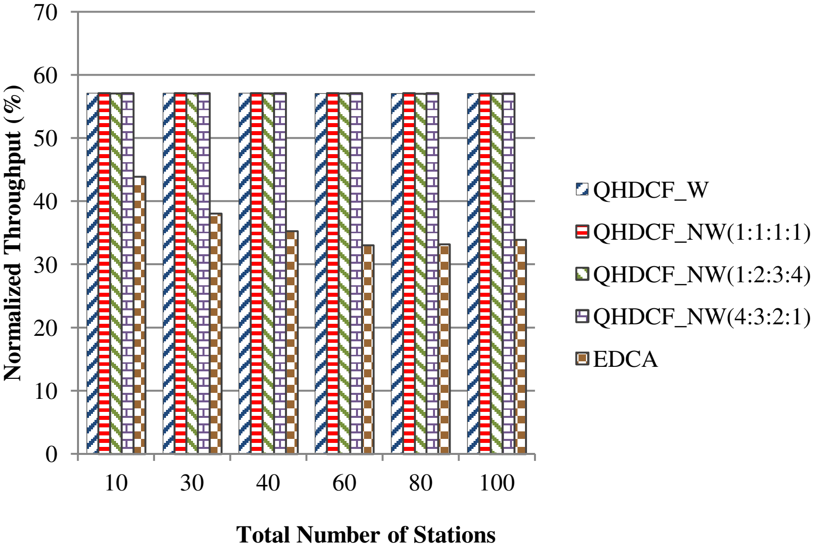


Fig. 2. Throughput comparisons between QHDCF and EDCA

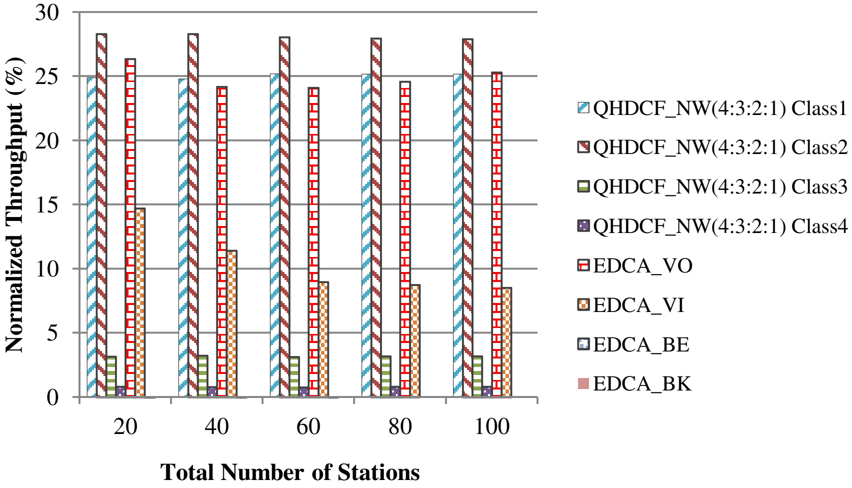


Fig. 3. Throughput comparisons between QHDCF_NW(4:3:2:1) and EDCA

5 Conclusions and Future Work

This study presents a new High-performance Distributed Coordination Function with QoS support to achieve weighted fairness in WLANs. The proposed scheme not only inherits the HDCF's merit to minimize collision rate and idle slots but also provides priority-based medium access control through the selection scheme to fairly share the wireless medium among stations. In QHDCF, the current transmitting station can transmit a burst of data frames back-to-back within the TXOP limit, and then determines the next transmission station according to the relative weights of traffic classes in its Active List. The stations with high-priority traffics are set to have high transmission opportunity, and the stations with low-priority traffics have low transmission opportunity, achieving the goal of weighted fairness. Thus, since a station with low-priority traffic does have a transmission opportunity, it will hardly suffer from starvation. Simulation results have shown that the QHDCF protocol outperforms IEEE 802.11e EDCA significantly in terms of both throughput and weighted fairness.

The throughput and fairness analysis presented in this paper are based on the assumption that all stations in the network always have pending data to transmit. Evaluating the performance of the QHDCF protocol in non-saturated networks is our future work. Besides, in the proposed protocol, the transmitting station updates other stations' Active Lists through a broadcast message and all stations have to spend additional costs on the maintenance of their own Active Lists. Thus, the maintenance mechanism of the Active List of each station in the QHDCF protocol needs further investigation in our future work.

References

1. IEEE Std for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, ISO/IEC 802-11:1999(E) (August 1999)
2. IEEE Std 802.11b-1999, wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer extension in the 2.4 GHz band (1999)
3. IEEE Std 802.11g-2003. Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications Amendment 4: further higher data rate extension in the 2.4 GHz band (2003)
4. IEEE Std 802.11e-2005. Part 11: wireless LAN medium access (MAC) and physical layer (PHY) specifications. Amendment 8: medium access control (MAC) quality of service enhancement (2005)
5. Mangold, S., Choi, S., May, P., Klein, O., Hiertz, G., Stibor, L.: IEEE 802.11e wireless LAN for quality of service. In: Proceedings of European Wireless 2002, vol. 2, pp. 32–39 (February 2002)
6. Choi, S., Prado, J.D., Mangold, S.: IEEE 802.11e contention-based channel access (EDCF) performance evaluation. In: Proceedings of IEEE International Conference of Communications (ICC), vol. 2, pp. 1151–1156 (May 2003)
7. Gu, D., Zhang, J.: QoS enhancement in IEEE 802.11 Wireless Local Area Networks. IEEE Communications Magazine 41(6), 120–124 (2003)
8. Lindgren, A., Almquist, A., Schel´en, O.: Quality of service schemes for IEEE 802.11 wireless LANs-An Evaluation. Mobile Networks and Applications (MONET) 8(3), 223–235 (2003)
9. Kong, Z., Tsang, D.H.K., Bensaou, B., Gao, D.: Performance analysis of IEEE 802.11e contention-based channel access. IEEE Journal on Selected Areas in Communications (JSAC) 22(10), 2095–2106 (2004)
10. Qiao, D., Shin, K.G.: Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF. In: Proceedings of 10th IEEE International Workshop on QoS (IWQoS), pp. 227–236 (May 2002)

11. Al-Mefleh, H., Chang, J.M.: High Performance Distributed Coordination Function for wireless LANs. In: Proceedings of International Federation for Information Processing (IFIP) Networking, pp. 812–823 (May 2008)
12. Bayraktaroglu, E., King, C., Liu, X., Noubir, G., Rajaraman, R., Thapa, B.: On the Performance of IEEE 802.11 under Jamming. In: Proceedings of INFOCOM, pp. 1265–1273 (March 2008)
13. Bharghavan, V., Demers, A., Shenker, S., Zhang, L.: MACAW: a media access protocol for wireless LAN's. In: Proceedings of SIGCOMM 1994 Conference on Communications Architecture, Protocols and Applications, vol. 24(4), pp. 212–225 (October 1994)
14. Ozugur, T., Naghshineh, M., Kermani, P., Olsen, C.M., Rezvani, B., Copeland, J.A.: Balanced media access methods for wireless networks. In: Proceedings of ACM MOBICOM, pp. 21–32 (October 1998)
15. Banchs, A., Perez, X.: Distributed weighted fair queuing in 802.11 wireless LAN. In: Proceedings of IEEE International Conference of Communications (ICC), vol. 5, pp. 3121–3127 (April 2002)
16. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications (JSAC) 18(3), 535–547 (2000)
17. Yang, Y., Wang, J., Kravets, R.: Distributed optimal contention window control for elastic traffic in wireless LANs. In: Proceedings of INFOCOM, vol. 1, pp. 35–46 (March 2005)
18. Vaidya, N.H., Bahl, P., Gupta, S.: Distributed fair scheduling in a wireless LAN. In: Proceedings of ACM MOBICOM, pp. 167–178 (August 2000)
19. Cheng, R.G., Chang, C.J., Shih, C.Y., Chen, Y.S.: A new scheme to achieve weighted fairness for WLAN supporting multimedia services. IEEE Transactions on Wireless Communications 5(5), 1095–1102 (2006)

A Cross-Layer Design for Energy Efficient Sleep Scheduling in Uplink Transmissions of IEEE 802.16 Broadband Wireless Networks

Jen-Jee Chen[1], Shih-Lin Wu[2,⋆], and Wei-Yu Lin[2]

[1] Dept. of Electrical Engineering, National University of Tainan, Tainan, Taiwan
jjchen@mail.nutn.edu.tw
[2] Dept. of Computer Science and Information Engineering, Chang Gung University,
Kweishan Taoyuan, Taiwan
slwu@mail.cgu.edu.tw, b9529016@stmail.cgu.edu.tw

Abstract. To support mobility, wireless mobile devices are powered by batteries; however, a battery can only store a limited amount of energy. According to the experiment, for a 3G handheld device, the wireless interface consumes the largest proportion of the total power (up to 40%). In particular, the 4G wireless communications will adopt OFDMA as the wireless access technology. In this case, the wireless interface in a mobile device will consume more power. IEEE 802.16 is one of the promising technologies for future wireless communications. To save power, some studies focus on how to schedule the sleep of mobile stations (MSs) to minimize the total active time; others try to reduce the transmission power of MSs with satisfying MSs' demands as the constraint. However, none of them considers both sleep scheduling and power control at the same time in IEEE 802.16 networks. To conserve energy, a thorough study of both MAC and physical layers is required. In this paper, we propose a cross-layer design to jointly schedule the sleep and control the transmission power of MSs in the uplink direction of IEEE 802.16 networks. Extensive simulation results verify that the proposed scheme significantly decreases the total power consumption of the MSs while guarantee their QoS requirements.

Keywords: IEEE 802.16, power allocation, MAC protocol, power management, power saving class, WiMAX, wireless network.

1 Introduction

The IEEE 802.16 [1] is a promising standard for providing broadband wireless access to mobile stations (MSs) with high mobility. Like most other wireless mobile systems, how to conserve energy for battery-powered MSs is a critical issue in IEEE 802.16. To save power in IEEE 802.16, most work focuses on either the physical layer power control or the MAC (Medium Access Control) layer sleep scheduling. For power control, the problem is to minimize the total

[⋆] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 635–644.
DOI: 10.1007/978-3-642-35452-6_64

transmission power of MSs subject to the demand and resource constraints. IEEE 802.16 supports multiple levels of modulation and coding schemes (MCSs). Usually, higher transmission power improves the received signal quality at the receiver side such that a higher level MCS can be used and which needs less resource, while lower transmission power degrades the received signal quality at the receiver side such that a lower level MCS has to be used and which requires more resource. Under the constraint of limited resource and QoS (Quality of Service) requirements of MSs, resource and power allocations of MSs have to be done in an energy efficient way. On the other hand, in IEEE 802.16 MAC layer, *Power Saving Classes (PSCs)* are defined to allow an MS to switch between active and sleep modes to reduce unnecessary energy consumption. The standard defines three types of PSCs for different traffic characteristics. Type I is designed for non-real-time traffic flows; it has exponentially increasing sleep windows if no packet comes. Type II is designed for real-time traffic flows; it has a fixed size of sleep windows. Type III is designed for multicast connections or management operations. The standard suggests that connections with similar characteristics can be associated with one PSC. An MS can turn off its radio interface when all its PSCs are in their sleep windows, but has to wake up when any PSC is in a listening window. So, how to assign PCSs to MSs to minimize the total active time of MSs is the major problem of the MAC layer sleep scheduling.

In the literature, prior studies design solely either physical layer-based [2, 3] or MAC layer-based [4–8] schemes to address the power saving issue. For physical layer-based methods, [2] reduces the energy conservation problem in IEEE 802.16 networks to the multiple-choice knapsack problem and designs a heuristic solution to allocate the resource and power of MSs in an energy efficient way. Reference [3] develops two energy-efficient heuristics, called demand-first allocation and energy-first allocation schemes, to allocate uplink resource and power for IEEE 802.16j transparent-relay networks. For MAC layer-based methods, a Maximum Unavailability Interval (MUI) scheme is proposed in [4] for selecting the optimal start frames of PSCs to maximize the unavailable time. This work does not answer how to decide PSCs' parameters. In [5], since one single PSC is applied to serve all real-time connections in an MS, the selection of sleep and listening windows must meet the strictest bandwidth and packet delay bound requirements of all connections. This could incur waste of bandwidth and extra listening windows. [6] proposes two Per-flow Sleep Scheduling (PSS) schemes to arrange the sleep flows according to each of their characteristics. This allows flows with larger packet delay bounds to have longer sleep cycles, thus leading to better sleep ratio and bandwidth utilization. Reference [7] proposes to serve each MS by a PSC of type II, but all of them share the same sleep cycle. This results in PSCs without overlapping in their active frames. However, since the common sleep cycle is bounded by the strictest delay bound of all MSs, this way causes some MSs to have too many active frames. Chen et al. [8] proposes an efficient tank-filling algorithm, which can allocate resources and sleep cycles to MSs according to their QoS characteristics with the least number of active frames. Thus, the total power consumption of MSs can be significantly reduced and the bandwidth utilization is high.

As can be seen, none of the previous works aims at co-designing both the MAC-layer sleep scheduling and the physical layer power control to improve energy conservation of MSs in IEEE 802.16 networks. From the MAC layer's point of view, the less active ratio MSs need, the fewer power MSs consume. However, less active time means that an MS has to pay more transmission power in order to finish its packet delivery during a short time, thus increasing the energy consumption of the MS. On the other hand, physical layer lowers down the transmission power to conserve energy for MSs. If MSs can enter sleep mode when packet transmissions are not needed, energy consumption of MSs is able to be further reduced. But, the schedule of resource allocation must follow the rule of IEEE 802.16 sleep operation. In this paper, we propose a cross-layer design for scheduling the sleep and allocating wireless resource and transmission power of MSs in the uplink directions over IEEE 802.16 networks. The sleep scheduling part arranges the sleep of MSs to reduce unnecessary power consumption, while the resource and power allocation part minimizes the transmission power of MSs when active. Both parts have to find a balance between the maximization of sleep ratio and the minimization of transmission power. The former reduces the active frames of an MS and the latter requires more resource (or frames) to decrease the MS's transmission power. To solve above problems, our cross-layer design evaluates the total traffic load of MSs and then schedule the active frames of MSs according to the evaluation and each of their QoS characteristics. Then, the resource and power allocation part energy efficiently assigns the uplink transmission power and wireless resource to MSs. Our design is compliant to the standard. Simulation results show that the proposed scheme can improve the energy efficiency of the MSs while guarantee the QoS requirements.

The rest of the paper is organized as follows. Section 2 gives some backgrounds related to our work and formally defines the problem. The proposed joint sleep scheduling and power allocation design is presented in Section 3. Simulation results are shown in Section 4. Section 5 concludes this paper.

2 Backgrounds and Problem Definition

2.1 The Structure of the Uplink Subframe

We consider the uplink communication in an IEEE 802.16 OFDMA network using the *TDD (Time Division Duplex)* mode (Our scheme can also be applied in the *FDD (Frequency Division Duplex)* mode, but here we choose to explain our method by using the TDD mode). An IEEE 802.16 BS supports multiple MSs in a point-to-multipoint manner and is responsible to the radio resource arrangement of the MSs according to their traffic demands. The resource is divided into frames, where a frame is a two-dimensional (subchannel time slot) array. Each frame is further divided into a downlink subframe and an uplink subframe. We show the uplink subframe in Fig. 1. An uplink subframe is with h subchannels and w time slots. Totally, there are $h \times w$ basic resource allocation units in an uplink subframe.

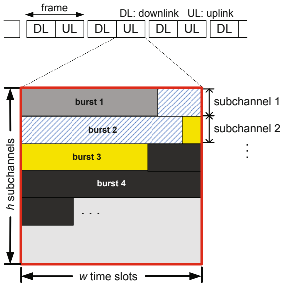


Fig. 1. The structure of the uplink subframe

In this work, we adopt the *PUSC (Partial Usage of Subchannel)* mode. Under the PUSC mode, *bursts* are the basic resource allocation units, where a burst is a sequence of slots arranged in a row-wise manner, as shown in Fig. 1. Note that a burst may cross multiple subchannels. The transmission power and rates of MSs are adjustable. However, the transmission rate of an MS within one burst should be fixed. The BS is responsible for allocating bursts for MSs. Since the BS is the only receiver, no two bursts can overlap.

2.2 Channel Model

Table 1 shows the available MCSs in IEEE 802.16 and their rates and required *SINRs (Signal to Interference plus Noise Ratios)*, denoted by $rate(\cdot)$ and $\delta(\cdot)$, respectively. As we can see, a high level MCS needs a high SINR while a low level MCS can sustain a low SINR.

Assume the transmission power of MS$_i$ is P_i. Let's consider any receiver j (which can be the BS). With power P_i, the received signal power $\tilde{P}(i,j)$ at receiver j can be modeled as

$$\tilde{P}(i,j) = \frac{G_i \cdot G_j \cdot P_i}{L(i,j)}, \tag{1}$$

where G_i and G_j are the antenna gains at MS$_i$ and receiver j, respectively, and $L(i,j)$ is the path loss from MS$_i$ to receiver j. Here, we adopt the *SUI (Stanford*

Table 1. MCSs supported by IEEE 802.16

MCS	scheme	$rate(MCS_k)$	$\delta(MCS_k)$
MCS_1	QPSK 1/2	48 bits/slot	6 dBm
MCS_2	QPSK 3/4	72 bits/slot	8.5 dBm
MCS_3	16QAM 1/2	96 bits/slot	11.5 dBm
MCS_4	16QAM 3/4	144 bits/slot	15 dBm
MCS_5	64QAM 2/3	192 bits/slot	19 dBm
MCS_6	64QAM 3/4	216 bits/slot	21 dBm

university interim) path loss model [9] to calculate $L(i,j)$, which is recommended by the 802.16 task group. So, the SINR perceived by receiver j is

$$SINR(i,j) = 10 \cdot \log_{10}\left(\frac{\tilde{P}(i,j)}{B \cdot N_o + I(i,j)}\right), \tag{2}$$

where B is the effective channel bandwidth, N_o is the thermal noise level, and $I(i,j)$ is the interference caused by other transmitters, which is evaluated by

$$I(i,j) = \sum_{l \neq i} \tilde{P}(l,j). \tag{3}$$

MS$_i$'s data (assume MCS_k is used) can be correctly decoded by receiver j if

$$SINR(i,j) \geq \delta(MCS_k). \tag{4}$$

By integrating Eqs. (1) and (2) into Eq. (4), the minimum power required for MS$_i$ to reach receiver j using MCS_k is

$$P_i \geq \frac{10^{\frac{\delta(MCS_k)}{10}}(B \cdot N_o + I(i,j)) \cdot L(i,j)}{G_i \cdot G_j}. \tag{5}$$

2.3 IEEE 802.16 Sleep Operation

Fig. 2 shows an example with three PSCs in an MS and the actual intervals that the MS can go to sleep. In this example, PSC 1 is a type II power saving class and has a listening window of 1 frame and a sleep window of 3 frames per cycle, PSC 2 is also a type II power saving class and has a listening window of 1 frame and a sleep window of 8 frames per cycle, while PSC 3 is a type II power saving class and has a listening window of 1 frame and a final sleep window of 8 frames. The actual unavailability intervals that the MS can sleep are the frames that all the three PSCs are in their sleep windows. Clearly, we observe that by increasing the overlapping of listening windows, the MS's energy consumption can be reduced. But the required resource of each PSC has to be taken into consideration. In this paper, we will focus on the type II power saving class.

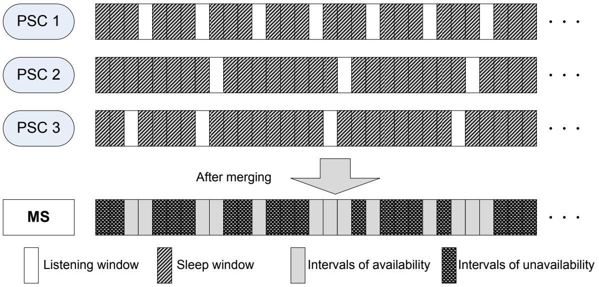


Fig. 2. Sleep frame determination of an MS

2.4 Problem Definition

We consider a BS serving n MSs MS_i, $i = 1..n$. Each MS_i has a maximum transmission power of P_i^{MAX} (mW per subchannel), an uplink real-time data arrival rate of r_i bits/frame, and each data arrival has a delay bound of d_i frames. We assume that MSs may move around within the BS's signal coverage, but the relative distances among BS and MSs can be estimated. The available bandwidth per uplink subframe is $h \times w$ slots. The goal is to assign each MS_i a PSC of type II with a sleep cycle of T_i, a listening window of T_i^L, and an offset of T_i^S. Then, for each uplink subframe, each active MS is allocated its uplink transmission power and MCS_k such that the total energy consumption of all MSs can be minimized. Also, there is an implicit requirement that whenever a listening window of an MS arrives, the BS should be able to serve all its backlog data that would be overdue otherwise.

3 The Proposed Cross-Layer Design

In this section, we illustrate the cross-layer design for the sleep scheduling MSs in the uplink directions over IEEE 802.16 networks. Our proposed scheme is composed of two sub-schemes. One is for the sleep scheduling, the other is for the resource and power allocations. *Sleep scheduling sub-scheme* arranges the sleep of MSs to reduce unnecessary active frames, while the *resource and power allocation sub-scheme* minimizes the transmission power of MSs when active. Sleep scheduling sub-schemes is executed when initializing the sleep mode of MSs, adding/deleting an MS from the sleep mode, or altering the sleep parameters of MS(s). On the other hand, after sleep mode is activated, the resource and power allocation sub-scheme executes repeatedly in a per-frame basis to dynamically allocate slot resource and transmission power to the active MSs.

3.1 Sleep Scheduling

In this section, we are to schedule the power saving classes of MSs. To schedule the sleep of MSs, the following parameters for each MS_i, $i = 1..N$, will be determined in the BS: (1) (T_i, T_i^L, T_i^S) and (2) amount of bandwidth $R[i, k]$ allocated in bits in the k-th active frame of each listening window, $k = 1..T_i^L$. Then these parameters are sent to each MS_i. These MSs will behave accordingly. The detail procedure is as follows:

1. Set T_i of each MS_i, $i = 1..N$, to

$$T_i = \min\{d_i, i = 1..N\} \text{ (in frames).} \tag{6}$$

2. Calculate the total required resource Ω (in slots) of MSs per T_i by

$$\Omega = \Sigma_{i=1}^{N} \frac{r_i \times T_i}{r_i^{CH}}, \tag{7}$$

where r_i^{CH} is the average channel rate in bits/slot of MS_i.

3. Assume $R[j] = \Omega/T_i$, $j = 1..T_i$, where $R[j]$ is to record the amount of remaining free resource in the j-th frame per cycle.
4. Assume $j = 1$, for each MS$_i$, $i = 1..N$, assign its power saving class as follows:
   (a) Set T_i^S to $T_i^S = j$.
   (b) If $R[j] \geq \frac{r_i \times T_i}{r_i^{CH}}$, set T_i^L and $R[i,1]$ to $T_i^L = 1$ and $R[i,1] = \frac{r_i \times T_i}{r_i^{CH}}$, respectively, update $R[j] = R[j] - \frac{r_i \times T_i}{r_i^{CH}}$, then go to step (d). Otherwise, go to next step.
   (c) If $R[j] < \frac{r_i \times T_i}{r_i^{CH}}$, allocate consecutive free resource to MS$_i$ frame by frame until enough resource is reserved. Then we set T_i^L to the number of frames from the first to the last frame where MS$_i$'s allocated resource is placed. Also, we set

$$R[i,k] = \begin{cases} R[j], & \text{if } k = 1 \\ \Omega, & \text{if } 1 < k < T_i^L \\ \frac{r_i \times T_i}{r_i^{CH}} - R[i,1] - \left(T_i^L - 2\right) \times \Omega, & \text{if } k = T_i^L \end{cases}. \tag{8}$$

   Then update $R[j+k] = 0$, for $k = 0..(T_i^L - 2)$, set $j = j + T_i^L - 1$ and $R[j] = R[j] - R[i, T_i^L]$.
   (d) If $R[j] = 0$, update $j = j + 1$.

Note that our sleep scheduling algorithm evenly distributes MSs' required resource in each frame of a cycle, i.e., for each frame, only $\Omega/(h \times w \times T_i) \times 100\%$ (total required resource of MSs per T_i over total slots per T_i) of resource is used. This is because the transmission power and frame resource is changeable, i.e., high power allows high level MCS and consumes less frame resource while low power permit only low level MCS and needs more frame resource. Therefore, the reserved free resource in each frame can help MSs to further reduce their transmission power when allocating resource and power in each uplink subframe. In the sleep scheduling procedure, we can guarantee the QoS of each MS$_i$, $i = 1..N$, because

$$T_i = \min\{d_i, i = 1..N\} \leq d_i. \tag{9}$$

3.2 Resource and Power Allocation

After assigning PSCs to MSs, BS will sent each MS its sleep parameters. Then each MS operates accordingly. For each uplink subframe, the BS will dynamically allocate resource and power for the active MSs according to their channel conditions. The resource and power allocation of each uplink subframe ϱ are as follows:

1. Suppose there are m active MSs in the ϱ-th uplink subframe. Without loss of generality, we reindex these MSs from 1 to m. For each MS$_i$, $i = 1..m$, assume this is its n_i-th active frame of the listening window, $1 \leq n_i \leq T_i^L$, the resource $\widehat{r}_i$ (in bits) allocated to MS$_i$ can be calculated by

$$\widehat{r}_i = \frac{R[i, n_i]}{\sum_{k=1}^{k=T_i^L} R[i,k]} \times (r_i \times T_i), \tag{10}$$

2. By referring to Eq. (5), the BS calculates the required transmission power of each MS$_i$ for MCS$_1$ ∼MCS$_6$. We present the required transmission power of MS$_i$ for MCS$_v$ by $P_i(v)$, $v = 1..6$.
3. To achieve the best slot resource efficiency, we first assign each MS$_i$ to use its highest allowable level of MCS, MCS$_{l_i}$, and then allocate the required power $P_i(l_i)$ accordingly.
4. After above steps, the BS repeatedly executes the following procedure to reduce the transmission power of MSs until all $h \times w$ slots are used up.
   (a) For each MS$_i$, $i = 1..m$, calculate the energy consumption $E(i, v)$ for each its allowable MCS$_v$, $v = 1..l_i$. We can derive $E(i, v)$ by

$$E(i,v) = \frac{10^{\frac{\delta(MCS_v)}{10}}(B \cdot N_o + I(i,j)) \cdot L(i,j)}{G_i \cdot G_j} \times \left\lceil \frac{\widehat{r}_i}{rate(MCS_v)} \right\rceil, \quad (11)$$

   where j is the BS.

   (b) For each MS$_i$, $i = 1..m$, calculate the benefit ratio $B(i, l_i, l_i')$, $l_i' = 1..(l_i - 1)$, by lowering down the MCS level of MS$_i$ from MCS$_{l_i}$ to MCS$_{l_i'}$. $B(i, l_i, l_i')$ can be derived by

$$B(i,l_i,l_i') = \frac{E(i,l_i) - E(i,l_i')}{\left\lceil \frac{\widehat{r}_i}{rate(MCS_{l_i'})} \right\rceil - \left\lceil \frac{\widehat{r}_i}{rate(MCS_{l_i})} \right\rceil}. \quad (12)$$

   (c) For all $B(i, l_i, l_i')$, $i = 1..m$, $l_i' = 1..(l_i - 1)$, choose the one with the largest $B(i^*, l_i^*, l_i'^*)$. Then, reduce the transmission power of MS$_{i^*}$ to $P_{i^*}(l_i'^*)$ and change MS$_{i^*}$'s MCS to MCS$_{l_i'^*}$.
   (d) If there's any free slot in the uplink subframe, go back to step (a); otherwise, finish the resource and power allocation of the subframe.

