Lecture Notes in Computer Science 4286

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Paul Spirakis Marios Mavronicolas
Spyros Kontogiannis (Eds.)

Internet and Network Economics

Second International Workshop, WINE 2006
Patras, Greece, December 15-17, 2006
Proceedings

Springer

Volume Editors

Paul Spirakis
Research Academic Computer Technology Institute (RACTI)
Department of Computer Engineering and Informatics
University of Patras
Patras 26500, Greece
E-mail: spirakis@cti.gr

Marios Mavronicolas
Department of Computer Science
University of Cyprus, Cyprus
E-mail: mavronic@cs.ucy.ac.cy

Spyros Kontogiannis
Computer Science Department, University of Ioannina
45110 Ioannina, Greece
Research Academic Computer Technology Institute
26500 Rio Patra, Greece
E-mail: kontog@cs.uoi.gr, kontog@cti.gr

Library of Congress Control Number: 2006937397

CR Subject Classification (1998): H.4, K.4.4, J.1, H.3, H.2, G.1.2

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-68138-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-68138-0 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11944874 06/3142 5 4 3 2 1 0

Preface

The present volume is devoted to the second edition of the international Workshop on Internet and Network Economics (WINE), an interdisciplinary conference intending to provide a forum for researchers as well as practitioners to exchange innovative ideas and to be aware of each other's efforts and results. This second edition of the conference (WINE 2006) was hosted by the Research Academic Computer Technology Institute, at the University of Patras, December 15–17, 2006.

The volume contains all contributed papers presented at WINE 2006 (ordered according to the Scientific Program of the workshop), together with the distinguished invited lectures of Abraham Neyman (Hebrew University of Jerusalem, Israel), Mihalis Yannakakis (Columbia University, USA) and Xiaotie Deng (City University of Hong Kong, Hong Kong, SAR, China). This year WINE was under the auspices of the European Association for Theoretical Computer Science (EATCS).

In response to the Call for Papers, the Program Committee received 79 submissions. Among the submissions, there were 15 papers with at least one coauthor that was also a PC member of WINE 2006. For these PC–coauthored papers, an independent subcommittee (Marios Mavronicolas Chair, Elias Koutsoupias, Eva Tardos) made the judgement, and eventually seven papers were proposed for inclusion in the scientific program. For the remaining 64 (non–PC–coauthored) papers, the PC of WINE 2006 conducted a thorough evaluation and electronic discussion, and eventually selected 25 papers for inclusion in the scientific program.

We wish to thank the creators of the EasyChair System, a free conference management system provided and supported by the group of Professor Voronkov, which significantly assisted the work of the Program Committee. Finally, we wish to thank the Research Academic Computer Technology Institute for kindly offering its facilities and human resources for the successful organization of WINE 2006.

December 2006

Paul Spirakis (Program and Organizing Chair)
Marios Mavronicolas (Program Co–chair)
Spyros Kontogiannis (Organizing Co–chair)

Organization

WINE 2006 was organized by the Research Academic Computer Technology Institute, University of Patras, Greece, in cooperation with EATCS.

Program Committee

Xi Chen	Tsinghua University, China
Xiaotie Deng	City University of Hong Kong, Hong Kong, SAR China
Daniel Grosu	Wayne State University, USA
Kamal Jain	Microsoft Research, USA
Ehud Kalai	Northwestern University, USA
Ming-Yang Kao	Northwestern University, USA
Spyros Kontogiannis	University of Ioannina, Greece
Elias Koutsoupias	University of Athens, Greece
Vangelis Markakis	University of Toronto, Canada
Marios Mavronicolas	University of Cyprus, Cyprus (Co–chair)
Igal Milchtaich	Bar-Ilan University, Israel
Dov Monderer	Technion, Israel
Burkhard Monien	University of Paderborn, Germany
Christos Papadimitriou	University of California at Berkeley, USA
Maria Serna	Universitat Politecnica de Catalunya, Spain
Paul Spirakis	Computer Technology Institute, Greece (chair)
Eva Tardos	Cornell University, USA
Vijay Vazirani	Georgia Institute of Technology, USA

Steering Committee

Xiaotie Deng	City University of Hong Kong, Hong Kong, SAR China
Ehud Kalai	Northwestern University, USA
Christos Papadimitriou	University of California at Berkeley, USA
Paul Spirakis	Computer Technology Institute, Greece
Yinyu Ye	Stanford University, USA

Organizing Committee

Paul Spirakis	Computer Technology Institute, Greece (chair)
Spyros Kontogiannis	Computer Technology Institute, Greece (Co–chair)

Panagiota Panagopoulou	Computer Technology Institute, Greece
Rozina Efstathiadou	Computer Technology Institute, Greece
Aggeliki Stamatopoulou	Computer Technology Institute, Greece
Lena Gourdoupi	Computer Technology Institute, Greece

External Reviewers

Babarro, Joaquim
Bleischwitz, Yvonne
Bu, Tianming
Busch, Costas
Carroll, Thomas E.
Chakrabarty, Deeparnab
Daskalakis, Constantinos
Devanur, Nikhil
Dobzinski, Shahar
Dumrauf, Dominic
Elkind, Edith
Feldmann, Rainer
Gairing, Martin
Georgiou, Chryssis
Gkantsidis, Christos
Goldberg, Paul
Golovin, Daniel
Hegde, Rajneesh
Jiang, Min
Kolliopoulos, Stavros
Lavi, Ron
Mehta, Aranyak
Mirrokni, Vahab
Mu'alem, Ahuva
Pai, Mallesh
Philippou, Anna
Qi, Qi
Sami, Rahul
Schoppmann, Florian
Sun, Wei
Tiemann, Karsten
Tscheuschner, Tobias
Vöcking, Berthold
Wagner, Klaus
Xhafa, Fatos
Chen, Jin
Yang, Guoming

Sponsoring Institutions

- Ministry of Natural Education and Religious Affairs, Greece
- University of Patras, Greece
- DELIS: Dynamically Evolving Large-Scale Information Systems (EU Research Project Within 6th Framework Programme Under Contract 001907)

Table of Contents

Recent Developments in Learning and Competition with Finite Automata (Extended Abstract)

Abraham Neyman

Institute of Mathematics and Center for the Study of Rationality,
The Hebrew University of Jerusalem, Givat Ram, Jerusalem 91904, Israel
aneyman@math.huji.ac.il
www.ratio.huji.ac.il/neyman

Consider a repeated two-person game. The question is how much smarter should a player be to effectively predict the moves of the other player. The answer depends on the formal definition of effective prediction, the number of actions each player has in the stage game, as well as on the measure of smartness. Effective prediction means that, no matter what the stage-game payoff function, the player can play (with high probability) a best reply in most stages. Neyman and Spencer [4] provide a complete asymptotic solution when smartness is measured by the size of the automata that implement the strategies.

Let $G = \langle I, J, g\rangle$ be a two-person zero-sum game; I and J are the set of actions of player 1 and player 2 respectively, and $g : I \times J \to \mathbb{R}$ is the payoff function to player 1. Consider the repeated two-person zero-sum game $G(k, m)$ where player 1's possible strategies are those implementable by an automaton with k states and player 2's possible strategies are those implementable by an automaton with m states. We say that player 2 can *effectively predict* the moves of player 1 if for every reaction function $r : I \to J$ player 2 has a strategy (in $G(k, m)$) such that for every strategy of player 1 the expected empirical distribution of the action pairs (i, j) is essentially supported on the set of action pairs of the form $(i, r(i))$. A recent result of Neyman and Spencer characterizes the asymptotic relation of $m = m_k$ and k so that player 2 can effectively predict the moves of player 1. This asymptotic relation is: $\liminf \frac{\log m_k}{k}$, as k goes to infinity, is at least the minimum of $\log |I|$ and $\log |J|$. It follows that the value of $G(k, m_k)$ converges to $\max_{i\in I} \min_{j\in J} g(i, j)$ as $k \to \infty$ and $\liminf_{k\to\infty} \frac{\log m_k}{k} \geq \min(\log |I|, \log |J|)$.

An open problem (see [2]) is the quantification of the feasible "level of prediction" when the limit of $\frac{\log m_k}{k}$ equals θ and $0 < \theta < \min(\log |I|, \log |J|)$. For example, do the values of $G(k, m_k)$ converge as $k \to \infty$ and $\lim_{k\to\infty} \frac{\log m_k}{k} = \theta$, and, for those values of θ for which the limit exists, what is the limit of the values as a function of the stage game G and θ? It is known that the value of $G(k, m_k)$ converges, as $m_k \geq k \to \infty$ and $\frac{\log m_k}{k} \to 0$, to the value of the stage game [1].

The level of prediction, where player 1 is either (an uncertain periodic) nature or a player that does not observe the moves of player 2, has a complete asymptotic characterization [3]. The value of the two-person zero-sum repeated game, where

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 1–2, 2006.
© Springer-Verlag Berlin Heidelberg 2006

player 1's possible strategies are those implementable by oblivious automata of size k and player 2's possible strategies are those implementable by automata of size m, converges, as k goes to infinity and $\frac{\log m}{k}$ goes to $\theta \geq 0$, to a limit $v(\theta)$. The limit $v(\theta)$ is characterized by the data of the stage game $G = \langle I, J, g\rangle$. It equals the maxmin of $E_Q g(i,j)$ where the max is over all mixed stage actions p and the min is over all distributions Q on action pairs with marginal p on I, denoted Q_I, and $H(Q_I) + H(Q_J) - H(Q) \leq \theta$, where H is the entropy function. This result remains intact when player 2's possible strategies are those implementable by automata with time-dependent mixed actions and mixed transitions.

Another question is how long it takes the smarter player to effectively predict the moves of the other player. We study this question by analyzing the T-stage repeated game $G^T(k,m)$ where player 1's (respectively, player 2's) possible strategies are those implementable by an automaton with k (respectively, m) states. It is known that when player 2 is "supersmart" ($m = \infty$) and $T \gg k \log k$, player 2 can effectively predict the moves of player 1 [5]. Formally, the values of the two-person zero-sum games $G^{T_k}(k,\infty)$ converge to $\max_{i\in I} \min_{j\in J} g(i,j)$ as $k \to \infty$ and $\limsup_{k\to\infty} \frac{k \log k}{T_k} = 0$. It is conjectured in [2] that the values of the two-person zero-sum games $G^{T_k}(k,\infty)$ converge to the value of the stage game G as $k \to \infty$ and $\limsup_{k\to\infty} \frac{k \log k}{T_k} = \infty$.

References

1. Ben-Porath, E. (1993). Repeated games with finite automata, *Journal of Economic Theory*, **59**, 17–32.
2. Neyman, A. (1997). Cooperation, repetition, and automata, in *Cooperation: Game-Theoretic Approaches*, S. Hart, A. Mas Colell, eds., NATO ASI Series F, Vol. 155. Springer-Verlag, 233–255.
3. Neyman, A. (2006). The strategic value of memory (tentative title). Forthcoming.
4. Neyman, A. and J. Spencer (2006). The complexity threshold for effective prediction (tentative title). Forthcoming.
5. Neyman, A. and D. Okada (2000). Two-person repeated games with finite automata, *International Journal of Game Theory*, **29**, 309–325.

Dynamic Mechanism Design*

Davide Bilò[1], Luciano Gualà[1], and Guido Proietti[1,2]

[1] Dipartimento di Informatica, Università di L'Aquila, Italy
[2] Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy
{davide.bilo, guala, proietti}@di.univaq.it

Abstract. In this paper we address the question of designing truthful mechanisms for solving optimization problems on dynamic graphs. More precisely, we are given a graph G of n nodes, and we assume that each edge of G is owned by a selfish agent. The strategy of an agent consists in revealing to the system the cost for using its edge, but this cost is not constant and can change over time. Additionally, edges can enter into and exit from G. Among the various possible assumptions which can be made to model how these edge-cost modifications take place, we focus on two settings: (i) the *dynamic*, in which modifications are unpredictable and time-independent, and for a given optimization problem on G, the mechanism has to maintain efficiently the output specification and the payment scheme for the agents; (ii) the *time-sequenced*, in which modifications happens at fixed time steps, and the mechanism has to minimize an objective function which takes into consideration both the quality and the set-up cost of a new solution. In both settings, we investigate the existence of exact and approximate truthful mechanisms. In particular, for the dynamic setting, we analyze the *minimum spanning tree* problem, and we show that if edge costs can only decrease, then there exists an efficient dynamic truthful mechanism for handling a sequence of k edge-cost reductions having runtime $\mathcal{O}(h \log n + k \log^4 n)$, where h is the overall number of payment changes.

Keywords: Algorithmic Mechanism Design, On-line Problems, Dynamic Algorithms, Approximate Mechanisms.

1 Introduction

Algorithmic mechanism design (AMD) is concerned with the computational complexity of implementing, in a centralized fashion, truthful mechanisms for solving optimization problems in multi-agents systems [13]. AMD is by now one of the hottest topic in theoretical computer science, especially since of the game-theoretic nature of Internet. As a result, many classic network optimization problems have been resettled and solved under this new perspective [3,4,7,8,9].

* Work partially supported by the Research Project GRID.IT, funded by the Italian Ministry of Education, University and Research.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 3–15, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Apparently, however, the canonical approach is that of dealing with these problems by means of one-shot mechanisms, whose natural computational counterpart are static graph problems. This is in contrast with the intrinsic dynamicity of Internet's infrastructure, where links and node can rapidly appear, disappear, or even change their characteristics. Thus, surprisingly enough, there is a lack of modeling for those situations in which agents need to adapt their strategies over time, according to sudden changes in their owned components. To the best of our knowledge, the only effort towards this direction has been done in the framework of the so-called *on-line mechanism design* (OMD) [6,14]. There, the dynamic aspect resides in the fact that agents arrive and depart once over time, and their strategy consist of a *single* announcement of a bidding value for a time interval included between the arrival and the departing time. However, the limitation of OMD is that agents are not allowed to play different strategies over time, thus preventing to model those situations in which bidding values need to be continuously adjusted.

In this paper, we aim exactly to fill this gap, by exploring the difficulties and the potentialities emerging in this new challenging scenario. In doing that, we combine some of the theoretical achievements of the AMD with techniques which are proper of dynamic and on-line algorithms. The result of this activity is what we call as *dynamic mechanism design* (DMD). As a paradigmatic framework, we consider the situation in which each agent owns an edge of a given underlying graph G of n nodes, and its strategy consists in revealing to the system the cost (which can change over time) for using its edge. We focus on two main realistic scenarios:

1. In the first scenario, we consider the case in which edge costs are subject to sudden changes, due to boundary conditions alterations. In the extreme case, an edge might become unavailable to the system, due to a failure for instance. On the opposite side, some new edge might become available. All these variations are presented *on-line* to the system, which is completely unaware of possible future changes. Moreover, we will assume that each agent is unaware about other agents' types and strategies, and thus it cannot observe the global status of the system.[1] We feel that this is particularly attractive in an Internet setting, where an agent may not even know which other agents are participating to the mechanism. From an algorithmic point of view, this translates into a continuously evolving input graph, over which a solution to a given optimization problem has to be maintained. In other words, we need to design a *fully dynamic* mechanism which updates efficiently both the output specification and the corresponding payment scheme for the agents. In the rest of the paper, we will refer to this as the *dynamic* scenario. What is interesting here is that while classic dynamic graph algorithms can be used for the maintenance of the output specification, as far as the payment scheme updating is concerned, this defines

[1] Notice that the case in which an agent can observe the strategies of the other agents transforms our problem into a *repeated game*, for which the existence of a dominating strategy is unknown.

novel dynamic graph problems, which would make no sense in a canonical centralized framework. In this paper, as a starting point, we deal with a basic graph problem that has served as a case study for several papers on AMD, namely the *minimum spanning tree* (MST) problem. After observing that efficient dynamic MST algorithms in [10] can be turned into an efficient dynamic mechanism for handling a sequence of k edge-cost modifications having runtime $\mathcal{O}(k\, n \log^4 n)$, we will show that for the case in which edges can only become less expensive, then the mechanism runtime drops to $\mathcal{O}(h \log n + k \log^4 n)$, where h is the overall number of payment changes. We emphasize that this edge-cost lowering scenario is interesting because of the competitive nature of Internet.

2. In the second scenario, we consider the case in which the graph evolves in a sequence of time steps, and every agent has a specific cost for using its edge in each of these steps. Here, the time-depending modifications of the graph suggest that the mechanism's goal should now be the composition of two objectives: maintaining a *good* (not necessarily optimal) solution at a *low* (not necessarily minimal) cost of setting it up. Thus, on a sequence of graph changes, the objective function is now given by the overall cost of the sequence of solutions, plus the overall set-up cost. This approach is inspired to that proposed in the past in [11] to model the fact that on an on-line sequence of changes, it is important to take care of the modifications on the structure of the solution, since radical alterations might be too onerous in terms of set-up costs. In the rest of the paper, we will refer to this as the *time-sequenced* scenario. Here, on a positive side, we will show that: (i) if each set-up cost is upper bounded by the initial one and changes are presented *on-line* to the system, then a ρ-approximate monotone algorithm for a given optimization problem Π on G, translates into an approximate truthful mechanism for Π which on a sequence of graph changes of size k has an approximation ratio of $\max\{k, \rho\}$; (ii) if the underlying graph optimization problem is utilitarian and polynomial-time solvable, and changes are presented *off-line* to the system, then there exists a VCG-like truthful mechanism for solving optimally the sequence, which can be computed in polynomial time by means of a dynamic programming technique. On the other hand, on a negative side, we will show that even if graph changes are presented off-line to the system and set-up costs are uniform, then any truthful mechanism which solves the problem by means of a *divide et impera* paradigm (as explained in more detail in Section 6) cannot achieve a better than k approximation ratio.

The paper is organized as follows: in Section 2 we give preliminary definitions; after, in Section 3 we present the dynamic mechanism for the MST problem, while in Section 4 we define formally the time-sequenced model; finally, in the last two sections we give, respectively, positive and negative results on the existence of time-sequenced truthful mechanisms.