Note that in step 4.(c), it is possible that the total amount of required slots will exceed $h \times w$ after the change of MS$_{i^*}$'s transmission power and MCS. If this situation occurs, we shall give up changing MS$_{i^*}$'s transmission power and MCS and try suboptimal benefit ratio to see if reducing the total transmission power is possible to further reduce the total power consumption of the system.

4 Simulation Results

In this section, we develop a simulator in C++ to verify the effectiveness of our proposed method. The system parameters of our simulator are listed in Table 2. Each MS has an admitted constant bit rate traffic flow. For each flow, its packet size, packet inter-arrival time, and delay bound are randomly generated in the initialization stage of the simulation. The network contains one BS and several MSs. To simulate the movement of MSs, we let each MS roam inside the BS's signal coverage (which is the largest area that the BS can communicate with each MS using the lowest QPSK1/2 MCS) and move following the random waypoint model with the maximal speed of 20 meters per second [10]. In the simulation, we compare our proposed method against the tank-filling scheme [8].

Table 2. The parameters in our simulator

parameter	value
channel bandwidth	10 MHz
FFT size	1024
zone category	PUSC with reuse 1
slot-time	200.94 μ s
uplink frame duration	2.5 ms
uplink subframe space	12 × 30
MCS	Table 1
path loss model	SUI
Tx/Rx antenna gain	BS: 16 dBi/16 dBi; MS: 8 dBi/8 dBi
antenna hight	BS: 30 m, MS: 2 m
thermal noise	-100 dBm
P_i^{MAX}	1000 mW (milliwatt)
threshold ∂	50

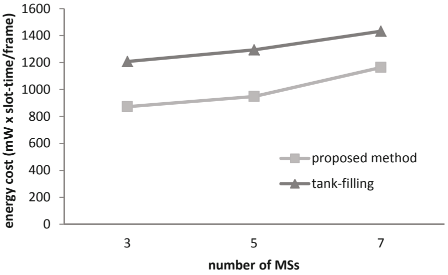


Fig. 3. The energy consumption of MSs under different numbers of MSs

In the experiment, we study the effect of the number of MSs on the average per frame total energy consumption. As shown in Fig. 3, the power consumption increases when the number of MSs increases. Our proposed method always performs better than the tank-filling. But the performances of the two schemes become closer as the number of MSs increases. This is because the total required resource Ω of MSs per T_i raises when the number of MSs increases, thus reducing the available free space and the possibility of further lowering down the transmission power of MSs.

5 Conclusion

In this paper, we propose a cross-layer design for energy efficient sleep scheduling in uplink transmissions of IEEE 802.16 networks such that the overall power consumption of the system is minimized while the QoS of each MS can be guaranteed. Compared to the previous work, we are the first to consider the co-design of both MAC layer sleep scheduling and physical layer power control. With the co-design, we can calculate MSs' power consumptions more accurately and turn off their radio interfaces when not needed. This leads to each MS can deliver their data in low power and conserve more energy. Also, the proposed scheme is compatible to the standard.

Acknowledgments. The authors would like to thank High Speed Intelligent Communication (HSIC) Research Center of Chang Gung University for facility and financial supports and the National Science Council of the Republic of China for financially supporting this research under Contract No. 100-2218-E-024-001-MY3, 101-2923-E-182-001-MY3, and 101-2221-E-182-030.

References

1. IEEE 802.16-2009, IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems, pp. C1-2004 (May 2009)
2. Yoon, J.-P., Kim, W.-J., Baek, J.-Y., Suh, Y.-J.: Efficient Uplink Resource Allocation for Power Saving in IEEE 802.16 OFDMA Systems. In: Proc. IEEE Vehicular Technology Conference (VTC 2008), pp. 2167–2171 (2008)
3. Liang, J.-M., Wang, Y.-C., Chen, J.-J., Liu, J.-H., Tseng, Y.-C.: Energy-efficient uplink resource allocation for IEEE 802.16j transparent-relay networks. Computer Networks 55(16), 3705–3720 (2011)
4. Chen, T.-C., Chen, J.-C., Chen, Y.-Y.: Maximizing unavailability interval for energy saving in IEEE 802.16e wireless MANs. IEEE Transactions on Mobile Computing 8(4), 475–487 (2009)
5. Tsao, S.-L., Chen, Y.-L.: Energy-efficient packet scheduling algorithms for real-time communications in a mobile WiMAX system. Computer Communications 31(10), 2350–2359 (2008)
6. Chen, J.-J., Wu, S.-L., Wang, S.-W., Tseng, Y.-C.: Per-Flow Sleep Scheduling for Power Management in IEEE 802.16e Networks. Computer Networks 55(16), 3721–3733 (2011)
7. Huang, S.-C., Jan, R.-H., Chen, C.: Energy efficient scheduling with QoS guarantee for IEEE 802.16e broadband wireless access networks. In: Proc. Int'l Conf. on Wireless Communications and Mobile Computing (IWCMC 2007), pp. 547–552 (2007)
8. Chen, J.-J., Liang, J.-M., Tseng, Y.-C.: An Energy Efficient Sleep Scheduling Considering QoS Diversity for IEEE 802.16e Wireless Networks. In: Proc. of IEEE Int'l Conf. on Communications (ICC 2010), pp. 1–5 (2010)
9. IEEE 802.16.3c-01/29, Channel models for fixed wireless applications (2003)
10. Bettstetter, C., Resta, G., Santi, P.: The node distribution of the random waypoint mobility model for wireless ad hoc networks. IEEE Transactions on Mobile Computing 2(3), 257–269 (2003)

Supporting Similarity Range Queries Efficiently by Using Reference Points in Structured P2P Overlays

Guanling Lee, Yi-Chun Chen, and Chung Chi Lee

Department of Computer Science and Information Engineering
National Dong Hwa University, Hualien, Taiwan, R.O.C
guanling@mail.ndhu.edu.tw

Abstract. In recent years, the research issues in peer to peer (P2P) systems have been discussed widely. In a P2P system, the role of each node is the same, and the nodes simultaneously function as both clients and servers to the other nodes on the network. Many studies have been proposed for solving different problems to improve the performance of P2P systems. To solve file availability and network flow problems, a method named distributed hash tables (DHT) has been proposed. However, these DHT-based systems are not able to support efficient queries such as similarity queries, range queries, and skyline queries.

In this paper, a novel method for supporting similarity searches in a structured P2P system is proposed. Compared to other existing works, our approach shows great improvement in precision and guarantees the file availability. The experimental results show the effectiveness of our approach.

Keywords: Peer to Peer, Similarity Search, Dimension Reduction, iDistance.

1 Introduction and Related Work

Structured P2P systems, such as CAN [6] and Chord [7], utilize a Distributed Hash Table (DHT) to direct searches to specific node(s) holding the requested data. This architecture mainly concentrates on the key values lookup to improve response efficiency and to guarantee file availability. Since keyword searches are not sufficient, how to support a general query in DHT-based systems has been widely discussed [4][9][10]. In [8] the problem of similarity discovery in a P2P system is discussed and a method called pSearch is proposed. It uses a peer Vector Space Model and peer Latent Semantic Indexing (LSI) to improve the traditional Vector Space Model. [1] proposed a scalable and distributed access structure for similarity searches in metric spaces. All the works discussed above proposed a certain mechanism to reduce the document dimension for supporting similarity queries in a structured P2P overlay. However, the *recall* of the query cannot be guaranteed. *Precision* and *recall* are two key statistics regarding the system's returned results for a query, and are usually used to measure the effectiveness of an information retrieval system. Precision is defined as the proportion of retrieved documents that are relevant and recall is defined as the proportion of relevant documents that are retrieved. In general, there is an inverse relationship between precision and recall.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 645–652.
DOI: 10.1007/ 978-3-642-35452-6_65

To solve the problem of recall, the M-Chord is proposed in [5]. The structure takes advantage of the idea of a vector index method iDistance[2] in order to transform the issue of a similarity search into the problem of an interval search in one dimension. However, it used only one reference point to calculate the response data which results in poor precision.

In this paper, the problem of how to support a similarity query in a structured P2P overlay is discussed. In our approach, by using the idea of iDistance, *m* reference points are selected to reduce a high dimensional document vector into an *m* dimensional vector. Additionally, the document is published in an *m* dimension CAN by mapping the *m* dimensional vector into a specific point in CAN. In query processing, the user's query will be calculated with reference points, which will get *m* bounds in every dimension. Those bounds will be used to fetch a set of object in CAN. The advantages of our work are that the recall and precision of a query can be guaranteed.

The remainder of this paper is organized as follows: The problem definition is described in section 2. Section 3 presents the main idea of our approach. Experimental results and analysis are discussed in Section 4. Section 5 summarizes our work.

2 Problem Definition and Preliminaries

In the similarity search system, each object can be represented by a *k*-element vector and in the vector, each dimension is standing for a term, which is related to the object. The similarity between the objects can be represented by the corresponding vectors. There are several similarity measures in the traditional database. In this paper, Euclidean distance is used to evaluate similarity between objects. The Euclidean distance between objects $X=(x_1, x_2,\ldots, x_k)$ and $Y=(y_1, y_2,\ldots, y_k)$, in Euclidean space, is defined as: $\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots+(x_k-y_k)^2}$. In a similarity discovery system, the range query is defined as Query(q, r) and it will retrieve all objects within distance r to q. Express as $\{x \in objects : dist(q,x) \le r\}$. In general, the number of dimensions is between 80,000 and 110,000. Even if LSI is used to reduce dimensionality, k is still between 50 and 350. To further reduce the dimensionality, the idea of iDistance is proposed. The advantage of iDistance is that the recall of search results can be guaranteed. However, the precision in the system is not good enough. That is, too many unrelated objects are retrieved. To solve the problem, in this paper, a document is mapped into a multi-dimension structure by applying a similar idea of iDistance.

In iDistance, the data space is partitioned into n clusters. For each cluster C_i, the centroid of C_i, said p_i, is selected as its reference point. Each data object is assigned a one-dimensional iDistance value according to the distance between itself and the reference object of the cluster it belongs to. By using a constant c to separate an individual cluster, the iDistance value for an object $x \in C_i$ is $iDist(x) = dist(p_i, x) + i * c$.

Expecting that *c* is large enough, all objects in cluster *i* are mapped to the interval [*i***c*, (*i*+1)**c*]. According to the basic concept of range query, for Query(*q*, *r*), only the clusters whose coverage overlap the query range may contain the satisfied objects. That is, only the cluster C_i which satisfies the inequality $dist(p_i, q) - r \le r_i$, where r_i is the radius of C_i, should be checked. Moreover, because iDistance maps each object into a one dimensional space, the search range of the satisfied cluster C_i can be summarized as [MAX($i*c$, $dist(q,p_i)-r+i*c$), MIN(dist(q,p_i)$+r+i*c$, r_i+i*c)].

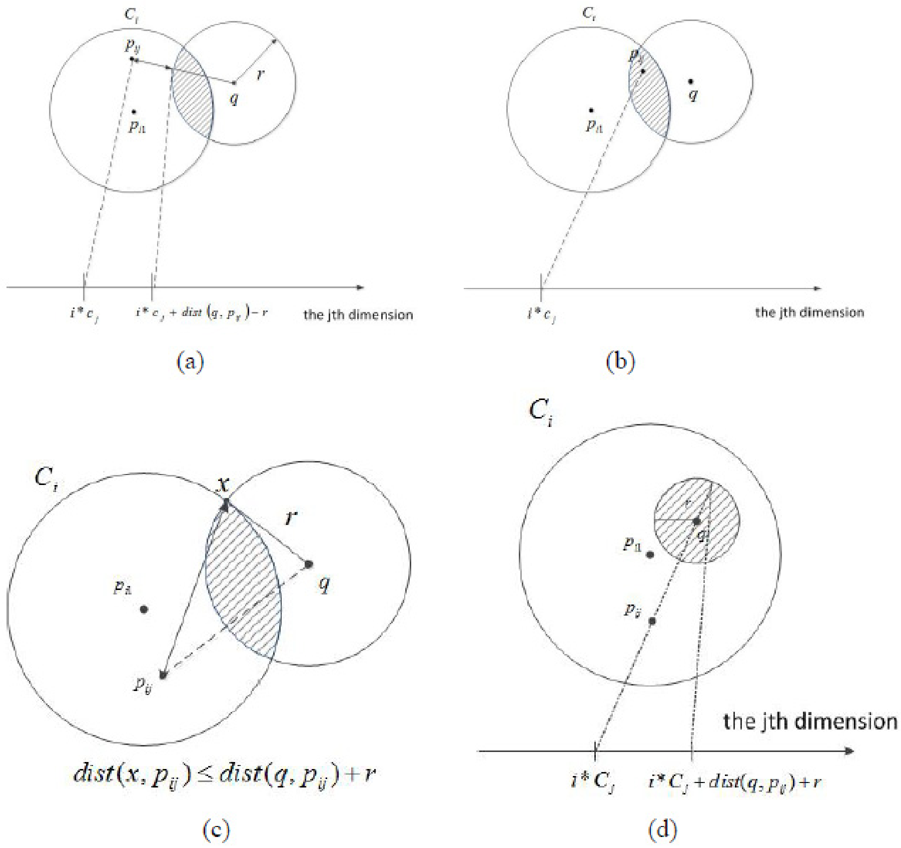


Fig. 1. Search bounds of reference point p_{ij}

Our system is based on CAN [6], which is designed to store and retrieve individual data objects and support exact match queries. To support a similarity search, in our approach the idea of iDistance is extended to map the document into CAN. Moreover, to increase the precision, for each C_i, m reference points $\{p_{i1}, p_{i2},\ldots, p_{im}\}$, where p_{i1} is centroid of C_i, are selected in this approach. For an object x belonging to cluster C_i, the m-dimension vector $< dist(x, p_{i1}) + i * c_1, dist(x, p_{i2}) + i * c_2, ..., dist(x, p_{im}) + i * c_m >$ can be used to represent x in an m dimensional space. In the equation, c_j is a constant used to separate the range covered by the clusters in dimension j. And after the transformation, x can be assigned into m-dimension CAN easily.

To process range query Query(q,r), only the cluster C_i which satisfies the inequality $dist(p_i, q) - r \le r_i$, where r_i is the radius of C_i, should be checked. The lower bound of the search range is the minimum distance from reference point p_{ij} to the overlap region which is $dist(q, p_{ij})$-r+i*c_i (Fig. 1(a)) or i*c_j (Fig. 1(b)). The upper bound of the search range is the longest distance from p_{ij} to the overlap region. Refer to Fig. 1(d), when query range is fully covered by the cluster C_i, the longest distance can be computed as $dist(p_{ij}, p_{i1}) + r_i + i * c_j$. Otherwise, refer to Fig. 1(c), the longest distance is bounded by $dist(q, p_{ij}) + r + i * c_j$. Summarizing the above discussions, the search range for C_i in dimension j, i.e., the dimension corresponding to p_{ij}, can be computed as follows:

$$[MAX(i * c_j, dist(q, p_{ij}) - r + i * c_j), MIN(dist(q, p_{ij}) + r + i * c_j, dist(p_{ij}, p_{i1}) + r_i + i * c_j)] \quad (1)$$

3 Our Approach

3.1 File Publishing

Initially, a distributed cluster method proposed in [3] is used to partition the objects into clusters. After getting the clusters, m reference points will be generated for each cluster. A reference point p_{ij} denotes the jth reference point in cluster i. Moreover, p_{i1} is the centroid of C_i. By using the equation proposed in section 2, a high dimensional object x can be transformed into an m dimensional vector, that is, $< dist(x, p_{i1}) + i * c_1, dist(x, p_{i2}) + i * c_2, ..., dist(x, p_{im}) + i * c_m >$. With the m dimensional vector, object x can be published in an m dimension CAN by using the publish function supported in CAN directly.

3.2 Range Query Processing

To process range query Query(q,r), we first find out the clusters whose coverage overlap the query range. That is, the cluster C_i which satisfies the inequality $dist(p_{i1}, q) - r \le r_i$ is a candidate cluster. And then, for each candidate cluster C_i, equation (1) proposed in section 2 is calculated to get m search bounds. By searching the objects within the search bounds in the corresponding dimension in CAN, the objects satisfy Query(q,r) are obtained. The detailed algorithms are shown in Fig. 2 and 3.

```
Algorithm: Range Query($q$,$r$)
Input: query point $q$
     query range $r$
Output: result S
S={}
For each cluster $C_i$
  If ($dist(p_{i1},q)-r \leqq r_i$)  // $C_i$ overlaps Query($q$,$r$)
    S←S $\cup$ GetClusterCandidate($C_i$,$q$,$r$)
  End If
End For
Return S
```

Fig. 2. Range query

```
Algorithm GetClusterCandidate($C_i$,$q$,$r$)
Input: cluster $C_i$
     query point $q$
     query range $r$
Output:result S
S={}
For each $p_{ij}$  // reference points of $C_i$
// calculate the search bound
  $L_j \leftarrow MAX(i * c, dist(q, p_{ij}) - r + i * c_i)$
  $U_j \leftarrow MIN(dist(q, p_{ij}) + r + i * c, dist(p_{ij}, p_{i1}) + r_i + i * c_i)$
End For
S←Search the range $\langle L_1,U_1\rangle$, $\langle L_2,U_2\rangle$,…,$\langle L_m,U_m\rangle$ in CAN
Return S
```

Fig. 3. GetClusterCandidate

3.3 Reference Points Generation

The precision of the result is truly affected by the reference points. Therefore, how to generate *m* reference points is an important issue. In this approach, we propose three methods to generate reference points. Their efficiency will be discussed in section 4.

Random selection: Select *m*-1 objects in cluster *i* randomly to be the reference points $p_{i2},\ldots, p_{im}$.

Cluster selection: When *m* is 2, the one reference point is the centroid of the cluster, and the other is selected randomly. When *m* > 2, the objects in the cluster are further partitioned into *m*-1 subclusters. The centroids of the subclusters are selected to be reference points.

Coverage selection: When *m* is 2, the one reference point is the centroid of the cluster, and the other is selected randomly. When *m* > 2, the cluster is partitioned into *m*-1 equal coverage space areas. And the centers of the *m*-1 areas are selected to be reference points.

4 Experimental Results

4.1 Simulation Setup

All programs were written in Java and run on a PC with 3.0G Pentium 4 processor and 1G memory. The objects are generated synthetically with normal distribution. The function which is generally used to measure precision in a traditional IR system is shown below. Moreover, because our approach can achieve a 100% recall, we do not show the result here. Table 1 lists the parameters setting of our simulation.

$$\text{Precision}=\frac{\text{Number of items retrieved that are relevant}}{\text{Total number of file retrieved}} \tag{2}$$

Table 1. Parameter settings

	default	range
Number of Documents	100,000	-
Number of Peers in CAN	2,000	-
Number of reference points	4	2~8
Reference points generation methods	Random selection	Random selection Cluster selection Coverage selection

4.2 Experimental Results

Fig. 4 shows the average hop counts for processing range query with different number of reference points. In CAN, average hop-count decreases as dimension increases. Because the routing protocol is based on CAN, Fig. 4 shows the similar result.

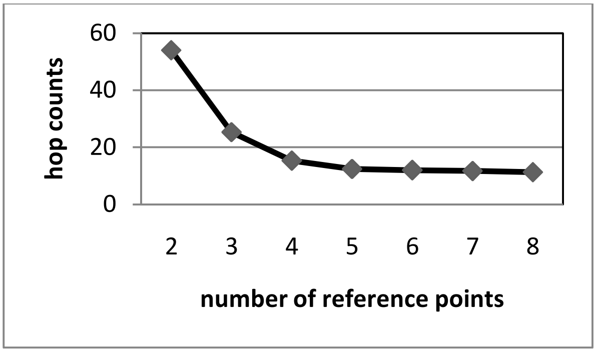


Fig. 4. Hop counts with different dimensions

Fig. 5 shows the precision with different number of reference points. As shown in the result, the precision of random selection and cluster selection methods increase as the number of reference point increases. And they always outperform M-chord. Moreover, random selection method performs almost the same as cluster selection method when the number of reference points larger than 5. This is because increase the number of reference points, the probability that the selected points cover the cluster increase. The precision of coverage selection method is very poor. The reason is that the distribution of objects in a cluster may not be uniform. Therefore, use the centers of the equal partition areas to be the reference points may lead to poor discrimination of the objects in the same cluster.

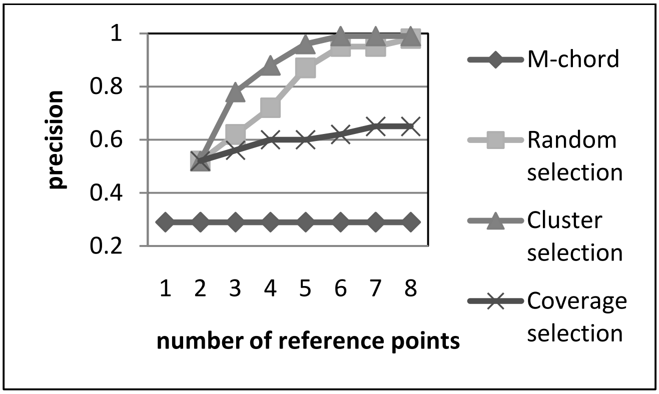


Fig. 5. Precision with different number of reference points

5 Conclusions

In this paper, the problem of how to support a similarity range query method in structured P2P overlays is discussed. In our approach, by using the idea of iDistance, m reference points are selected to reduce a high dimensional vector into an m dimensional vector which can be located in CAN. To process a range query, m search bounds are calculated according to m reference points. By using the bounds, the satisfied objects can be retrieved in CAN efficiently. In the simulation, a set of evaluations is performed to show the benefits of our approach. In comparison with previous approaches, our approach has a great improvement in precision and guarantees file availability.

References

1. Falchi, F., Gennaro, C., Zezula, P.: A Content–Addressable Network for Similarity Search in Metric Spaces. In: Moro, G., Bergamaschi, S., Joseph, S., Morin, J.-H., Ouksel, A.M. (eds.) DBISP2P 2005 and DBISP2P 2006. LNCS, vol. 4125, pp. 98–110. Springer, Heidelberg (2007)
2. Jagadish, H.V., Ooi, B.C., Tan, K.-L., Yu, C., Zhang, R.: iDistance: An adaptive B+-tree based indexing method for nearest neighbor search. ACM Transactions on Database System (TODS 2005) 30, 364–397 (2005)
3. Li, M., Lee, G., Lee, W.-C., Sivasubramaniam, A.: PENS: An Algorithm for Density-Based Clustering in Peer-to-Peer Systems. In: Proceedings of INFOSCALE 2006. IEEE Computer Society, Hong Kong (2006)
4. Mass, Y., Sagiv, Y., Shmueli-Scheuer, M.: KMV-peer: a robust and adaptive peer-selection algorithm. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, Hong Kong, China, pp. 157–166 (2011)
5. Novak, D., Zezula, P.: M-Chord: A Scalable Distributed Similarity Search Structure. In: Proceedings of the 1st International Conference on Scalable Information Systems, New York, USA, pp. 1–10 (May 2006)
6. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content-Addressable Network. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, San Diego USA, pp. 161–172 (August 2001)
7. Stoica, I., Morris, R., Karger, D., Kaashoek, M., Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In: Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, San Diego, USA, pp. 149–160 (2001)
8. Tang, C., Xu, Z., Mahalingam, M.: pSearch: Information Retrieval in Structured Overlays. ACM SIGCOMM Computer Communication Review 33, 89–94 (2003)
9. Tang, Y., Xu, J., Zhou, S., Lee, W.C.: A Lightweight Multidimensional Index for Complex Queries over DHTs. IEEE Transactions on Parallel and Distributed Systems 22(12), 2046–2054 (2011)
10. Tang, Y., Zhou, S., Xu, J.: Light: A Query-Efficient Yet Low-Maintenance Indexing Scheme over Dhts. IEEE Transactions on Knowledge Data Engineering 22(1), 59–75 (2010)

The Universal Local IP Access for Small Cells

Yung-Chun Lin[*] and Chai-Hien Gan

Information and Communications Research Laboratories, ITRI, Hsinchu, Taiwan
{linyj,chgan}@itri.org.tw

Abstract. Today, more than 50% of voice calls and 60% of the wireless traffic data originate in the indoor environment, and to improve the indoor radio coverage and transmission quality becomes a major issue. The deployment of the Small Cells (SCs) could easily and efficiently increase the indoor radio access quality and capacity. The SCs also provide the universal Local IP Access (uLIPA) feature allowing outside UEs connect to and access their home Local IP Access (LIPA) networks from the visited SCs. This paper proposed four schemes to support the uLIPA service. We compared their performance with the well-known prior solution, Virtual Private Network (VPN). Our study indicates that except the IPAM approach, the other uLIPA schemes bring significant overhead for those services with small-sized IP packets, e.g., the popular G.729 VoIP service. This study quantitatively shows that for both radio and wire transmission, the IPAM outperforms the other approaches.

1 Introduction

In recent years, the indoor traffic grows rapidly. Since about 50% of voice calls and more than 60% of the wireless traffic data originate in the indoor environment [4], to improve the radio coverage and transmission quality indoors becomes an efficient way for better performance. To deploy the *Small Cells* (SCs) [1,2,3,13] (generally containing the femtocells and picocells) is a popular solution to easily and efficiently increase the indoor radio access quality and capacity. An SC operates as a low-cost and low-power cellular base station (BS) typically designed for indoor usage (e.g., in a home or small business). Today, the SC mechanism is considered in all the 3G and/or 4G wireless communication systems, such as *Universal Mobile Telecommunications System* (UMTS), CDMA 2000, *Long Term Evolution* (LTE) and *Worldwide Interoperability for Microwave Access* (WiMAX). In this paper, we introduce the SCs based on the UMTS and LTE specification, and the SC *Access Point* (AP) is referred as the Home (e)NodeB (H(e)NB). Figure 1 shows the usage of SC Access environment and usage scenarios [2].

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 653–662.
DOI: 10.1007/978-3-642-35452-6_66

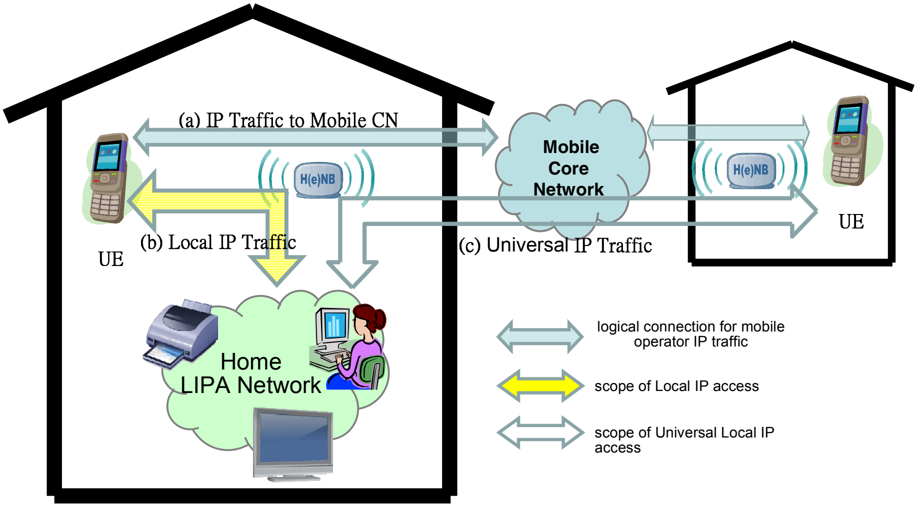


Fig. 1. The Small Cell Access Environment

Besides traditional traffic routing service provided by the *Core Networks* (CNs; see Figure 1 (a)), this Figure further illustrates two characteristic features that are specified as the requirements in 3GPP/3GPP2 specification [1,2,3]: *Local IP Access* (LIPA) and *Universal LIPA* (uLIPA; also referred as the *Remote IP Access* (RIPA)). The LIPA allows local data traffic for subscriber' home LIPA networks (e.g., corporate intranets, residential or enterprise IP networks) or for the public Internet to be directly routed and offloaded by the home H(e)NB (see Figure 1 (b)). On the other hand, the uLIPA capability makes it possible that an outside UE can connect to and access the home LIPA network at his H(e)NBs from a visited H(e)NB (see Figure 1 (c)) . These two characteristic features attract great interest from both academia and industry.

For the LIPA feature, many papers and patents proposed their approaches, and the 3GPP specifications also introduced several schemes to achieve the functionality [4]. However, for the uLIPA aspect, there is no standardized solution proposed in specifications. Fewer papers and patents proposed the well-recognized uLIPA mechanism. Some studies [11,12] suggest a trivial uLIPA solution, the *Virtual Private Network* (VPN) and VPN-like tunnels [6]. For VPN approach, the UE should be one of the tunneling endpoints that encapsulate and decapsulate the uLIPA packets, and the precious radio resources are wasted to carry the additional tunneling headers. Furthermore, the traditional VPN tunneling protocols, such as *IP Security* (IPSec), *Layer 2 Tunneling Protocol* (L2TP) and *Point to Point tunneling Protocol* (PPTP), have to add lots of extra headers to protect the secure and authenticated transmission. Patent [7] proposes both LIPA and uLIPA schemes. For LIPA, the *Local PDN gateway* (L-GW) in home H(e)NB is further equipped with the *serving gateway* (S-GW) data bearer maintainance function to route local packets through the data bearer, where the

L-GW is originally introduced in 3GPP specification for the LIPA services [1,2]. The in-home UE could then directly access the local services through local S-GW in its home H(e)NB (Figure 1 (b)) without the core network involved. For the uLIPA services (as Figure 1 (c)), patent [7] enhances the L-GW in home H(e)NB with the *PDN gateway* (P-GW) IP connectivity functionality to interwork with the S-GW in CN to route and forward user data packets for the outside UE. Then, multiple GTP tunnels would be established across the CN to carry data between the P-GW (in UE's home H(e)NB) and his S-GW, and between the S-GW to his visited (e)UTRAN. Those GTP tunnels through mobile CN not only bring a lot of traffic load and routing overheads, but also cause lots of tunneling overhead for the large-sized GTP encapsulation headers. To reduce the tunneling overheads, we proposed four uLIPA schemes in this paper. The numerical results are provided as well to study the performance for these approaches.

This paper is organized as follows. In Section 2, we first describe a well-known existing uLIPA solution, the VPN approach, and our proposed schemes in details. In Section 3, we introduce the input parameters and output measures for these uLIPA approaches. Section 4 discusses the numerical results to study the performance. In Section 5, we conclude this study. This study quantitatively shows that one of our proposed schemes outperforms the others.

2 The uLIPA Approaches

In this section, we first introduce the existing VPN approach that establishes the VPN tunnel between both endpoints (i.e., UE and his home H(e)NB) to carry the user data from and to UE's home LIPA network. Then, we describe our proposed schemes: the *Enhanced Home Gateway* (eHG) schemes 1 and 2 (referred as eHG1 and eHG2), and the *IP Address Mapping* (IPAM) scheme.

The VPN scheme utilizes the VPN tunnels [6] for uLIPA services. The eHG1 carries the uLIPA traffic by the 3G GTP tunnels, while the eHG2 utilizes the UDP tunnels [5] and IP-in-IP tunnels [8]. The IPAM scheme supports the uLIPA packet transmission by utilizing the *Network Address and Port Translation* (NAPT) technologies [10]. Figures 2-4 illustrate how these approaches work to provide the uLIPA services, where the UE (a) leaves his home H(e)NB (d) and is currently served by a visited H(e)NB (b). Through the uLIPA mechanism, the UE accesses an application server (AS) (e) in his home LIPA network through the visited and home H(e)NBs. Note that the UE has a "home address" (from its home H(e)NB) for the home LIPA network and a "visited address" (from its mobile CN) for visited network and internet access, where the home address could be statically assigned or dynamically obtained when the LIPA service is requested, e.g., the *Packet Data Protocol* (PDP) context activation procedure with a specific LIPA-related *Access Point Name* (APN).

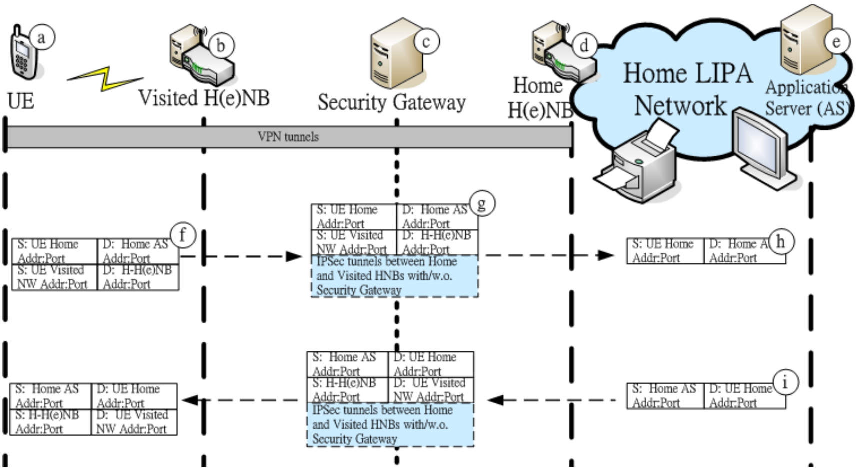


Fig. 2. The VPN Scheme for the uLIPA Service

- **VPN:** As shown in Figure 2, for the visited UE (a) to access his home LIPA network, a VPN tunnel is established between the UE and his home H(e)NB (d) to carry the user data. The VPN tunnel endpoints, UE (a) and his home H(e)NB (d), are responsible for the encapsulation and decapsulation of the tunneled packets. For example, for the packet (f) from the UE to the application server in home LIPA network, its IP address is set from UE home address to the AS address. But before UE sends the packet, the packet is first encapsulated with the VPN header containing both the addresses of the tunnel endpoints (i.e., from: the UE visited address and to: the home H(e)NB's address).

  Note that for all uLIPA schemes, depending on the usage requirements or the physical SCs deployment, the tunneled packet (see Figure 2 (g)) might be further encapsulated by IPSec tunnels and passed through the security gateway (c) between the visited and home H(e)NBs. While the tunneled packet arrives the home H(e)NB, the packet is decapsulated and then forwarded to the AS (see Figure 2 (h)). Similarly, in the vise direction, the packet (i) from the AS to the UE would be first encapsulated by the home H(e)NB and then decapsulated by the UE itself.
- **Enhanced Home Gateway Scheme 1 (eHG1):** The eHG1 is an improvement of the previous home gateway scheme [7]. In eHG1, we respectively import the S-GW functionality in visited H(e)NB and the P-GW functionality in home H(e)NB for the uLIPA service, and the GTP tunnel would be established directly between the UE's visited H(e)NB (b) and home H(e)NB (d) (see Figure 3) through the standard 3GPP PDP context activation procedure. As the GTP tunnel endpoints are the UE's visited and home H(e)NBs, they are responsible for the encapsulation and decapsulation for uLIPA traffic. The encapsulation/decapsulation process is similar to that of VPN, and the details are omitted.

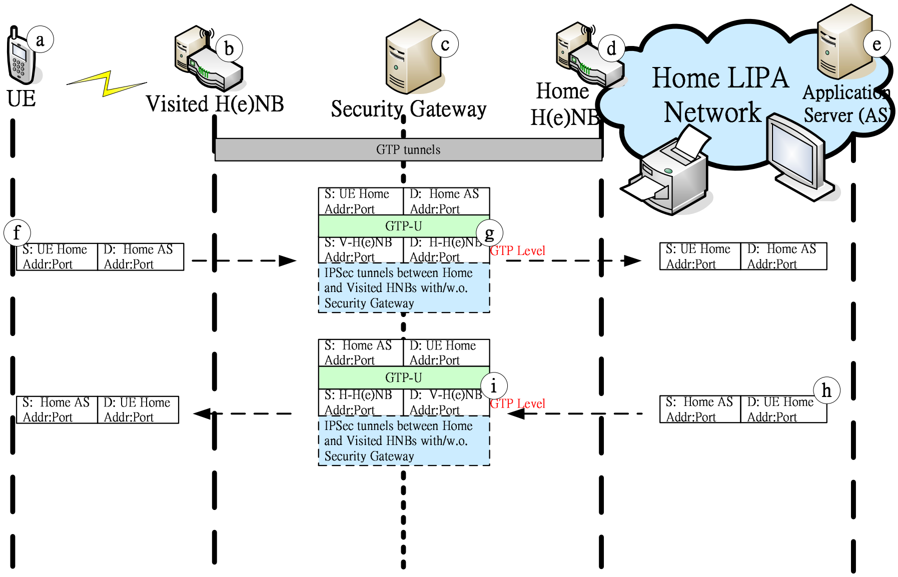


Fig. 3. The Enhanced Home Gateway Scheme 1 (eHG1) for the uLIPA Service

- **Enhanced Home Gateway Scheme 2 (eHG2)**: The eHG2 is a modification for eHG1. Note that the 3GPP GTP packets are carried by the UDP/IP protocol. To further mitigate the tunneling overheads, we provide two kinds of modifications: **eHG2-UDP** and **eHG2-IPIP**. eHG2-UDP approach removes the GTP-U header, and user traffic is carried by the underlying UDP tunnels. eHG2-IPIP approach further skips the UDP protocol from eHG2-UDP and directly carries the uLIPA data by the IP-in-IP tunnel. The eHG2 schemes are similar to eHG1, and the details are omitted.

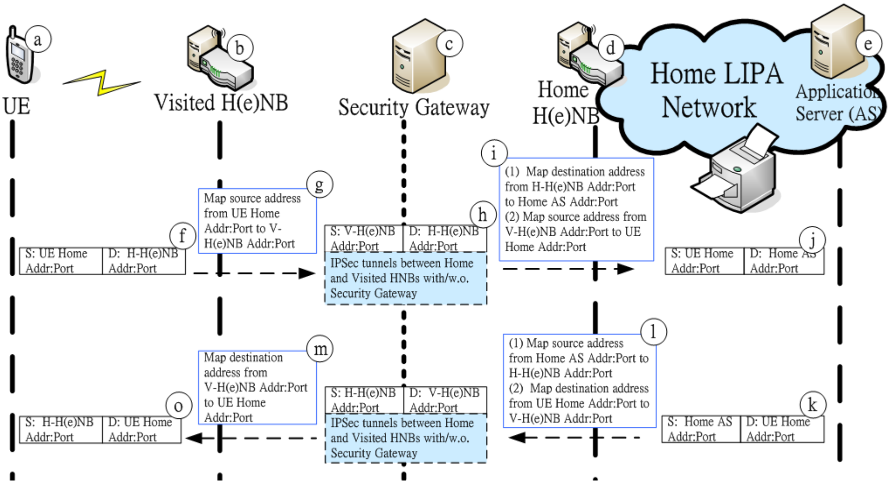

a
UE
b
Visited H(e)NB
c
Security Gateway
d
Home H(e)NB
Home LIPA Network
e
Application Server (AS)
g
Map source address from UE Home Addr:Port to V-H(e)NB Addr:Port
f
S: UE Home Addr:Port
D: H-H(e)NB Addr:Port
h
S: V-H(e)NB Addr:Port
D: H-H(e)NB Addr:Port
IPSec tunnels between Home and Visited HNBs with/w.o. Security Gateway
i
(1) Map destination address from H-H(e)NB Addr:Port to Home AS Addr:Port
(2) Map source address from V-H(e)NB Addr:Port to UE Home Addr:Port
j
S: UE Home Addr:Port
D: Home AS Addr:Port
m
Map destination address from V-H(e)NB Addr:Port to UE Home Addr:Port
o
S: H-H(e)NB Addr:Port
D: UE Home Addr:Port
S: H-H(e)NB Addr:Port
D: V-H(e)NB Addr:Port
IPSec tunnels between Home and Visited HNBs with/w.o. Security Gateway
l
(1) Map source address from Home AS Addr:Port to H-H(e)NB Addr:Port
(2) Map destination address from UE Home Addr:Port to V-H(e)NB Addr:Port
k
S: Home AS Addr:Port
D: UE Home Addr:Port

Fig. 4. The IPAM Scheme for the uLIPA Service

- **IPAM:** As shown in Figure 4, the IPAM approach makes some mappings and translations to the packets' IP addresses as follows. In this scheme, the home H(e)NB acts like but more than an NAT server. To send the uLIPA packet (see Figure 4 (f)) to an AS (e), the visited UE set the destination address as a specific home H(e)NB's address:port pair that represents specific application in AS (e) behind the home H(e)NB. When the visited H(e)NB receives the packet whose source address is UE's home address (i.e., the uLIPA packet), it replaces the source address by its own address (i.e., visited H(e)NB's address; see Figure 4 (g)) and send the packet (h) to the home H(e)NB. When the home H(e)NB receives the packet (h), it first maps the destination address:port pair to the correspondent AS address:port pair (see Step 1 in Figure 4 (i)), and then the source address translation is made from the visited H(e)NB address:port to the UE home address:port (Figure 4 (i) Step 2). Finally, the packet (j) is forwarded the AS. In the vise direction, the packet address of the packet (k) from the AS to the UE would be first translated by the home H(e)NB, i.e., the source address:port is mapped from AS address:port to home H(e)NB address:port (Step 1 at Figure 4 (l)), while the destination addresses:port is mapped from UE home address:port to visited H(e)NB address:port (Step 2 at Figure 4 (l)) that presents the UE. When the packet is delivered to the visited H(e)NB, the visited H(e)NB maps the destination address:port to the correspondent UE home address:port (Figure 4 (m)), and sent back to the UE.

3 Input Parameters and Output Measures

This section describes the input parameters and output measures for the uLIPA approaches. The following two input parameters are considered.

- **The Packet Size *k*.** *k* is the original IP packet size for the uLIPA traffic data that is transferred between the home LIPA network and the UE through a visited H(e)NB.
- **The Transmission Overhead** *h*. *h* is the additional transmission overhead for the delivery of the uLIPA traffic data between the UE and his home H(e)NB. As shown in Figure 5, the value of transmission overhead *h* depends on the uLIPA approaches:
  - **VPN:** For VPN uLIPA scheme, the VPN tunnel is established to carry the uLIPA traffic. The most popular VPN solution contains the IPSec, L2TP and PPTP technologies. To encapsulate an IP packet, the IPSec tunnel (see Figure 5 (a.1)) adds at least 50-byte size IPSec tunneling headers that contain the new IP header (20 bytes), the *Encryption Security Protocol* (ESP) header (8 bytes), the ESP *Initialization Vector* (IV; 8 bytes), the ESP Trailer for Padding (2 - 257 bytes) and the ESP authentication data (AUTH; 12 bytes). The L2TP tunnel (see Figure 5 (a.2)) brings 40-byte size extra headers that contain the new IP header (20 bytes), the new UDP header (8 bytes), the L2TP header (8 bytes) and the PPP header (4 bytes). The PPTP

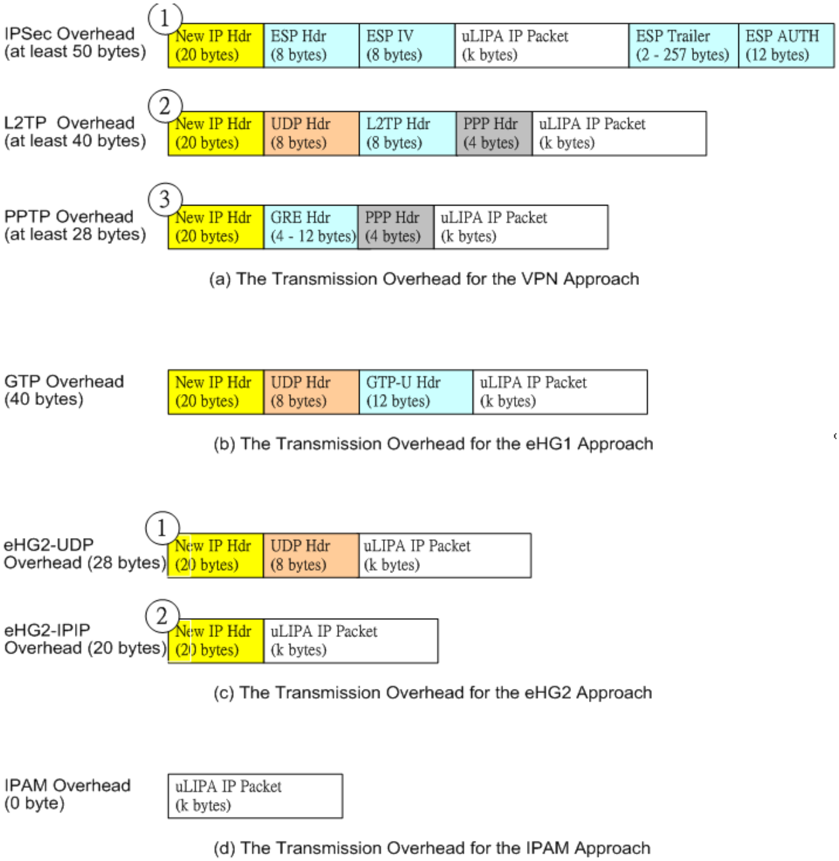


Fig. 5. The Transmission Overhead for the uLIPA Approaches

tunnels have 28 - 40 bytes additional tunneling overhead for the new IP header (20 bytes), the GRE header (4 bytes at least and usually 8 - 12 bytes), and the PPP header (4 bytes). To archive the best performance, the PPTP tunnel with smallest tunneling overhead (i.e., 28 bytes) is considered for the VPN uLIPA scheme (see Figure 5 (a.3)).

- **eHG1:** In eHG1, the S-GW and the P-GW are respectively assigned in visited H(e)NB and in home H(e)NB while providing the uLIPA service, and the GTP tunnels are established between the UE's visited and home H(e)NBs. In this case, each uLIPA IP packet would be encapsulated with extra 40-byte GTP tunneling headers (i.e., 20-byte IP header, 8-byte UDP header and 12-byte GTP-U header; see Figure 5 (b)).
- **eHG2:** The eHG2 is a modification of eHG1 by removing the GTP-U header (i.e., eHG2-UDP in Figure 5 (c.1)) or further removing both the GTP-U and UDP headers (eHG2-IPIP in Figure 5 (c.2)), and respectively results in 28-byte and 20-byte tunneling overhead for each uLIPA packet.

- **IPAM:** Instead of establishing the tunnels, the IPAM scheme makes mappings and translations to the source and the destination addresses for uLIPA packets, and no extra tunneling headers are needed (Figure 5 (d)).

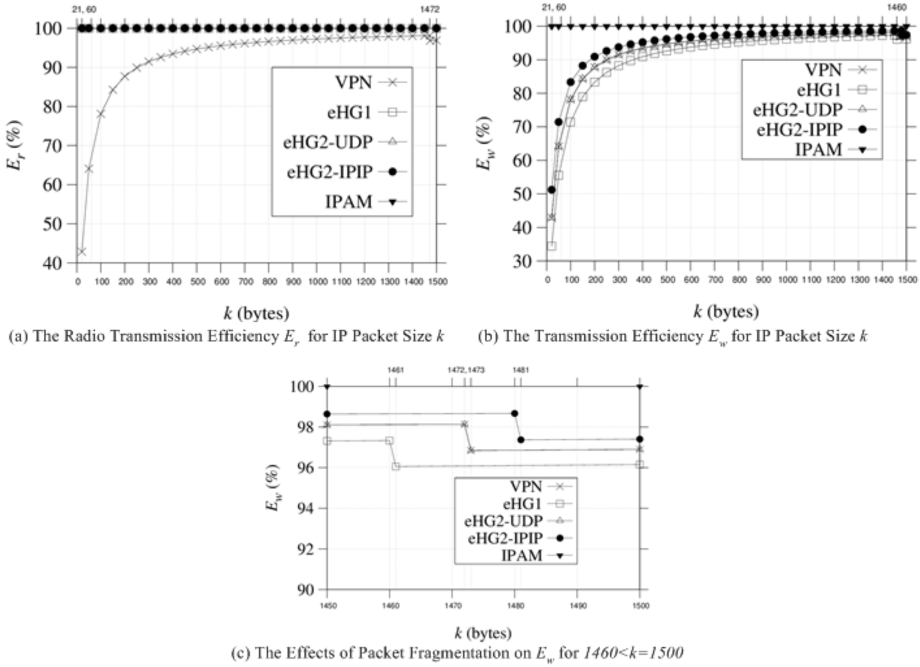
To study performance of these uLIPA approaches, the transmission efficiency E is defined as follows:

$$E = \frac{packet\_size \quad k}{packet\_size \quad k + transmission\_overhead \quad h} \times 100\% \cdot \qquad (1)$$

We further define the efficiency E_r for radio transmission and the efficiency E_w for the wire (conductive) transmission through the internet, cables or other network infrastructures. Then we numerically compare both transmission efficiencies in Section 4.

4 Results Comparison and Discussion

In this section, we compare the performance for the uLIPA schemes (VPN, eHG1, eHG2 and IPAM schemes) in terms of the transmission efficiencies E_r and E_w (equation (1)). Note that because the maximum Ethernet packet size is 1500 bytes, the IP packets with the sizes larger than 1500 bytes are always fragmented to avoid the Ethernet segmentation. Therefore, we could consider the IP packet size from 21 (including the 20-byte IP header) to 1500 for the uLIPA packets (see Figure 6).



(a) The Radio Transmission Efficiency E_r for IP Packet Size k

(b) The Transmission Efficiency E_w for IP Packet Size k

(c) The Effects of Packet Fragmentation on E_w for *1460<k=1500*

Fig. 6. The Transmission Efficiencies E_r and E_w for IP Packet Size k

The Effects of Packet Size *k* on Radio Transmission Efficiency E_r. Figure 6 (a) plots the radio transmission efficiency E_r as a function of IP packet size k for all uLIPA approaches. As the packet size k increases, E_r of VPN approach increases for $21 \le k \le 1472$, while the other approaches have the same E_r with fixed value *100%*. Each uLIPA packet is added with a 28-byte VPN header during transmission over the VPN tunnel. From equation (1), we know that the smaller the packet size k, the more significant the effect of the tunneling overhead. As k increases, the effect of VPN tunneling header becomes insignificant, and E_r increases. For $k \ge 1473$, the encapsulated uLIPA IP packet exceeds 1500 bytes (e.g., $1473+28=1501$) and should be fragmented. *1473* is referred as the fragment k value for VPN scheme. It requires another 20-byte IP header to carry the fragmented packet, and therefore E_r first decreases at $k=1473$ and then increases when k increases as described above. The other uLIPA approaches bring no extra overheads for the radio transmission, and E_r =*100%* is better than the VPN approach.

The Effects of Packet Size *k* on Wire Transmission Efficiency E_w. Figure 6 (b) depicts the wire transmission efficiency E_w as the function of IP packet size k for all uLIPA approaches. For IPAM approach, the E_w=*100%* for all k values because the IPAM approach brings no extra tunneling headers for uLIPA packets. For the other approaches, as k increases, the E_w first increases, then decreases at their correspondent fragment k values (i.e., $k=1461$ for eHG1, $k=1473$ for both VPN and eHG2-UDP, and $k=1481$ for eHG2-IPIP; see Figure 6 (c)), and finally increases. Figures 6 (b) and (c) also show that E_w(IPAM)> E_w(eHG2-IPIP)> E_w(eHG2-UDP),E_w(VPN)> E_w(eHG1). The IPAM approach outperforms the other approaches.

Figure 7 compares the uLIPA approaches on efficiencies E_w for both the web traffic and the popular VoIP streaming services. The result is constructed based on the VoIP protocols and the web traffic statistics in study [9], and the details are omitted.

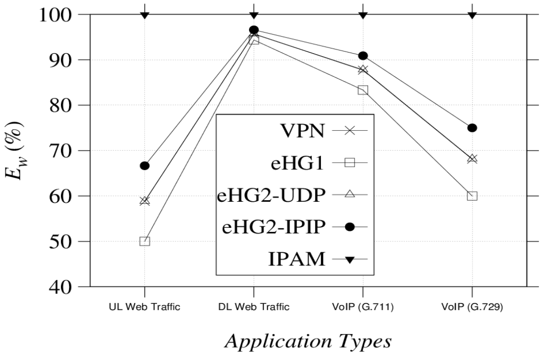


Fig. 7. The E_w for the Web Traffic and VoIP Streaming Services

5 Conclusions

This paper studied the efficiencies for both radio and wire transmission for the uLIPA approaches: VPN, eHG1, eHG2-UDP, eHG2-IPIP and IPAM. For the radio transmission, only the VPN approach brings extra overhead and results in a worst E_r. For the wire transmission efficiency, the smaller the packet size k, the more significant the header overhead for the tunneling uLIPA approaches (VPN, eHG1 and eHG2). Therefore, the tunneling uLIPA approaches would bring significant overhead for those services with small-sized packets (e.g., G.729). This study quantitatively shows that for both radio and wire transmission, the IPAM outperforms the other approaches.

Acknowledgements. This paper was supported in part by the ITRI/NCTU JRC Research Project, the ITRI Advanced Research Program under B301EA3300, B301AR2R10, and B352BW1100.

References

1. 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Local IP Access and Selected IP Traffic Offload (Release 10). 3GPP TS 23.829 V10.0.1 (2011)
2. 3GPP. 3rd Generation Partnership Project; Technical Specification Group Services and Sys-tem Aspects; Service requirements for Home Node B (HNB) and Home eNode B (HeNB) (Release 11) 3GPP TS 22.220 V11.0.0 (2010)
3. 3GPP2. cdma2000 Femtocell Network: Packet Data Network Aspects. 3GPP2 X.S0059-100-0 V1.0 (January 2010)
4. Chandrasekhar, V., Andrews, J., Gatherer, A.: Femtocell Networks: A Survey. IEEE Communication Magazine 49(9), 59–67 (2008)
5. Levkowetz, H., Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices. IETF RFC 3519 (April 2003)
6. Metz, C.: The latest in virtual private networks: part I. IEEE Internet Computing 7(1), 87–91 (2003)
7. Parsons, E.W., Reininger, P., Stojanovski, S., Barnowski, B.: Ubiquitous Access to femto-connected network. U.S. Patent Application (2011)
8. Perkins, C.: IP Encapsulation within IP. IETF RFC 2003 (October 1996)
9. Reyes-Lecuona, A., González, E., Casilari, E., Casasola, J.C., Díaz-Estrella, A.: A Page-Oriented WWW Traffic Model for Wireless System Simulations. In: Proc. of International Teletraffic Congress (ITC-16), Edinburgh (UK), vol. 3.b, pp. 817–826 (June 1999)
10. Srisuresh, P., Egevang, K.: Traditional IP: Network Address Translator (Traditional NAT). IETF RFC 3002 (January 2001)
11. Tinnakornsrisuphap, P., Palanigounder, A., Jayaram, R.S., Dondeti, L.R., Wang, J.: Remote access to local network. U.S. Patent Application (2010)
12. Tinnakornsrisuphap, P., Palanigounder, A., Jayaram, R.S., Dondeti, L.R., Wang, J.: Remote access to local network via security gateway. U.S. Patent Application (2010)
13. Small Cell Forum. Interference Management in UMTS Femtocells (2010)

Implementation of Radar Map Using GPS in Vehicular Networks

Chun-I Kuo[1], Po-Ching Wang[1], Chih-Hsun Lin[1], Ce-Kuen Shieh[1],
and Ming-Fong Tsai[2]

[1] Institute of Computer and Communication Engineering,
Department of Electrical Engineering, Nation Cheng Kung University, Taiwan
{chuni.kuo,willydog029,fashionshow0}@gmail.com,
shieh@ee.ncku.edu.tw

[2] ICT Design and Validation for Vehicles Department, Division for Telematics and
Vehicular Control System, Information and Communications Research Laboratories,
Industrial Technology Research Institute, Hsinchu, Taiwan
mingfongtsai@gmail.com

Abstract. In the highway environment, many applications require localization. It is common to use the Global Positioning System (GPS) to find the current position, but GPS is easily influenced by the external environment and has many internal errors. Dead Reckoning (DR) is a common method for correcting the GPS position. However, the related applications do not have a notice system that can show the number of and distances to neighboring vehicles. In this paper, we design a map application on the Android operating system for vehicular networks. In this map application, the road map would be shown on a monitor, and the neighboring vehicles could also be drawn on this map. Users can easily understand the road conditions. Moreover, the application can exchange traffic information with other vehicles, and the map can also show the vehicles which are out of range. Furthermore, the application can show the time to collision between two vehicles. The experimental results will show the adjustment of position corrected by DR on Google Map, and the radar map on the Android operating system.