2 Preliminaries

Let a communication network be modeled by a graph $G = (V, E)$ with n nodes and m edges. We will deal with the case in which each edge $e \in E$ is controlled by a selfish agent a_e holding a private information t_e, namely the *true type* of a_e. Only agent a_e knows t_e. Each agent has to declare a public *bid* b_e to the mechanism. We will denote by t the vector of types, and by b the vector of bids.

For a given optimization problem Π defined on G, let $\textsc{Sol}(\Pi)$ denote the corresponding set of feasible solutions. We will assume that $\textsc{Sol}(\Pi)$ does not depend on the agents' types. For each $x \in \textsc{Sol}(\Pi)$, an objective function is defined, which depends on the agents' types. A *mechanism* for Π is a pair $\mathcal{M} = \langle g(b), p(b) \rangle$, where $g(b)$ is an algorithm that, given the agents' bids, computes a solution for Π, and $p(b)$ is a scheme which describes the payments provided to the agents. For each solution x, a_e incurs a cost $\nu_e(t_e, x)$ for participating to x (also called *valuation of* a_e *w.r.t.* x). Each agent tries to maximize its *utility*, which is defined as the difference between the payment provided by the mechanism and the cost incurred by the agent w.r.t. the computed solution. On the other hand, the mechanism aims to compute a solution which minimizes the objective function of Π w.r.t. to the agents' types, but of course it does not know t directly. In a *truthful* mechanism this tension between the agents and the system is resolved, since each agent maximizes its utility when it declares its type, regardless of what the other agents do.

Given a positive real function $\varepsilon(n)$ of the input size n, an $\varepsilon(n)$-*approximate* mechanism returns a solution whose measure comes within a factor $\varepsilon(n)$ from the optimum. A mechanism has a runtime of $\mathcal{O}(f(n))$ if $g(\cdot)$ and $p(\cdot)$ are computable in $\mathcal{O}(f(n))$ time. Moreover, a mechanism design problem is called *utilitarian* if the objective function of Π is equal to $\sum_{e \in E} \nu(t_e, x)$. For utilitarian problems, there exists a well-known class of truthful mechanisms, i.e., the *Vickrey-Clarke-Groves (VCG) mechanisms.*

In [2], Archer and Tardos have shown how to design truthful mechanisms for another well-known class of mechanism design problems called *one-parameter.* A problem is said one-parameter if (i) the true type of each agent a_e can be expressed as a single parameter $t_e \in \mathbb{R}$, and (ii) each agent's valuation has the form $\nu_e(t_e, x) = t_e\,\omega_e(b)$, where $\omega_e(b)$ is called the *work curve* for agent a_e, i.e., the amount of work for a_e depending on the output specified by the mechanism algorithm, which in its turn is a function of the bid vector b. When, for each agent a_e, $\omega_e(b)$ can be either 0 or 1, then the problem is also called *binary demand* [12]. In [2] it is shown that for one-parameter problems, a sufficient condition for truthfulness is given by a monotonicity property of the mechanism algorithm. In particular, for a binary demand problem, such property reduces to the following. Let b be the vector of bids of the agents, and let b_{-e} denote the vector of all bids besides b_e; the pair (b_{-e}, b_e) will denote the vector b. If we fix b_{-e}, a *monotone* algorithm $\mathcal{A}$ defines a threshold value $\theta_e(b_{-e}, \mathcal{A})$ such that if a_e bids no more than $\theta_e(b_{-e}, \mathcal{A})$, then e will be selected, while if a_e bids above $\theta_e(b_{-e}, \mathcal{A})$, e will not be selected.[2] Sometimes, we will write $\theta_e(b_{-e})$ when the algorithm $\mathcal{A}$ is clear from the context. The results in [2] imply that the only truthful mechanism for a binary demand problem using an algorithm $\mathcal{A}$ is the one-parameter mechanism $\mathcal{M} = \langle \mathcal{A}, p^{\mathcal{A}}(\cdot)\rangle$, where $\mathcal{A}$ is required to be monotone, and the payment $p_e^{\mathcal{A}}(b)$ for each agent a_e is defined as its threshold value if it owns a selected edge, and 0 otherwise.

3 An Efficient Dynamic Mechanism for the MST Problem

We start by addressing the problem of designing an efficient mechanism for the fully dynamic MST problem. Since we assume that agents' types change over time, we allow the agents to declare a new bid to the mechanism at any time. Recall that edge-cost changes are presented on-line to the system, which is unaware of possible future changes, and that the agents do not know other agents' bids. The mechanism works as follows. At any time, whenever it receives a new bid from an agent, it computes a new MST w.r.t. the new bid profile, and it updates the payments exactly as the one-parameter mechanism for the MST problem. Concerning the truthfulness of the mechanism, this follows from the truthfulness of the one-parameter mechanism for the MST problem, and from the fact that every agent is completely unaware of other agents' bids.

On the other hand, concerning the time complexity, the mechanism has to maintain: (i) an MST of G, and (ii) the corresponding payments. Moreover, it has to support a payment query in $\mathcal{O}(1)$ time. To dynamically maintain an MST, one can use the algorithm proposed in [10], which takes $\mathcal{O}(k \log^4 n)$ time for processing k edge-cost updates (deletions of edges are simulated by setting to $+\infty$ the cost of an edge). Thus, it remains to manage the payment scheme. We remind that for binary demand problems, the payment provided to a_e is equal to $\theta_e(b_{-e})$ if e is selected, and zero otherwise. This means it suffices to maintain the threshold value $\theta_e(b_{-e})$ for each e belonging to the current solution. We emphasize that the algorithm in [10] can be straightforwardly used to accomplish such a task, and from this it follows that there exists a truthful mechanism for the fully dynamic MST which runs in $\mathcal{O}(k\, n \log^4 n)$ time for processing k updates. Improving this latter result is a challenging open problem. In the following, we show that for the *edge-cost decreasing* case, in which edge costs are only allowed to decrease, a significant improvement is possible. We argue this is not a very special case, as it includes the well-known *partially dynamic* scenario, where only edge insertions are allowed.

How to Maintain the Payments. Let G be a graph, and let T be an MST of G. For each *non-tree edge* $f = (u, v) \in E \setminus E(T)$, $T(f)$ will denote the set of tree edges belonging to the (unique) path in T between u and v. For each $e \in E(T)$,

[2] As usual, we will assume that there always exists a feasible solution not containing e, which implies that $\theta_e(b_{-e}, \mathcal{A})$ is bounded.

let $C_T(e) = \{f \in E \setminus E(T) \mid e \in T(f)\}$. We denote by *swap*$(e)$ the cheapest non-tree edge in $C_T(e)$.[3] Note that $\theta_e(b_{-e}) = b_{swap(e)}$.

Clearly, if a tree edge decreases its cost, no payment changes. Consider now the situation in which a non-tree edge f decreases its cost from b_f to b'_f. Denote by T' the new MST, i.e., the MST computed w.r.t. the cost profile $b' = (b_{-f}, b'_f)$. We have two cases:

Case 1: $T' = T$. Clearly, only the threshold of edges in $T(f)$ may change, since for each $e' \notin T(f)$, no edge in $C_T(e')$ has changed its cost. Moreover, the threshold of e changes iff $\theta_e(b_{-e}) > b'_f$, and in this case the new threshold value becomes $\theta_e(b'_{-e}) = b'_f$.

Case 2: $T' \neq T$. Clearly $T' = T \setminus \{e\} \cup \{f\}$. Moreover, the payment for a_e becomes 0, while that for a_f will be $\theta_f(b'_{-f}) = b_e$, since $C_{T'}(f) \subseteq C_T(e) \cup \{e\}$.

Lemma 1. *For every $e' \in E(T') \setminus T'(e)$, $\theta_{e'}(b'_{-e'}) = \theta_{e'}(b_{-e'})$.*

Proof. The lemma trivially follows from the fact that for each $e' \in E(T') \setminus T'(e)$, $C_{T'}(e') = C_T(e')$ and $f \notin C_T(e')$. □

Lemma 2. *The threshold of an edge $e' \in T'(e)$ changes iff $\theta_{e'}(b_{-e'}) > b_e$. In this case, $\theta_{e'}(b'_{-e'}) = b_e$.*

Proof. Let $e' \in T'(e)$ be such that $\theta_{e'}(b_{-e'}) > b_e$. Since $e \in C_{T'}(e')$, then $\theta_{e'}(b'_{-e'}) \leq b_e$. We have to show that $\nexists f' \in C_{T'}(e')$ with $b_{f'} < b_e$. For the sake of contradiction, suppose that $\exists f' \in C_{T'}(e')$ such that $b_{f'} < b_e$. Then, we show T was not an MST by proving that $f' \in C_T(e)$. Suppose that $f' \notin C_T(e)$; then $T(f') = T'(f')$, which implies $\theta_{e'}(b_{-e'}) < b_e$.

It remains to show that if $\theta_{e'}(b_{-e'}) \leq b_e$, then $\theta_{e'}(b'_{-e'}) = \theta_{e'}(b_{-e'})$. Notice that if *swap*$(e') \in C_T(e)$, then $\theta_{e'}(b_{-e'}) \geq b_e$ from the minimality of T, which implies $\theta_{e'}(b_{-e'}) = b_e$. Otherwise, *swap*$(e') \in C_{T'}(e')$. In both cases $\theta_{e'}(b'_{-e'}) \leq \theta_{e'}(b_{-e'})$. Moreover, since $C_{T'}(e') \subseteq C_T(e') \cup C_T(e) \cup \{e\}$, then

$$\begin{aligned}\theta_{e'}(b'_{-e'}) &= \min_{f \in C_{T'}(e')}\{b_f\} \geq \min_{f \in C_T(e') \cup C_T(e) \cup \{e\}}\{b_f\} \\ &= \min\{b_{swap(e')}, b_e\} = \theta_{e'}(b_{-e'}).\end{aligned}$$
□

Implementation. To update the payments, we use a *top tree*, a data structure introduced by Alstrup *et al.* [1] to maintain information about paths in trees. More precisely, a top tree represents an edge-weighted forest $\mathcal{F}$ with weight function $c(\cdot)$. Some operations defined for top trees are:

- `link`$((u, v), x)$, where u and v are in different trees. It links these trees by adding the edge (u, v) of weight $c(u, v) = x$ to $\mathcal{F}$.
- `cut`(e). It removes the edge e from $\mathcal{F}$.
- `update`(e, x), where e belongs to $\mathcal{F}$. It sets the weight of e to x.
- `max`(u, v), where u and v are connected in $\mathcal{F}$. It returns the edge with maximum weight among the edges on the path between u and v in $\mathcal{F}$.

[3] If there are more than one such cheapest edges, we pick one of them arbitrarily.

In [1,5,15], it is shown how to implement a top tree (by using $\mathcal{O}(n)$ space) for supporting each of the above operations in $\mathcal{O}(\log n)$ time.

To our scopes, we use a top tree $\mathcal{T}$ as follows. $\mathcal{T}$ maintains the current MST where the cost of each edge $e \in E(T)$ is $\theta_e(b_{-e})$. Concerning Case 1, we only need to update the threshold of some edges in $T(f)$. So, let $f = (x, y)$ be the edge which has decreased its cost. While $c(e') > b'_f$, where $e' = \texttt{max}(x, y)$, then we (i) update the payment for $a_{e'}$ to b'_f, and (ii) perform $\texttt{update}(e', b'_f)$. For what concerns Case 2, let $e = (x, y)$ be the edge in T not in T'. First, we update the MST by performing $\texttt{cut}(e)$ and $\texttt{link}(f, b_e)$. Next, we update the payment for a_e (resp., a_f) to 0 (resp., b_e). Finally, while $c(e') > b_e$, where $e' = \texttt{max}(x, y)$, then (i) we update the payment for $a_{e'}$ to b_e, and (ii) we perform $\texttt{update}(e', b_e)$.

The above discussion yields the following:

Theorem 1. *There exists a dynamic mechanism supporting a sequence of k edge-cost decreasing operations in $\mathcal{O}(h \log n + k \log^4 n)$ time, where h is the overall number of payment changes.* □

4 Time-Sequenced Scenario: Problem Statement

Let $G = (V, E)$ be a graph with a positive real weight $w(e)$ associated with each edge $e \in E$. Henceforth, unless stated otherwise, by Π we will denote a *communication network* problem on (G, w), which asks for computing a subgraph $H \in \textsc{Sol}(\Pi)$ of G by minimizing an objective function $\phi(H, w)$ of the form

$$\phi(H, w) = \sum_{e \in E(H)} w(e) \cdot \mu_H(e),$$

where $\mu_H(e)$ depends only on the topology of H. Notice that this definition embraces the quasi-totality of communication network problems, like the MST problem, the *shortest-paths tree* problem, and so on.

Let k be a positive integer. We assume that the type of each agent a_e is $t_e = \langle t_e^1, \ldots, t_e^k \rangle$, while its bid is $b_e = \langle b_e^1, \ldots, b_e^k \rangle$. Intuitively, t_e^i represents the true cost incurred by a_e for using its link e at time i. We will denote by $t^i \in \mathbb{R}^m$ the vector of agents' types at time i, and by t the vector $\langle t^1, \ldots, t^k \rangle$.

Given a communication network problem Π, we want to design a truthful mechanism for the optimization problem that we will denote by $\textsc{Seq}(\Pi)$. This latter problem asks for computing a sequence $\mathcal{H} = \langle H_1, \ldots, H_k \rangle$, where $H_i \in \textsc{Sol}(\Pi)$, $i = 1, \ldots, k$, by minimizing the following objective

$$\Psi(\mathcal{H}, t) = \Phi(\mathcal{H}, t) + \Gamma(\mathcal{H}),$$

where $\Phi(\mathcal{H}, t)$ is a function measuring the quality of the solution $\mathcal{H}$, and $\Gamma(\mathcal{H})$ is a function measuring the overall set-up cost. For a given sequence $\mathcal{H}$, we will naturally assume that the valuation of a_e w.r.t. $\mathcal{H}$ is:

$$\nu_e(\mathcal{H}, t_e) = \sum_{i=1}^{k} \nu_e^i(H_i, t_e^i), \quad \text{where} \quad \nu_e^i(H_i, t_e^i) = \begin{cases} t_e^i & \text{if } e \in E(H_i); \\ 0 & \text{otherwise.} \end{cases}$$

Depending on the cost model to be adopted, the functions $\Phi(\cdot)$ and $\Gamma(\cdot)$ can be defined accordingly. In this paper, we will consider the prominent *additive* cost model, in which

$$\Phi(\mathcal{H}, t) = \sum_{i=1}^{k} \phi(H_i, t^i), \qquad \Gamma(\mathcal{H}) = \sum_{i=1}^{k} \gamma(i, \mathcal{H}),$$

where

$$\gamma(i, \mathcal{H}) = \begin{cases} \gamma_1 \in \mathbb{R}^+ & \text{if } i = 1; \\ \gamma_i \in \mathbb{R}^+ & \text{if } H_i \neq H_{i-1},\ i = 1, \ldots, k; \\ 0 & \text{otherwise.} \end{cases}$$

For any $1 \leq i \leq j \leq k$, by $[i, j]$ we will denote the *interval* $\{i, \ldots, j\}$. We will write $[i, j)$ instead of $[i, j-1]$. Given two intervals $[i, j], [i', j']$, we write $[i, j] \prec [i', j']$ if $j < i'$. An *interval vector* $s = \langle I_1, \ldots, I_h \rangle$ is a vector of pairwise disjoint intervals whose union is $\{1, \ldots, k\}$, and such that $I_1 \prec \cdots \prec I_h$. Given an interval I, let b^I be the vector defined as $b^I_e = \sum_{i \in I} b^i_e$, for each edge $e \in E$. Moreover, we will denote by H^*_I an optimum solution for Π when the input is (G, b^I). Finally, given two sequences $\mathcal{H} = \langle H_1, \ldots, H_i \rangle, \mathcal{H}' = \langle H'_1, \ldots, H'_j \rangle$, by $\mathcal{H} \odot \mathcal{H}'$ we denote the sequence $\langle H_1, \ldots, H_i, H'_1, \ldots, H'_j \rangle$.

5 Time-Sequenced Mechanisms: Positive Results

In this section we first define the class of *time-sequenced single-parameter (TSSP) mechanisms*, and we prove that any mechanism in this class is truthful for $\textsc{Seq}(\Pi)$. Moreover, for the case in which each set-up cost is upper bounded by γ_1, we show that there exists an on-line $\max\{k, \rho\}$-approximate TSSP mechanism, where ρ is the approximation ratio of a monotone algorithm for Π. Then, we turn our attention to the special case in which Π is utilitarian and polynomial-time solvable, and we show that if the graph changes are presented off-line to the system, then there exists a VCG-like truthful mechanism for solving optimally $\textsc{Seq}(\Pi)$, which can be computed in polynomial time by means of a dynamic programming technique.

5.1 On-Line Sequences with Bounded Set-Up Costs

From now on, by $\tilde{s}$ we will denote the interval vector $\langle [1, 1], \ldots, [k, k] \rangle$.

Definition 1. *Given a communication network problem Π, and a monotone algorithm $\mathcal{A}$ for Π, a* TSSP mechanism $\mathcal{M}(s) = \langle g_s(b), p(b) \rangle$ with interval vector $s = \langle I_1, \ldots, I_h \rangle$ *for* $\textsc{Seq}(\Pi)$ *is defined as follows:*

1. *$g_s(\cdot)$ returns a sequence $\mathcal{H} = \mathcal{H}_1 \odot \cdots \odot \mathcal{H}_h$, in which*

$$\forall j = 1, \ldots, h, \quad \mathcal{H}_j = \langle \hat{H}_j, \ldots, \hat{H}_j \rangle \quad \textit{has size } |I_j|,$$

where $\hat{H}_j$ is the solution returned by $\mathcal{A}$ with input (G, b^{I_j});

2. *For each agent* a_e

$$p_e(b) = \sum_{j=1}^{h} p_e^{\mathcal{A}}\big(b^{I_j}\big),$$

where $p_e^{\mathcal{A}}(b^{I_j})$ *is the payment provided to* a_e *by the one-parameter mechanism* $\langle \mathcal{A}, p^{\mathcal{A}}(\cdot)\rangle$ *for the problem* Π *when the input is* (G, b^{I_j}).

Notice that, by definition, $\mathcal{M}(\tilde{s})$ is the only on-line TSSP mechanism.

Proposition 1. $\mathcal{M}(s)$ *is a truthful mechanism for* $\textsc{Seq}(\Pi)$.

Proof. The mechanism breaks the problem in h instances $(G, b^{I_1}), \ldots, (G, b^{I_h})$ which are independent each other. Then it uses the one-parameter mechanism $\langle \mathcal{A}, p^{\mathcal{A}}(\cdot)\rangle$ for each of them in order to locally guarantee the truthfulness. □

The main result of this section, whose proof is omitted due to lack of space, is the following:

Theorem 2. *Given a* ρ*-approximate monotone algorithm* $\mathcal{A}$ *for* Π*, the mechanism* $\mathcal{M}(\tilde{s})$ *applied to* $\textsc{Seq}(\Pi)$ *with the assumption that each set-up cost is upper bounded by* γ_1*, has a performance guarantee of* $\max\{k, \rho\}$. □

5.2 Off-Line Utilitarian Problems

In this section we show how to design an exact off-line mechanism when Π is utilitarian. Before defining our mechanism, we show how to compute an optimal sequence by using dynamic programming.

Let $\mathcal{H}^*$ denote an optimal solution for $\textsc{Seq}(\Pi)$, and let $\mathcal{H}^*_{[1,i]}$ be an optimal solution for $\textsc{Seq}(\Pi)$ when the input is restricted to the interval $[1, i]$, i.e. we have i time steps and the bid vector is $\langle b^1, \ldots, b^i\rangle$. In order to lighten the notation, we will write $\Psi(\mathcal{H}_{[1,i]}, b)$ instead of $\Psi(\mathcal{H}_{[1,i]}, \langle b^1, \ldots, b^i\rangle)$, where $\mathcal{H}_{[1,i]}$ is a solution for $\textsc{Seq}(\Pi)$ restricted to the interval $[1, i]$. Notice that $\mathcal{H}^*_{[1,1]} = \langle H^*_{[1,1]}\rangle$, and $\Psi(\mathcal{H}^*_{[1,1]}, b) = \phi(H^*_{[1,1]}, b^1) + \gamma_1$. Moreover, $\mathcal{H}^*_{[1,k]} = \mathcal{H}^*$.

The dynamic programming algorithm computes $H^*_{[i,j]}$, for every $1 \le i \le j \le k$. Next, starting from $i = 1$ to k, it computes $\mathcal{H}_{[1,i]} = \mathcal{H}_{[1,h_i)} \odot \langle H^*_{[h_i,i]}\rangle$, with

$$h_i = \arg\min_{h=1,\ldots,i} \left\{ \Psi'(b, h, i) := \Psi(\mathcal{H}_{[1,h)}, b) + \phi\big(H^*_{[h,i]}, b^{[h,i]}\big) + \gamma_h \right\},$$

where $\mathcal{H}_{[1,1)}$ is the empty sequence, and $\Psi(\mathcal{H}_{[1,1)}, b)$ is assumed to be 0.

The following lemma, whose proof is omitted due to lack of space, holds:

Lemma 3. *For any* $i = 1, \ldots, k$*, the dynamic programming algorithm computes a solution* $\mathcal{H}_{[1,i]}$ *such that* $\Psi(\mathcal{H}_{[1,i]}, b) = \Psi(\mathcal{H}^*_{[1,i]}, b)$. □

We are now ready to define our VCG-like mechanism. Let $\mathcal{M}_{\text{VCG}}$ be a mechanism defined as follows:

1. The algorithmic output specification selects an optimal sequence (w.r.t. the bids b) $\mathcal{H}_G^*$;
2. Let $G-e = (V, E\backslash\{e\})$, and let $\mathcal{H}_{G-e}^*$ be an optimal solution (w.r.t. the bids b) in $G-e$. Then, the payment function for a_e is defined as

$$p_e(b) = \Psi(\mathcal{H}_{G-e}^*, b) - \Psi(\mathcal{H}_G^*, b) + \nu_e(\mathcal{H}_G^*, b_e).$$

From the above discussion, it is easy to prove the following

Theorem 3. *Let Π be utilitarian and solvable in polynomial time. Then, $\mathcal{M}_{\text{VCG}}$ is an exact off-line truthful mechanism for* Seq(Π) *which can be computed in polynomial time.* □

6 Time-Sequenced Mechanisms: Inapproximability Results

In this section we consider a natural extension of TSSP mechanisms named *adaptive TSSP mechanisms*, and we prove a lower bound of k to the approximation ratio that can be achieved by any truthful mechanism in this class.

Definition 2. *Let δ be a function mapping bid vectors to interval vectors. An* adaptive time-sequenced single-parameter (ATSSP) mechanism *$\mathcal{M}_\delta$ for Π is the mechanism which, for a given vector bid b, is defined exactly as $\mathcal{M}(\delta(b))$.*

Lemma 4. *Let t^i be a type profile for Π, and let $\mathcal{A}$ be an optimal algorithm for Π. Then, $\forall \eta \in \mathbb{R}^+$, $\theta_e^i(\eta \cdot t_{-e}^i) = \eta \cdot \theta_e^i(t_{-e}^i)$.*

Proof. Observe that $\forall H \in$ Sol(Π)

$$\phi(H, \eta \cdot t^i) = \sum_{e \in E(H)} \eta \cdot t_e^i \, \mu_H(e) = \eta \sum_{e \in E(H)} t_e^i \, \mu_H(e) = \eta \cdot \phi(H, t^i).$$ □

Theorem 4. *For any mapping function δ, for any optimal algorithm $\mathcal{A}$ for Π, and for any $c < k$, there exists no c-approximate truthful ATSSP mechanism using $\mathcal{A}$ for* Seq(Π)*, even when set-up costs are uniform.*

Proof. The proof is by contradiction. Let $M = \gamma_1 = \cdots = \gamma_k$. Let $\mathcal{M}_\delta$ be a c-approximate truthful ATSSP mechanism for Seq(Π). For the sake of clarity, we denote by $H(w)$ an optimum solution for Π with input (G, w). Let $t^1 = (t_{-e}^1, t_e^1)$, with $t_{-e}^1 = \langle 0, \ldots, 0 \rangle$, and $t^2 = (t_{-e}^2, 0)$ be two type vectors for Π such that the following three conditions hold:

(i) $2t_e^1 < \theta_e^2,\ t_e^1 > 0$, where $\theta_e^2 = \theta_e(t_{-e}^2)$;
(ii) $\phi(H(t_{-e}^2, +\infty), (t_{-e}^2, +\infty)) \geq (k^2 - 1)M$;
(iii) $\phi(H(t_{-e}^2, x), (t_{-e}^2, x))$ does not depend on M, for any $x < \theta_e^2$ not depending on M.

Lemma 5. *There always exist t_e^1 and t_{-e}^2 satisfying the above conditions.*

Proof. Let $H \in \textsc{Sol}(\varPi)$ be such that $E(H') \not\subset E(H), \forall H' \in \textsc{Sol}(\varPi)$. Let e be an edge of H. Now for each $e' \in E(H) \setminus \{e\}$, let $t^2_{e'} = \frac{1}{\mu_H(e')}$. Moreover, for each $e' \in E \setminus E(H)$, let $t^2_{e'}$ be defined as follows

$$t^2_{e'} = \max_{H' \in \textsc{Sol}(\varPi)} \frac{(k^2-1)M}{\mu_{H'}(e')}.$$

By construction, condition (ii) holds. For M large enough, it is easy to see that θ^2_e is at least $(k^2-1)M - |E(H)| > 0$, from which (i) follows as well. Finally, condition (iii) follows by observing that $\mu_H(e)$ does not depend on M. □

Let t be the type profile defined as follows:

$$\forall i = 1, \dots, k, \quad t^i = \begin{cases} t^1 & \text{if } i \text{ is odd;} \\ t^2 & \text{otherwise.} \end{cases}$$

Lemma 6. *For M large enough, $\delta(t) \neq \tilde{s}$.*

Proof. The proof is by contradiction. Let $\mathcal{H}$ be the solution computed by the mechanism corresponding to the interval vector $\tilde{s}$. Notice that $\varPsi(\mathcal{H}, t) \geq kM$, since $H(t^1) \neq H(t^2)$. Consider now the solution $\mathcal{H}'$ corresponding to the interval vector $\langle [1,k] \rangle$. It is easy to see that for t^1_e small enough, $\varPsi(\mathcal{H}', t) = M + \phi\big(H\big(t^{[1,k]}\big), t^{[1,k]}\big) \leq M + k\,\phi(H(t^2_{-e}, t^1_e), (t^2_{-e}, t^1_e))$. It follows that the approximation ratio achieved by the mechanism is at least

$$\frac{\varPsi(\mathcal{H}, t)}{\varPsi(\mathcal{H}', t)} \geq \frac{kM}{M + k\,\phi(H(t^2_{-e}, t^1_e), (t^2_{-e}, t^1_e))},$$

which, from (iii), goes to k when M goes to $+\infty$. This contradicts the fact that $\mathcal{M}_\delta$ is c-approximate. □

Lemma 7. *For M large enough, the utility of a_e in the solution $g_{\delta(t)}(t)$ computed by the mechanism $\mathcal{M}_\delta$ is less than $\left\lfloor \frac{k}{2} \right\rfloor \theta^2_e$.*

Proof. Let $\delta(t) = \langle I_1, \dots, I_h \rangle$ be the interval vector computed by δ, and let $\mathcal{H}$ be the corresponding solution. For each $j = 1, \dots, h$, let $I_j = [x_j, y_j]$ be the j-th interval, and let η_j be the number of occurrences of t^2 in $\langle t^{x_j}, \dots, t^{y_j} \rangle$. Notice that $t^{I_j} = (\eta_j\, t^2_{-e}, (|I_j| - \eta_j)\, t^1_e)$. It is easy too see that $(|I_j| - \eta_j) \leq \eta_j + 1$. Moreover, notice that e belongs to $H\big(t^{I_j}\big)$ iff $\eta_j > 0$. Indeed, whenever $\eta_j > 0$, $(\eta_j + 1)\, t^1_e < \eta_j\, \theta^2_e$ holds from (i), and from Lemma 4 this implies that e belongs to $H\big(t^{I_j}\big)$. Finally, notice that whenever $|I_j| > 1$, a_e incurs a cost of at least t^1_e.

Then, from Lemma 4, the payment provided to a_e is $\sum_{j=1}^{h} \eta_j\, \theta^2_e = \left\lfloor \frac{k}{2} \right\rfloor \theta^2_e$, while concerning the cost incurred by a_e, it is at least $t^1_e > 0$, since from Lemma 6 there must exist an index j^* such that $|I_{j^*}| > 1$. □

Consider now the following new type profile $\hat{t}$ which is equal to t except for $\hat{t}^i_e$ that is set to $+\infty$ for every odd i.

Lemma 8. *For M large enough, $\delta(\hat{t}) = \tilde{s}$.*

Proof. For the sake of contradiction, assume that $\delta(\hat{t}) \neq \tilde{s}$. Then, there must exist an index j for which the solution $\mathcal{H}$ computed by the mechanism does not change at time j. Hence, since either $\hat{t}_e^j = +\infty$ or $\hat{t}_e^{j-1} = +\infty$, from (ii) it must be $\Psi(\mathcal{H}, \hat{t}) \geq k^2 M$. Consider the solution $\mathcal{H}'$ corresponding to the interval vector $\tilde{s}$. Then, the approximation ratio achieved by the mechanism is at least

$$\frac{\Psi(\mathcal{H}, \hat{t})}{\Psi(\mathcal{H}', \hat{t})} \geq \frac{k^2 M}{kM + k\,\phi(H(t^2), t^2)},$$

which, from (iii), goes to k when M goes to $+\infty$. This contradicts the fact that $\mathcal{M}_\delta$ is c-approximate. □

To conclude the proof, observe that when the type profile is t, a_e has convenience to bid b_e defined as

$$\forall i = 1, \ldots, k, \quad b_e^i = \begin{cases} t_e^2 & \text{if } i \text{ is even;} \\ +\infty & \text{otherwise.} \end{cases}$$

Indeed, in this case, from Lemma 8, its utility becomes equal to $\left\lfloor \frac{k}{2} \right\rfloor \theta_e^2$, which is better than the utility it gets by bidding truthfully (see Lemma 7). □

Notice that, since in the uniform set-up cost case each set-up cost is upper bounded by γ_1, and since $\mathcal{M}(\tilde{s})$ belongs to the class ATSSP, then Theorem 4 implies that the upper bound in Theorem 2 is tight, when $\mathcal{A}$ is optimal (i.e., $\rho = 1$).

References

1. S. Alstrup, J. Holm, K. de Lichtenberg, and M. Thorup, Minimizing diameters of dynamic trees, *Proc. 24th International Colloquium on Automata Languages and Programming (ICALP'97)*, Vol. 1256 of LNCS, Springer-Verlag, 270–280.
2. A. Archer and É. Tardos, Truthful mechanisms for one-parameter agents, *Proc. 42nd IEEE Symp. on Foundations of Computer Science (FOCS'01)*, 482–491, 2001.
3. D. Bilò, L. Gualà, and G. Proietti, On the existence of truthful mechanisms for the minimum-cost approximate shortest-paths tree problem, *13th Coll. on Structural Information and Communication Complexity (SIROCCO'06)*, Vol. 4056 of LNCS, Springer-Verlag, 295-309.
4. P. Briest, P. Krysta, and B. Vöcking, Approximation techniques for utilitarian mechanism design, *Proc. 37th Ann. ACM Symp. on Theory of Computing (STOC'05)*, 39–48, 2005.
5. G.N. Frederickson, Data structures for on-line updating of minimum spanning trees, with applications, *SIAM J. Comput.*, 14(4):781–798, 1985.
6. E.J. Friedman and D.C. Parkes, Pricing WiFi at Starbucks: issues in online mechanism design, *4th ACM Conf. on Electronic Commerce (EC'03)*, 240–241, 2003.
7. L. Gualà and G. Proietti, A truthful (2-2/k)-approximation mechanism for the Steiner tree problem with k terminals, *Proc. 11th Int. Computing and Combinatorics Conference (COCOON'05)*, Vol. 3595 of LNCS, Springer-Verlag, 390–400.

8. L. Gualà and G. Proietti, Efficient truthful mechanisms for the single-source shortest paths tree problem, *Proc. 11th Int. Euro-Par Conf. (Euro-Par'05)*, Vol. 3648 of LNCS, Springer-Verlag, 941-951.
9. J. Hershberger and S. Suri, Vickrey prices and shortest paths: what is an edge worth?, *Proc. 42nd IEEE Symp. on Foundations of Computer Science (FOCS'01)*, 252–259.
10. J. Holm, K. de Lichtenberg, and M. Thorup, Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity, *J. of the ACM*, 48(4):723–760, 2001.
11. B. Kalyanasundaram and K. Pruhs, On-line network optimization problems, in *Online Algorithms: The State of the Art*, Vol. 1442 of LNCS, Springer-Verlag, 268–280, 1998.
12. M.-Y. Kao, X.-Y. Li, and W. Wang, Towards truthful mechanisms for binary demand games: a general framework, *Proc. 6th ACM Conf. on Electronic Commerce 2005*, 213–222, 2005.
13. N. Nisan and A. Ronen, Algorithmic mechanism design, *Games and Economic Behaviour* 35:166–196, 2001.
14. D.C. Parkes, S.P. Singh, and D. Yanovsky, Approximately efficient online mechanism design, *Adv. in Neural Information Processing Syst. 17 (NIPS'04)*, 2004.
15. D.D. Sleator and R.E. Tarjan, A data structure for dynamic trees. *J. Comput. Syst. Sci.*, 26(3):362–391, 1983.

Unconditional Competitive Auctions with Copy and Budget Constraints*

Tian-Ming Bu[1,**], Qi Qi[2], and Aries Wei Sun[2]

[1] Department of Computer Science and Engineering
Intelligent Information Processing Laboratory
Fudan University
Shanghai, 200433, P.R. China
031021063@fudan.edu.cn

[2] Department of Computer Science
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong SAR
qi.qi@student.cityu.edu.hk, sunwei@cs.cityu.edu.hk

Abstract. This paper investigates a new auction model in which bidders have both copy and budget constraints. The new model has extensive and interesting applications in auctions of online ad-words, software licenses, etc. We consider the following problem: Suppose all the participators are rational, how to allocate the objects at what price so as to guarantee auctioneer's high revenue, and how high it is.

We introduce a new kind of mechanisms called *win-win mechanisms* and present the notion of *unconditional competitive auctions*. A notably interesting property of *win-win mechanisms* is that each bidder's self-interested strategy brings better utility not only to himself but also to the auctioneer. Then we present *win-win mechanisms* for multi-unit auctions with copy and budget constraints. We prove that these auctions are *unconditional competitive* under the situation of both limited and unlimited supply.

1 Introduction

In recent years, great progresses have been made in electronic commerce, especially in internet auctions, to which various theoretical and practical studies have been conducted. Besides governments use auctions to sell rights and assets, such as Federal Communication Commission, a lot of companies also use internet auctions to conduct business. There even exist some companies whose revenue depends almost on certain types of auctions. Over 98% of Google's revenue and 50% of Yahoo!'s revenue are derived from sales via keywords advertising auctions [7].

* Research is supported by a research grant of City University of Hong Kong (Proj. No. 7001838).