1 Introduction

With the deployment of Dedicated Short-Range Communications (DSRC) and wireless technologies such as 802.11p, a vehicle can exchange traffic information with other vehicles through wireless communication and form a vehicular network [1]–[3]. When you have lost your way, location is quite important and will usually be obtained through Global Positioning System (GPS) equipment. GPS equipment, through the use of satellites as reference positions, provides absolute vehicle positions. But its accuracy and availability are affected by the satellite geometry, selective availability, environmental effects, and propagation delay through the ionosphere or any other complex unknown factors [4]–[6].

The Dead Reckoning (DR) method is an often used vehicle positioning technique [7]–[9]. In this method, heading and distance sensors are used to measure the displacement vectors, which are then used in a recursive manner to determine the current

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 663–672.
DOI: 10.1007/978-3-642-35452-6_67

vehicle position. However, the accuracy of the DR method constantly degrades over time, since the errors in the measurements accumulate and affect the current position estimation. For vehicles travelling on a road network, the map-matching technique is often used to correct positioning errors. In this method, the vehicle's travelling path is constantly compared with the road network. Through the matching process, the most likely location of the vehicle with respect to the map is determined. Several papers devoted to achieving accuracy in vehicle positioning have integrated GPS and DR, and applied this with the map-matching technique. The estimation of the actual vehicle location from the GPS position together with the use of the road information contained in the digital map using the spatial relationship between the GPS position and the roads has performed well [10]. However, there is as yet no application which is able to show the distance between two vehicles using the idea of a radar map which is constantly displayed: it is hard for the existing methods to draw locations and distances accurately on a map since the GPS is always jumping and the error is about 5–10 meters.

The present paper proposes a radar map based on the Android operating system, one which is able to show the locations of the neighboring vehicles in the vehicular network. The speed and heading information from the National Marine Electronics Association (NMEA) are used to minimize the GPS measurement errors which may result from signal noise or changes in the satellite geometry. Hence, in addition to providing the relative distance between other vehicles, an adaptive correction algorithm is proposed to minimize the cumulative position errors. Moreover, the warning of time to collision (TTC) is very important for the driver. The application, on the Android operating system, can show the TTC between two vehicles. In the experimental results presented, the radar map shows the neighboring vehicles, and this helps decrease the chances of a car accident due to dead line of sight.

The remainder of this paper is organized as follows. In Section 2, we present the implementation of our radar map. The experimental results and performance evaluations are given in Section 3. The conclusions are given in Section 4.

2 The Implementation of the Radar Map System

2.1 The Industrial Technology Research Institute's WAVE/DSRC Communications Unit (IWCU)

The Industrial Technology Research Institute (ITRI) WAVE/DSRC Communications Unit (IWCU) [11] is an integrated wireless communication system designed for deploying Intelligent Transportation System (ITS) applications and improving traffic safety on roadways, and is shown in Fig. 1. The IWCU device not only supports the IEEE 801.11p/1609 standard but also integrates the 3G and Wi-Fi protocols. With these capabilities, it can support many different types of communications, including intra-vehicle, vehicle-to-vehicle (V2V), and vehicle-to-roadside (V2R), and can support additional vehicle-to-infrastructure (V2I) communications. The IWCU device

Fig. 1. ITRI WAVE/DSRC communication unit

also provides measurements of the GPS, such as the heading and speed of a vehicle. Moreover, the IWCU also provides a software development kit (SDK). Using the SDK tools, many ITS applications can be developed easily on the flexible open platform.

2.2 Dead Reckoning

On a two-dimensional planar space for vehicular travel, it is possible to calculate a vehicle's position at any time, provided that the starting location and all subsequent displacement vectors are available. The DR method is a popular technique. The equation of the vehicle's position (x_k, y_k) at time t_k can be expressed by

$$x_k = x_0 + \sum_{i=0}^{k-1} s_i \cos\theta_i \,, y_k = y_0 + \sum_{i=0}^{k-1} s_i \sin\theta_i \tag{1}$$

where (x_0, y_0) is the initial vehicle location at time t_0, and s_i and θ_i are, respectively, the length and the absolute heading of the displacement vector from the vehicle's position (x_i, y_i) at time t_i to (x_{i+1}, y_{i+1}) at time t_{i+1}. In Fig. 3, the relative heading is defined as the difference between the absolute headings at two consecutive instances and is denoted by ω_i. Given the relative heading measurements ω_i for time $t_0, t_1, \dots, t_k$ the absolute heading of the vehicle θ_k, at time t_k, can be calculated from $\theta_k = \sum_{i=0}^{k} \omega_i$. As shown in Equation (1), the dead reckoning method is a cumulative process. Consequently, all previous sensor errors will be accumulated, and this decreases the accuracy of the current calculated position. As we all know,

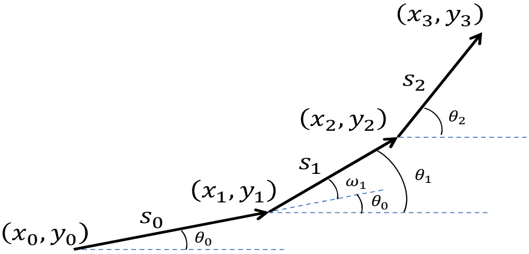


Fig. 2. Dead reckoning method

GPS often jumps because of multipath interference and changes in the satellite constellation. The calculation is accompanied by greater errors as time passes. The gap between the current position and the estimated position will be greater when a small error is included in the calculated heading and the mileage. Many researchers use the DR method when the GPS signal is cut off. However, we use the speed and heading information from the GPS to implement the DR method in this paper.

2.3 GPS

The Global Positioning System (GPS) is a satellite based radio navigation system realized by the U.S. Department of Defense [12]. When the GPS is fully operational, there will be a constellation of 24 Navstar satellites in six orbital planes. The satellite orbits will be arranged so that a minimum of 5 satellites are visible to the user anywhere in the world at all times. Whether 2-dimensional or 3-dimensional navigations, the GPS receiver selects a set of at least 3 or 4 satellites that forms the best satellite geometry, and calculates the distances between the receiver and the satellites. Given these distance measurements, the user's position with respect to the earth's inertial coordinates can be obtained [13]. However, it also possesses unexpected effects: large position errors may cause the GPS position received in the receiver to be inaccurate when the GPS satellite geometry and visibility are less than perfect. For road vehicles that constantly travel in urban areas where tunnels, overpasses, tall buildings, and trees are abundant, perfect reception conditions are rarely possible. Even when there are enough satellites available for the position calculation, the satellite geometry may not be good enough to obtain an accurate position.

2.4 Error Revision

It has become clear that a superior positioning system can be obtained if the advantages of the dead reckoning system and the GPS can be combined. The absolute positional accuracy of GPS can be used to provide feedback to calibrate the dead reckoning system, while the smoothness of the dead-reckoning signals can be used to correct the errors of the GPS position signals due to multi-path interference or other problems [14]. There have been several approaches that have combined GPS and dead reckoning. This integration is usually done either through a switching algorithm that switches from one system to another if the latter has a better signal condition. In such approaches, the integrated system usually performs as a stand-alone system if either the GPS signal or the dead reckoning signal is absent [15]–[18]. Given the availability and accuracy problems with GPS signals, an integrated system uses dead reckoning as its primary positioning technique. In our designed module, when the GPS signals are available, they will be used not only to correct the drifted dead reckoning position, but also to calibrate the dead reckoning sensors. When the GPS is not functioning well, the calibrated dead reckoning system will provide better performance than before.

2.5 The Implementation of Radar Map

The radar map is designed on the Android system, a Linux-based operating system for mobile devices, such as smartphones and tablet computers. It was developed by the Open Handset Alliance, led by Google. The error revision algorithm is designed in the IWCU SDK, and the system architecture is shown in Fig. 3. When the IWCU receives signals which contain GPS headings and speeds, the error revision will use the dead reckoning algorithm to calibrate any jumping GPS signals. Moreover, the IWCU will send the new position to the Android system through 802.11g. After receiving the new position, the mobile application is designed to transfer the position to a pixel system and plot a radar map in the Android operating system. The experimental evaluation will discuss the difference between the radar map with and without the DR method. The users can input the GPS position in Google Map, which points to the nearest position on the road. In the next section, the experimental evaluation will show the Google Map results and the radar map warning messages on an Android simulation.

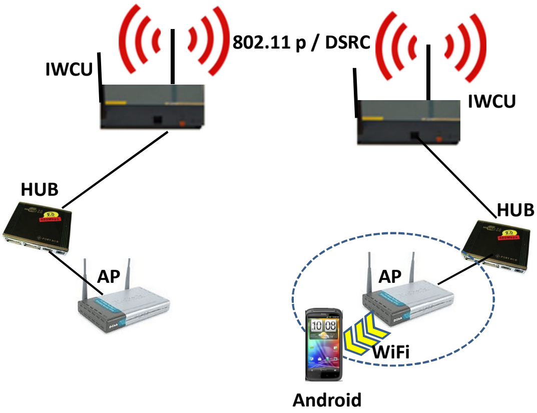


Fig. 3. System architecture

In Fig. 4, the IWCU can get the GPS information, such as latitude, longitude, speed, and heading, every 100 milliseconds. By comparing this with the history of the GPS information, the system will detect whether the GPS is reasonable or not. If the GPS information is not reasonable, the system will use a simple dead reckoning algorithm to adjust it by comparing it with the history of the GPS data, and broadcast the adjusting GPS information to nearby vehicles every 100 milliseconds. After receiving, the IWCU will send the information (including id, latitude, longitude, speed, heading, and time) to the Android system through Wi-Fi. In order to achieve real-time service, the Android system can distinguish who is sending the GPS information by unequal identification, and compute each data by thread. Using the received data, the relative distance (in the Cartesian coordinate system) from the sender to the receiver can be calculated, and the system can translate the suitable scale by the road information. In our experiment, the radar map can be drawn on the Android system in real-time, and the more detailed experimental results will be discussed in the next section. Moreover, in order to discuss the performance of the DR algorithm, the caution function is also implemented in our system. The warning function will notify the driver based on the speed difference, direction, and distance. When a red “Dangerous” is shown on the Android screen, the driver will know that another vehicle is too close. On the other hand, “Safe” means that there is a safe state with respect to the other cars.

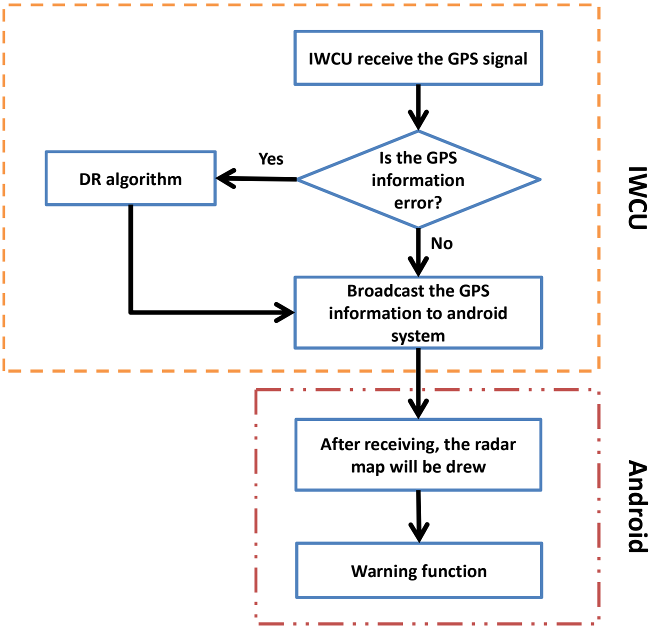

IWCU receive the GPS signal
Is the GPS information error?
Yes
DR algorithm
No
Broadcast the GPS information to android system
After receiving, the radar map will be drew
Warning function
IWCU
Android

Fig. 4. Flow chart

3 Experimental Evaluation

3.1 Experimental Setup

In Fig. 5, each car (car 1 and car 2) implements the DR method to correct the position error as described in Section 2.2. Each car will broadcast its own GPS information and then the other cars will receive this information to provide a radar map to the driver as described in Section 2.5. The experimental area was in Ximen Rd., Tainan, Taiwan. The average speed of car 1 and car 2 was 35 km/h. The WAVE/DSRC device was the IWCU 3.0 as described in Section 2.1.

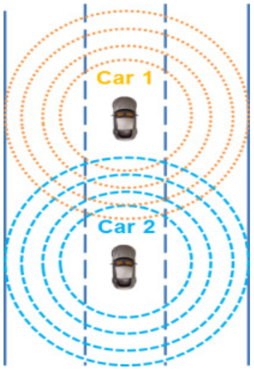
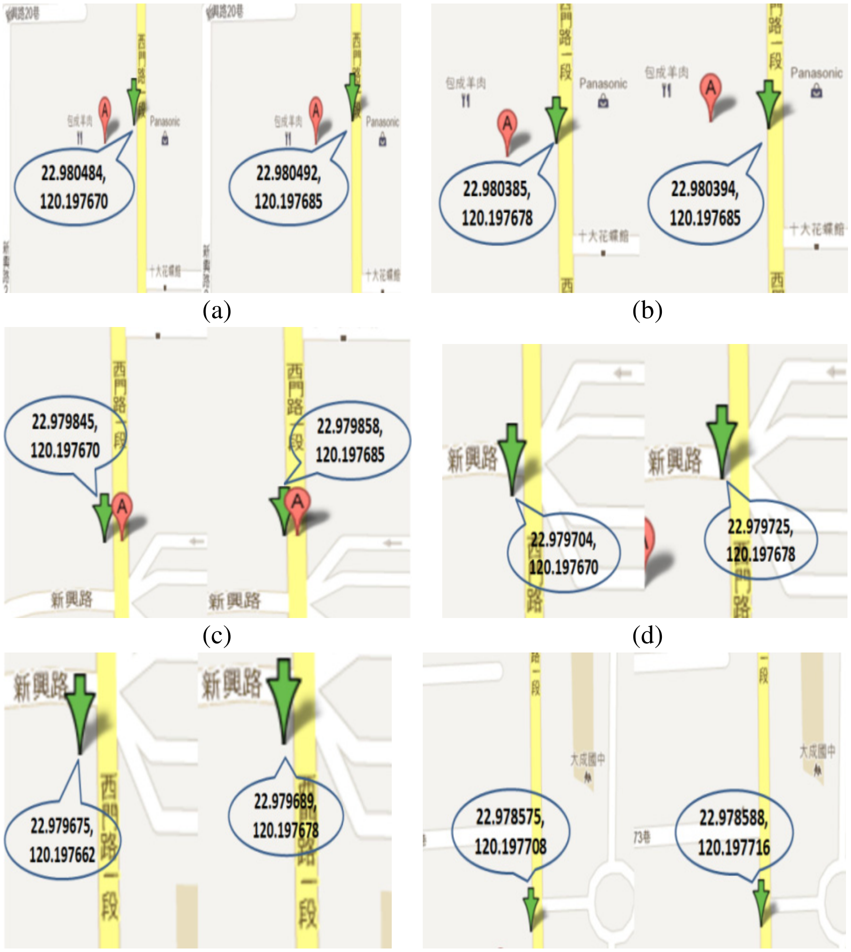


Fig. 5. Experimental topology



(a) (b)

(c) (d)

(e) (f)

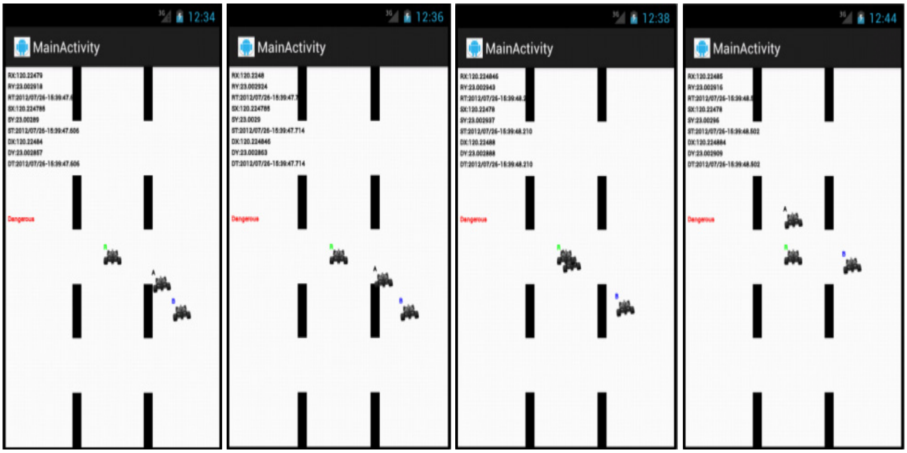
Fig. 6. Performance of DR method

3.2 Performance of GPS DR

We present the DR performance in this section. Fig. 6 (a), Fig. 6 (c) and Fig. 6 (e) are without DR. Fig. 6 (b), Fig. 6 (d) and Fig. 6 (f) are with DR. Many researchers have used gyroscope and accelerometer hardware to implement the DR method. In this paper, we used the speed and heading of the GPS information to implement a software DR method. Hence, we based it on the speed and heading from the GPS information to modify the position. We found that the speed and heading of the GPS information can help the GPS position as shown in Fig. 6. We can calculate the distance between two GPS values [19] by website [20]. Hence, we can obtain a DR-enhanced performance as follows. With DR, the position is enhanced 1.8 m, 1.2 m, 2.1 m, 2.5 m, 2.3 m, and 1.7 m, over the position without DR, in Fig. 6.

3.3 Performance of Radar Map

We present the warning message performance in this section. The cars A and B are the same as in Fig. 7. Car A is without DR and car B is with DR. In this scenario, the receiver car always drives in the middle lane and car A (car B) always drives in the right lane. In Fig. 4, "Dangerous" is shown on the Android screen, which means another vehicle is very close. However, car A (without DR) has the wrong GPS information for warning the driver. Hence, the driver (receiver) receives a false alarm. In Fig. 7 (a) and Fig. 7 (b), car A has been very close. In Fig. 7 (c) and Fig. 7 (d), car A has been positioned in another lane.



```
RX:120.22479
RY:23.002918
RT:2012/07/26-15:39:47.6
SX:120.224785
SY:23.00289
ST:2012/07/26-15:39:47.606
DX:120.22484
DY:23.002857
DT:2012/07/26-15:39:47.606
Dangerous
```

(a)

```
RX:120.2248
RY:23.002924
RT:2012/07/26-15:39:47.7
SX:120.224785
SY:23.0029
ST:2012/07/26-15:39:47.714
DX:120.224846
DY:23.002863
DT:2012/07/26-15:39:47.714
Dangerous
```

(b)

```
RX:120.224846
RY:23.002943
RT:2012/07/26-15:39:48.2
SX:120.22479
SY:23.002937
ST:2012/07/26-15:39:48.210
DX:120.22488
DY:23.002888
DT:2012/07/26-15:39:48.210
Dangerous
```

(c)

```
RX:120.22485
RY:23.002916
RT:2012/07/26-15:39:48.5
SX:120.22478
SY:23.00296
ST:2012/07/26-15:39:48.502
DX:120.224884
DY:23.002909
DT:2012/07/26-15:39:48.502
Dangerous
```

(d)

Fig. 7. Performance of warning message

4 Conclusions

This paper proposed a map application in the Android operating system. A road map is shown on a monitor and the neighboring vehicles are drawn on this map. Users can easily understand real-time road conditions. The proposed application can exchange traffic information with the other vehicles, hence, the map can also show vehicles which are out of range. Moreover, the proposed application can show the TTC between two vehicles, relying on software dead recknoning (DR). The experimental results have shown that adjusting the GPS position by corrections from DR can enhance the accuracy by from 1.7 m to 2.3 m. Moreover, the proposed radar map can support a collision warning which can avoid false alarms by using the software DR method.

References

1. Fernandez, J., Borries, K., Cheng, L., Kumar, B., Stancil, D., Bai, F.: Performance of the 802.11p physical layer in vehicular-to-vehicular environments. IEEE Transactions on Vehicular Technology 61(1), 3–14 (2012)
2. Hassan, M., Vu, H., Sakurai, T., Andrew, L.: Effect of retransmissions on the performance of the IEEE 802.11 MAC protocol for DSRC. IEEE Transactions on Vehicular Technology 61(1), 22–34 (2012)
3. Hsieh, T., Tseng, Y.: Vehicular traffic information dissemination for WAVE/DSRC networks. In: IEEE International Symposium on Wireless and Pervasive Computing, pp. 1–6 (2011)
4. Alam, N., Balaei, A., Dempster, A.: A DSRC Doppler-based cooperative positioning enhancement for vehicular networks with GPS availability. IEEE Transactions on Vehicular Technology 60(9), 4462–4470 (2010)
5. Lim, H., Chung, T.: Performance evaluation of relative positioning based on low-cost GPS and VANET. In: IEEE International Conference on Internet Technology and Secured Transactions, pp. 542–543 (2011)
6. He, Z., Ma, Y., Tafazolli, R.: Cooperative localization in a distributed base station scenario. In: IEEE Vehicular Technology Conference, pp. 1–5 (2011)
7. Cho, B., Koo, J., Yoon, B., Kim, B.: The research of dead reckoning stabilization algorithm using different kinds of sensors. In: IEEE International Conference on Control, Automation and Systems, pp. 1089–1092 (2010)
8. Yan, Z., Peng, S., Zhou, J., Zu, J., Jia, H.: Research on an improved dead reckoning for AUV navigation. In: IEEE Chinese Control and Decision Conference, pp. 1793–1797 (2010)
9. Fouque, C., Bonnifait, P., Betaille, D.: Enhancement of global vehicle localization using navigable road maps and dead-reckoning. In: IEEE Position, Location and Navigation Symposium, pp. 1286–1291 (2008)
10. Wen, J., Lu, X.: Research on low-cost vehicular integrated navigation. In: IEEE International Conference on Advanced Mechatronic Systems, pp. 287–290 (2011)
11. Li, H., Lin, K.: ITRI WAVE/DSRC communication unit. In: IEEE Vehicular Technology Conference, pp. 1–2 (2010)
12. Shimonaka, Y., Tasaka, S., Hatta, Y., Wada, T., Okada, H.: Accuracy improvement of vehicular collision avoidance support system (VCASS) for the next generation ITS. In: IEEE Wireless Communications and Networking Conference, pp. 2517–2522 (2007)
13. Savasta, S., Pini, M., Marfia, G.: Performance assessment of a commercial GPS receiver for networking applications. In: IEEE Consumer Communications and Networking Conference, pp. 613–617 (2008)

14. Yaacob, N., Abdullah, M., Ismail, M.: Ionospheric modelling: improving the accuracy of differential GPS (dGPS) in equatorial regions. IEEE Asia-Pacific Applied Electromagnetics, 1–5 (2007)
15. Bevly, D., Parkinson, B.: Cascaded Kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles. IEEE Transactions on Control Systems Technology 15(2), 199–208 (2007)
16. Berman, Z., Powell, J.: The role of dead reckoning and inertial sensors in future general aviation navigation. In: IEEE Position Location and Navigation Symposium, pp. 510–517 (1998)
17. Krakiwsky, E., Harris, C., Wong, R.: A Kalman filter for integrating dead reckoning, map matching and GPS positioning. In: IEEE Position Location and Navigation Symposium, pp. 39–46 (1988)
18. Yang, Y., Ye, H., Fei, S.: Integrated map-matching algorithm based on fuzzy logic and dead reckoning. In: IEEE International Conference on Control Automation and Systems, pp. 1139–1142 (2010)
19. Kuo, W.: Integration of GPS and dead-reckoning navigation systems. In: IEEE Vehicle Navigation and Information Systems Conference, pp. 635–643 (1991)
20. GPS distance calculate tool, china.cos@gmail.com. this tool is powered by Cosbeta, http://www.storyday.com/wp-content/uploads/2008/09/latlung_dis.html

Design and Implementation of a Software Development Kit for LLRP Readers

Yu-Shin Chang, Sheng-Pang Kuo, and Chua-Huang Huang

Department of Information Engineering and Computer Science
Feng Chia University
Taichung, Taiwan
{yushin,bowden,chh}@rfidlab.iecs.fcu.edu.tw

Abstract. In this paper, we design and implement a software development kit (SDK) for developing application programs of Low Level Reader Protocol (LLRP) readers. LLRP is one of the specification standards in the EPCglobal architecture framework. The purpose of LLRP standard is to unify reader interface and functions. Since LLRP standard provides the details of reader control, the user needs to understand syntax and semantics of the parameters and instruction format in order to develop application programs. The LLRP SDK includes LLRP function library to simplify the development process. The user only needs to understand the basic procedures of reader operations. The LLRP SDK is designed as a three-layer architecture, including the reader layer, the message layer, and the transport layer. A graphical interface tool is used to guide the user to set SDK function parameters with an intuitive way. The parameter values are stored in XML files then which are retrieved and converted to byte date following the LLRP standard by SDK library.

Keywords: Radio Frequency Identification, Class-1 Genenration-2 protocol, Low Level Reader Protocol, EPCglobal architecture framework, software development kit.

1 Introduction

Radio frequency identification (RFID) is a non-contact automatic identification technology which has been widely applied in various areas of application, such as track and trace the product in supply chain management [8, 14], tracking the carbon footprint of products [1], air cargo management [2], medical security [12], electronic ticket [9], etc. The architecture of an RFID system is usually composed of tags, readers, a middleware system, and applications. For ultra high frequency (UHF) RFID technology, most applications use on shelf commercial readers manufactured by different vendors such as Alien, AWID, Impinj, Intermec, Omron, *etc*. The readers from different manufactures all have their own proprietary reader interfaces and commands. This industrial phenomenon causes tremendous difficulty for middleware and application developers that they must tailor their programs if readers of different vendors are deployed.