** This work was done while the author was at City University of Hong Kong.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 16–26, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we study a quite general yet very practical type of auctions, where both budget and copy constraints are present. In our model, a single seller sells multiple copies of a single kind of digital goods. During the process, each buyer reports one *private* unit value he is willing to pay for the item, one *private* number of copies he demands and one *private* budget that he is able to pay. We investigate the model from the perspective of the seller, and our aim is to present some auctions, whose performance can be theoretically guaranteed, that maximize the seller's revenue, with the basic assumption that buyers are rational and want to maximize their preferences.

Different from traditional models which shield complex factors by making certain assumptions, there are some additional considerations that distinguish our model in terms of real life applicability:

1. Our model is especially suitable for digital goods, which can produce unlimited copies with marginal cost zero, such as license sales, mp3 copies, online advertisements, etc.
2. We consider both copy and budget constraints. We argue that under most realistic circumstances, the demand of a buyer is limited. Redundant allocation will bring not profit, but resource waste. For example, issue 1000 copies of software license to a company with only 100 computers is no doubt an undesirable allocation result.

Besides studying the new auction model, this paper also has the following contributions:

– We introduce a new kind of mechanisms called *win-win mechanisms*. As we know, in two-sided markets the famous VCG mechanisms maximize the buyers' utilities and contrarily minimize the sellers' revenue. Interestingly, in our *win-win mechanisms*, each buyer's self-interested strategy brings better utility not only to himself but also to the seller.
– The concept of competitive ratio is first introduced by [9]. However, [9]'s competitive ratio is only available to mechanisms with dominant strategy. We generalize the concept to *unconditional competitive ratio*, which is applicable to any mechanisms with equilibria.
– For the model with both limited and unlimited supply goods, we present two *win-win mechanisms* with *unconditional competitive ratio.*

For the auctions with constraints, all the previous papers only considered budget constraint. [4,12,14] studied the model of *one item* to sell under the Bayesian-Nash budget constraint. In the last two years, [2] and [1] began to study the model of *multiple units, multiple bidders* with budget constraint. Our framework is inspired by the work of [8,1]. Specifically, [8] studies the auctions with unlimited supply of digital goods. Each buyer in that model wants at most one copy without any constraint. [1] studies the auctions with limited supply. But the buyers have budget constraint and their demands are unlimited. Since our model allows both unlimited and limited supply of goods and each buyer's bid consists of three parameters, clearly our model is substantially more complicated and general than these previous models. There are many unique properties of our model requiring delicate mechanism designs and proofs. Indeed, our auction is robust against bidders' any strategic behavior.

The paper is organized as follows. In Section 2, we formally give the definitions of the concepts and the model. Section 3 describes a *win-win mechanism* for unlimited supply model with copy constraint. In Section 4, we prove the equilibria of it. Section 5 proves the unconditional competitive ratio of the auction. In Section 6, we generalize the model to limited supply with copy and budget constraint. Furthermore, we present another *win-win mechanism* for it with *unconditional competitive ratio*. Finally we conclude with Section 7. Details of all the proofs can be found in the full version [3].

2 Preliminaries

2.1 Win-Win Mechanisms and Unconditional Competitive Ratio

Before giving its definition, it is necessary for us to introduce some basic knowledge of mechanism design first. We follow the assumption in economics that all agents are rational, i.e, each of them chooses its strategy to maximize its own utility selfishly.

A standard model for mechanism design is as follows. Assume there are n agents, each agent i has its private value t_i (termed its *true type*) which is only known to itself. Furthermore, each agent i is given a set of strategies A_i such that agent i can perform any strategy $a_i \in A_i$. For any input vector $(a_1, \cdots, a_n)$, the mechanism $\mathcal{M}(\mathcal{O}, \{\mathcal{P}_1, \cdots, \mathcal{P}_n\})$ should provide an output function $\mathbf{o} = \mathcal{O}(a_1, \cdots, a_n)$ and a payment function $p_i = \mathcal{P}_i(a_1, \cdots, a_n)$ to each agent. All the output function and payment functions are open to all. In a specific mechanism, if $p_i \geq 0$, agent i needs to pay p_i, as often happens in auctions. If $p_i < 0$, p_i is the money given to agent i. Without loss of generality, we will always assume $p_i \geq 0$ for any i in the context of auctions in the paper.

For any output, each agent i's preference is given by a valued function: $v_i(t_i, \mathbf{o})$, called its valuation. Then its (*quasi linear*) utility can be defined as $u_i(t_i, \mathbf{o}) = v_i(t_i, \mathbf{o}) - \mathcal{P}_i(a_1, \cdots, a_n)$. Accordingly, the auctioneer's revenue should be $\sum_{i=1}^{n} \mathcal{P}_i(a_1, \cdots, a_n)$.

One of the most famous mechanisms is called Vickrey-Clarke-Groves (VCG) mechanism by Vickrey [17], Clarke [6], and Groves [10]. Despite VCG mechanism has the attractive virtue that it is *incentive compatible*, namely, each agent maximizes its utility when it reports its true type, it also has several weaknesses, for instance, the revenue may be very low, even zero. Actually, for a two-sided market in which a product with large, indivisible units is exchanged for money, VCG mechanisms maximize the buyers' payoff and contrarily minimize the sellers' revenue [16,13], which results from the attractive *dominant strategy* property.

In this paper, we develop a new kind of mechanisms called *win-win mechanisms*, in which each agent's self-interested strategy brings better utility not only to himself, but also to the auctioneer. The rigorous definition is as follows.

Definition 2.1 (Win-Win Mechanisms). For any Nash equilibrium $(a_1^*, \ldots, a_n^*)$ in the mechanism, the auctioneer's revenue must be

$$\sum_{i=1}^{n} \mathcal{P}_i(a_1^*, \cdots, a_n^*) \geq \sum_{i=1}^{n} \mathcal{P}_i(t_1, \cdots, t_n)$$

So under any equilibrium state of a *win-win mechanism*, the auctioneer's revenue must be at least as high as that when all the agents tell the truth.

In accord with the notion of *win-win mechanisms*, we generalize the concept of *competitive ratio* first appeared in Goldberg et al.'s paper [9] as follows.

Definition 2.2 (Unconditional Competitive Ratio). An auction $\mathcal{A}$ has some *unconditional competitive ratio* β, if for any set of *rational* bidders and their private true value vector $\mathbb{b}$,

$$\mathsf{Revenue}_{\mathcal{A}}(\mathbb{b}^*) \geq \frac{\mathcal{F}(\mathbb{b})}{\beta}$$

where $\mathbb{b}^*$ is any Nash equilibrium in the auction, $\mathsf{Revenue}_{\mathcal{A}}$ represents the (expect) revenue of auction $\mathcal{A}$ and $\mathcal{F}(\mathbb{b})$ denotes the optimal single price revenue that the auctioneer could have obtained if the true types of the bidders were known in advance.

2.2 Auctions with Copy Constraint: Model and Notation

In the model, the auctioneer sells an idiosyncratic commodity with unlimited copies to n buyers, denoted by $i = 1, 2, \ldots, n$. Each buyer i has two kinds of privately known information: $u_i \in \mathbb{R}^+$, $c_i \in \mathbb{N}$. u_i represents the *unit value* buyer i is willing to pay for the commodity, c_i represents the *number of copies* i demands.

Each buyer i *simultaneously* submits his bid, denoted by (u_i, c_i) to the auctioneer. When receiving all the submitted bids, the auctioneer decides how many copies each buyer will get and how much he should pay. Actually, it is the *one-round sealed-bid* auction.

We use $\mathcal{F}$ represent the optimal revenue the auctioneer could get if the true types of the bidders were known in advance, under the consideration that the auctioneer can only set an identical unit price. Formally,

Definition 2.3. Given bids $\mathbb{b} = ((u_1, c_1), \ldots, (u_n, c_n))$ sorted in decreasing order according to the unit value,

$$\mathcal{F}(\mathbb{b}) = \max_{1 \leq k \leq n} u_k \sum_{1 \leq i \leq k} c_i \tag{2.1}$$

denotes the maximum single price revenue the auctioneer can achieve, and such corresponding price u_k is denoted by $p_{\mathcal{F}(\mathbb{b})}$.

Furthermore, we use $\mathcal{F}^{(2)}(\mathbb{b})$ represent the optimal single price revenue that there are at least 2 winners. I.e.,

$$\mathcal{F}^{(2)}(\mathbb{b}) = \max_{2\leq k\leq n} u_k \sum_{1\leq i\leq k} c_i$$

Definition 2.4 (Utility). Each buyer i's utility for his allocation $\mathsf{allocation}_i$ and payment $\mathsf{payment}_i$ is defined as:

$$\mathcal{U}_i = u_i \times \min\{c_i, \mathsf{allocation}_i\} - \mathsf{payment}_i \tag{2.2}$$

where u_i, c_i are bidder i's true type and $\mathsf{allocation}_i$ is the copies he gets corresponding to his submitted bid.

In addition, for convenience and simplicity, we will use the following notations in the entire paper.

Definition 2.5. Given bidders' **true type** vector $\mathbb{b}$,

$$\mathcal{B} = \left\{(u_i, c_i) | u_i \geq p_{\mathcal{F}(\mathbb{b})}\right\} \tag{2.3}$$

denotes the set of winners whose unit values are not lower than $p_{\mathcal{F}(\mathbb{b})}$. And

$$\alpha = \frac{\max_{\{i|u_i\geq p_{\mathcal{F}(\mathbb{b})}\}}\{c_i\}}{\sum_{u_i\geq p_{\mathcal{F}(\mathbb{b})}} c_i} \tag{2.4}$$

denotes the ratio of maximum demanded copies among winners to the number of demanded copies of all the winners in the single price optimal auction.

3 Algorithm: Random Partition with Revenue Share Auction

In the following, based on the bidders' input vector, we present a *win-win auction* with *unconditional competitive ratio* called *Random Partition with Revenue Share* Auction to obtain the allocation and the payment. It is inspired by the *Cost Sharing Mechanism* in [8]. But due to the differences of our model and input vector, some properties of the *Cost Sharing Mechanism* are unavailable. We shall develop a new and technically involved analysis for it.

Definition 3.1. Given bids $\mathbb{b} = ((u_1, c_1), \ldots, (u_n, c_n))$ and R, find the smallest $p \in \mathbb{R}^+$ such that

$$p = \frac{\mathsf{R}}{\sum_{u_i\geq p} c_i}$$

such p is denoted by p_{R}.

Obviously, such p can be found in $\mathcal{O}(n)$ time.

Lemma 3.2. *Bidders will tell their true unit values in RevenueShare*$_{\mathsf{R}}$ *Auction.*

Auction 1. *RevenueShare$_R$* Auction

1: Calculate p_{R}.
2: For any winner i such that $u_i \geq p_{\mathsf{R}}$, sell c_i copies to bidder i at unit price p_{R}.
3: Other bidders lose.

The following auction is based on Auction 1.

Auction 2. *Random Partition with Revenue Share* Auction (RPRS)

1: Partition bids $\mathbb{b}$ uniformly at random into two bid sets $\mathbb{S}'$ and $\mathbb{S}''$.
2: Compute $\mathcal{F}(\mathbb{S}')$ and $\mathcal{F}(\mathbb{S}'')$.
3: Run $RevenueShare_{\mathcal{F}(\mathbb{S}')}$ on $\mathbb{S}''$ and $RevenueShare_{\mathcal{F}(\mathbb{S}'')}$ on $\mathbb{S}'$ respectively.

Theorem 3.3. *Bidders will tell their true unit values in* Random Partition with Revenue Share Auction*.*

We have proved that bidders will tell their true unit values in RPRS Auction. Then will they tell their true copies? Assume all bidders are rational, then they will lie on copies as long as they can get more benefit. Suppose there are three bidders in RPRS Auction and their true bids are (1,1),(1.1,1),(0.54,1). Now the expect utility of the first bidder is 0.43. However, if he bids (1,2) instead of (1,1), his expect utility will increase to 0.45. In the next section, we will further talk about this issue.

4 Nash Equilibrium and Copy Bounds of RPRS Auction

Consider the counter example above which implies that in RPRS Auction a bidder may have motivation to lie on the number of copies.

In that example, if the first bidder wants to obtain more profit, he has to increase his input number of copies. Although this change results in waste of copies, he may attain more profit as long as the new price is low enough.

We assume that bidders choose their input number of copies to maximize their expect utility given the bids made by the other bidders. If there exists an equilibrium, then in the equilibrium, each bidder has no reason to change his bid, which motivates the following definition.

Definition 4.1. In RPRS Auction, a Nash equilibrium is a set of input parameters such that for any bidder i and his strategy c_i^* in the equilibrium, there does not exist c_i' such that:

$$u_i \min\{c_i, \mathsf{allocation}_i^*\} - p^* \times \mathsf{allocation}_i^* < u_i \min\{c_i, \mathsf{allocation}_i'\} - p' \times \mathsf{allocation}_i'$$

where c_i is bidder i's true copy demand, p^* is the price under the equilibrium while p' is the unit price when i bids c_i'. $\mathsf{allocation}_i^*$ is the copies that i gets under the equilibrium and $\mathsf{allocation}_i'$ is the copies that i gets when he bids c_i'.

Lemma 4.2. *In RevenueShare$_{\mathsf{R}}$ Auction, a bidder will never tell a smaller number of copies than his true copy demand.*

Theorem 4.3. *In RPRS, a bidder will never tell a smaller number of copies than the true copies he demands.*

Suppose bidders are sorted in decreasing order according to the unit value. $\mathbb{b}_{-i} = \big((u_1, c'_1), \ldots, (u_{i-1}, c'_{i-1}), (u_{i+1}, c'_{i+1}), \ldots, (u_n, c'_n)\big)$ denotes the set of all bidders except i's input vectors. $\mathbb{b}_i = (u_i, c_i)$ is bidder i's true type. In *RevenueShare*$_{\mathsf{R}}$ Auction, we use p_{R} represent the price corresponding to the input vector $(\mathbb{b}_i, \mathbb{b}_{-i})$. Now, suppose bidder i changes his copy demand to c'_i and we use p'_{R} represent the new price. In order to increase his utility, bidder i may lie on his copy demand. Then how large could bidder i lie on his copies? The following will answer this question by giving the bounds.

Lemma 4.4. *In RevenueShare$_{\mathsf{R}}$ Auction, if bidder i wants to benefit from increasing his copy demand, his cheating must make at least one loser after him become a winner.*

Assume $C = \{c'_1, \cdots, c'_n\}$ is the set of current copies of all bidders in the auction. For any equilibrium in the equilibrium set, bidder i's copy demand is denoted by c^*_i. From Lemma 4.4, if all bidders become winners, no one can benefit from increasing copy demand, so we can get the upper bound of submitted copies for *RevenueShare*$_{\mathsf{R}}$ Auction: $c^*_i - c_i \le \frac{\mathsf{R}}{u_n} - \sum_{j=1}^{n} c'_j$. Combined with lemma 4.2, we have the following theorem:

Theorem 4.5. *In RevenueShare$_{\mathsf{R}}$ Auction, the copy bounds is as follows:*

$$c_i \le c^*_i \le c_i + \frac{R}{u_n} - \sum_{j=1}^{n} c_j$$

In RPRS, first we partition the bids into two bid sets. Assume the optimal revenue for the two bid sets is $\mathcal{F}'$ and $\mathcal{F}''$ respectively, and the revenue for all bids is $\mathcal{F}$. Since $\mathcal{F}' \le \mathcal{F}$ and $\mathcal{F}'' \le \mathcal{F}$, then we have the theorem:

Theorem 4.6. *In RPRS Auction, the copy bounds in Nash equilibrium is:*

$$c_i \le c^*_i \le \frac{\mathcal{F}}{u_n}$$

5 Revenue Bounds of RPRS Auction

In this section, we focus on the auctioneer's revenue. Here we will prove that RPRS Auction is a *win-win auction* with *unconditional competitive ratio*. We use the optimal single price auction as the benchmark to compute the revenue bounds of RPRS Auction.

Theorem 5.1. *RPRS Auction is a* win-win auction*.*

The following definition and lemmas are prepared for computing the competitive ratio of RPRS Auction.

Definition 5.2. Given any set $S = \{e_1, \ldots, e_n\}$ where $e_i \in \mathbb{R}^+$, partition S uniformly at random into two sets S_1 and S_2 such that $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = S$. Let

$$g(S) = \mathbf{E}\left[\min\left\{\sum_{i\in S_1} e_i, \sum_{i\in S_2} e_i\right\}\right]$$

which is the expectation of the minimum sum between subset S_1 and S_2.

Lemma 5.3. *$\forall i, j$, if $S' = (S \backslash \{e_i, e_j\}) \cup \{e_i + e_i\}$, $g(S') \le g(S)$*

Lemma 5.4. *Given any set $S = \{e_1, \ldots, e_n\}$ where $e_i \in \mathbb{R}^+$ and $|S| \ge 2$, $g(S)/\sum_{j\in S} e_j \ge (1-\alpha')/4$, where $\alpha' = \max_i\{e_i/\sum_{j\in S} e_j\}$.*

Theorem 5.5. *Random Partition with Revenue Share auction is $4/(1-\alpha)$ competitive against α defined in definition 2.5 if there are at least two bidders win.*

6 Limited Supply with Copy and Budget Constraints

The previous model only considers copy constraint. Since some bidders may have limited purchasing power, such as in ad-words auction, here we present another *unconditional competitive auction* for the model with copy and budget constraints. Obviously, when budget tends to infinite, the model is the same as that with copy constraint only. What's more, in this section we talk about limited supply instead of unlimited supply. In fact, unlimited supply is a special case of limited supply. In limited model, if the supply exceeds bidders' total demands, then it is equivalent to the unlimited supply. So here we are talking about a more general model.

Definition 6.1. Given bids $\mathbb{b} = ((u_1, c_1, b_1), \ldots, (u_n, c_n, b_n))$, where u_i represents the *unit value* buyer i is willing to pay for the commodity, c_i represents the *number of copies* i demands and b_i represents i's *budget*. If $\mathbb{b}$ is sorted in decreasing order according to the unit value, then $\mathcal{F}_m$ represents the optimal single price revenue subject to the constraint that there are at most m copies sold. I.e.,

$$\mathcal{F}_m(\mathbb{b}) = \max_{p\in\mathbb{R}^+}\left\{p \cdot \min\left\{\sum_{u_i \ge p} \min\left\{c_i, \frac{b_i}{p}\right\}, m\right\}\right\} \tag{6.1}$$

Definition 6.2. Given bids $\mathbb{b} = ((u_1, c_1, b_1), \ldots, (u_n, c_n, b_n))$, R, and limited supply m, find the largest integer $k \in [1, m]$, such that

$$\sum_{u_i \ge \mathsf{R}/k} \min\left\{c_i, \frac{b_i}{\mathsf{R}/k}\right\} \ge k$$

Let $p_{\mathsf{R},m} = \frac{\mathsf{R}}{k}$.

Auction 3. *RevenueShare*$_{\mathsf{R},m}$ Auction

1: Calculate $p_{\mathsf{R},m}$.
2: Set $r = \sum_{u_i \geq p_{\mathsf{R},m}} \min\left\{c_i, \frac{b_i}{p_{\mathsf{R},m}}\right\}$.
3: If $r \leq m$, for any winner i such that $u_i \geq p_{\mathsf{R},m}$, sell $\min\left\{c_i, \frac{b_i}{p_{\mathsf{R},m}}\right\}$ copies to i at unit price $p_{\mathsf{R},m}$;
If $r > m$, sell m units from r units randomly to the winners at unit price $p_{\mathsf{R},m}$ under the constraint that any winner i should get at most $\min\left\{c_i, \frac{b_i}{p_{\mathsf{R},m}}\right\}$ copies.
4: Other bidders lose.

Obviously, *RevenueShare*$_{\mathsf{R},m}$ Auction sells no more than m units.

Lemma 6.3. *Bidders will tell truth on unit value and budget in* RevenueShare$_{\mathsf{R},m}$ *Auction.*

Based on Auction 3, we get the following auction.

Auction 4. Random Partition with Revenue *Share*$^{(m)}$ Auction (*RPRS*$^{(m)}$)

1: Partition bids $\mathbb{b}$ uniformly at random into two bid sets $\mathbb{S}'$ and $\mathbb{S}''$.
2: Compute $\mathcal{F}' = \mathcal{F}_{\frac{m}{2}}(\mathbb{S}')$ and $\mathcal{F}'' = \mathcal{F}_{\frac{m}{2}}(\mathbb{S}'')$.
3: Run *RevenueShare*$_{\mathcal{F}',\frac{m}{2}}$ on $\mathbb{S}''$ and *RevenueShare*$_{\mathcal{F}'',\frac{m}{2}}$ on $\mathbb{S}'$ respectively.

Similarly, we get the following two theorems:

Theorem 6.4. *In RPRS*$^{(m)}$ *Auction, bidders always tell their true types of unit value and budget.*

Theorem 6.5. *RPRS*$^{(m)}$ *Auction is a* win-win auction*.*

In this general model, we can still get the competitive ratio $4/(1-\alpha)$, however, the proof in the previous section can not apply to this model.

Theorem 6.6. *RPRS*$^{(m)}$ *is* $4/(1-\alpha)$ *competitive against* α *if there are at least two bidders win.*

7 Conclusion and Discussions

This paper investigates multi-unit auctions with copy and budget constraints. We introduce a new kind of mechanisms called *win-win mechanisms*. Then we design *win-win mechanisms* with the same *unconditional competitive ratio* of $\frac{4}{1-\alpha}$ for both unlimited and limited supply goods. For any auction with dominate strategy, possibly there exists an alternate win-win auction with better competitive ratio. So it is worth trying to improve their competitive ratios by designing performance guaranteed win-win auctions.

It is worthy of emphasis that our novel *win-win mechanisms* shed light on the following scenario. Sometimes, in order to maximize the revenue, the optimal auction has to be executed inefficiently. I.e., the optimal solution of underlying allocation and payment will have to be found in exponential time. When it happens, we can relax the auction as long as bidders' strategic behaviors must also lead to larger total revenue to the auctioneer. In other words, the auction makes use of the bidders' computational power to increase the auctioneer's revenue. This idea also has emerged in the mechanism design in [15].

Coincidentally, a new kind of mechanisms called *output truthful mechanisms* is raised these days in [5,11]. In *output truthful mechanisms*, what concerns us is whether the output under the equilibria in the mechanisms is the same as the result under the truthful input, while our *win-win mechanisms* are concerned about whether the revenue under the equilibria in the mechanism is higher than the revenue under the truthful input. Actually the motivation of both mechanisms is to improve otherwise performances by relaxing the constraint of dominant strategy.

References

1. Z. Abrams. Revenue maximization when bidders have budgets. In *Proceedings of the seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-06)*, pages 1074–1082, Miami, FL, Jan. 22–26 2006. ACM Press.
2. C. Borgs, J. T. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *ACM Conference on Electronic Commerce (EC-05)*, pages 44–51, 2005.
3. T.-M. Bu, Q. Qi, and A. W. Sun. Unconditional competitive auctions with copy and budget constraints. Full version, availble at `http://www.cs.cityu.edu.hk/~deng/`.
4. Y.-K. Che and I. Gale. Standard auctions with financially constrained bidders. *Review of Economic Studies*, 65(1):1–21, Jan 1998.
5. X. Chen, X. Deng, and B. J. Liu. On incentive compatible competitive selection protocol. In *Lecture Notes in Computer Science*, volume 4112, pages 13–22, Taipei, Taiwan, Aug. 15 2006.
6. E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:11–33, 1971.
7. B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of dollars worth of keywords. Available at `http://www.nber.org/papers/w11765.pdf`.
8. A. Fiat, A. V. Goldberg, J. D. Hartline, and A. R. Karlin. Competitive generalized auctions. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 72–81, New York, May 19–21 2002. ACM Press.
9. A. V. Goldberg, J. D. Hartline, and A. Wright. Competitive auctions and digital goods. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-01)*, pages 735–744, New York, Jan. 7–9 2001. ACM Press.
10. T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
11. M.-Y. Kao, X.-Y. Li, and W. Wang. Output truthful versus input truthful: A new concept for algorithmic mechanism design, 2006.
12. J.-J. Laffont and J. Robert. Optimal auction with financially constrained buyers. *Economics Letters*, 52:181–186, 1996.

13. H. B. Leonard. Elicitation of honest preferences for the assignment of individuals to positions. *The Journal of Political Economy*, 91:461–479, 1983.
14. E. S. Maskin. Auctions, development, and privatization: Efficient auctions with liquidity-constrained buyers. *European Economic Review*, 44(4–6):667–681, 2000.
15. N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, pages 242–252, N.Y., Oct. 17–20 2000. ACM.
16. L. S. Shapley and M. Shubik. The assignment game i: the core. *International Journal of Game Theory*, 1:111–130, 1972.
17. W. Vickery. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, XVI:8–37, 1961.

Truthful Auctions with Optimal Profit

Pinyan Lu[1], Shang-Hua Teng[2,⋆], and Changyuan Yu[1]

[1] Department of Computer Science and Technology
Tsinghua University, Beijing, China
and Microsoft Research Asia
{lpy, yucy05}@mails.tsinghua.edu.cn
[2] Department of Computer Science, Boston University
steng@cs.bu.edu

Abstract. We study the design of truthful auction mechanisms for maximizing the seller's profit. We focus on the case when the auction mechanism does not have any knowledge of bidders' valuations, especially of their upper bound. For the Single-Item auction, we obtain an "asymptotically" optimal scheme: for any $k \in Z^+$ and $\epsilon > 0$, we give a randomized truthful auction that guarantees an expected profit of $\Omega(\frac{OPT}{\ln OPT \ln\ln OPT \cdots (\ln^{(k)} OPT)^{1+\epsilon}})$, where OPT is the maximum social utility of the auction. Moreover, we show that no truthful auction can guarantee an expected profit of $\Omega(\frac{OPT}{\ln OPT \ln\ln OPT \cdots \ln^{(k)} OPT})$.

In addition, we extend our results and techniques to Multi-units auction, Unit-Demand auction, and Combinatorial auction.

1 Introduction

Auction has become an active area of research in Computer Science both for its commercial applications in the rapid expanding space of Internet Economy and for its algorithmic and game-theoretical appeals. A typical auction problem consists of one or more sellers who have several items to sell and a collection of bidders who want to buy what they would like to have with as little price as possible. An auction mechanism then determines who gets which items and at what price. As the participants (sellers and bidders) in an auction have their own self-incentive and private information, an auction problem can be viewed as a game among its participants.

The concept of *truthful* or *incentive compatible* mechanism captures the notion of reasonable auctions — a reasonable auction should encourage its bidders to show their true valuations. Truthfulness is a quite strong game-theoretical requirement, stating that for each bidder, bidding his/her true valuation is among the optimal strategies, no matter how other bidders behave. In another word, in a truthful auction, the decision and pricing scheme are such that there is no reason for any bidder to lie.

⋆ This work is done at Microsoft Research Asia.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 27–36, 2006.
© Springer-Verlag Berlin Heidelberg 2006

1.1 Related Work and Motivations

Many auction problems have truthful mechanisms. An example is the famous Vickrey-Clarke-Groves (VCG) mechanism that maximizes the social utility [13,3,7]. However, in VCG, the maximization of the social utility might be achieved at the expense of the seller's profit — generally, the VCG scheme provides no guarantee on the seller's profit. A natural step is to design a truthful auction mechanism that maximizes profits.

Assuming that the distribution of valuations are known or can be gathered by some statistical means, VCG mechanism with a properly chosen reserved price can obtain very tight bounds on the expected profits [12,11,10]. However, there are reasons to consider profit-maximization auction without full knowledge of the valuation distributions[5].

A possible scenario is that the range of bidders' valuations is known. Given an upper bound h on the valuations, truthful auction mechanisms have been developed to achieve a profit of $\Omega\left(\frac{OPT}{\log h}\right)$, where OPT is the optimal social utility of the auction [8,9].

In absence of any valuation information, Goldberg, Hartline, Wright introduced a notion of competitive auctions in [6]. They proposed to measure the quality of the profit-maximization scheme using a worst-case competitive analysis against $F^{(2)}$, the optimal single-price auction that sells at least two items. Since then, several truthful auction schemes with constant competitive ratios have been developed [5,1,2].

Note that $F^{(2)}$ is a relatively lower bentchmark compared to OPT. In some cases, one can not bound $F^{(2)}$ with OPT. In this paper, we compare the profit directly with OPT.

1.2 Our Results

For auctions with a single item, we present a randomized truthful profit-maximization scheme and prove that it is "asymptotically" optimal. In particular, for $\forall k \in Z^+, \epsilon > 0$, we give a randomized truthful auction that guarantees an expected profit of $\Omega(\frac{OPT}{\ln OPT \ln\ln OPT \cdots (\ln^{(k)} OPT)^{1+\epsilon}})$. Moreover, we show that no truthful auction can always achieve a profit of $\Omega(\frac{OPT}{\ln OPT \ln\ln OPT \cdots \ln^{(k)} OPT})$.

Furthermore, we extend our technique for Single-Item auction to more complex auction problems such as multi-units auction, AdWords auction (Unit-Demand auction), and combinatorial auction. For multi-units and AdWords auctions, both our upper and lower bounds can be generalized. All our schemes also guarantee that the expected social utility are within a constant fraction of the optimal social utility.

For the general combinatorial auction, we build a profit-oriented auction scheme on the truthful approximation scheme of Dobzinski, Nisan, and Schapira [4]. We can achieve a profit of $\Omega(\frac{OPT}{\sqrt{m} \ln OPT \ln\ln OPT \cdots (\ln^{(k)} OPT)^{1+\epsilon}})$, where m is the number of items. When the bidders' utility functions are submodular, a profit of $\Omega(\frac{OPT}{(\log m)^2 \ln OPT \ln\ln OPT \cdots (\ln^{(k)} OPT)^{1+\epsilon}})$ can be obtained.

2 Notations

We assume that there are n bidders, and a set M of distinct items, $M = \{1, 2, \cdots, m\}$. In addition, the seller has c_j copies (c_j may be $+\infty$) of item $j \in M$. A bundle of items can be specified as a vector $(d_1, d_2, \cdots, d_m)$, where $0 \le d_j \le c_j, \forall j \in M$, and we denote the collection of all the bundles with $\mathcal{D}$. Each bidder i has a private valuation function v_i, which assigns a non-negative value to each bundle of items.

Each bidder submits a bid $b_i = \{b_i(S), S \in \mathcal{D}\}$. An *auction mechanism* then outputs an allocation $(S_1, S_2, \cdots, S_n)$, where $S_i \in D$, and a price $(p_1, p_2, \cdots, p_n)$. A feasible output of the mechanism must satisfy the following two conditions:

- **Limited Supply:** For each item $j \in M$, there are at most c_j copies in $(S_1, S_2, \cdots, S_n)$.
- **Individual Rationality:** For each bidder $i \in [n]$, $p_i \le b_i(S_i)$.

A deterministic mechanism is *truthful* if for each bidder, truth-telling is a dominant strategy, which means that her utility is maximized when she bids truthfully no matter how others bid. For randomized mechanisms, there are two extensions of truthfulness, universally truthful and truthful in expectation. A randomized mechanism is *universally truthful* if it is a distribution of truthful deterministic mechanisms. *Truthfulness in expectation* means that the expected utility of a bidder is maximized when bidding truthfully.

In this paper, we focus on several special cases.

1. Single-Item auction: M consists of a single item, possibly with multiple copies. Each bidder would like to buy at most one copy.
2. Unit-Demand auction: multiple items, each item with one copy. Each bidder would like to buy at most one item and is considering a number of different options. In another words, each bidder only bids for Single-Item bundle.
3. Combinatorial auction: multiple items, each item with one copy. The bidder bids for subsets of M.

3 Single-Item Auction

In this section, we focus on the Single-Item auction. We first consider the case when there is one copy of the item, and give a profit-optimal truthful mechanism. We then extend this result to the case of multiple copies.

3.1 Single-Copy Auction

Without loss of generality, we assume the bids are $b_1 \ge b_2 \ge \cdots \ge 1$. Let $g(x) = \ln x + 1$, and $\tilde{g}^{(k)}(x) = \prod_{i=1}^{k} g^{(i)}(x)$, Recall that $g^{(i)}(x) = g(g^{(i-1)}(x)), \forall i \ge 2$.

Algorithm: `SingleCopyAuction`

INPUT: $k \in Z^+, \epsilon > 0$ and $\delta > 0$.

1. If there is only one bidder, we set $b_2 = 1$.
2. With probability $1 - \delta$, we use the second price auction, that is, the highest bidder wins the item at a price of the second highest price.
3. With probability δ, the seller chooses a reserved price r according to the distribution with density:

$$f_{k,\epsilon}(x) = \frac{\epsilon}{x\tilde{g}^{(k-1)}(\frac{x}{b_2})\left(g^{(k)}(\frac{x}{b_2})\right)^{1+\epsilon}}, x \in [b_2, +\infty)$$

Then if $b_1 \geq r$, the highest bidder wins the iterm with price r. Otherwise, the item remains unsold.

It is well known that the second price auction is truthful. Because the highest bidder is the only potential recipient of the item and the reserved price r is chosen independently of her bid b_1, the auction of step 3 is a distribution of truthful mechanism. Thus, our acution scheme is universally truthful.

The algorithm above uses reserved price auction to guarantee the seller's profit while uses second price auction to enhance the social utility. The parameter δ provides a tradeoff between these two objectives.

Theorem 1 (Profit Guarantee). *Let $E(R)$ be the expected profit of the auction and $E(SU)$ be its expected social utility. Let OPT denote the maximum social utility. Then we have*

$$E(R) = \Omega\left(\frac{OPT}{\tilde{g}^{(k-1)}(OPT)\left(g^{(k)}(OPT)\right)^{1+\epsilon}}\right)$$
$$E(SU) \geq (1-\delta)OPT$$

Proof: For simplicity, we give a proof for $k = 1, \epsilon = 1$, and $\delta = \frac{1}{2}$. The proof is essentially the same for general k, ϵ, and δ.

In Single-Item auction, the optimal profit OPT is equal to the maximum bid b_1. So it is obvious that $E(SU) \geq \frac{1}{2}OPT$ because with probability of 1/2, we use the second price auction and get a social utility of OPT.

For the seller's profit, we have:

$$\begin{aligned} E(R) &= \frac{1}{2}b_2 + \frac{1}{2}\int_{b_2}^{b_1} xf(x)dx \\ &= \frac{1}{2}b_2 + \frac{1}{2}\int_{b_2}^{b_1} \frac{1}{(\ln\frac{x}{b_2}+1)^2}dx \end{aligned}$$

$$\geq \frac{1}{2}b_2 + \frac{1}{2}\frac{b_1 - b_2}{(\ln\frac{b_1}{b_2} + 1)^2}$$
$$\geq \frac{b_1}{2(\ln\frac{b_1}{b_2} + 1)^2}$$
$$\geq \frac{OPT}{2(\ln OPT + 1)^2} \qquad \square$$

We now prove that the bound in Theorem 1 is essentially tight. To show this, we first give two technical lemmas.

Lemma 1. *Let Φ be the distribution of a bid with $Pr(b_1 = 2^j) = \frac{1}{2^{j+1}}, j = 0, 1, 2, \cdots$. Then no truthful (even in expectation) mechanism can extract revenue greater than 1 on Φ.*

Proof: The distribution we used here is a modified version of distribution used in [12], and the proof is similar. $\square$

Lemma 2. *For a fixed $k \in Z^+$, $\sum_{j\geq 0}\frac{1}{\tilde{g}^{(k)}(2^j)}$ goes to infinity.*

Proof: We show that for any i, there exists constant C_i and $N_i > 0$, such that for any $x \geq N_i$, we have $g^{(i)}(x) \leq C_i \ln^{(i)} x$. This is shown by induction on i.