To solve this problem, EPCglobal has released a specification standard Low Level Reader Protocol (LLRP) on the purpose of unifying reader interface and functions [6]. LLRP is one of the standard documents of EPCglobal Architecture Framework [5]

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 673–682.
DOI: 10.1007/978-3-642-35452-6_68

and is based on Class-1 Generation-2 UHF RFID protocol [4]. LLRP is a protocol interface between reader and clients, including middleware and application systems, and interacts with the air protocol interface. LLRP provides the details of reader control and definition of the messages, parameters, and fields of LLRP instructions. If a system developer is going to design and implement applications of LLRP readers without any assistant tool, he/she needs to understand the detailed LLRP functions that may greatly increase the cost and time of the application project.

Two vendors, Impinj and Intermec, provide LLRP support on their readers [10, 11]. Fosstrak, an open source RFID platform developer, presents an LLRP Commander for configuring and managing LLRP readers [7]. The purpose of the LLRP Commander is to provide a graphical interface that an application developer can pack messages for LLRP reader. However, if a programmer wants to use LLRP Commander to develop his/her applications, he/she still needs to know the details of message flow of LLRP specification. The LLRP Toolkit Project develops open source libraries in different languages that programmers can use it to build and parse LLRP messages [13]. Fosstrak, Impinj, and Intermec are users of the LLRP Toolkit.

In this paper, we design and implement a software development kit (SDK) of LLRP reader to provide a graphical user interface and library functions for developing LLRP reader application programs. It aims to simplify the LLRP development process that an application developer only needs to understand the basic procedures of reader operations. The LLRP SDK is designed as a three-layer architecture, including the reader layer, the message layer, and the transport layer. LLRP SDK provides different class function libraries at each layer. Reader Layer is designed to setting of client operations; message layer is responsible for building up the LLRP messages and parse the reports from readers; and transport Layer implements the TCP/IP connection between client and reader, and receives the response of reader.

This paper is organized as follows. The LLRP data model is briefly overviewed in Section 2. In Section 5, we the explain design and implementation of the LLRP SDK. The conclusions and future research are given in Section 4.

2 Overview of LLRP Operations and Messages

Low Level Reader Protocol (LLRP) standard is a specification for the network interface between the reader and its controlling software or hardware [6]. This specification is based on the tag and reader air interface protocol with the UHF Class-1 Generation-2 protocol [4]. The major role of the LLRP specification is to define data format and operation procedure between an LLRP reader and clients. The data unit of an LLRP reader is message. A message from a client to the reader includes getting and setting reader configurations, discovering reader capabilities, and managing the inventory and access operations. Messages from the reader to clients include reports of reader status, results of RF surveys, and results of inventory and access operations.

2.1 LLRP Operations

The two basic LLRP operations are: reader operations and access operations. A reader operation is to perform RF survey and/or antenna inventory. An access operation is to perform data access to and/or from a tag. Fig. 1 shows the structure of specification of LLRP reader operations and access operations.

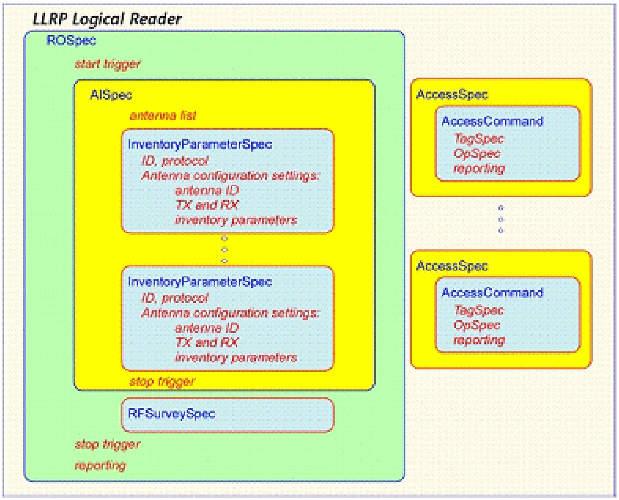


Fig. 1. Specification of LLRP reader and access operations (Source: EPCglobal, Taiwan)

Reader operations specify how antennas of an LLRP reader to be operated. A reader operation is denoted as an ROSpec which consists of one or more AISpec and RFSuveySpec. An ROSpec contains a specification identifier, boundary specification for the reader operation, specification priority, and one or more AISpecs and/or RFSurveySpecs. In addition, an ROSpec may contain a report specification which describes what are in the RO report and how it is delivered.

An access operation specification (AccessSpec) describes tag inventory and access operations. An AccessSpec is composed of tag specification (TagSpec), specifying the select condition, and operation specification (OpSpec) specifying what operations to be performed. Optionally, a report specification can be added to return the operation results. An AccessSpec is able to reconfigure antenna capability.

2.2 LLRP Messages

The data transmitted between a client and an LLRP reader are byte frames, called messages. In this paper, we refer to the object-oriented implementation of LLRP for the data model [3]. The LLRP functions of SDK library are called LLRP messages which are classified as the followings:

1. Protocol version management: Messages from client to reader for enquiring the protocol version or to set the protocol version for the current connection.
2. Reader device capabilities: Messages from client to reader for querying reader capabilities.
3. Reader operation control: Messages from client to reader for specifying antenna inventory and RF operations
4. Access control: Messages from client to reader for describing the access operations performed by tags.

5. Reader device configuration: Messages from client to reader for querying/setting reader configuration and LLRP connection.
6. Reports: Messages from reader to client for carrying different reports. A report may be about reader device status, data from tag, and/or RF analysis result.
7. Custom extension: Messages from client to reader of vendor defined content.
8. Errors: Messages from reader to client for reporting errors.

Typically, a software application client communicates with an LLRP reader with a sequence of LLRP messages to complete reading and/or writing tag data:

1. The client requests the reader to obtain its capability information.
2. The client sets the reader configuration according to the operation needs.
3. The client issues ROSpec commands to the reader. An ROSpec may contain a list of inventory commands, i.e., AISpecs.
4. The client sends AccessSpecs to the reader so it will read and/or write tag data.
5. The client receives reports from the reader.

The purpose of this paper is to build a software development kit so a system engineer can easily develop programs to perform the above process.

3 LLRP Software Development Kit Architecture

In this section, we present the architecture of the LLRP software development tools, including the functions for executing LLRP messages and the graphical interface tool for generating parameters of LLRP messages.

3.1 LLRP SDK Functions

The LLRP SDK functions are designed to execute LLRP messages, i.e., each function will generate or parse the bit string of a command frame as defined in the LLRP specification standard. Based on the RFID air protocol and LLRP messages, the main functions of SDK are described as below:

- Inventory setting: Client can set the inventory parameters.
- Access operator setting: Client can set the access operation parameters. To access the target tag memory, such as read, write, kill, and lock.
- Access control: When client want to access the tag memory, SDK shall check client's controlling authority, such as password management.
- Reader configuration: Get and set the configuration of reader, such as RF configuration, data transmission rate.
- Report management: SDK will receive reports from a reader.
- Error control: SDK shall provide the basic fool-proofing function, such as parameter format error, scheduling conflicts, and error control.

We design and implement the LLRP SDK functions as a three-layer architecture: reader layer, message layer, and transport layer, as shown in Fig. 2. The graphical user interface (GUI) is the tool to set function parameters. We will explain the graphical tool later.

Reader Layer

Reader Layer is responsible for client settings and deal with the the reports from reader. There are four modules in this layer: reader configuration module, inventory setting module, access operator module, and trigger setting module. We list the functions with explanation as follows.

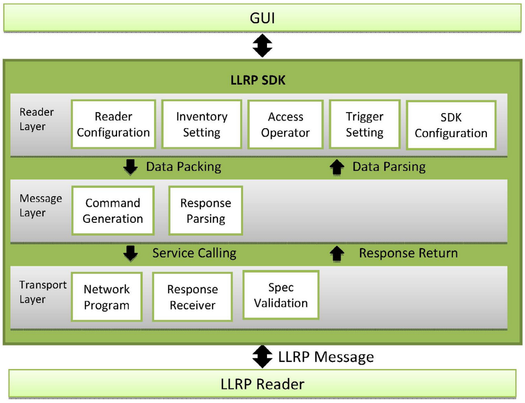


Fig. 2. SDK architecture

- Reader configuration module:

Return Type	Function Definition
void	LLRP_Capabilities(int RequestData): To get the reader function and ability.
void	Get_Reader_Config(int RequestData): To get the reader configuration.
void	Set_Reader_Config(int ConfigurationData): To set reader configuration.

- Inventory setting module:

Return Type	Function Definition
void	Inventory_Setting(byte[ ] rospec): This function is to set ROSpec parameters. The ROSpec parameters will transfer to Message_Encoding function of the message layer to pack the LLRP message. The message will be extended to LLRP commands of ADD_ROSPEC, ENABLE_ROSPEC, and START_ROSPEC.
void	EPC_Memory_Selector(): When the client is to read a specific tag. This function is used to specify the tag memory address and its mask rule.

- Access operator module:

Return Type	Function Definition
void	Access_Command_TagSpec(): This function is to set the access specification of tag memory data. If client do not set this parameter, it means to access all tag memory.
void	Access_Command_OpSpec(opType): This function is used to set the action of the access tag memory. The opType parameter means read, write, kill, and/or lock operation.

void	Check_Password(int password): Before performing the memory data access, it is necessary to check whether the client's password is correct.
void	Access_Command_Setting(byte[] acspec): This function is to set the AccessSpec parameters. The parameters will be transferred to the Message_Encoding function of the message layer to pack the LLRP message. The message will be extended to LLRP commands of ADD_ACCESSSPEC and ENABLE_ACCESSSPEC.

– Trigger setting module:

Return Type	Function Definition
void	Trigger_Setting(SpecType, SpecType, TriggerType, TriggerType, TriggerAction TriggerAction): This function is to set the LLRP start and stop triggers type and corresponding actions.

– SDK configuration module:

Return Type	Function Definition
void	Connection_Time_Setting(time): This function is used to set the connection time with the reader. During the period of the setting time, SDK will maintain connection with the reader.
boolean	Recovery(): This function sets all operation settings returning to the default values.
void	New_Tag_Data(String pramname, String membank, String address, String value, String description): This function is to specify a new tag memory data and store it in an XML file, e.g., TagData.xml. This XML file can be used in application program to retrieve tag memory information. Parameter name pramname is used to search XML for the tag memory data.
void	Modify_Default_Data(byte[] specData): This function is to modify the default value of LLRP specifications.

We give an example of XML file for specifying tag memory data. The client can set milk product information in SDK with an XML file as below.

```
<ParmInfoList>
  <ParmInfo>
       <TagMemList>
            <TagMen>
               <ParmName> MilkManu </ParmName>
               <MemBank> UserBank </MemBank>
               <Address>
                  <Pointer> 0 </Pointer>
                  <Length> 32 </ Length >
               </Address>
               <Data> </Data>
            </TagMen>
            <TagMen>
               <ParmName> MilkExp </ParmName>
               <MemBank> UserBank </MemBank>
```

```
                <Address>
                    <Pointer> 32 </Pointer>
                    <Length> 32 </ Length >
                </Address>
                  <Data> </Data>
                </TagMen>
                <TagMen>
                   <ParmName> MilkEPC </ParmName>
                   <MemBank> EPCBank </MemBank>
                   <Address>
                      <Pointer> 32 </Pointer>
                      <Length> 58 </ Length >
                   </Address>
                   <Data> 12045067890 </Data>
                 </TagMen>
             </TagMemList>
      </ParmInfo>
    </ParmInfoList>
```

The tag elements of the XML file are defined as the followings:

- ParmInfo: parameter information.
- TagMenList: a list of tag memory specification.
- TagMen: tag memory specification.
- MemBank: tag memory bank
- Address: address of tag memory.
- Pointer: starting bit of the address.
- Length: data length.
- Data: data value. If the data value is missing, it is provided by the program or read from the reader. In the example, MilkEPC is the 58-bit data, including header, filter value, partition, company prefix and item reference, but not serial number. This tag memory specification can be used to search tags of the specified milk product.

When the reader layer gathers all necessary parameters of reader operations, it will request the message layer to generate byte frames according to LLRP specification.

Message Layer

Message Layer is used to build up the LLRP messages and parameters and parse reports from reader. There are two modules in this layer: command generation module and message parsing module. We list the functions with explanation as follows.

– Command generation module:

Return Type	Fuction Definition
Byte[ ]	Message_Encoding(int MessageID): This function shall be given a message ID to find out corresponding message setting file. As the result, the byte array of an LLRP command will be enconded. Then, the command is transferred the reader via the transport layer.

Byte[ ]	Parameter_Encoding(int ParameterID) : This function shall be given a parameter ID to find out corresponding parameter setting file. As the result, the byte array of the parameter will be enconded. Then, the parameter in the bit format is returned to the reader layer for the followup work.

— Response parsing module:

Return Type	Function Definition
void	Message_Parsing(byte[] data): Parsing the bit frame of a report returned from the reader.

When the message layer finishes generation of byte frames, it will be transmitted to the reader via the transport layer. The message layer also receives report byte frames from the transport layer.

Transport Layer

In the transport layer, we implement the TCP/IP connection between client and reader. The byte arrays of command and report are transmitted to and received from the reader using functions in this module. There are two modules in this layer: networking module and specification validation module. We list the functions with explanation as follows.

— Networking module:

Return Type	Method Definition
void	LLRP_Connection(String address, int port): This function is to start a TCP/IP connection between the client and the reader.
void	Send_Message(byte[] message): This function is to send the command byte array generated by the message layer to the reader.
void	Read_Buff(): This function is to read the buffer of a report byte array and pass it the message layer.

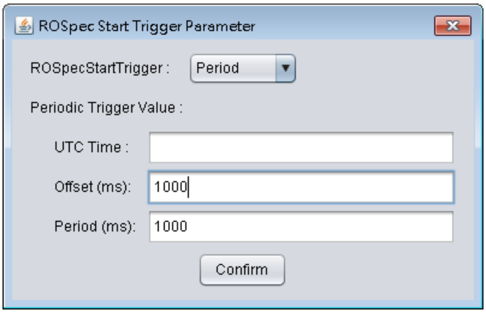
— Specification validation model:

Return Type	Function Definition
void	Check_Schedule_Time(): This function to check whether the LLRP processing schedule is conflict. If a conflicting schedule is detected, the message is considered invalid.

After the message generation is completed, the message layer will call functions of the transport layer to implement the data transfer between the client and the reader. However, most of these functions require input parameters that it is a cumbersome job if a programmer needs to fill in the parameter values manually during the development time. In order to easy programmer's effort, we develop a graphical interface tool that will guide the programmer to generate parameter values.

3.2 Graphical Interface Tool

The graphical interface tool is designed to set parameter values that a programmer will fill in values in the way intuitive to him/her, not as the binary strings required by the LLRP specification standard. We give an example to demonstrate the graphical interface tool.



ROSpec Start Trigger Parameter

ROSpecStartTrigger : Period

Periodic Trigger Value :

UTC Time :

Offset (ms): 1000

Period (ms): 1000

Fig. 3. Start trigger setting

Fig. 3 shows the graphic interface for setting start trigger of an ROSpec. A set of parameters must be given a unique parameter name, e.g., StartTrigger1. UTC time is the starting time the ROSpec is to be executed. If UTC time is missing, as in Fig. 3, the starting time is the time when ROSpec is received by the reader plus the offset time. Period is how often the inventory operation is executed. For example, in Fig. 3, if an ROSpec is received by the reader at 3:30pm, than it is executed at 3:35pm. The inventory operation is executed every 30 minutes (1800 seconds) starting at 3:35pm.

The parameters of the trigger start specified in Fig. 3, will be stored in an XML file as below:

```
<ParmInfoList>
  <ParmInfo>
        <StartTrigger>
              <ParmName> StartTrigger1 </ParmName>
              <TriggerType> Periodic </Periodic>
              <UTC> </UTC>
              <Offset> 300 </Offest>
         <Period> 1800 </Period>
        </StartTrigger>
  </ParmInfo>
</ParmInfoList>
```

When a program create an ROSpec with this start trigger, the SDK will convert the parameters to corresponding strings, e.g., 1800-second period time is converted to 0X17BB40 which is 1,800,000 milliseconds as specified in LLRP standard.

4 Conclusion

In this paper, we focus on Low Level Reader Protocol (LLRP) in the EPCglobal architecture framework, design and implement a software development kit (SDK) for

LLRP. The LLRP SDK provides a graphical user interface and LLRP function library. It aims to simplify the LLRP development process. Client can set the specifications default file with demand. Client also can set its tag data for product commercial information. In order to use in the follow-up the LLRP operations on the development of RFID-related applications. In future, the LLRP SDK can be added higher composite functions, provide a more complete development tools.

In the future development of RFID systems, LLRP SDK can be added to the analysis of the ALE middleware. Due to the overlapping functions of ALE and LLRP, we can integrate some of the functions to reduce the RFID system development time and integration costs.

References

1. Auto-ID Labs. The Potential of the EPC Network to Monitor and Manage the Carbon Footprint of Products. Auto-ID Labs, White Paper WP-BIZAPP-047 (2009)
2. Chang, Y.S., Son, M.G., Oh, C.H.: Design and Implementation of RFID Based Air-Cargo Monitoring System. Advanced Engineering Informatics 25(1), 41–52 (2011)
3. Constantinou, F.: An Object Oriented Implementation of LLRP. Technical report, MIT EPC Net (2007)
4. EPCglobal. EPC Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz — 960 MHz Version 1.0.9. EPCglobal, Inc. (2005)
5. EPCglobal. The EPCglobal Architecture Framework, Version 1.4. EPCglobal, Inc. (2010)
6. EPCglobal. Low Level Reader Protocol (LLRP), Version 1.0.1. EPCglobal, Inc. (2007)
7. Fosstrak, Fosstrak LLRP Commander. Fosstrak Open Source RFID Platform, `https://code.google.com/p/fosstrak/wiki/LlrpMain`
8. Foster, P., Sindhu, A., Blundell, D.: A Case Study to Track High Value Stillages Using RFID for an Automobile OEM and Its Supply Chain in the Manufacturing Industry. In: IEEE International Conference on Industrial Informatics, pp. 56–60 (2006)
9. Guihao, B., Minggao, Z., Jiuwen, L., Yin, L.: The Design of an RFID Security Protocol Based on RSA Signature for E-ticket. In: Proceedings of the 2nd IEEE International Conference on Information Management and Engineering, pp. 636–639 (2010)
10. Impinj. LLRP—Reader Control Simplified. Impinj, lnc. (2007)
11. Intermec, Low-Level Reader Protocol Programmer's Reference Manual. Intermec Technologies Corporation (2009)
12. Lee, B., Kim, H.: Ubiquitous RFID based Medical Application and the Security Architecture in Smart Hospitals. In: 2007 International Conference on Convergence Information Technology, pp. 2359–2362 (2007)
13. LLRP Tool Kit, `http://www.llrp.org/index.html`
14. Tan, H.: The Application of RFID Technology in the Warehouse Management Information System. In: Proceedings of 2008 International Symposium on Electronic Commerce and Security, pp. 1063–1067 (2008)
15. Thiesse, F., Floerkemeier, C., Harrison, M., Michahelles, F., Roduner, C.: Technology, Standards, and Real-World Deployments of the EPC Network. IEEE Internet Computing 13(2), 36–43 (2009)

The Performance Comparison of European DTTV Standards with LDPC-Encoded-DVB-T Standard under AWGN Channel

Oktay Karakuş

Department of Electrical and Electronics Engineering
Izmir Institute of Technology
Izmir, Turkey
oktaykarakus@iyte.edu.tr

Abstract. In this study, in addition to previous work which is a performance comparison of European DTTV Broadcasting standards which are known as DVB-T and DVB-T2, DVB-T2's inner encoder/interleaver block which has LDPC Encoder/Bit Interleaver, is integrated into DVB-T instead of its own inner encoder/interleaver block. This Proposed System's performance under AWGN channel is compared to the DVB-T/T2 standards. The Proposed System achieves better performance results according to DVB-T and very close results according to DVB-T2. It achieves nearly from 4 to 7 decibels SNR gain and up to 13 Mb/s data rate gain then DVB-T results according to different code rate and modulation parameters.

Keywords: European DTTV Standards, DVB-T, DVB-T2, AWGN Channel, LDPC Encoder, Performance Comparison.

1 Introduction

Depending on the rapid advancement of technology in recent years, significant improvements have been observed in television broadcasting standards. The current technologies allow broadcasting in High Definition (HD) and 3-Dimension (3D) and watching this content at fixed or mobile receivers. In addition to this, the development on every area of wireless communications affects analog terrestrial television broadcasting technologies, and gives rise to an idea of digital terrestrial television broadcasting. With these developments, since the beginning of the 90s, digital terrestrial television broadcasting has become an important study area in the whole world.

Digital Video Broadcasting (DVB) project started to prepare specifications about digital terrestrial television in Europe in 1993. DVB can be defined as an industry-led consortium of broadcasters, manufacturers, network operators, software developers and regularity bodies committed to designing open technical standards for digital television and data services [1]. DVB Organization has prepared and published digital transmission standards for Europe. These are;

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 683–691.
DOI: 10.1007/978-3-642-35452-6_69

- DVB-S, for Satellite Systems [2],
- DVB-C, for Cable Systems [3],
- DVB-H, for Handheld Systems [4],
- DVB-T, for Terrestrial Systems [5].

Although DVB-T has many advantages and is a great success in digital broadcasting area, it requires some improvements. These improvements should be in the area of adaptation for portable devices that moves faster and that is inside of the buildings. These facts provide a new and updated standard for DVB-T and this led to the standardization of DVB-T2. DVB-T2 specification was firstly published in September 2009 and in this document, new baseline transmission system is described in [6].

In a previous work [7], both of the standards are simulated under the effects of Additive White Gaussian Noise (AWGN) Channel and the acquired results are compared according to the Target Bit Error Rate (BER) value which is stated in standards. The results showed that DVB-T2 standard outperforms DVB-T standard under AWGN Channel and achieves up to 7 decibels power gain according to code rate and modulation parameters.

Low Density Parity Check (LDPC) Codes usage may be most important advantage of DVB-T2 Systems. In this study, it's going to be searched an answer to the question, "What will be the performance results for DVB-T system with LDPC Encoder?"

In literature, a study [8] applied this kind of idea to the DVB-T. In that study, writer uses just 16QAM modulation technique and substitute LDPC block with RS-Convolutional Coding block. In the present study, LDPC Coding is used instead of Convolutional Coding. RS Coding block is also applied after inner interleaving block and in addition to 16QAM, QPSK and 64QAM Modulation Schemes are also used in this study.

This paper is organized as follows. First of all, DVB-T/T2 system architectures are briefly explained in section 2. In section 3, the proposed system is explained. Also the performance comparison and results is given in section 4. Paper ends by concluding the study.

2 DVB-T/T2 System Architectures

Transmitter structures of DVB-T and DVB-T2 systems are defined and published by DVB organization. DVB-T and DVB-T2's system architectures are briefly explained in this section but deeply defined in [5] and [6] respectively.

Receiver structures of DVB systems have no any specifications but "Implementation Guidelines" [9], [10]. These documents give information about the main blocks of systems' receivers and different ways of implementation for some specific blocks.

System architecture for DVB-T and DVB-T2 is provided in this section and in addition theoretical comparison of both is illustrated in table 1.

2.1 DVB-T Transmitter System Architecture

DVB-T systems are designed to perform adaptation of the baseband TV signals from the output of the MPEG-2 transport multiplexer to terrestrial channel characteristics.

Data stream which is the output of MPEG-2 block is processed in following processes;

- Transport multiplex adaptation and randomization for energy dispersal
- Outer coding & interleaving
- Inner coding & interleaving
- Mapping and modulation
- OFDM transmission

DVB-T's main transmitter block diagram is given in Figure 1 below.

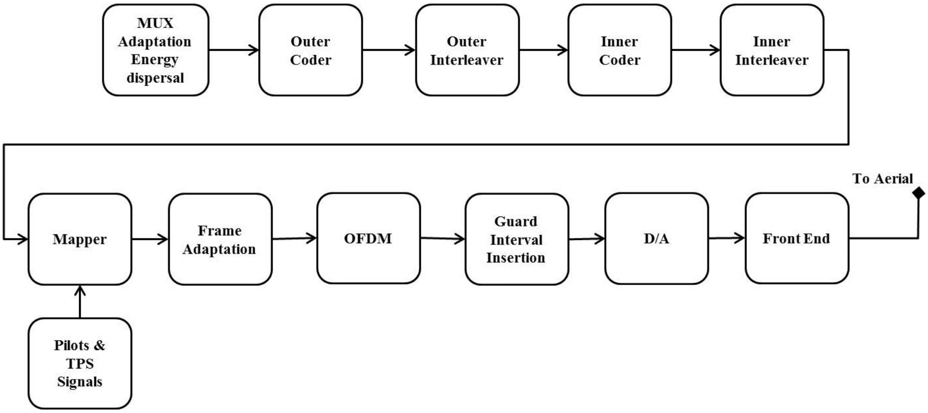


Fig. 1. DVB-T Transmitter Block Diagram[5]

2.2 DVB-T2 Transmitter Architecture

DVB-T2 system's general high level block diagram is represented in Figure 2. The inputs of the system may be MPEG-2 Transport Streams (TS) and/or Generic Streams (GS).

The systems output can be a single signal to be transmitted or optionally can be two signals to be conveyed two sets of antennas which are called MISO transmission mode. All input data streams are then carried in individual Physical Layer Pipes (PLPs).

A DVB-T2 system mainly consists of four blocks: Input Processing, Bit Interleaved Coding & Modulation, Frame Builder, Orthogonal Frequency Division Multiplexing (OFDM) Generation.

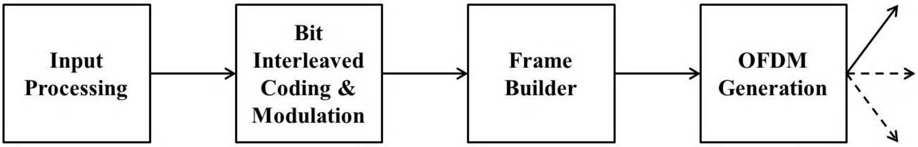


Fig. 2. DVB-T2 Transmitter Block Diagram[6]

Input processing block consists of two main sub block which are;

- Mode Adaptation: This block operates separately for each PLP and slice the input data into data fields.
- Stream Adaptation: This block prepares data streams for Bit Interleaved Coding & Modulation (BICM) block by applying padding and scrambling.

Bit Interleaved Coding & Modulation (BICM) block provides some important sub-blocks as seen in Figure 3.

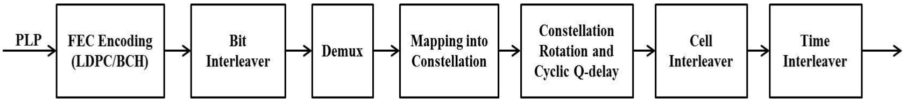


Fig. 3. Bit Interleaved Coding and Modulation (BICM) [6]

The function of the OFDM generation block is to insert the reference information and allow the receiver to compensate for distortions introduced by the channel, and to produce time domain signal for transmission. After that process, guard interval is inserted and if possible PAPR reduction is applied and finally completed T2 signal is produced.

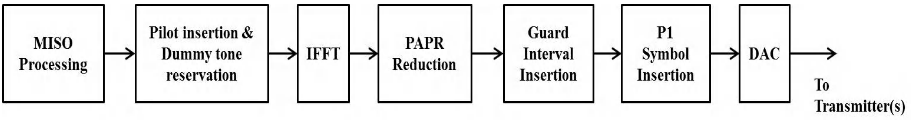


Fig. 4. OFDM Generation [6]

3 Proposed System Architecture

In chapter II, a very simple explanation of the DVB Organizations Terrestrial TV Standards is given and shown in figures 1 and 2.