For $i = 1$, $g(x) = \ln x + 1 \leq 2\ln x, \forall x \geq e$.

$$\begin{aligned}
g^{(i)}(x) &= g^{(i-1)}(\ln x + 1) & \\
&\leq g^{(i-1)}(2\ln x) & \text{for } x \geq e \\
&\leq C_{i-1}\ln^{i-1}(2\ln x) & \text{for } 2\ln x \geq N_{i-1} \\
&\leq C_i \ln^{(i)}(x) & \text{exists } C_i, \text{ and } N_i
\end{aligned}$$

For a fixed $k \in Z^+$, let $C = c_1 c_2 \cdots c_k$ and J be the smallest integer such that $J \in Z^+$, $2^J > max\{N_i : 1 \leq i \leq k\}$. Then we have $g^{(i)}(2^j) \leq C_i \ln^{(i)}(2^j) \leq C_i \ln^{(i-1)} j$. So we have

$$\sum_{j\geq 0}\frac{1}{\tilde{g}^{(k)}(2^j)} \geq \sum_{j=0}^{J}\frac{1}{\tilde{g}^{(k)}(2^j)} + \frac{1}{C}\sum_{j>J}\frac{1}{j\ln j\cdots\ln^{(k-1)} j} = +\infty. \qquad \square$$

Theorem 2 (Impossibility Result). *For any $k \in Z^+$, there is no truthful (even in expectation) mechanism with an expected profit of $\Omega(\frac{OPT}{\tilde{g}^{(k)}(OPT)})$.*

Proof: Assume there is a truthful auction, with an (expected) profit of $\Omega\left(\frac{OPT}{\tilde{g}^{(k)}(OPT)}\right)$. That is to say, $\exists c > 0, N > 0$, s.t. $E(R) \geq c\frac{OPT}{\tilde{g}^{(k)}(OPT)}$, when $OPT > N$. Let J be the smallest integer such that $2^J > N$. Considering the bid distribution Φ, we have

$$E(R) \geq c \sum_{j \geq J} \frac{1}{2^{j+1}} \frac{2^j}{\tilde{g}^{(k)}(2^j)}$$
$$\geq \frac{c}{2} \sum_{j \geq J} \frac{1}{\tilde{g}^{(k)}(2^j)}$$

By lemma 2, $E(R)$ goes to infinity, which contradicts with lemma 1. □

3.2 Multi-copy Auction

In a multi-copy auction, there is one item with c copies (c may be unbounded). We give a similar mechanism as Single-Copy auction. Our analysis can be extended to this case. Since there is no difference between the case $c > n$ ($c = +\infty$) and the case $c = n$, we can assume that $c \leq n$, and $b_1 \geq b_2 \geq \cdots \geq b_n$. We use the following auction scheme.

Algorithm: `MultiCopyAuction`

INPUT: $k \in Z^+, \epsilon > 0$, and $\delta > 0$.

1. If $c = n$, we set $b_{n+1} = 1$.
2. With probability $1 - \delta$, we use the VCG mechanism: sell c items to the c highest bidder at the price of the $(c+1)$-th highest bidder.
3. With probability δ we sell the items to the highest c bidders with a reserved price r chosen according to the distribution with density:

$$f_{k,\epsilon}(x) = \frac{\epsilon}{x\tilde{g}^{(k-1)}(\frac{x}{b_{c+1}}) \left(g^{(k)}(\frac{x}{b_{c+1}})\right)^{1+\epsilon}}, x \in [b_{c+1}, +\infty)$$

Similarly to the Single-Copy auction, we can obtain the following lower and upper bounds on the expected profit for multi-copy auctions.

Theorem 3. *Let OPT denote the optimal social utility and b_{max} be the highest bid (By our assumption $OPT = \sum_{j=1}^{c} b_j$ and $b_{max} = b_1$). Then we have:*

$$E(R) = \Omega\left(\frac{OPT}{\tilde{g}^{(k-1)}(b_{max})\left(g^{(k)}(b_{max})\right)^{1+\epsilon}}\right)$$
$$E(SU) \geq (1-\delta)OPT$$

In addition, no truthful auction can obtain an expected profit of $\Omega\left(\frac{OPT}{\tilde{g}^{(k)}(b_{max})}\right)$.

4 Unit-Demand Auction

We now consider the auction of multiple items, as in the keywords auction. Assume there are n bidders (advertisers), and m slots on the web page to place advertisements. The advertiser bids for each slot on the web page, and the search engine must decide which m bidders win slots, as well as the order to place the advertisements and the prices. We focus on the case when the search engine does not place two identical advertisements on the same page, which is the so-called Unit-Demand auction.

As each bidder i has a bid for each item j, we can express their valuations by a matrix $B = (b_i(j))$.

Algorithm: `AdWordAuction`

1. Choose a reserved price r according to the following distribution: with a probability of $1-\delta$, set $r=1$; with a probability of δ, pick r according to the distribution with density
$$\frac{\epsilon}{x\tilde{g}^{(k-1)}(x)\left(g^{(k)}(x)\right)^{1+\epsilon}}, x \in [1, +\infty)$$
.
2. Compute prices $\mathbf{p}$ and allocation S by running VCG on input B with reserved prices $\mathbf{r} = (r, \cdots, r)$. The *reserved price VCG* works as follows: add m virtual bidders with bid $\mathbf{r} = (r, \cdots, r)$ into the auction, then run VCG to determine the allocation and price of each item. If an item is sold to a virtual bidder, then it is in fact unsold in the original auction.

Recall the VCG scheme for Unit-Demand auction computes a maximum weighted matching betwen bidders and items and allocates the items accordingly. The price of each item is set to be the bidding price of its recipient minus the difference of the total weights of this matching and of the maximum weighted matching without this recipient. Clearly, VCG runs in polynomial time in the number of bidders and items. Therefore, the algorithm above is a polynomial-time auction scheme.

Theorem 4. *The Unit-Demand auction is truthful and has an expected profit of* $E(R) = \Omega\left(\frac{OPT}{\tilde{g}^{(k-1)}(b_{max})\left(g^{(k)}(b_{max})\right)^{1+\epsilon}}\right)$*, where* $b_{max} = max_{i,j}\{b_i(j)\}$*. The expected social utility* $E(SU) \geq (1-\delta)OPT$*.*

Proof: Again for simplicity, we prove the theorem for $k=1, \epsilon=1$, and $\delta = \frac{2}{3}$. Let M be a maximum weighted matching between the n bidders and m items, $p_1 \geq p_2 \geq \cdots \geq p_m$ be the prices of the items sold in M, and $n_x = argmax_j\{p_j \geq x\}$. Using the similar technique in [8], we know that when the reserved price is picked at x, there are n_x items with prices higher than x sold in M, and at least half of them can be sold by the reserved price auction. So we have:

$$\begin{aligned} E(R) &= \frac{1}{3}m + \frac{2}{3}\int_1^{+\infty} \frac{n_x}{2} x f(x) dx \\ &\geq \frac{1}{3}m + \frac{1}{3}(\sum_{i=1}^{m} \int_{p_{i+1}}^{p_i} n_x \frac{1}{(\ln x + 1)^2} dx) \end{aligned}$$

$$
\begin{aligned}
&\geq \frac{1}{3}m + \frac{1}{3}\frac{1}{(\ln p_1 + 1)^2}\sum_{i=1}^{m} i(p_i - p_{i+1}) \\
&= \frac{1}{3}m + \frac{1}{3}\frac{1}{(\ln p_1 + 1)^2}(p_1 + p_2 + \cdots + p_m - m) \\
&\geq \frac{OPT}{3(\ln b_{max} + 1)^2}
\end{aligned}
$$

With a probability of 1/3, we use the VCG with a reserved price of 1 and can obtain the optimal social utility OPT. So $E(SU) \geq \frac{1}{3}OPT$. □

5 Combinatorial Auction

We modify the algorithm in [4]. In step 3, they use a second-price auction for M, the bundle of all items, with a reserved price p_0, however, we use a randomly chosen reserved price. To be self-contained, we include the basic steps of this algorithm.

Algorithm: `CombinatorialAuction`

- **Phase I:** Partitioning the Bidders
  1. Assign each bidder to exactly one of the following three sets: SEC-PRICE with probability $1-\epsilon$, FIXED with probability $\frac{\epsilon}{2}$, and STAT with probability $\frac{\epsilon}{2}$.
- **Phase II:** Gathering Statistics
  2. Calculate the value of the optimal fractional solution in the combinatorial auction with all m items, but only with the bidders in STAT. Denote this value by OPT^*_{STAT}.
- **Phase III:** A Second-Price Auction with reserved price.
  3. Randomly pick a reserved price r according to the following density function: $f_{k,\epsilon_1}(x) = \frac{\epsilon_1}{x\tilde{g}^{(k-1)}(\frac{x}{p_0})\left(g^{(k)}(\frac{x}{p_0})\right)^{1+\epsilon_1}}, x \in [p_0, +\infty)$ where $p_0 = OPT^*_{STAT}$.
- **Phase IV:** A Fixed-Price Auction
  4. Let $R = M, p = \epsilon OPT^*_{STAT}/(8m)$.
  5. For each bidder $i \in$ FIXED, in some arbitrary order:
  (a)Let S_i be the demand of bidder i given the following prices: p for each item in R, and $+\infty$ for each item in $M - R$.
  (b) Allocate S_i to bidder i, and set his price to be $p|S_i|$.
  (c) Let $R = R \setminus S_i$.

Theorem 5. *In the general combinatorial auction,*

$$E(R) = \Omega\left(\frac{OPT}{\sqrt{m}(\tilde{g}^{(k-1)}(OPT))\left(g^{(k)}(OPT)\right)^{1+\epsilon_1}}\right)$$

When the bidders' utility functions are submodular, we have

$$E(R) = \Omega\left(\frac{OPT}{(\log m)^2(\tilde{g}^{(k-1)}(OPT))\left(g^{(k)}(OPT)\right)^{1+\epsilon_1}}\right)$$

Proof: The combinatorial auction can be formulated as a linear program. let OPT^* be the optimal fractional solution. As mentioned in [4], there are two cases:

- There is a bidder i such that $v_i(M) \geq \frac{OPT^*}{\sqrt{m}}$. This is similar to the Single-Copy auction. Let $v_{max} = max_i v_i(M)$, then $OPT \geq v_{max} \geq \frac{OPT^*}{\sqrt{m}} \geq \frac{OPT}{\sqrt{m}}$.

$$\begin{aligned} E(R) &= \Omega\left(\frac{v_{max}}{\tilde{g}^{(k-1)}(v_{max})\left(g^{(k)}(v_{max})\right)^{1+\epsilon_1}}\right) \\ &= \Omega\left(\frac{OPT}{\sqrt{m}(\tilde{g}^{(k-1)}(OPT))\left(g^{(k)}(OPT)\right)^{1+\epsilon_1}}\right) \end{aligned}$$

- For each bidder i, $v_i(M) \geq \frac{OPT^*}{\sqrt{m}}$. As shown in [4], $E(R)$ is $\Omega(\frac{OPT^*}{\sqrt{m}})$ Thus the expected profit is $\Omega\left(\frac{OPT}{\sqrt{m}(\tilde{g}^{(k-1)}(OPT))\left(g^{(k)}(OPT)\right)^{1+\epsilon_1}}\right)$.

The proof is similar for the case when the bidders' utility functions are submodular. □

6 Discussions and Future Work

In the scenario that an upper bound h of the valuations is given, we can give a mechanism which improves the profit guarantee in [8,9] by a constant factor $\log e$. The algorithm is a VCG scheme with a reserved price, which is randomly picked according to the density function $f(x) = \frac{1}{x \ln h}, x \in [1, h]$. This scheme guarantees an expected profit of $\frac{OPT}{\ln h}$, which is proved to be optimal in [12].

All the randomized VCG scheme with reserved price mentioned in our algorithms can be translated into a **Randomized-Fixed-Price Auction**. The fixed price is picked from the same distribution as that of the reserved price. Then we sell items with the fixed price to the bidders in a random order. All the profit guarantees and the proofs above still apply. Using this Randomized-Fixed-Price scheme, we can extend our results to the online auctions[1].

The Unit-Demand auction is in fact a matching problem between bidders and items. The maximum social utility are achieved by the maximum weighted matching. A natural generalization of Unit-Demand auction is the following **multi-pattern auction**: Given t_1 groups of items, the bidders have their valuations for all items. The auction mechanism then chooses one of the groups and allocates its items to the bidders.

From the view of matching, the valuations define t_1 sets of matching problems between the bidders and items. The multi-pattern auction could be useful in

Internet advertising. For example, the search engine can offer several kinds of patterns for sponsored advertising, each with several slots to place the advertisements. Each advertiser (bidder) could submit a bid for each slot in every pattern.

Assuming that there are t_1 groups and each group has t_2 items, we can extend our Unit-Demand auction scheme to obtain the following result.

Theorem 6. *For any $k \in Z^+, \epsilon > 0$, there is a truthful auction scheme with an expected profit of*

$$E(R) = \Omega\left(\frac{1}{t}\frac{OPT}{\tilde{g}^{(k-1)}(b_{max})\left(g^{(k)}(b_{max})\right)^{1+\epsilon}}\right)$$

where $b_{max} = max_{i,j}\{b_i(j)\}, t = min\{t_1, t_2\}$.

Open Problem: Can we improve the factor of $\frac{1}{t}$ in the bound?

References

1. A. Blum, J. D. Hartline. Near-Optimal Online Auctions. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1156-1163, 2005.
2. A. Blum, V. Kumar, A. Rudra, and F. Wu. Online Learning in Online Auctions. *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms* , pages 202-204, 2003.
3. E. H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 11:17-33, 1971.
4. S. Dobzinski, N. Nisan, M. Schapira. Truthful Randomized Mechanisms for Combinatorial Auctions. *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 644-652, 2006.
5. A. Goldberg, J. D. Hartline, A. Karlin, A. Wright, and M. Saks. Competitive auctions. *In submitted to Games and Economic Behavior*, 2002.
6. A. Goldberg, J. D. Hartline, and A. Wright. Competitive auctions and digital goods. *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages $735 - 744, 2001$.
7. T. Groves. Incentives in Teams. *Econometrica*, 41:617- 631, 1973.
8. V. Guruswami, J. D. Hartline, A. Karlin, D. Kempe, K. Kenyon, and F. McSherry. On Profit Maximizing Envy-free Pricing. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1164-1173, 2005.
9. A. Likhodedov and T. Sandholm. Approximating Revenue-Maximizing Combinatorial Auctions. *In Proceedings of the National Conference on Artificial Intelligence (AAAI)*,pages 267-274, 2005.
10. R. B. Myrson. Optimal auction design. *Mathematics of operational research*, $6 : 58 - 73, 1981$.
11. A. Ronen. On approximating optimal auctions. *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 11-17, 2001.
12. A. Ronen and A. Saberi. Optimal auctions are hard. *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science* , pages 396-405, 2002.
13. W. Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, $16 : 8 - 37, 1961$.

Mechanisms with Verification for Any Finite Domain⋆

Carmine Ventre

Dipartimento di Informatica ed Applicazioni "R.M. Capocelli", Università di Salerno, Via Ponte Don Melillo I-84084 Fisciano (SA), Italy
ventre@dia.unisa.it

Abstract. In this work we study *mechanisms with verification*, as introduced by Nisan and Ronen [STOC 1999], to solve problems involving selfish agents. We provide a technique for designing *truthful* mechanisms with verification that optimally solve the underlying optimization problem. Problems (optimally) solved with our technique belong to a rich class that includes, as special cases, *utilitarian* problems and many others considered in literature for so called *one-parameter* agents (e.g., the make-span). Our technique extends the one recently presented by Auletta et al [ICALP 2006] as it works for *any* finite multi-dimensional valuation domain. As special case we obtain an alternative technique to optimally solve (though not in polynomial-time) Scheduling Unrelated Machines studied (and solved) by Nisan and Ronen. Interestingly enough, our technique also solves the case of *compound* agents (i.e., agents declaring more than a value). As an application we provide the *first* optimal truthful mechanism with verification for Scheduling Unrelated Machines in which every selfish agent controls more than one (unrelated) machine. We also provide methods leading to approximate solutions obtained with polynomial-time truthful mechanisms with verification. With such methods we obtain polynomial-time truthful mechanisms with verification for smooth problems involving compound agents composed by one-parameter valuations. Finally, we investigate the construction of mechanisms (with verification) for infinite domains. We show that existing techniques to obtain truthful mechanisms (for the case in which verification is not allowed), dealing with infinite domains, could completely annul advantages that verification implies.

1 Introduction

Many computer scientists look at the world from a new perspective: they study problems assuming there are *selfish* entities working for their own interests rather than for community interests. This implies that one has to design new *algorithms* that have to deal, not just with the combinatorial structure of the problem, but also, and perhaps *mainly*, with *private* interests conflicting

⋆ Research partially supported by the European Project FP6-15964, Algorithmic Principles for Building Efficient Overlay Computers (AEOLUS).

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 37–49, 2006.
© Springer-Verlag Berlin Heidelberg 2006

with the aim of *optimizing*. The new perspective is motivated by many real-life situations. Consider, for example, computations over the Internet. They often involve self-interested parties (*selfish agents*) which may manipulate the system by misreporting a fundamental piece of information they hold (their own *type* or *valuation*). The system runs some algorithm which, because of the misreported information, is no longer guaranteed to return a "globally optimal" solution (optimality is naturally expressed as a function of agents' types) [19]. Since agents can manipulate the algorithm by misreporting their types, one augments algorithms with carefully designed payment functions which make disadvantageous for an agent to do so. A *mechanism* consists of an algorithm (also termed *social choice function*) and payment functions which associate a payment to every agent. Payments should guarantee that it is in the agent's interest to report her type correctly. A social choice function is *implementable* if the *utility* that an agent derives from the chosen outcome and from the payment she receives is maximum when this agent reports her type correctly (see Sect. 1 for a formal definition of these concepts). When a social choice function A is implementable we refer to the pair (A, P) to as *truthful mechanism*. The only known general technique for designing truthful mechanisms is the classical Vickrey-Clarke-Groves (VCG) paradigm [25,12,9]. These mechanisms suffer from two main limitations: (i) they can be used only for a limited family of optimization functions (see e.g. [19]) and (ii) they require the algorithm to compute exact solutions which, in many cases, is unfeasible if the mechanism has to run in polynomial time (see e.g. [20]). In their seminal work, Nisan and Ronen [19] introduce a mechanism design approach to computer science problems having *non-utilitarian* optimization functions, and show that even *exponential-time* mechanisms cannot achieve optimal solutions (in contrast with the unselfish counterpart where a $(1+\varepsilon)$-approximation can be obtained in polynomial-time). An alternative to VCG mechanisms is to restrict the *domain* of the agents (i.e., possible values they can report). For example the so-called *one-parameter* agents have been studied in [18,2]. Unfortunately, these domains are rather limited: for instance, although they can model scheduling problems on related machines [2], they cannot model the unrelated case in [19], nor the case of agents owning more than one machine. A quite innovative mechanism design approach has been introduced by Nisan and Ronen [19] in order to overcome the above mentioned difficulties for their scheduling problem: the mechanism can observe the job release time and provides payments *after* the solution has been implemented. These mechanisms are called *mechanisms with verification*. More "classical" mechanisms without verification award the payment associated to an agent unconditionally (i.e., without performing any kind of verification and solely based on the agents reported types). There are several reasons for being interested in mechanisms with verification. First of all, there are specific optimization problems for which verification allows to overcome certain impossibility results for mechanisms without verification [19,4,5] (which holds also for one-parameter agents). Moreover, mechanisms with verification are very natural and many real-life applications require and/or already implement this kind of approach: reputation is used by the e-Bay system to measure credibility of sellers; service providers offer connectivity to the Internet with the guarantee of a minimal rate. Finally, verification helps in designing truthful mechanisms.

Our contribution. In this work we prove the first general result on mechanisms with verification and show that, for any finite domain, there is a mechanism that optimizes *any* minimization (resp. maximization) function monotone non-increasing (resp. non-decreasing) in the agents' valuations. This result applies to *any finite domain* and extend to a "multidimensional scenario" called *compound agents* (see Def. 6). We indeed provide a social choice function (i.e., an optimization algorithm) which is implementable with verification on finite domains and that maximizes any optimization function $\mu(\cdot)$ which is monotone in the agents valuations (i.e., $\mu(v_1(X), \ldots, v_n(X))$ is non-decreasing in each agent valuation $v_i(X)$). With our "always implementable" social choice function we are able to construct truthful mechanisms (with verification) optimizing the underlying optimization function (Cor. 1). Observe that VCG mechanisms [25,9,12] can only deal with particular functions of this form called *affine maximizers* (basically, the case $\mu(v_1(X), \ldots, v_m(X)) = \sum_i \beta_i v_i(X)$, with constants β_i defined by the mechanisms). The $Q||C_{\max}$ scheduling problem is an example of an optimization problem involving a monotone non-increasing function (thus our result applies to $Q||C_{\max}$) that is not an affine maximizer. Our result gives an alternative proof of the existence of an exact truthful mechanism with verification for unrelated machines [19]. Interestingly enough, our results extend to the case of compound agents (see Sec. 3). To the best of our knowledge, these are the first results/techniques on mechanisms with verification for such "multidimensional" scenario (it should be noticed that already the "one-dimensional" case is a generalization of both one-parameter [2] and comparable types [5]). Such results give us powerful tool to solve very general problems. Indeed, we present the first truthful exact mechanism with verification for scheduling unrelated machines when agents control *more than one machine* (thus generalizing the "one machine per agent" scenario/results in [19]). These exact mechanisms (and in general those obtained with our technique above) could not run in polynomial time. We thus move our attention towards approximation polynomial-time mechanisms, and investigate the implementation of classical approximation algorithms. In particular we consider compound agents in which each "dimension" is a one-parameter valuation. In this setting, we show that any approximation algorithm for a smooth problem (Def. 8) can be transformed into a truthful mechanism (with verification) for the problem. The resulting mechanism (essentially) preserves the approximation ratio of the algorithm. In order to guarantee a polynomial running time, we require a constant number of compound agents with constant dimensions (Th. 6). The most relevant application is a polynomial-time $c(1+\varepsilon)$-approximation mechanism for scheduling related machines when the agents control *more than one machine* (see Def. 9) (given a c-approximating algorithm for the problem). To the best of our knowledge, no solution was known for this natural extension of the problem studied in [2]. The assumption of finite domains deserves some more discussions. From a practical point of view, many real-life applications involve agents with types from a finite and discrete domain (e.g., when costs or execution times are expressed as multiples of a given monetary or time unit, and an upper bound is known). From a theoretical point of view, it is interesting to investigate how to use verification to overcome impossibility results proved for infinite domains (with no verification allowed). The only case in which the assumption of finite domains has been removed is for problems involving one-parameter agents (see [11,4]). In Sec. 4 we study truthful mechanisms with verification for infinite domains. We show that known techniques, developed for mechanisms with no verification, seem to "cancel" the advantages given by the verification. Besides one-parameter agents, no result (neither positive nor negative) was known on the design of mechanism with verification for infinite domains. We stress that for finite domain several impossibility results for mechanisms without verification are known [2,7], some of them applying to our optimization functions in the "multi-dimensional" scenario. This shows that verification does help for finite domains. Due to lack of space we omit some proofs. These proofs can be found in the full version of the paper [24].

Related Works. Affine maximizers (see above) can be implemented for quasi-linear utility functions (i.e., payment received plus agent's monetary valuation) using the celebrated VCG mechanism [25,12,9]. Roberts [21] showed that VCG mechanisms are essentially the only truthful mechanisms if no hypothesis is made on the domains of the agents. Mechanisms for *one-parameter* agents have been characterized in [18,2]. For one-parameter agent domain there exists truthful mechanisms for scheduling to minimize the makespan [2,3,1] and for some types of combinatorial auctions [15]. Lavi, Mu'alem and Nisan showed that a *weak monotonicity* condition (W-MON) characterizes *order-based* domains with range constraints [14]. Similar results hold for *linear inequality constraints* on the domain [13] and, more in general, for *convex domains* [23] (each class extending the prior one and the result for one-parameter agents). These results concern mechanisms which do *not* use verification and cannot be applied to our case (indeed, one wishes to use mechanisms with verification to solve problems which the other mechanisms cannot solve [19,4,5]). The study of social choice functions implementable with verification was started by Nisan and Ronen [19], who gave a truthful $(1+\varepsilon)$-approximate mechanism for scheduling on (a constant number) of unrelated machines to minimize the make-span. Similar results have been obtained by Auletta *et al* [4] for scheduling on any number of related machines. These results are based on a characterization of mechanisms with verification [4] for one-parameter agents. Mechanisms with verification for one-parameter agents also appear in [11] where the main contribution is in providing payment schemes, computable in constant time, working with infinite domains. These schemes have the advantage of not conditioning execution time and approximation ratio of mechanisms using them. As already mentioned, a recent work [5] characterizes mechanisms with verification for a rich class of finite domains. Afterwards, they extend their result to the class of one-parameter agents providing different mechanisms for several scheduling problems. The work [8] presents a general technique for constructing polynomial-time approximation mechanisms for *utilitarian* problems (those affine maximizers in which one seeks to maximize the social welfare, i.e., sum of the valuation of the agents). The technique they use is similar to the one we use to obtain truthful mechanisms for multidimensional agents. Similar methods were already used in [5]. This approach derives from [17], where agents' types are even simpler than the one-parameter case (this kind of agents are called KSM bidders). Polynomial-time mechanisms which approximate the social welfare for certain auctions are given in [10]. Mechanisms in [8,10] do not use verification, but all problems are utilitarian and solutions rely on VCG mechanisms.

Preliminaries. We have a finite set $\mathcal{O} = \{X_1, \ldots, X_K\}$ of K possible *alternative outcomes*. We also have m selfish agents, each of them having a *valuation* (or *type*) $v_i \in D_i$, with D_i being the *domain* of agent i. Domains are multi-dimensional in the sense of [6]: The domain of v_i is $D_i \subset \mathbf{R}^K$ with the kth coordinate of type v_i being $v_i(X_k)$, this type's utility for outcome X_k (i.e., $v_i = (v_i(X_1), \ldots, v_i(X_K))$).[1] The valuation v_i is known to agent i only. A *social choice function* $A : D \to \mathcal{O}$ maps the agents' valuations into a particular outcome $A(v_1, \ldots, v_m)$, where $D = D_1 \times \cdots \times D_m$ is the domain of function A. A *mechanism* $M = (A, P)$ is a social choice function A augmented with a *payment scheme* $P = (P_1, \ldots, P_m)$, where each P_i is a function $P_i : D \to \mathbf{R}$. The mechanism elicits from each agent its valuation; an agent i can misreport her valuation to any $b_i \in D_i$. The mechanism, on input the *reported* valuations $\mathbf{b} = (b_1, \ldots, b_m)$, selects outcome X as $X = A(\mathbf{b})$ and assigns payment $P_i(\mathbf{b})$ to agent i. The *utility* of agent i, when receiving a payment $P_i(\mathbf{b})$, with valuation v_i is thus $P_i(\mathbf{b}) + v_i(X)$. This kind of utilities are commonly denoted to as *quasi-linear* utilities. We let $b_i \in D_i$ denote the valuation (or type) *reported* by agent i and by $\mathbf{b}_{-i}$ the vector $(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_m)$ of all valuations (or types) reported by the other agents. We stress that both the outcome and the payments depend on the *reported* valuations $\mathbf{b} = (b_1, \ldots, b_m)$. In particular, given $\mathbf{b}_{-i}$, the reported type b_i determines the outcome $A_{\mathbf{b}_{-i}}(b_i) := A(\mathbf{b})$ and the payment $P_i(\mathbf{b})$. For a vector $\mathbf{x} = (x_1, \ldots, x_m)$, we let $\mathbf{x}_{-i}$ denote the vector $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$ and $(y, \mathbf{x}_{-i})$ the vector $(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_m)$; similarly, $D_{-i} := D_1 \times \cdots \times D_{i-1} \times D_{i+1} \times \cdots \times D_m$. A *mechanism with verification* can detect whether $b_i \neq v_i$ if and only if $v_i(A_{\mathbf{b}_{-i}}(b_i)) < b_i(A_{\mathbf{b}_{-i}}(b_i))$; in this case, agent i will *not* receive the associated payment. This verification model generalizes the concept of verification introduced in [19].

Definition 1 ([19]). *A social choice function A is* implementable with verification *if there exists $P = (P_1, \ldots, P_m)$ such that, for all i, all $\mathbf{b}_{-i} \in D_{-i}$ the utility of agent i with type v_i is maximized by reporting $b_i = v_i$.*

---

[1] An alternative, but completely equivalent, way to define these multidimensional valuations is the following: Each valuation v_i is a function $v_i : \mathcal{O} \to \mathbf{R}$ representing a monetary valuation $v_i(X)$ that agent i associates to outcome $X \in \mathcal{O}$. This definition has been used, for example, in [5].

In this case, the pair (A, P) is called a *truthful mechanism with verification.* A different way to read Def. 1 is that there exists $P = (P_1, \ldots, P_m)$ such that, for all $v_i, b_i \in D_i$ and $\mathbf{b}_{-i} \in D_{-i}$, the following inequality holds:

$$v_i(A_{\mathbf{b}_{-i}}(v_i)) + P_i(v_i, \mathbf{b}_{-i}) \geq v_i(A_{\mathbf{b}_{-i}}(b_i)) \tag{1}$$

if $v_i(A_{\mathbf{b}_{-i}}(b_i)) < b_i(A_{\mathbf{b}_{-i}}(b_i))$, and the following inequality holds:

$$v_i(A_{\mathbf{b}_{-i}}(v_i)) + P_i(v_i, \mathbf{b}_{-i}) \geq v_i(A_{\mathbf{b}_{-i}}(b_i)) + P_i(b_i, \mathbf{b}_{-i}) \tag{2}$$

if $v_i(A_{\mathbf{b}_{-i}}(b_i)) \geq b_i(A_{\mathbf{b}_{-i}}(b_i))$. We are interested in social choice functions which are implementable with verification and that optimize some *objective function* $\mu(\cdot)$ which depends on the agent valuations $\mathbf{v} = (v_1, \ldots, v_m)$. For maximization (respectively, minimization) functions, we let $\text{OPT}_\mu(\mathbf{v})$ be $\max_{X \in \mathcal{O}} \mu(X, \mathbf{v})$ (respectively, $\min_{X \in \mathcal{O}} \mu(X, \mathbf{v})$). An outcome $X \in \mathcal{O}$ is an α-approximation of μ for $\mathbf{v} \in D$ if $\max\left\{\frac{\mu(X,\mathbf{v})}{\text{OPT}_\mu(\mathbf{v})}, \frac{\text{OPT}_\mu(\mathbf{v})}{\mu(X,\mathbf{v})}\right\} \leq \alpha$. A social choice function A is α-*approximate* for μ if, for every $\mathbf{v} \in D$, $A(\mathbf{v})$ is an α-approximation for μ and $\mathbf{v}$. We stress that, in this paper we consider social choice functions that are implementable with verification and either optimize or α-approximate a function μ. The approximation only refers to how good the selected outcome is and not to the utilities of the agents (which are always maximized by reporting the true valuation). Recall that we assume domains to have finite cardinality.

Truthful Mechanisms with Verification. For fixed i and $\mathbf{b}_{-i}$, Eq.s 1 and 2 give a system of linear inequalities with unknowns $P^x := P_i(x, \mathbf{b}_{-i})$, for $x \in D_i$. This system of inequalities is compactly encoded by the following graph.

Definition 2 (verification-graph). *Let A be a social choice function. For every i and $\mathbf{b}_{-i} \in D_{-i}$, the* verification-graph $\mathcal{V}(A_{\mathbf{b}_{-i}})$ *has a node for each type in D_i. The set of edges of $\mathcal{V}(A_{\mathbf{b}_{-i}})$ is defined as follows. For every $a, b \in D_i$ add an edge (a, b) whenever the solution $Y = A_{\mathbf{b}_{-i}}(b)$ is such that $a(Y) \geq b(Y)$. The weight of edge (a, b) (if any) is $\delta(a, b) := a(X) - a(Y)$ where $X = A_{\mathbf{b}_{-i}}(a)$.*

The definition of the verification-graph is a modification of the graph introduced in [16] (and after used by [5]) to study the case in which verification is not allowed.

Theorem 1. *A social choice function A is implementable with verification if and only if, for all i and $\mathbf{b}_{-i} \in D_{-i}$, the graph $\mathcal{V}(A_{\mathbf{b}_{-i}})$ does not have negative weight cycles.*

The theorem follows from the observation that the system of linear inequalities involving the payment functions is the linear programming dual of the shortest path problem on the verification-graph. Therefore, a simple application of Farkas lemma shows that the system of linear inequalities has solution if and only if the verification-graph has no negative weight cycle. The same argument has been used for the case in which verification is not allowed albeit of a different graph (see [22,16,13]). Payments computation deserves a last remark. When verification is not allowed the graph is complete and, if the system is feasible, one solution is to set each P^x equal to the length of the shortest path from an arbitrarily chosen root vertex. The fact that the graph is complete implies that all P^x assume a finite value. When verification is allowed the graph is not complete and, possibly, the graph is not completely connected.[2] Thus, setting payments to shortest paths could lead to unbounded payments. However, it is always possible to set unbounded payments to bounded payments satisfying truthfulness conditions. Simple rules (on the existence of ingoing and outgoing edges) ensure to bound payments preserving truthfulness. We remark that these rules can be implemented in polynomial time.

2 $\mathbf{MAX}_{\mu}$ Social Choice Function

In this section we present our technique to obtain truthful mechanisms with verification for any finite domain. We use next social choice function.

Definition 3 ([5]). *Let $\mu(\cdot)$ be any maximization function monotone non-decreasing in each of its arguments $b_1(X), \ldots, b_m(X)$, with $X \in \mathcal{O}$ and $b_i \in D_i$. For any $X_1, \ldots, X_\ell \in \mathcal{O}$, let MAX_μ be the social choice function that, on input $(b_1, \ldots, b_m) \in D$, returns the solution X_j of minimum index that maximizes the value $\mu(b_1(X_j), \ldots, b_m(X_j))$.*

Notice that MAX_μ social choice function uses a precedence relation among outcomes in $\mathcal{O}$. Indeed, when more solutions lead to the same value of the objective function μ, MAX_μ always selects the "minimum" solution. Therefore, it holds the following straightforward fact.

Fact 1. *For any i and any $\mathbf{b}_{-i} \in D_{-i}$, let a and b two valuations in D_i and denote $X = MAX_\mu(a, \mathbf{b}_{-i})$ and $Y = MAX_\mu(b, \mathbf{b}_{-i})$. If $\mu(a(X), \mathbf{b}_{-i}(X)) = \mu(b(Y), \mathbf{b}_{-i}(Y))$ then $X = Y$.*

The MAX_μ function can be also used to minimize a function μ which is non-increasing in each of its arguments: in fact, given such a μ, simply running $\mathrm{MAX}_{-\mu}$ one obtains the minimum of μ. As observed MAX_μ social choice function has been introduced in [5]. In that work it is shown that MAX_μ function is implementable with verification for comparable types.