When designing the Proposed System, the important success of DVB-T2 which is LDPC Encoder is taken into account. DVB-T2 has new coding and modulation technique according to DVB-T. One of the most important techniques here is LDPC Encoding. So an idea appeared; what would be the performance results of DVB-T, if LDPC Encoder were used as its inner encoder? This novel idea shows the way to this study.

In Figure 5 and Figure 6, Inner Encoder/Interleaver blocks of both systems are shown.

In proposed system, DVB-T's inner encoder/interleaver block is replaced by DVB-T2's inner encoder/interleaver block with an additional Stream adaptation block. All the other operations of DVB-T remain the same. In Figure 7, The Proposed System's block diagram is given.

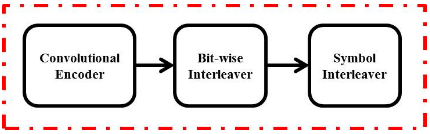


Fig. 5. DVB-T's Inner Encoder/Interleaver Block

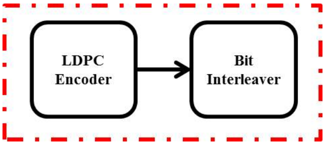


Fig. 6. DVB-T2's Inner Encoder/Interleaver Block

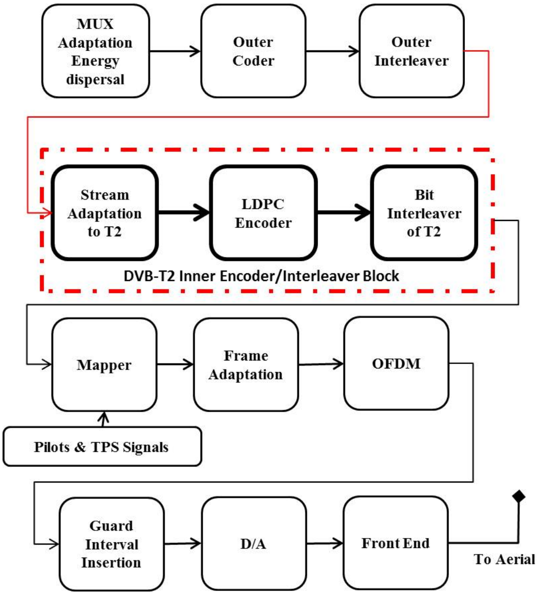


Fig. 7. The Proposed Systems Block Diagram

4 Performance Comparison and Results

In this chapter, the performance comparison of the systems which are defined briefly in chapter 2 and chapter 3 is measured. The comparison of DVB-T and DVB-T2 is studied in [7] and in addition to this, The Proposed System is simulated and results are taken according to the simulation parameters which are given in table 1.

Table 1. Simulation Parameters

Modulation Schemes	QPSK – 16QAM – 64QAM
Inner Encoder Code Rates	1/2 – 2/3 – 3/4 – 5/6
Transmission Mode (OFDM)	2K
Transmission Type	SISO
Channel Bandwidth	8 MHz
Target BER after Inner Encoders	2 x 10^{-4}
Target BER after Outer Encoders	0
LDPC Code Length	64800
LDPC Max. Iterations	50

Carrier-to-Noise Ratio (C/N) values in decibels, which are needed to achieve Target BER under the effect of AWGN Channel, are shown in Table 2 below.

Table 2. C/N Values for the Systems to achieve Target BER under AWGN

Modulation	Code Rate	DVB-T	Proposed System	DVB-T2
QPSK	1/2	*4,1*	**-0,3**	*-0,4*
QPSK	2/3	*5,7*	**1,8**	*1,7*
QPSK	3/4	*6,8*	**2,7**	*2,6*
QPSK	5/6	*7,9*	**3,8**	*3,8*
16QAM	1/2	*10,5*	**5,1**	*4,8*
16QAM	2/3	*12,1*	**7,7**	*7,2*
16QAM	3/4	*13,3*	**8,8**	*8,5*
16QAM	5/6	*14,7*	**10,2**	*10*
64QAM	1/2	*15,9*	**9,6**	*9,4*
64QAM	2/3	*17,8*	**12,7**	*12,3*
64QAM	3/4	*19,1*	**14,2**	*13,9*
64QAM	5/6	*20,7*	**15,8**	*15,7*

Figure representation of Table 2 is shown in Figures 8 to 10. In these figures, for the Target BER of 2x10^{-4}, the proposed system has nearly the same C/N values with DVB-T2 standard. However The Proposed System has same architecture with DVB-T, it outperforms DVB-T system.

Depending on the results in Figures 8 to 10, Figure 11 presents SNR gain of The Proposed System from DVB-T for different modulation type and code rate parameters. According to this figure, The Proposed System achieves up to 7 dBs SNR gain than DVB-T.

Figure 12 presents the maximum bitrate gain of The Proposed System from DVB-T. According to this figure, it's concluded that up to 13Mb/sec bitrate gain achieved by The Proposed System than DVB-T. The results in Figure 12 depend on values which are given in Table 3.

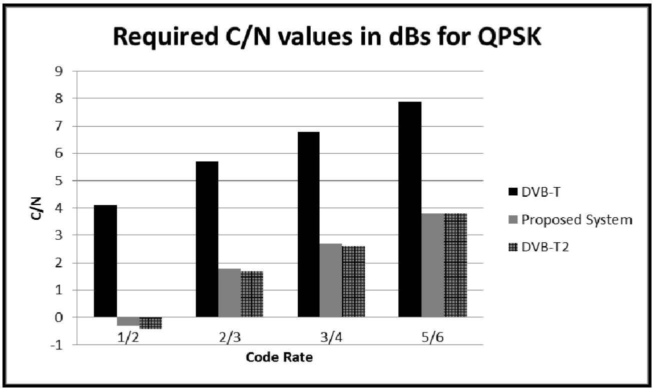


Fig. 8. Required C/N values for QPSK

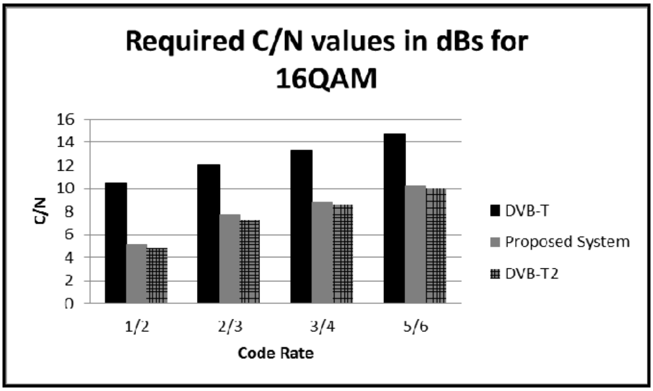


Fig. 9. Required C/N values for 16QAM

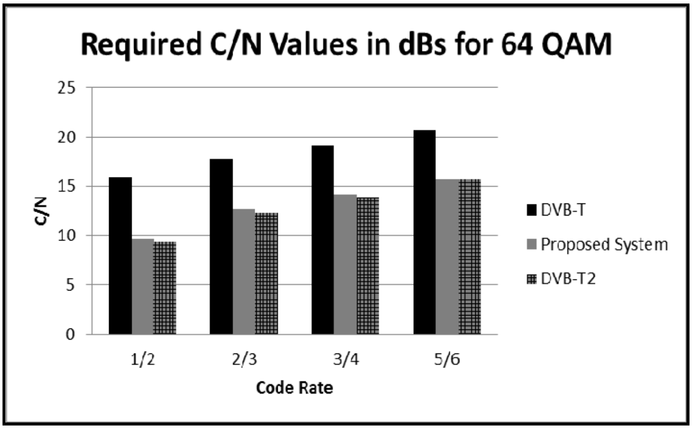


Fig. 10. Required C/N values for 64QAM

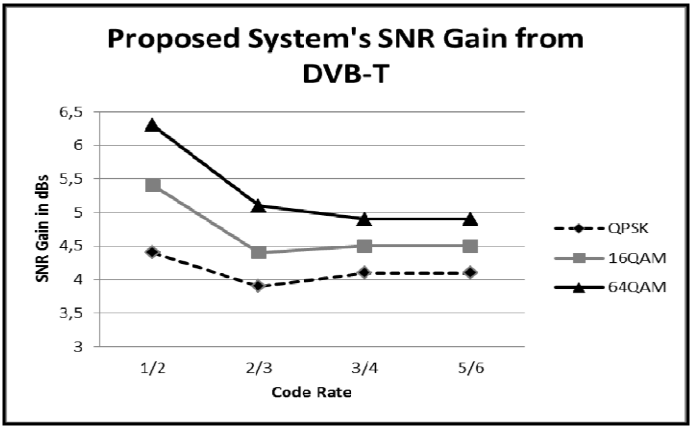


Fig. 11. SNR Gain of The Proposed System in dBs

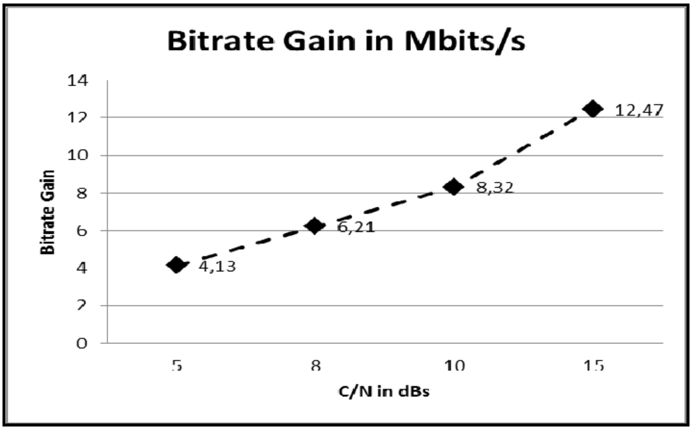


Fig. 12. Maximum Bit Rate Gain of the Proposed System

Table 3. Maximum Bit Rate values of DVB-T and the Proposed System

Max. Bitrates in Mb/s		
C/N	**DVB-T**	**The Proposed System**
5 dB	8,29	12,42
8 dB	10,37	16,58
10 dB	12,42	20,74
15 dB	18,63	31,1

5 Conclusions

According to the given performance comparison results, these can be concluded that;

- The Proposed System achieves nearly the same results with DVB-T2 in spite of it uses DVB-T architecture.
- The Proposed System achieves up to 7 dBs SNR Gain to perform a communication at corresponding Target BER value. By using this result it can be said that LDPC Coding outperforms Convolutional Coding.
- The Proposed System achieves up to 13 Mb/s Bit rate gain at same C/N value.
- The Proposed System may be an alternative approach which offers same performance with DVB-T2 for countries uses DVB-T standard.

The Proposed System performance under multipath channel and mobility effect both in SNR Gain and Maximum Bit Rate Gain is planned to be studied as Future Work.

References

1. ETSI, TS 102 685: Digital Video Broadcasting (DVB); High-level Technical Requirements for QoS for DVB Services in the Home Network (2011)
2. ETSI, EN 300 421: Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services (1997)
3. ETSI, EN 300 429: Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Cable Systems (1998)
4. ETSI, TR 102 377: Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines (2009)
5. ETSI, EN 300 744: Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television (2009)
6. ETSI, EN 302 755: Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2) (2011)
7. Karakuş, O., Özen, S.: European Terrestrial Digital Television standards performance comparison under AWGN channel. In: 20th Signal Processing and Communications Applications Conference (SIU), pp. 1–4 (2012) (in Turkish)
8. Jokela, T.: Performance analysis of substituting DVB-S2 LDPC code for DVB-T error control coding system. In: 2008 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, pp. 1–5 (2008)
9. ETSI, TR 101 190: Digital Video Broadcasting (DVB); Implementation Guidelines for DVB Terrestrial Services; Transmission Aspects (2011)
10. ETSI, DVB Document A133: Digital Video Broadcasting (DVB); Implementation Guidelines for a Second Generation Digital Terrestrial Television Broadcasting System, DVB-T2 (2010)

Coding Scheme Improved Query Efficiency of Hierarchical MIIS Servers with Cache in Heterogeneous Networks

Pei-Chen Tseng[1], Jai-Yan Tsai[2], and Wen-Shyang Hwang[2,*]

[1] Department of Information Engineering and Informatics,
Tzu Chi College of Technology, Hualien, Taiwan
peichen@tccn.edu.tw

[2] Department of Electrical Engineering, National Kaohsiung,
University of Applied Sciences, Kaohsiun, Taiwan
1098304123@kuas.edu.tw, wshwang@mail.ee.kuas.edu.tw

Abstract. The rapid development of wireless network technologies trending toward seamless heterogeneous networks, query efficiency of handover process while accessing neighboring networks for seamless mobility is a major concern issue. Our previous paper proposes Hierarchical MIIS Servers in Heterogeneous Networks by Caching (HMSHNc) in limited network, in which a hierarchical network classifies the stored information by network level (class) and distributes the location of the MIIS server cache to reduce query cost. To deal with increasing larger network, this paper proposes an enhanced method called HMSHNcC, which using the coding scheme to the ID assignment of all MIIS servers—ZMIIS, LMIIS and GMIIS. Simulation result shows the proposed HMSHNcC can satisfy to reduce query cost and to improve query efficiency.

Keywords: MIIS server, MIH, IEEE 802.21.

1 Introduction

The wireless communication revolution brings fundamental changes to data networking, Internet, telecommunication, and makes integrated networks a reality. The wireless devices are becoming increasingly multimodal, containing multiple communication interfaces such as the Wireless Local Area Network (WLAN) and the Worldwide Interoperability for Microwave Access (WiMAX) as shown in Fig. 1 [1]. Without the geographical coverage limitations of individual communication systems, this allows users to communicate and to choose an optimum wireless network interface in accordance with the desired requirements in terms of transmission rate, quality of service (QoS), communication price, and so on. In the new generations of wireless networks, seamless mobility support across heterogeneous networks is

[*] Corresponding author.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 693–701.
DOI: 10.1007/978-3-642-35452-6_70

referred to as the event when all sessions of an MN continue to maintain their connection even as an MN changes its point of attachment (PoA). An MN can roam across heterogeneous networks and keep its connections active for seamless mobility.

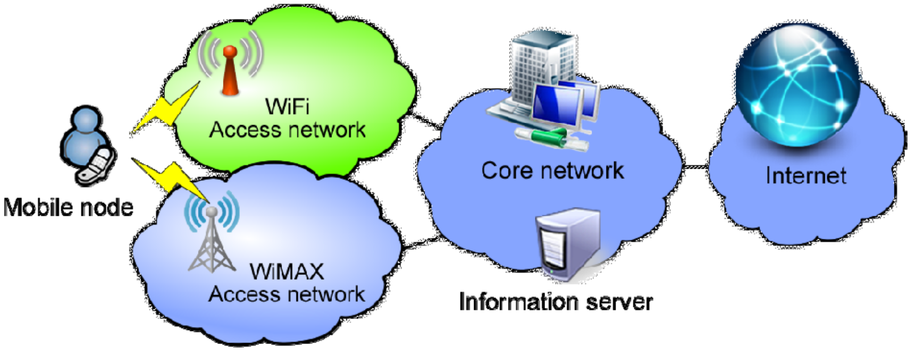


Fig. 1. Sample topology of heterogeneous networks

Because of the different wireless technologies, mobile nodes (MN) cannot be utilized. The IEEE 802.21 [2] working group recently finalized a standard called Media-Independent Handovers (MIH) for dealing with handovers in heterogeneous networks (Fig. 1), which provides Media Independent Information Service (MIIS) servers residing at the network layer. This provides MNs with seamless services in heterogeneous networks by vertical handover supported by the MIH framework. With the aid of MIIS, IEEE 802.21 defines a set of handover-enabling functions which are specified with respect to network elements in the protocol stack. IEEE 802.21 also allows the reception of dynamic information about the performance of the serving network. IEEE 802.21 further allows the reception of dynamic information about the performance of the other networks in range. This information can be used to assist in making decisions about which handover target to choose and in making preliminary preparations for a handover.

IEEE 802.21 provides methods for continuous monitoring of available access conditions. However, IEEE 802.21 does not specify any methods for collecting this dynamic information at the link layer. Moreover, IEEE 802 transmission technology provides no seamless interconnection mechanism, even in the mature IEEE 802.11 and 802.16 technologies. Nevertheless, seamless mobility of users is demanded in environments with multiple access technologies, each with different levels of coverage, QoS, protocols and cost to the end user. Thus, inter-handover in wireless network technology is a critical capability, implying that MIH will be receiving great attention in the near future.

Keeping in mind that from a conventional, IP-centered point of view, changing the Point of Attachment (PoA) calls for mobility management actions, although in practice there may be no physical mobility whatsoever. Vertical handover is more challenging under mobile conditions, with seamless handover being particularly important in heterogeneous networks. Many issues are involved in the mobility heterogeneous network to keep connections anytime, anywhere:

- Network Discovery
- Handover Decision
- Mobility Management
- Quality of Service (QoS)

Network discovery is the most important thing among them. Therefore, this paper focus on the issue of network discovery . The IEEE 802.21 MIH [2] uses a central architecture divided into classes: class 1 = edge router (ZMIIS); class 2 = core router (LMIIS); class 3 = global router (GMIIS). In our previous proposed system [3], all three levels have their own MIIS servers. The edge router's MIIS servers have an added cache. The cache has its edge router's AP information and different core router (network operator) information, but does not duplicate the information of the other edge routers of the same core network. Here it is assumed that different network operators have different core routers. The intent is to shorten the time for MN query when accessing neighbor networks. However, this also makes it easy for invalid node issues to occur. In large networks, it may be difficult for the MIIS server to get the correct network information, possibly due to delay or network congestion.

The HNDSHO, Hierarchical Neighbor Discovery Scheme for Handover Optimization [4], improves on the standard problem [2] by giving the MIIS server the ability to distinguish the stored information hierarchically, thereby enhancing query request efficiency for the given neighbor network conditions. But this causes MN query overtime because of layer-by-layer search rules. It also increases the MN handover time. Therefore, Our previous paper proposes HMSHNc [3], Hierarchical MIIS Servers in Heterogeneous Networks by Caching, which seeks to improve the query time for crossing the core network. HMSHNc works well only in a limited network. It got difficult for the MIIS server to get the correct network information because of delay or network congestion in large networks. Herein, this paper proposes the coding scheme to the ID assignment of all MIIS servers—ZMIIS, LMIIS and GMIIS, so-called HMSHNcC, to deal with increasing larger network. This kind of coding scheme is something like the codec method used in the telecommunication. The proposed coding scheme can effectively reduce the MN cost for querying the same PoA, or for the different PoA.

The rest of this paper is structured as follows: section 2 introduces the related study. Section 3 describes our proposed structure and operation scenario. Simulation results are in section 4. A final summary is presented in a concluding section.

2 Related Work

2.1 HNDSHO

Buiati considered many access networks and network operators: when a response message exceeds a maximum size, some information must be removed from the MIIS response. Therefore, HNDSHO (Hierarchical Neighbor Discovery Scheme for Handover Optimization) [4], in which the network coverage area is divided into mobility zones, is managed by different MIIS servers as illustrated in Fig. 2.

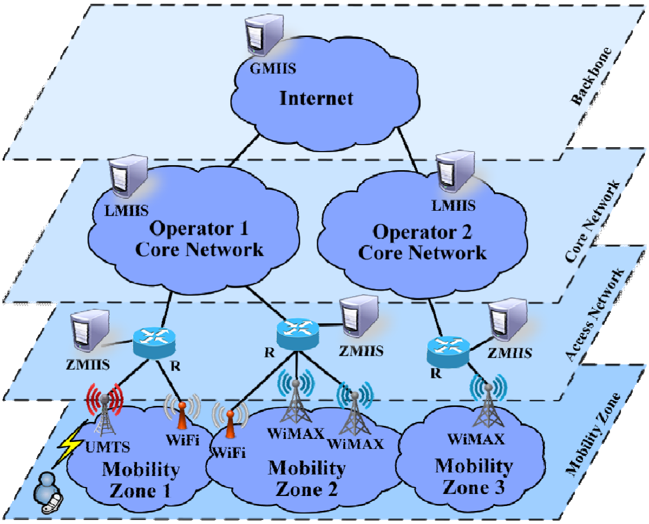


Fig. 2. HNDSHO topology

From bottom upward, the first level of the hierarchy is composed of mobility zones defined by the number of existing networks, users, and also considering areas where networks overlap. The second level contains the Zone MIIS servers (ZMIIS) which are in charge of supplying highly detailed information about specific PoAs in a particular mobility region. The third level contains the Local MIIS servers (LMIIS), managing the information of the different mobility zones which belong to the same operator. Finally, a Global MIIS server (GMIIS) is specified for use in multi-operator environments.

When higher servers are used, the cost increases. Using only the ZMIIS level results in improved efficiency, i.e. the system network has the best performance when dealing with the vertical handover in heterogeneous networks. Use of the LMIIS level results in average cost. Use of the GMIIS has a negative effect on cost. In general, The HNDSHO which giving the MIIS server the ability to distinguish the stored information hierarchically, enhances query request efficiency for the given neighbor network conditions. It has high processing time, and transmission cost.

2.2 HMSHNc

Our previous proposed HMSHNc (Hierarchical MIIS Servers in Heterogeneous Networks by Caching) considering the network topological environment and MIIS assignment [3]. HMSHNc improves the query efficiency of Hierarchical MIIS Servers in Heterogeneous Networks (HMSHN) by caching, is shown in Fig. 3. It uses a hierarchical structure like HNDSHO to avoid removal of MIIS response information when the response message exceeds a maximum size. This is important because removal of important information may cause sub-optimal handover decisions, resulting in ABC failure of the MN [5]. However, in the proposed network environment, different class MIIS servers are allotted to the different network levels. A cache is added to the ZMIIS servers.

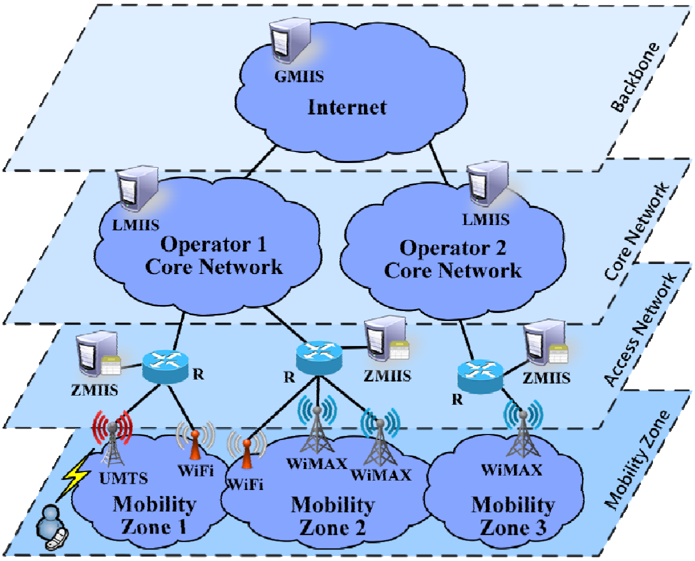


Fig. 3. HMSHNc topology

When a MN requests a query that crosses the core network service, the query needs to pass through the global router to the target MIIS server (edge router) to obtain the needed information. The obtained information is stored in the serving MIIS server's cache. That is, MN queries of PoA information located in neighboring networks with different network operators pass through the GMIIS, the target ZMIIS replies with needed information prior to inter-system handover (vertical handover) execution. The next MN query is the same, the information can be obtained from the serving MIIS server's cache directly, with no need to use the global router. A copy of the reply information for successful queries is stored in the cache of the local ZMIIS server, so the time-consuming transit of the GMIIS need not be repeated. This can reduce the MN cost for querying the same PoA.

From bottom upward, the bottom PoA of the overlapping access networks consists of WiFi AP, WiMax BS, etc. The ZMIIS server database stores the complete PoA information of its serving area. The LMIIS server is responsible for managing the corresponding ZMIIS of the same core network of the network operator, and keeps the related ZMIIS information in its server database. The GMIIS server stores the network information of the specified network operators.

3 HMSHNcC

We consider the network topological environment and MIIS assignment as the same with HMSHNc [3]. A hierarchical storage structure is adopted. The various class network assignments of the MIIS server are located in a local cache in the network access area. HMSHNc works well only in a limited network. To deal with increasing larger network, this paper proposes an enhanced method called HMSHNcC as shown in Fig. 4, which using the coding scheme to the ID assignment of all MIIS servers—ZMIIS, LMIIS and GMIIS. This kind of coding scheme is something like the codec method used in the telecommunication. This kind of codec provides security, flexible, stable, transparency, and so on. The codec of MIIS server is as Fig. 5, this paper proposes the coding scheme occupied 4n bits, here GMIIS 1 bit, LMIIS 3 bits, ZMIIS 4 bits. The first digit 0 of each MIIS ID reserved for signal control. The max. frame size of no fragment is 128 bits in IEEE 802.21 MIH standard. This paper we set the min. frame size to 33 bits.

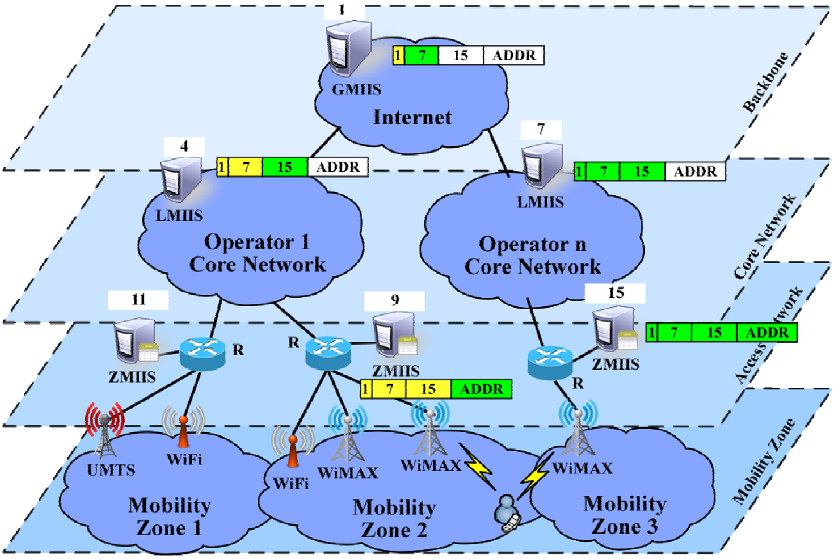
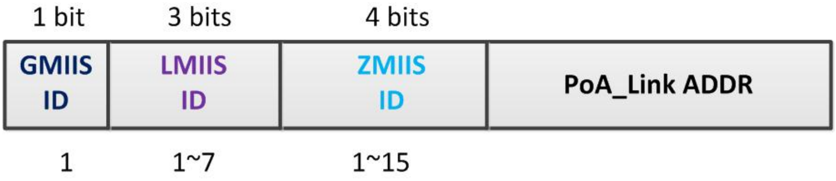


Fig. 4. HMSHNcC topology



1 bit	3 bits	4 bits	
GMIIS ID	LMIIS ID	ZMIIS ID	PoA_Link ADDR
1	1~7	1~15	

Fig. 5. The HMSHNcC ID codec of MIIS server

In the HMSHNcC topology in Fig. 4, the WiMax MN of Mobility Zone 2 is not the first MN to pass through to the destination WiMax of Mobility Zone 3. The MN obtained the related handover information stored in the cache of the serving ZMIIS server #9. The destination is to the ZMIIS server #15, by way of the ZMIIS #7. Therefore, this MN is travelling from the ZMIIS #9 to its LMIIS #4 of Operator 1, then goes through to the GMIIS #1, then send to the LMIIS #7 of Operator n, and finally to the ZMIIS #15.

3.1 HMSHNcC v.s. HMSHNc

Frankly speaking, the HMSHNcC mechanism is based on the HMSHNc mechanism [3]. Herein, we compare the performance cost of HMSHNcC with HMSHNc. Performance is evaluated as Query Cost [6]. The cost is calculated as equation (1) and contains Queuing Time, Transmission Time, Propagation Time and Processing Time. This paper assumes the network load is low. Therefore the Queuing Time is treated as 0. The Processing Time is Tp. The wired transmission cost of the control packets between nodes X and Y is the average number of hops ($d_{x,y}$) multiplied by the per hop transmission cost unit (τ) as shown in equation (2). The wireless transmission cost adds the weight factor (κ) as in equation (3).

$$\text{Cost} = \text{Queuing Time} + \text{Transmission Time} + \text{Propagation Time} + \text{Processing Time} \quad (1)$$

$$C_{x,y} = \tau\, d_{x,y} \quad (2)$$

$$C_{MN,PoA} = \tau\, \kappa \quad (3)$$

Therefore in HMSHNc, the first query cost of MN is as the equation (4) to (6) for ZMIIS, LMIIS, GMIIS, respectively. The next query cost of MN is as equation (7).

$$C_{IQ}^{ZMIIS} = t_{req} + t_{resp} = 2\tau(\kappa + C_{PoA,ZMIIS}) + TP_{ZMIIS} \quad (4)$$

$$C_{IQ}^{LMIIS} = 2\tau(\kappa + C_{PoA,ZMIIS} + C_{ZMIIS,LMIIS} + C_{LMIIS,ZMIIS}) + TP_{ZMIIS} + TP_{LMIIS} \quad (5)$$

$$\begin{aligned} C_{IQ}^{GMIIS} = {} & 2\tau(\kappa + C_{PoA,ZMIIS} + C_{ZMIIS,LMIIS} + C_{LMIIS,GMIIS} + C_{GMIIS,LMIIS} + C_{LMIIS,ZMIIS}) \\ & + TP_{ZMIIS} + TP_{LMIIS} + TP_{GMIIS} \end{aligned} \quad (6)$$

$$C_{IQ}^{Cache} = 2\tau(\kappa + C_{PoA,ZMIIS}) + TP_{ZMIIS} + TP_{Cache} \quad (7)$$

In HMSHNcC, the first query cost of MN is as the equation (8) to (10) for ZMIIS, LMIIS, GMIIS, respectively. The cost of ZMIIS as equation (7) is the same as equation (4).

$$C_{IQ}^{ZMIIS} = t_{req} + t_{resp} = 2\tau(\kappa + C_{PoA,ZMIIS}) + TP_{ZMIIS} \quad (8)$$

$$C_{IQ}^{LMIIS} = 2\tau(\kappa + C_{PoA,ZMIIS} + C_{ZMIIS,LMIIS} + C_{LMIIS,ZMIIS}) + 2\phi_{ZMIIS} + \phi TP_{LMIIS} \quad (9)$$

$$\begin{aligned} C_{IQ}^{GMIIS} = {} & 2\tau(\kappa + C_{PoA,ZMIIS} + C_{ZMIIS,LMIIS} + C_{LMIIS,GMIIS} + C_{GMIIS,LMIIS} + C_{LMIIS,ZMIIS}) \\ & + 2\phi TP_{ZMIIS} + 2\phi TP_{LMIIS} + \phi TP_{GMIIS} \end{aligned} \quad (10)$$

4 Simulation Results

To evaluate the performance of the presented HMSHNcC (Fig. 4) codec hierarchical MIIS system, the HNDSHO and HMSHNcC functionality was implemented by C# for numerical analysis. Compared with HNDSHO topology (Fig. 2), Fig. 6 shows the first MN to access the heterogeneous network, Fig. 7 shows the next MN that the HMSHNcC topology produces superior results with regard to query number. Because there is no need to query upper servers for different network operators, and with better codec feature, the ZMIIS server's cache reduces the net query cost.

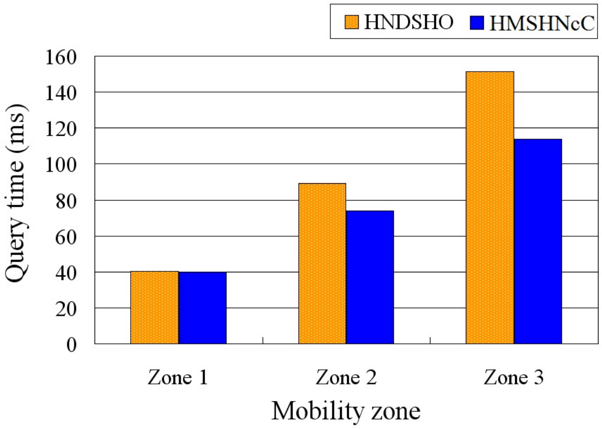


Fig. 6. Query time comparison for the first MN

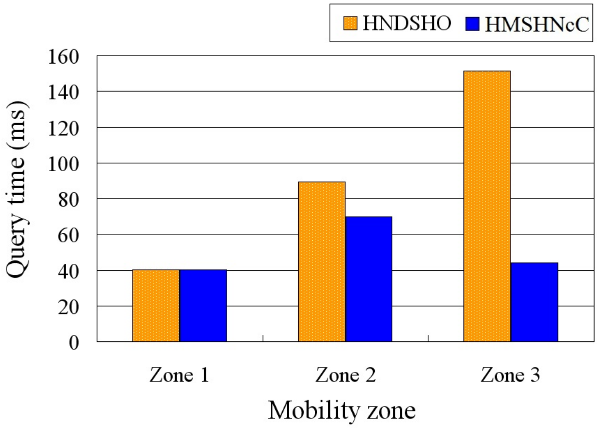


Fig. 7. Query time comparison for the next MN

5 Conclusions

The proposed HMSHNcC based on the HMSHNc mechanism [3], adopts a hierarchical network to classify stored information by network level (class). A cache is added to ZMIIS servers and using ID codec of all MIIS server, both for MN querying of PoA information located in neighbor but different network operators. If a query message has have passed through the GMIIS, then the target ZMIIS reply information is stored in the serving ZMIIS server's cache. This reduces the need for queries to involve the higher levels in the system, thereby reducing Query cost and improving system performance in heterogeneous get larger and larger networks.

Acknowledgments. This research was supported by National Science Council of Taiwan under project number NSC 100-2221-E-277-001 and NSC 100-2221-E-151-036.

References

1. Yoo, S.J., Cypher, D., Golmie, N.: Timely Effective Handover Mechanism in Heterogeneous Wireless Networks. Wireless Personal Communications, 449–475 (2010)
2. IEEE 802.21 Working Group: IEEE Standard for Local and metropolitan area networks-Part 21: Media Independent Handover. In: IEEE Std 802.21-2008, pp. c1-301 (2009)
3. Tseng, P.C., Tsai, J.Y., Hwang, W.S., Chen, Y.S.: Improved query efficiency of hierarchical MIIS servers in heterogeneous networks by caching (HMSHNc). In: 7th International Symposium on Wireless and Pervasive Computing, ISWPC 2012, pp. 1–6 (2012)
4. Buiati, F., Villalba, L.J.G., Corujo, D., Soares, J., Sargento, S., Aguiar, R.: Hierarchical Neighbor Discovery Scheme for Handover Optimization. IEEE Communications Letter (2010)
5. Gustafsson, E., Jonsson, A.: Always best connected. IEEE Communications Magazine 10, 49–55 (2003)
6. Makaya, C., Pierre, S.: Enhanced fast handoff scheme for heterogeneous wireless networks. Computer Networks, 2016–2029 (2008)

Message Relaying with Data Aggregation for Drive-Thru Internet Services in Vehicular Networks

Chih-Lin Hu, Chung-Chun Wang, and Jiun-Yu Tu

Department of Communication Engineering, National Central University, Taoyuan,
Taiwan 32001, R.O.C
clhu@ce.ncu.edu.tw
http://www.ce.ncu.edu.tw/~clhu/index.html

Abstract. Since the drive-thru Internet service becomes one of promising vehicular applications, it can be easily affected by the problem of system overload when unpredictable traffic of request and response workloads exceeds the service capacity that a roadside unit can stand for. To mitigate this situation, this paper proposes an efficient data aggregation mechanism which is able to perform request fusion and response diffusion among vehicles to reduce message traffic in the network. Functionally, a roadside unit can delegate relay vehicles to merge requests from neighboring vehicles into a compound request, and later reply with a batch of response messages. Relay vehicles then deliver responses to requesting vehicles in proximity in a broadcasting manner. By simulation results, the effort of this proposed data aggregation mechanism can significantly reduce message traffic. This mechanism can defer the critical time of system overload to some extent and meanwhile moderate the service starvation situation under heavy request workload.

Keywords: message relay, data aggregation, drive-thru service, vehicular network.

1 Introduction

The advances of IEEE 802.11p dedicated short range communication (DSRC) and radios, wireless and mobile network technologies in intelligent transportation systems have encouraged many appealing scenarios and deployments of mobile data and information services in vehicular ad hoc networks, or so-called vehicular networks for brevity. Typical examples include driving safety, mobile access to Internet, intelligent transport, infrastructure monitoring, and in-time dissemination for emergency information sharing such as accidents and weather hazards [1]. Vehicles are capable of powerful computation, energy, storage, and mobility capacities. A vehicle can communicate with neighboring vehicles during movement, that is, vehicle-to-vehicle (V2V) communications; at the same time, a vehicle can communicate with roadside unites over backend infrastructures, that is, vehicle to infrastructure (V2I) communications.

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 703–712.
DOI: 10.1007/978-3-642-35452-6_71

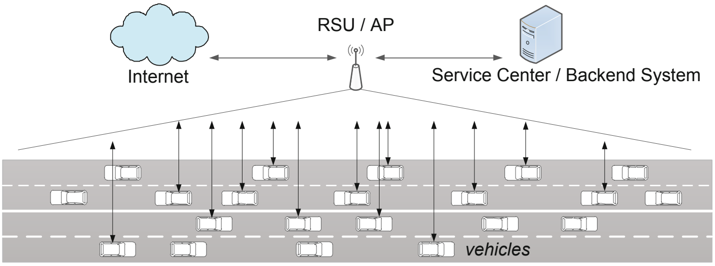


Fig. 1. Drive-Thru Internet services in a vehicular networks

A vehicular network can be a special but practical variant that essentially inherits many functions from traditional mobile ad hoc networks and delay tolerant networks. Data forwarding in vehicle networks reckon on a number of vehicles serving as mobile relay nodes that in turn forward stored-and-carried messages in an opportunistic routing way from a source to a destination. Also, vehicles on the roads can have Internet access through the roadside unites (RSU) over backend infrastructures. Regardless, vehicle networks are characterized by frequent network disruption, fast topological change, high-speed movement and mobility uncertainty, thus rendering a new challenge to achieve efficient data dissemination in vehicular networks.

Fig. 1 shows a generic view of a vehicular network that contains vehicles and roadside units. A vehicle is able to interact with encountered vehicles directly in its short radio range, or communicate with other vehicles indirectly by the RSUs. Vehicles can collectively perform in-network data services, such emergency notification. In addition, vehicles can have external information services, which are often termed as drive-thru Internet [5] through RSUs that can serve as proxies or gateways to the Internet and backend systems.

As one of the most promising services widely discussed in the vehicular application domain, this paper considers the external information services in vehicular networks. Data dissemination in vehicular networks is mainly affected by two issues, including timely transmission and delivery efficiency, making this study apart from conventional data dissemination services in wireless and mobile networks. The former is due to very short contact duration in ten seconds on average when a vehicle passes through a stationary RSU, as reported in [2]. Thus, any data request must be responded in time before the vehicle of this request will move out of the serving range of the RSU in charge. The latter issue further considers dynamic traffic generated by a varied vehicle population within an RSU's serving range. An RSU can inevitably run into overload when request workload and data traffic exceed the service capacity that the RSU can offer [9].

This study proposes a data aggregation mechanism for drive-thru Internet services in vehicular networks. By taking advantage of both V2V and V2I benefits, ideally, the data aggregation mechanism consists of two functions: request fusion from many vehicles in a V2V context, and response diffusion from an RSU to many vehicles in a V2I context. Thus, this mechanism can be used as necessary to reduce message traffic among an RSU and relay vehicles. With simulation-based vehicular networks, several experiments highlight the performance degradation by the incremental request workload. Compared with the naive setting, the data aggregation mechanism comes out a larger amount of successful response messages against a variety of request workload. Performance results show that this mechanism is able to enhance bandwidth efficiency of an RSU, thereby alleviating the service starvation situation especially under high vehicular density, for example, in rush hours or traffic jam.

The remainder of this paper is organized as follows. Section 2 reviews some related works. Section 3 describes the system model. Section 4 specifies the data aggregation mechanism. Section 5 presents the performance evaluation with simulation-based experiments. The concluding remarks are given in Section 6.

2 Related Work

Considering the drive-thru Internet access, many previous research works investigated service processing and bandwidth utilization upon the verge of overloading at the RSU side. In [8], the work designed a network-coding-based MAC protocol that can alternatively take one-hop and two-hop communications to augment network efficiency according to vehicular density under highway traffic loads. The one-hop communication is used as the vehicular density is heavy. Otherwise, the service base station, e.g., RSU, turns on the two-hop relay that can extend the coverage of the non-safety data services to the two-hop area, so as to increase network efficiency. In [9], the work proposed a predictive scheduling scheme to minimize the number of uncompleted file download jobs. By estimating the remaining times that all vehicles will move out the coverage of an RSU, an RSU can determine whether or not the total size of pending file segments in the data queue is larger than the amount of data that it can serve to those vehicles. In case of a system overload, this work employs a scheduling tree to select data in queue in order to maximize the number of download jobs to be completed. Note that [8][9] contributed to the network services under the condition of system overload; however, their efforts did not mitigate the overloaded situation itself.

To prevent the critical time of system overload, this paper refers to several technologies resolving similar overload problems in cellular networks and sensor networks. In [4][10], several efficient strategies were designed to deploy an appropriate amount of relay nodes to offload the strict system load when the number of active users is more than the tolerable maximum at a base station. Another technology is the data aggregation methodology in sensor networks [6]. When sensor nodes collaborate on forwarding messages, strategic nodes can merge several message data into one message to reduce the message workload over the network. This way can reduce the message forwarding and acknowledge flows, thereby maximizing the lifetime of the sensor systems. Recently, a few studies in vehicular networks attempted to use the notion of data aggregation to improve specific performance. For example, [1] designed a road information sharing architecture (RISA) which performs in a distributed approach for road condition detection and dissemination in vehicular networks. This architecture supports in-network aggregation and dissemination of event information detected by multiple vehicles in a timely manner for improved information reliability and bandwidth efficiency. Note that the work in this paper exploits the functionality of data aggregation to sustain the system performance for drive-thru Internet services. The effects of our proposed mechanism can moderate the system load and so improve network bandwidth efficiency between an RSU and vehicles in response to dynamic request workload. Hence, this study could be distinguishable from prior ones that were contributed to in-network data dissemination between moving vehicles on the road.

3 System Model

To ease exposition, Fig. 2 illustrates a vehicular network environment where this paper designs the proposed data aggregation mechanism. For simplification without loss of generality, such an environment includes two basic roles: an RSU and vehicles. An RSU has its own service area that is not overlapped with other RSUs' ones. Every vehicle in an RSU's coverage can communicate with the responsible RSU with reciprocal data transmissions. Vehicles on the road can directly communicate with neighboring vehicles during their contacts in a short time. In addition, vehicles can indirectly interconnect with other vehicles through intermediate vehicles' voluntary forwarding services. Among devices, a special role of relay vehicle is assigned by an RSU; namely, an RSU can select some relay vehicles from the set of vehicles that currently connect with the RUS. Through periodic beacon messages [7], an RSU can advertise the list of relay vehicles in its serving coverage. Definitely, a concise beacon representation, including an RSU's location reference and a list of relay vehicles, suffices for the usage of data aggregation as intended. In addition to directly send data requests to an RSU, vehicles can alternatively send their data requests to any relay vehicle that can further submit these requests to their corresponding RSU.

Assume that the slotted time model is adopted to formulate data transmission in a vehicular network environment; it takes one time slot to transmit each data item. An RSU is able to process one request or response at a certain time slot. Then, an RSU is able to process the response immediately as receiving a request, provided that the database is associated with the RSU or data objects are already there at the RSU with any appropriate data pre-fetching techniques. Thus, the data fetching time can be ignored herein to concentrate on the examination of bandwidth efficiency.

Let data rate from the RSU to the vehicles be identical, denoted as R Mbps, at any time unit T in the time horizon. The concurrent transmissions of data requests and responses share the common bandwidth capacity, denoted as C Mbps for $C \leq R$. An RSU maintains a simple queue to keep pending requests in a first-in-first-out manner, and the queue length is the number of pending requests, denoted as Q. For simplicity, either a request or a response message is

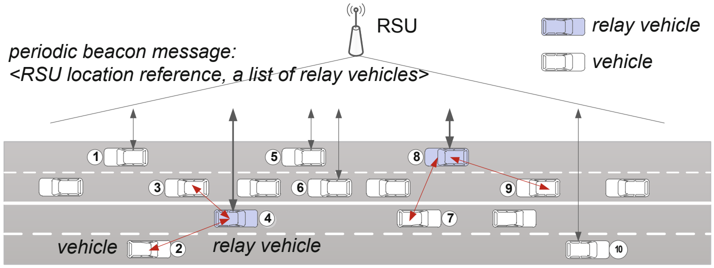


Fig. 2. A simple vehicular network model

formed in a uniform size, denoted as S. At every transmission moment, an RSU can calculate the amount of required data throughput as the value of S times Q. Accordingly, it is able to determine whether the system overload occurs or not, by comparing C with $S \times Q$. Incidentally, the queue size is not limited purposely for determining the number of uncompleted requests as comparing the results between the request-response manner and the proposed method.

4 Mechanism Design

This section firstly abstracts the proposed mechanism in Section 4.1, and then details the functional components, as well as their specifications in Section 4.2.

4.1 Design Abstraction

The proposed data aggregation mechanism takes advantage of both V2V and V2I benefits and accordingly performs data aggregation to sustain system performance for external information services in vehicular networks. This mechanism consists of two main functions: *request fusion* from many vehicles in a V2V context, and *response diffusion* from an RSU to many vehicles in a V2I context. Ideally, an RSU with a finite transmission bandwidth can provide a limited number of data channels that vehicles can access to communicate with the RSU. When all channels have been allocated, any relay vehicle that already occupies one channel can merge some requests from neighboring vehicles along with its request into a compound request delivered to the RSU. Then, the RSU can receive a batch of requests at once and later encapsulate all the results in a compound response. When the relay vehicle obtains the response, the results will be propagated to respective vehicles to somehow share the service loading on behalf of the RSU.

4.2 Functional Design

The vehicular network contains three functional components, i.e., an RSU, relay vehicles, and vehicles, which collaboratively execute the data aggregation

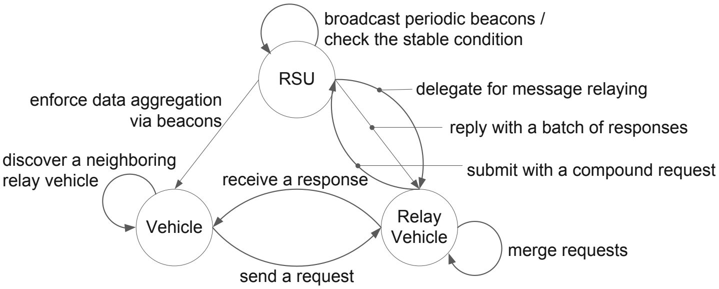

broadcast periodic beacons /
check the stable condition
RSU
enforce data aggregation
via beacons
delegate for message relaying
reply with a batch of responses
submit with a compound request
discover a neighboring
relay vehicle
receive a response
Vehicle
Relay
Vehicle
merge requests
send a request

Fig. 3. A data aggregation model

mechanism in two phases. The first phase is the initial detection phase that an RSU triggers when it detects the imminent situation of system overload. The second phase is to enforce all vehicles to process data aggregation process in a distributed manner. Fig. 3 shows the interactive behavior and functionalities among these components.

In the initial detection phase, an RSU is responsible for monitoring dynamic traffic workload in its serving coverage. Specifically, an RSU repeatedly checks the stable condition subject to the inequality: $C > S \times Q$ by referring to the varied queue length Q, and constant C and S. When the available bandwidth capacity is larger than the accumulation of requested data in buffer, an RSU can immediately output all data traffic without delay. However, as $C < S \times Q$, the RSU invokes the second phase to execute the data aggregation mechanism to cope with the increasing data traffic in a vehicular network. It is noted herein that notwithstanding the amount of incoming requests varies dynamically, this initial detection phase performs in a straight fashion to determine system overload instantly without regard to the preceding or the following request workload. This design is simplified but reasonable because of unpredictable traffic workload in vehicular networks of high dynamics.

The second phase of data aggregation contains request fusion and response diffusion that are mutual processes in the V2V and V2I contexts, respectively, as mentioned below with the graphic presentations in Figs. 2 and 3.

Request Fusion. Request fusion is a canonical approach to reducing the number of request messages by exploiting the V2V paradigm in a vehicular network. Specifically, an RSU selects some vehicles from active vehicles (that is, the set of vehicles that currently occupy channels) to form a set of relay vehicles. Then, the RSU updates the content of a periodic beacon message with a list of relay vehicles' identifiers, and broadcasts new beacons to all vehicles in its coverage. Every active vehicle receiving the broadcast beacon can know whether it is selected as a relay vehicle or not. Meanwhile, any non-active or ordinary vehicles (that is, the set of vehicles that are waiting for free channels) can learn the set of relay vehicles and then figure out the closest relay vehicle in its surrounding. Herein, the examination of search techniques is not the goal of this study. Any suitable search technique in vehicular, mobile ad hoc, or delay-tolerant networks can be used for this purpose. Without loss of generality, this design adopts a common neighboring table method to determine a relay vehicle in charge of message relaying from/to an ordinary vehicle in vehicular networks.

Therefore, ordinary vehicles can forward requests to neighboring relay vehicles in place rather than wait for submitting requests to the RSU till some active vehicle releases its channel. On behalf of the RSU, a relay vehicle likely receives a number of requests from ordinary vehicles in a short time. It can append received requests with its own request, resulting in a compound request submitted to the RSU by only one message transmission. Referring to an example as shown in Fig. 2, an RSU can simultaneously serve six vehicles (no. 1, 4, 5, 6, 8, 10) at most in a naive response-response method. Relatively, if the data aggregation is used, the RSU can delegate some active vehicles (no. 4, 8) as relay vehicles, so ordinary vehicles (no. 2, 3, 7, 9) can ask for the relay vehicles to forward their requests to the RSU. Therefore, the request fusion is able to significantly decrease the amount of requests to the RSU. Meanwhile, service starvation to ordinary vehicles can be alleviated to some extent.

Response Diffusion. Response diffusion lightens the system load of an RSU with better bandwidth usage of V2I communication channels in a vehicular network. An RSU sends out a batch of response messages once with only one transmission, instead of unicasting responses to individuals. Although the size of a batch is larger, the transmission time at one or several consecutive time slots, in effect, is still shorter than the transmission duration required by multiple transmissions to different vehicles. Hence, it is cost-effective to perform the response diffusion after the request fusion. The RSU delivers the relay vehicle all the responses corresponding to those requests prescribed in the compound request. The relay vehicle further distributes the responses to respective vehicles in a V2V context, same as the message forwarding manner in request fusion. As the example in Fig. 2, the relay vehicles (no. 4 and 8) symmetrically forward responses to ordinary vehicles (no. 2 and 3, and no. 7 and 9) like in a round-trip manner. Incidentally, although the route to some vehicle may be changed during the short time by response diffusion, the response can be forwarded to the target vehicle through the same neighboring table method used in request fusion.

Hence, all parties involved in the data aggregation procedure can perform in a distributed fashion in compliance with the state flowchart in Fig. 3. It is noteworthy that all messaging flows of relayed requests converge toward the only RSU, and all messaging flows of relayed responses originate from this RSU. An RSU is able to promptly respond to dynamics of network and traffic situations in vehicular networks. More importantly, an RSU is able to synchronize all vehicles using periodic beacons over the vehicular broadcast medium, while all vehicles are autonomous in the coverage of an RSU. Likewise, an RSU is able to update the cached information on vehicles. As a result of design simplicity, the proposed data aggregation mechanism could be amenable with respect to bandwidth efficiency, as well as the treatment of service starvation and system overload for the drive-thru Internet services in vehicular networks.

5 Performance Results

This section conducts synthetic simulations to investigate the effects of the proposed data aggregation mechanism for drive-thru Internet services in terms of three measurement metrics, including average throughput, successful delivery rate, and delay time. Section 5.1 describes the simulation environment, and performance results are mentioned in Section 5.2.

5.1 Simulation Environment

To formulate a drive-thru Internet application scenario, this study conducts the simulation based on NS 2.35 network simulator with VanetMobiSim modules [3] that can generate vehicle movement patterns in vehicular networks. In this simulation, vehicular communications are referred to IEEE 802.11p model with basic specifications, e.g., 5.9 G medium spectrum, 10 MHz channel bandwidth, 3 Mbps BPSK modulation, and Nakagami propagation mode. The simulation generates a two-way road with 1000 m in a direct distance wherein only one RSU lies to serve any autonomous vehicles that freely move in or out. To simplify the experimental settings, all vehicles and the RSU have the same transmission distance in 230 radius meters. The simulator controls the number of vehicles in the coverage of an RSU as 50 vehicles on average per second; among these vehicles, 5 vehicles are randomly delegated as relay vehicles by an RSU to deal with the treatment of request and response messages as previously mentioned in Section 3. Every vehicle can generate a request message per second. Either request or response messages are of the same message size in 500, 1000, 1500, 2000, or 2300 bytes. Regarding traffic generation and measurement, the simulation runs in 100 seconds to produce data workload and repeats all experiments in 6 times to have the average results for performance examination.

5.2 Results and Discussion

This section investigates the effects of message relaying with data aggregation on channels between an RSU and relay vehicles, in comparison with the ordinary approach that all vehicles contend for channels to the RSU. Notice that in practical situations, vehicles in V2V and V2I contexts can use different dedicated channels; the study thus emphasizes on relative performance under the V2I context where the likelihood of data aggregation is mainly considered. The comparative results of throughput, successful delivery rate, and delay time are exhibited in two parts: Fig. 4 shows the results in the aspect of message delivery from vehicles to an RSU, and Fig. 5 shows the results in the reverse aspect from an RSU to vehicles.

On applying the proposed data aggregation mechanism, the enhancement of service capability and stability on the channels between an RSU and vehicles is obvious. As Figs. 4(a) and 5(a) depict, the proposed mechanism achieves the continuous increase of data throughput in response to ascending traffic with larger message sizes. Relatively, the ordinary approach can fall into performance

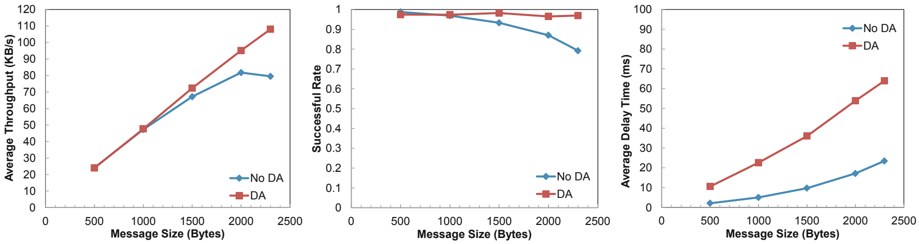


Fig. 4. The performance results of throughput, successful rate, and delay time from vehicles to an RSU

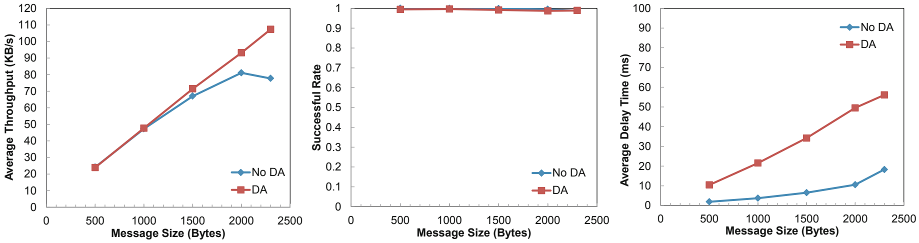


Fig. 5. The performance results of throughput, successful rate, and delay time from an RSU to vehicles

saturation earlier in both requesting and responding directions, while the message size is close to the marginal size, i.e., 2000. Such a critical phenomenon jeopardizes service reliability in the coverage of an RSU. By contrast, data aggregation can efficiently defer the saturation situation and simultaneously attain a very high successful delivery rate. Particularly, as depicted in Figs. 4(b) and 5(b), the successful rate by the proposed mechanism is very close to 1, thereby manifesting its service reliability. On the other hand, the proposed mechanism can have performance compromise in part from the viewpoint of delay time. Figs. 4(c) and 5(c) show that the performance of delay time by the proposed mechanism is less than the ordinary approach. This sort of delay can be easily understood for the process of data aggregation that is additionally employed to reduce traffic load. Nonetheless, it is noteworthy in Fig. 4(b) that the ordinary approach can cause a considerable number of dropped requests; comparatively, all requests are almost answered when the proposed mechanism is used instead. Therefore, the proposed data aggregation mechanism with more reliable performance improvement is amenable for message delivery services against traffic dynamics in vehicular network environments.

6 Concluding Remarks

This paper proposes a data aggregation mechanism for drive-thru Internet services in vehicular networks. The proposed mechanism comprises two specific processes, request fusion and response diffusion, which can moderately mitigate the problem of service overload on communication channels between an RSU and vehicles under heavy traffic loads in the network. Performance studies examine the proposed mechanism in terms of data throughput, successful delivery rate, and delay time in synthetic vehicular network environments. Simulation results show that this mechanism is able to achieve reliable performance improvement with better bandwidth utilization and service capacity for drive-thru message delivery in vehicular networks.

Acknowledgement. This work was supported in part by the National Science Council of Taiwan, R.O.C., under Contract No. NSC100-2221-E-008-085-MY3; also, Dr. Hu's work in this paper was partially supported by ITRI under Grant No. B352BR1N3A.

References

1. Ahn, J., Wang, Y., Yu, B., Bai, F., Krishnamachari, B.: Risa: Distributed road information sharing architecture. In: Proceedings of IEEE INFOCOM 2012, pp. 1494–1502 (March 2012)
2. Bychkovsky, V., Hull, B., Miu, A., Balakrishnan, H., Madden, S.: A measurement study of vehicular internet access using in situ wi-fi networks. In: Proceedings of ACM MOBICOM 2006, pp. 50–61 (September 2006)
3. Fiore, M., Harri, J., Filali, F., Bonnet, C.: Vehicular mobility simulation for vanets. In: Proceedings of the 40th IEEE Annual Simulation Symposium (ANSS 2007) (March 2007)
4. Kolios, P., Friderikos, V., Papadaki, K.: Load balancing via store-carry and forward relaying in cellular networks. In: Proceedings of IEEE GLOBECOM 2010 (December 2010)
5. Ott, J., Kutscher, D.: Drive-thru internet: IEEE802.11b for "automobile" users. In: Proceedings of IEEE INFOCOM 2004, pp. 362–373 (March 2004)
6. Rajagopalan, R., Varshney, P.K.: Data-aggregation techniques in sensor networks: a survey. IEEE Communications Surveys & Tutorials 8(4), 48–63 (2006)
7. Ros, F.J., Ruiz, P.M., Stojmenovic, I.: Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks. IEEE Transactions on Mobile Computing 11(1), 33–46 (2012)
8. Su, H., Zhang, X.: Network-coding-based relay mac protocols for drive-thru internet services in vehicular networks. In: Proceedings of IEEE GLOBECOM 2010 (December 2010)
9. Yang, S., Yeo, C.K., Lee, B.S.: Predictive scheduling in drive-thru networks with flow-level dynamics and deadlines. In: Proceedings of IEEE ICC 2011 (June 2011)
10. Yanmaz, E., Tonguz, O.K.: Dynamic load balancing and sharing performance of integrated wireless networks. IEEE Journal on Selected Areas in Communications 22(5), 862–872 (2004)

Design and Analysis of Hybrid Wireless Mesh Networks for Smart Grids

Philip Huu Huynh and C. Edward Chow

Department of Computer Science, University of Colorado at Colorado Springs, Colorado, USA
{phuynh,cchow}@uccs.edu

Abstract. In this paper the use of a Hybrid Wireless Mesh Network (HWMN) technologies for the smart grid of Advanced Metering Infrastructure (AMI), which enables the collecting of meter data in real-time, was proposed and analyzed. A Google Maps mashup was developed to read the real GIS data from a local utility company and display the locations of the meters and light poles, which can be selected for mounting the WiMAX or Wi-Fi network devices. A NS-3 simulator was developed to simulate the network traffic of meter data collection over the HWMN and to allow us to evaluate different topologies and to see if their capacities are adequate to report all meter values to the data center within one second. The preliminary simplified simulation results show that with the proper antennae selection for the HWMN, it is feasible to collect meter data from hundreds of thousands smart meters within one second.

Keywords: AMI, WiMax, Wi-Fi, Smart Grid, Simulation, Network Planning.

1 Introduction

Recently many utilities deployed smart grids for collecting meter data [1]. The main reasons are to reduce the cost by not sending people to read the meter data and by avoiding generating excess power through correct prediction of the load profile using the aggregated meter values. To correctly predict the load profile and perform load forecasting, utilities need to collect meter data in real time.

Utilities have hundreds of thousands of meters installed in their service areas, and want to network these meters for metering collection. The wireless communication technologies have been popularly deployed in the local areas and the metropolitan areas because of their conveniences in the cost, network installation and maintenance. Taking advantages of the wireless technologies, utilities can network their meters and the data center for data communication. However, if the underlined wireless technologies do not provide enough bandwidth, then the meter data cannot be delivered to the data center in time. The WiMAX technology allows us networking the meters and the data center with the broadband data transmission at long distance and higher bandwidth [2, 3]. Therefore, the wireless networking solution for real-time metering collection is feasible. The important question is to design of the wireless infrastructure and their topology so that it can be scaled up and meet the cost and real-time perfor-

R.-S. Chang et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 20, pp. 713–721.
DOI: 10.1007/978-3-642-35452-6_72

mance requirements, given huge wireless meters to be served in large areas. For example, The Wi-Fi mesh technology can be employed as part of a hybrid wireless infrastructure with WiMAX and Wi-Fi to allow the deployment at the reasonable low cost [4].

The wireless technologies such as WiMAX, and Wi-Fi are high performance, scalable, and secured [4]. Taking the advantages of these network technologies, utilities can deploy the smart grid wireless communications infrastructure for the real-time metering collection. The real-time meter data can save the operation costs and reduce the electricity market price.

When the WMN technology is applied in AMI solutions, it can bring new components to the electrical grids, such as self-managing and self-healing mesh networking, intelligent meters, and bridging to Home Area Networks (HAN) [5] for connectivity with energy consuming appliances. The real time communication between the smart meters and the utility's data center provides detailed usage data while also receives and display Time-of-Use (TOU) pricing information, and offers other on-demand abilities such as remote connect or disconnect, unrestricted monitoring and control, etc. Customers are able to access the usage data for tailoring consumption and minimizing energy expenses while helping balance overall network demand.

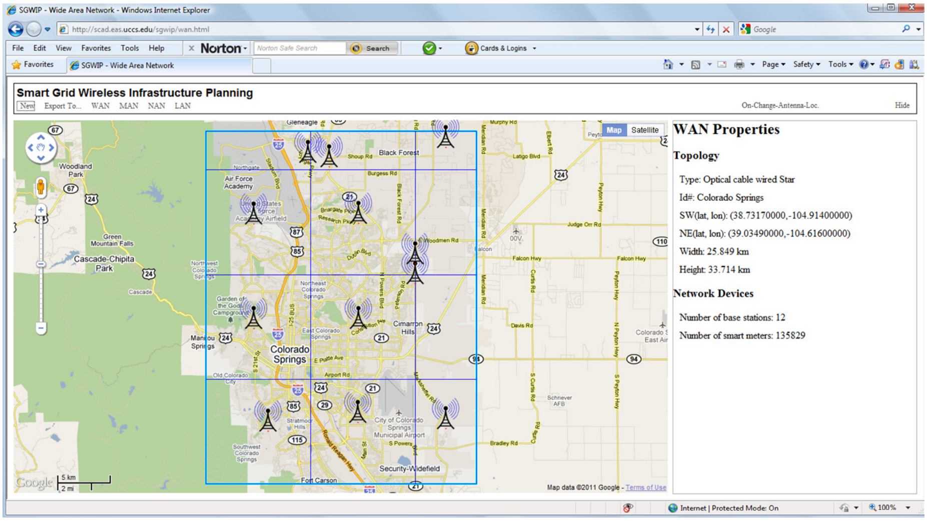
When WMNs are used in the AMI, they can provide the following features [6, 7]:

- *Low cost of management and maintenance* - WMNs are self-organizing and require no manual address/route/channel assignments. It is simple to manage thousands or millions of devices resulting in the lowest total cost of ownership.
- *Increased reliability* – The WMN routing mechanisms provide the redundant paths between the sender and receiver of the wireless connection. Communication reliability is significantly increased because of the eliminations of single point failures and potential bottleneck links. Network robustness against potential problems, e.g., node failures and path failures due to RF interferences or obstacles, can also be ensured by the existence of multiple possible alternative routes.
- *Scalability, flexibility and lower costs* - WMNs are self-organizing and allow true scalability. Nodes and gateways are easily added at a very low cost with:
  - No limitation on number of hops
  - No network address configuration
  - No managed hierarchical architecture
  - No hard limitation on number of nodes per gateway
- Robust security – WMNs have the security standards that allows all communications in AMI are protected by mutual device authentication and derived per-session keys using high bit rate AES encryption. This hardened security approach allows for authentication as well as confidentiality and integrity protection in each communication exchange between every pair of network devices – Smart meters, Relays, or Wireless Gateways.

In this paper, we focus on the AMI solutions using the hybrid wireless mesh network technologies such as WiMAX and Wi-Fi that are capable of collecting data in real-time. A wireless network planning tool was developed based on the existing utility infrastructure and GIS information and the performance of the large wireless network infrastructures was evaluated using the network simulation software NS-3. Section 2 shows the design and implementation of the smart grid wireless planning tool. Section 3 shows the simulation results of the HWMN for smart grids. Section 4 is the conclusion remarks.

2 Smart Grid Wireless Infrastructure Planning (SG-WIP) Tool

The SG-WIP is a wireless network topology planning application. We have developed this planning tool to assist the planning and designing phase of the AMI wireless network infrastructure. Figure 1 shows the GUI of SG-WIP.



Smart Grid Wireless Infrastructure Planning

WAN Properties

Topology

Type: Optical cable wired Star

Id#: Colorado Springs

SW(lat, lon): (38.73170000,-104.91400000)

NE(lat, lon): (39.03490000,-104.61600000)

Width: 25.849 km

Height: 33.714 km

Network Devices

Number of base stations: 12

Number of smart meters: 135829

Fig. 1. SG-WIP tool for planning AMI wireless infrastructure network in Colorado Springs

The SG-WIP is a Google Maps mashup [8, 9]. It reads in the GIS data of a utility company and provides the information about the geographical locations of the network topologies, network devices, and the residential housing units in the service areas of a utility. Real data were provided by a local utility and used in this study.

With the real utility GIS data, we also conducted related researches that use the SGWIP tool, including antenna placement problem and housing unit density statistics.

The research for antenna placement of the WiMAX/Wi-Fi networks utilizes the SG-WIP platform as a tool to extract information of the geographical network topologies such as housing unit locations, or street light poles. Utilities typically use their street light poles for installing wireless base stations if they own and manage them. It turns out some light poles were not suitable as antennae placement locations due to their structure and power source limitation. Therefore logistically they can not be used in real situations.

Figure 2 shows the planning antennae placement for the smart meters and the WiMAX/Wi-Fi gateway on the Google Maps. The Google Maps with satellite images help verify the locations of wireless devices and related placement decisions.

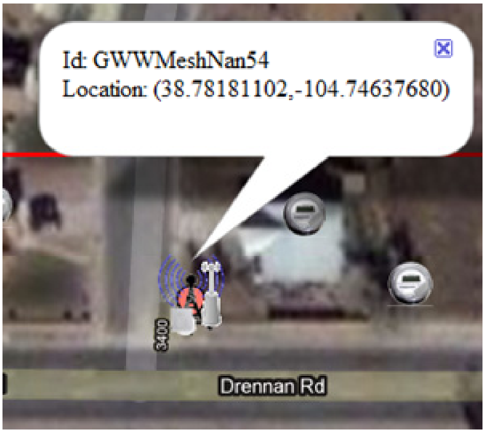


Fig. 2. Using SG-WIP tool for planning the antennae position. A WiMAX/Wi-Fi gateway was placed on a streetlight pole.

The research about housing unit density of the designing wireless networks also used the SGWIP tool to gather the distribution of the housing units. It enables us to consider smart meter density as a factor in our simulation studies.

Table 1 shows the range of number housing units in the LAN, NAN, WAN topologies. The dimensioning information is helpful for the designing of smart grid network simulation. For example, it helps deciding the sizes for the grids.

Table 1. The ranges of housing unit density of the LAN, NAN, MAN topologies in Colorado Springs

	Low Bound (housing units)	High Bound (housing units)	Simulation
LAN	0	51	50
NAN	0	1,054	950
MAN	0	40,501	27,000

Figure 3 shows the WLAN topology size 100x100 square meters that has fifty one housing units. The SIGWIP tool overlays the meter symbols on the houses based on the GIS locations of the meters.

The information about the network topologies from SG-WIP tool, as well as the research results about the housing unit density, and the antenna locations can help the AMI network infrastructure researchers and designers in their simulation and analysis of the wireless network infrastructure of the AMI.

The SG-WIP will be made available for academic research.

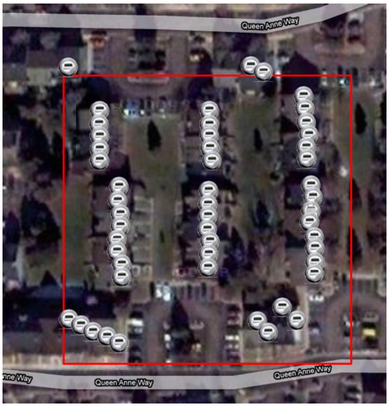


Fig. 3. The WLAN topology (100x100 square meters) has a high density of resident housing units (51)

3 HWMN Simulation

In the proposed HWMN network, we assume Wi-Fi base stations (BS) are used to collect meter data from the smart meters, On top of that, WiMax gateways (GW) are used to provide routing paths from these Wi-Fi base stations to a group of "take-out" points. From the handful of take-out points, the huge volume of meter data are sent over high speed optical fiber communication links to the utility data center (DC).

In this simplified hierarchical network simulation study, we assume the whole metro area served by the utility were divided into layers of regular square areas. NS-3 simulation studies were conducted on Wi-Fi network with each Wi-Fi network served a small number smart meters based on the house density statistics collected with network planning studies conducted with SG-WIP. We then feed the traffic statistics to simulate the WMAN network with variable GWs servicing a region of the metro area. We also investigated the impact of fiber cable length from the take-out points to the Data Center. Here are some of the simulation results.

In Figure 4, the number of the UDP packets sent and received in every one second for the simulation duration versus the number of GWs for different simulation experiments. The total processing delay is also plotted on the chart.

We can see that the network has successfully transmitted the UDP packets in every one second from the senders (or GWs) to the receiver (or BS).

Moreover, we can see that the total processing delay is between 930 and 960 milliseconds. It does not increase with the number of the GWs. This is due to the WiMAX network or 802.16d standard has a fixed frame time (5ms) that is independent to the number of the subscribers.

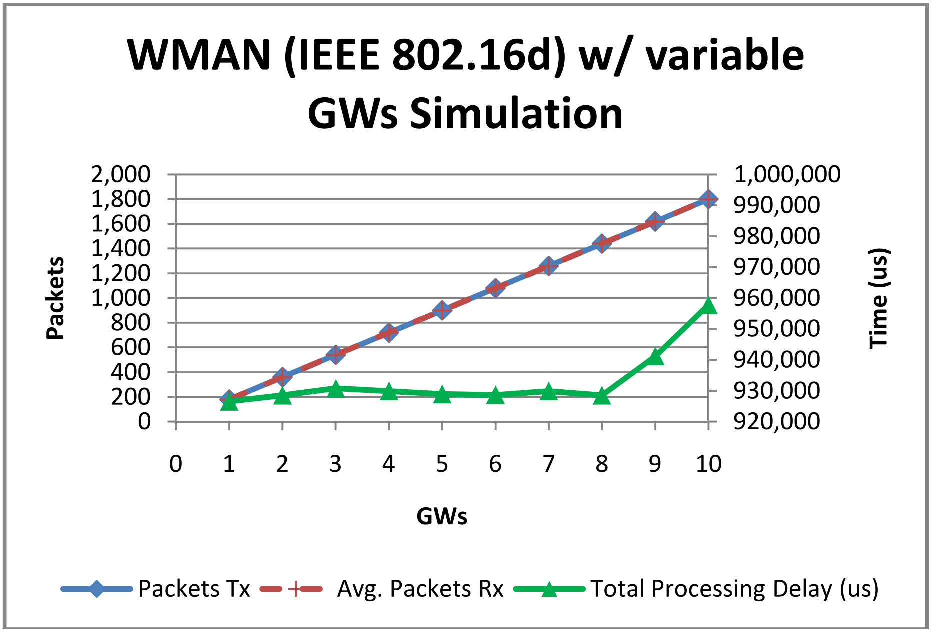


Fig. 4. The simulation results for the WMAN topology. The number of GWs is assigned from one to ten in the experiments to evaluate the changing of total processing delay at the GWs.

3.1 Impact on the Network Performance by Aggregating Meter Data

Figure 5 shows the simulation results for the WMAN point-to-multipoint topology from many experiments. There are one BS and ten GWs in the network. Each GW sent 180 packets to the BS in every second. The number of meter data packets put in a transmitted UDP packet was assigned from the one to seventeen packets to evaluate the changing of total processing delay at the BSs in the experiments. Figure 5 shows the improvement of network performance when the number of meter data packets are aggregated in a transmitted UDP packet or the length of loaded data in one WiMAX frame. The number of meter data packet was increased until the network going to the overloaded state. As we can see on the chart, when the number of meter packets is less than 16, the network successfully transmitted all of the UDP packets. This is due to the sending UDP packet that contains a designated number of meter data packets, is not fragmented in the transmission. Moreover, the traffic application was programmed to send out in every second the number of UDP packets that can be delivered completely by the network in one second.

However, the network is overloaded when the number of embedded meter data packets is equal or greater than 16. This is due to the UDP packets were fragmented in the transmission. That caused the number of received packets in one second to be less

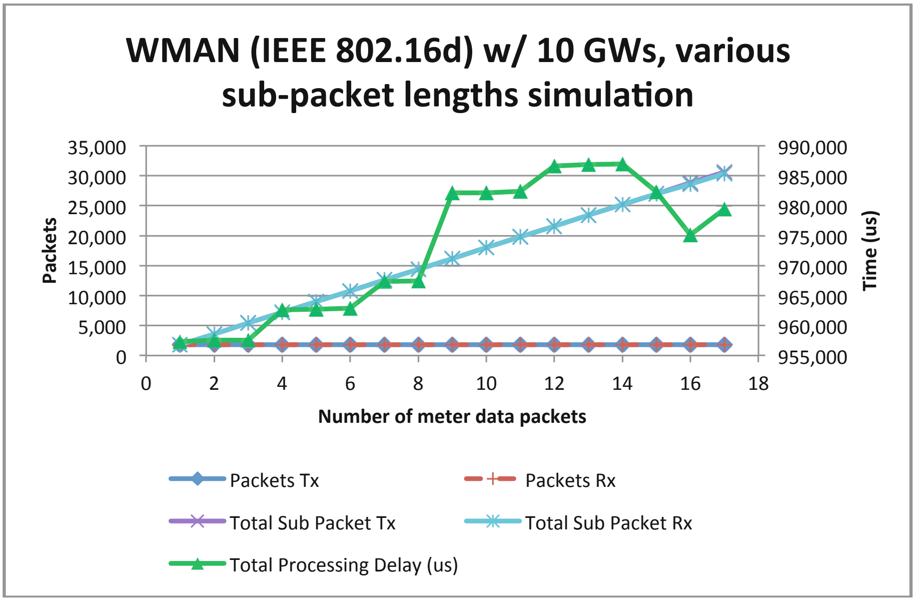


Fig. 5. Impact on the network performance by aggregating meter data

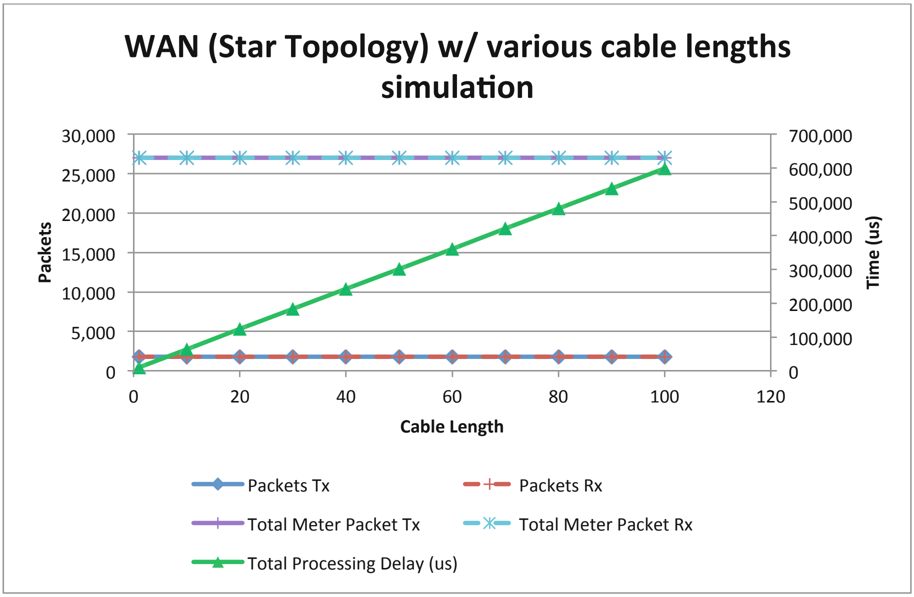


Fig. 6. The simulation results for the WAN star topology with one DC and one BS. The length of the optical fiber cable was assigned from one to 100 km.

than the number of sent packets. As a result, the average transmission delay of a packet was increased.

It is also observed that the total processing delay is independent from the number of the BSs. This is due to the BSs were connected to the data center in the point-to-point connections. The network can transmit over 180,000 meter data packets that sent from seven BSs, and the total processing time at each BS is less than 600 milliseconds. However, the average delay is affected by the distribution of the BSs around the data center (DC).

Figure 6 shows the simulation results for the WAN star topology from many experiments. There was one DC and one BS in the network. GW sent 1,800 packets to the DC in every second. The length of the optical fiber cable that connects the DC and BS was assigned from the one to 100 kilometers to evaluate the changing of total processing delay at the BS in the experiments. We can see that the total processing time was linearly increased with the length of the optical fiber cables that connect the BSs and the data center.

4 Simulation Limitations and Future Work

In this simplified simulation study, we did not include the 3D model and terrain information into the considerations for signal interference and degradation. Future work includes better integration of SG-WIP with the NS-3 simulation modules. Demand response is one of the important goals of the AMI deployment. In contrast to the metering collection, the demand response supporting applications will request the meter data from the data center for the consumer's demand analysis. It is expected that the demand response applications can help the consumers utilize their energy more efficiently. Although this subject is out of the scope of this paper, it is an important area of research, where the AMI researcher community can contribute through the design and performance evaluation of the HWMN infrastructure.

5 Conclusion

AMI is being implemented by many utilities around the world. AMI contributes the benefits not only to the utilities but also to the consumers. AMI real-time meter data collection can give the utilities and consumers the ability to access the real-time meter data. Consumers are benefits from the sharing real-time meter data because they can monitor and actively adjust their demand of electric, gas, and water to save money. Utilities are benefits from the real-time meter data because they can use the real-time meter data to improve the quality of load profile charts, and load prediction. So the utilities can save the fuel usage of power plants and reduce the price of electricity.

Many utilities have implemented the AMI wireless infrastructures for collecting meter data automatically. However, most of the deployed wireless infrastructure did not support or have not supported yet the real-time meter data collection. The intervals for meter data collection are typically higher than one minute. The current meter data collecting period is often in the range between fifteen and forty five minutes.

We have developed a software tool SG-WIP for planning and designing the AMI wireless infrastructure using the real utility light poles and meters GIS data from the city of Colorado Springs, Colorado. SG-WIP allows us to examine the actual geographical distribution of smart meters and a real utility. It also served as an education and training tool for discussing the smart grids and related issues.

We proposed a parameterized WiMAX/Wi-Fi network model and implemented it in the NS-3 platform. Experiments were conducted using the network simulation process, including the WLAN (Wi-Fi) simulation, the WNAN (Wi-Fi Mesh) simulation, the WMAN (WiMAX) simulation, and the WAN (optical fiber point-to-point connection) simulation. The simulation results show that the proposed WiMAX/Wi-Fi Hybrid WMN infrastructure can transport the meter data from 160,000 smart meters in the CSU service areas to the data center in one second.

From the simulation result analysis, we can conclude that the high scalability property of WiMAX/Wi-Fi WMN helps flexibly extend the coverage area of the AMI wireless infrastructure without degrading the network performance.

References

1. U.S. Department of Energy, Smart Grid, http://www.oe.energy.gov/smartgrid.htm
2. IEEE Standard 802 Part 16: Air Interface for Broadband Wireless Access Systems (2009)
3. Zhao, S., Raychaudhuri, D.: On the Scalability of Hierarchical Hybrid Wireless Networks. In: Proceedings of the Conference on Information Sciences and Systems (CISS 2006), pp. 711–716 (March 2006)
4. Akyildiz, I.F., Wang, X., Wang, W.: Wireless Mesh Networks: A Survey. Computer Networks Journal 47(4), 445–487 (2005)
5. National Energy Technology Laboratory, white paper. Advanced Metering Infrastructure (February 2008)
6. Liu, B., Liu, Z., Towsley, D.: On the capacity of hybrid wireless networks. In: Proceedings of IEEE INFOCOM, vol. 2, pp. 1543–1552 (March 2003)
7. Krishnamurthy, S.: Smart AMI Network Solutions Enable the Smart Grid., ElectricEnergyOnline.com, http://www.electricenergyonline.com/?page=show_article&mag=55&article=395
8. Purvis, M., Sambells, J., Turner, C.: Beginning Google Maps Applications with PHP and Ajax. Apress (2006)
9. Google Maps JavaScript V3, http://code.google.com/apis/maps/documentation/javascript/

Author Index