Definition 4. *Agent i's domain D_i is* comparable *if for any $a, b \in D$ either $a \leq b$ or $b \leq a$ (where $a \leq b$ means that for all $X \in \mathcal{O}$, $a(X) \leq b(X)$).*

Theorem 2 ([5]). *Let $\mu(\cdot)$ be any maximization function monotone non-decreasing in each of its arguments $b_1(X), \ldots, b_m(X)$, with $X \in \mathcal{O}$ and $b_i \in D_i$. MAX_μ is implementable with verification for comparable types.*

[2] This is not the case for comparable types. Indeed, for comparable types, the graph is always connected.

Theorem above summarizes some of the results in [5] in a concise form. Its proof essentially exploits monotonicity of MAX_μ, monotonicity of μ and monotonicity of comparable types. It turns out that, monotonicity of comparable types can be overcome using properties of $\mathcal{V}_{\mathbf{b}_{-i}}(\mathrm{MAX}_\mu)$, and thus MAX_μ is *always* implementable with verification for *any* finite domain. Let us proceed more formally. The main theorem of the section is the following:

Theorem 3. *Let $\mu(\cdot)$ be any maximization function monotone non-decreasing in each of its arguments $b_1(X), \ldots, b_m(X)$, with $X \in \mathcal{O}$ and $b_i \in D_i$, then MAX_μ is implementable with verification.*

Theorem above follows from the following observation: so that a cycle in verification-graph exists then it has to involve valuations mapped by MAX_μ in the same outcome (thus obtaining a cycle of weight 0).

Notice that MAX_μ always maximizes a monotone non-decreasing objective function μ. In other words, a direct consequence of last theorem is the following.

Corollary 1. *Let $\mu(\cdot)$ be any monotone non-decreasing function in each of its arguments $b_1(X), \ldots, b_m(X)$, with $X \in \mathcal{O}$ and $b_i \in D_i$. Then, there exists a social choice function OPT_μ which maximizes $\mu(\cdot)$ and is implementable with verification.*

If the set $\mathcal{O}$ of outcomes is very large, then social choice function MAX_μ could not be efficiently computable. Our next result can be used to derive efficiently-computable social choice functions which approximate the objective function by restricting to a suitable subset of the possible outcomes.

Definition 5 ([5]). *A set $\mathcal{O}' \subseteq \mathcal{O}$ is α-*approximation preserving for *μ if, for every $\mathbf{b} \in D$, the set $\mathcal{O}'$ contains a solution X' which is an α-approximation of μ for $\mathbf{b}$.*

Corollary 2. *Let $\mu(\cdot)$ be any optimization function to maximize. Assume $\mu(\cdot)$ is monotone non-decreasing in its arguments $b_1(X), \ldots, b_m(X)$, with $X \in \mathcal{O}$ and $b_i \in D_i$. For any α-approximation preserving set $\mathcal{O}'$ the social choice function $APX_\mu := MAX_{X \in \mathcal{O}'}\{X\}$ is an α-approximation for μ and is implementable with verification. Moreover, social choice function $APX_\mu(\mathbf{b})$ can be computed in time proportional to the time needed for computing values $\mu(X, \mathbf{b})$, for $X \in \mathcal{O}'$.*

Corollary above is implied by Theorem 3. Last results generalize results presented in [5]. Indeed, they show that the technique works for *any* valuation domain and not just for comparable types. Moreover, these results lead to interesting applications in studying (and solving) problems like Scheduling Unrelated Machines. The valuations we are studying model well the case of scheduling unrelated machines (see [19]). Looking for mechanisms that are truthful but that do not run in polynomial time, then Cor. 1 gives an alternative (to the one proposed in [19]) technique to *optimally* solve the $Q||C_{max}$ problem on unrelated machines. Unfortunately, these mechanisms do not run in polynomial-time (as in [19]) since, in the case of unrelated machines, the number of solutions to examine are exponentially many.

3 Compound Agents

In this section we show that our technique presented in Sec. 2 solves the problem of designing truthful mechanisms with verification for what we call *compound* agents.

Definition 6 (Compound types). *The type $\bar{v} = (v^1, v^2, \ldots, v^d)$ of a d-dimensional compound agent has d components each of which is a single valuation. A* d-dimensional compound type set *is simply the cross product of d type sets and we will denote by $D_i = D_i^1 \times \ldots \times D_i^d$ the type set of agent i. We call, an agent with compound type set, compound agent.*

A solution $X \in \mathcal{O}$ consists of d components (i.e., $X = (X^1, \ldots, X^d)$). The valuation $\bar{v}$ for a given solution X is a function of v^1, v^2, $\ldots$, v^d (e.g., $\bar{v}(X) = v^1(X) + \ldots + v^d(X)$). More specifically, given $f : \mathbf{R}^d \to \mathbf{R}$, $\bar{v}(X) = f(v^1(X), \ldots, v^d(X))$. We assume that the mechanism is able to verify each of the d coordinates independently. This implies that if an agent is caught lying over one of the d components (for example, the agent declared b_1 as first component of its type instead of v_1 and the solution computed by the mechanism X is such that $b_1(X) > v_1(X)$) then the agent receives no payment.[3] We stress that we do not require here that valuation along one dimension is one-parameter, i.e., for all $1 \leq i \leq d$, v^i is *any* valuation in a finite domain defined in Sec. 1. Using a technique similar to the one used in Sec. 2 we are able to prove a theorem similar to Th. 3 for objective functions of the form $\mu(b_1^1(X), \ldots, b_1^d(X), \ldots, b_m^1(X), \ldots, b_m^d(X))$ that are monotone non-decreasing in any of its inputs.

Theorem 4. *Let $\mu(\cdot)$ be any maximization function monotone non-decreasing in each of its arguments $b_1^1(X), \ldots, b_m^d(X)$, with $X \in \mathcal{O}$ and $\bar{b}_i \in D_i$, then MAX_μ is implementable with verification.*

Observe that Th. 4 applies also to the case in which each agent i has (potentially) different dimension d_i. Indeed, slightly changing the definition of compound-verification-graph (edges are added only between pairs of valuations of the same dimension), the proof of theorem above still holds. Obviously, Cor.s 1 and 2 hold also for compound types. Last theorem gives a powerful method to solve the following problem.

Definition 7 (Scheduling Unrelated Compound Machines). *There are n jobs that need to be allocated to M machines. Machines are owned by $m \leq M$ agents. Each agent owns d_i machines, with $\sum_{i=1}^m d_i = M$. Agent i's valuation $\bar{v}_i$ is the vector $(v_i^1, \ldots, v_i^{d_i})$. Each v_i^j is equal to the vector $(v(X_1), \ldots,\ v(X_{M^n}))$*

[3] One could also define mechanisms with verification for compound agents as mechanisms able to verify only the whole valuation. That is, an agent would be able to cheat along as many coordinates as he wants in a way that the d-dimensional reported valuation is undetectable for the mechanism. In this case, the approach of Sec. 2 works: indeed we can consider these agents as ones defined in Sec. 1. But, with such definition, the concept of truthful mechanism would be quite strange.

for some $v \in D_i^j$ *where* $v(X_k)$ *is the opposite of the completion time of the work assigned to* j*th machine controlled by agent* i *by the solution* X_k*. The goal is to minimize the completion time of the last assignment (the makespan).*

Theorem 5. *There exists an exact truthful mechanism with verification for the problem Scheduling Unrelated Compound Machines.*

The theorem above bases on the technique of Th. 4. The resulting mechanism does not run in polynomial-time as the solutions to work on are in exponential number (namely M^n).

Implementing Classical Algorithms for Compound One-Paramater Agents. Next we will consider compound one-parameter agents, that is, compound agents in which each coordinate is a one-parameter valuation (i.e., for agent i and solution X, $v_i(X) = -t_i \cdot w_i(X)$ where t_i is i's type and $w_i(X)$ is the work assigned to i by solution X). Observe that, for one-parameter agents any valuation (as defined in Sec. 1) is in the form $(-t_i \cdot w_i(X_1), \ldots, -t_i \cdot w_i(X_K))$ given the type t_i. Thus any vector is just represented by the type t_i.

We next introduce the class of smooth functions, for which there exists small α-approximation preserving set of outcomes.

Definition 8. *Fix* $\varepsilon > 0$ *and* $\gamma > 1$*. A function* μ *is* (γ, ε)–smooth *if, for any pair of valuations* $\bar{\mathbf{b}}$ *and* $\tilde{\bar{\mathbf{b}}}$ *such that, for any* $1 \leq j \leq d$, $b_i^j \geq \tilde{b}_i^j \geq \gamma b_i^j$ *for* $i = 1, 2, \ldots, m$*, and for all possible outcomes* X*, it holds that* $\mu(X, \bar{\mathbf{b}}) \leq \mu(X, \tilde{\bar{\mathbf{b}}}) \leq (1+\varepsilon) \cdot \mu(X, \bar{\mathbf{b}})$.

For smooth functions μ, on types we are studying, we can transform any α-approximate polynomial-time algorithm A (which is not necessarily implementable with verification) into a social choice function for a constant number of d-dimensional agents which is computable in polynomial-time (with d being constant), implementable with verification and $\alpha(1+\varepsilon)$-approximates μ.

Theorem 6. *Let* A *be a polynomial-time* α*-approximate algorithm for non-decreasing (in each input)* (γ, ε)*-smooth objective function* $\mu(\cdot)$ *to maximize. If the problem involves compound one-parameter agents then, for any* $\varepsilon > 0$*, there exists an* $\alpha(1+\varepsilon)$*-approximate social choice function* $A^\star$ *implementable with verification. If the number of* d*-dimensional agents is constant and* d *is constant,* $A^\star$ *can be computed in polynomial time.*

The proof of Th. 6 bases on the technique given by Corollary 2 on compound types. Moreover, it applies also to the case in which each agent i has (potentially) different dimension d_i. In this more general case, the social choice function $A^\star$ runs in polynomial time if m is constant and $\sum_{i=1}^m d_i = O(1)$. Next, we provide a small but nice application of technique presented in Th. 6. We solve the following problem.

Definition 9 (Scheduling Related Compound Machines). *There are* n *jobs that need to be allocated to* M *machines. Machines are owned by* $m \leq M$ *agents. Each agent owns* d_i *machines, with* $\sum_{i=1}^m d_i = M$*. Each agent* i*'s* *valuation $\bar{v}_i$ is, for each machine j he controls, the opposite of the completion time of the work assigned to machine j. That is, $\bar{v}_i = (v_i^1, \ldots, v_i^{d_i})$ and $v_i^j(\cdot) = -t_j w_j(\cdot)$. The goal is to minimize the completion time of the last assignment.*

We present a polynomial-time truthful mechanism with verification for the problem above when M is constant.

Theorem 7. *Let A be a c-approximating algorithm for the make-span problem. There exists a truthful $c(1+\varepsilon)$-approximating mechanism with verification for the Scheduling Related Compound Machines using algorithm A. When the number of the machines is constant then the mechanism runs in polynomial time.*

4 Mechanisms with Verification for Infinite Domains

It should be clear that verification helps in defining payments. In fact, mechanism is able to fine some kind of (detectable) lie. As stated in Sec. 1, payment is a function going from D to $\mathbf{R}$. It is well known that, for mechanisms without verification, defining payments on $\mathcal{O} \times D_{-i}$ is completely equivalent to our payment functions definition. This consideration leads to the classical technique used, designing mechanisms without verification, for dealing with infinite domains: the use of the so-called *allocation graph* (which we generalize to the verification setting to as allocation-verification-graph). In this section we show that this technique, in general, cancels advantages of verification and, thus, cannot be used, at least tout-court, in the verification setting.

Definition 10 (allocation-verification-graph). *Let A be a social choice function. For every i and $\mathbf{b}_{-i} \in D_{-i}$, the* allocation-verification-graph *$\mathcal{G}(A_{\mathbf{b}_{-i}})$ has a node for each outcome in $\mathcal{O}$. The set of edges of $\mathcal{G}(A_{\mathbf{b}_{-i}})$ is defined as follows. For every $X, Y \in \mathcal{O}$ add an edge (X, Y) if there exists valuations $a, b \in D_i$ such that $A_{\mathbf{b}_{-i}}(a) = X$, $A_{\mathbf{b}_{-i}}(b) = Y$ and $a(Y) \geq b(Y)$ with $X \neq Y \in \mathcal{O}$. The weight of edge (X, Y) (if any) is $\delta(X, Y) := \inf_{a \in \mathcal{R}_X^Y}\{a(X) - a(Y)\}$, where $\mathcal{R}_X^Y = \{a \in \mathcal{R}_X | \exists b \in \mathcal{R}_Y \text{ s.t. } a(Y) \geq b(Y)\}$ with $\mathcal{R}_X = \{a \in D_i | A_{\mathbf{b}_{-i}}(a) = X\}$.*

Definition 11. *An outcome $X \in \mathcal{O}$ is fully reachable w.r.t. a social choice function A, if for any i, any $\mathbf{b}_{-i}$ and any $X \neq Y \in \mathcal{O}$ there exist valuations $a, b \in D_i$ such that $A_{\mathbf{b}_{-i}}(a) = X$ and $A_{\mathbf{b}_{-i}}(b) = Y$ with $b(X) \geq a(X)$.*

Notice that, it is, in general, possible that a social choice function A admits fully reachable outcomes. It is also possible that a social choice function has neutral domains.

Definition 12. *Agent i's domain D_i is neutral w.r.t. a social choice function A if for all $X, Y \in \mathcal{O}$ s.t. (X, Y) belongs to $\mathcal{G}(A_{\mathbf{b}_{-i}})$ it holds that $\inf_{a \in \mathcal{R}_X^Y}\{a(X) - a(Y)\} = \inf_{a \in \mathcal{R}_X}\{a(X) - a(Y)\}$.*

Next theorem states that verification's advantages are canceled (on allocation-verification-graph) for some particular social choice functions.

Theorem 8. *Let A be a social choice function. If any outcome in $\mathcal{O}$ is fully reachable w.r.t. A and any agent domain is neutral w.r.t. A then, using the allocation-verification-graph, A is implementable with verification if and only if A is implementable without verification.*

Acknowledgements. We wish to thank Paolo Penna for helpful discussions.

References

1. N. Andelman, Y. Azar, and M. Sorani. Truthful approximation mechanisms for scheduling selfish related machines. In *Proc. of STACS*, 2005.
2. A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of FOCS*, 2001.
3. V. Auletta, R. De Prisco, P. Penna, and G. Persiano. Deterministic truthful approximation mechanisms for scheduling related machines. In *Proc. of STACS*, 2004.
4. V. Auletta, R. De Prisco, P. Penna, and G. Persiano. The power of verification for one-parameter agents. In *Proc. of ICALP*, 2004.
5. V. Auletta, R. De Prisco, P. Penna, G. Persiano, and C. Ventre. New constructions of mechanisms with verification. In *Proc. of ICALP*, 2006.
6. S. Bikhchandani, S. Chatterji, R. Lavi, A. Mualem, N. Nisan, and A. Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 2006. To appear.
7. D. Bilò, L. Gualà, and G. Proietti. On the existence of truthful mechanisms for the minimum-cost approximated shortest-paths tree problem. Technical report, 2006.
8. P. Briest, P. Krysta, and B. Vöcking. Approximation techniques for utilitarian mechanims design. In *Proc. of STOC*, 2005.
9. E.H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, pages 17–33, 1971.
10. S. Dobzinski, N. Nisan, and M. Schapira. Approximation Algorithms for Combinatorial Auctions with Complement-Free Bidders. In *Proc. of STOC*, 2005.
11. A. Ferrante, G. Parlato, F. Sorrentino, and C. Ventre. Improvements for truthful mechanisms with verifiable one-parameter selfish agents. In *Proc. of WAOA*, 2005.
12. T. Groves. Incentive in Teams. *Econometrica*, 41:617–631, 1973.
13. H. Gui, R. Müller, and R. V. Vohra. Dominant strategy mechanisms with multidimensional types. Technical report, 2004.
14. R. Lavi, A. Mu'Alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *In Proc. of FOCS*, 2003.
15. D. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002.
16. A. Malakhov and R. V. Vohra. Single and multi-dimensional optimal auctions - a network approach. Technical report, 2004.
17. A. Mu'Alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proc. of AAAI*, 2002.
18. R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.
19. N. Nisan and A. Ronen. Algorithmic Mechanism Design. In *Proc. of STOC*, 1999.
20. N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *Proc. of EC*, 2000.

21. K. Roberts. The Characterization of Implementable Choice Rule. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*, 1979.
22. J.-C. Rochet. A condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16:191–200, 1987.
23. M. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proc. of EC*, 2005.
24. C. Ventre. Mechanisms with verification for any finite domain. Technical report, 2006.
25. W. Vickrey. Counterspeculation, Auctions and Competitive Sealed Tenders. *Journal of Finance*, pages 8–37, 1961.

Pure Nash Equilibria in Player-Specific and Weighted Congestion Games*

Heiner Ackermann, Heiko Röglin, and Berthold Vöcking

Department of Computer Science
RWTH Aachen, D-52056 Aachen, Germany
{ackermann, roeglin, voecking}@cs.rwth-aachen.de

Abstract. Unlike standard congestion games, weighted congestion games and congestion games with player-specific delay functions do not necessarily possess pure Nash equilibria. It is known, however, that there exist pure equilibria for both of these variants in the case of *singleton congestion games*, i. e., if the players' strategy spaces contain only sets of cardinality one. In this paper, we investigate how far such a property on the players' strategy spaces guaranteeing the existence of pure equilibria can be extended. We show that both weighted and player-specific congestion games admit pure equilibria in the case of *matroid congestion games*, i. e., if the strategy space of each player consists of the bases of a matroid on the set of resources. We also show that the matroid property is the maximal property that guarantees pure equilibria without taking into account how the strategy spaces of different players are interweaved. In the case of player-specific congestion games, our analysis of matroid games also yields a polynomial time algorithm for computing pure equilibria.

1 Introduction

Congestion games are a natural model for resource allocation in large networks like the Internet. It is assumed that n players share a set $\mathcal{R}$ of m resources. Players are interested in subsets of resources. For example, the resources may correspond to the edges of a graph, and each player may want to allocate a spanning tree of this graph. The delay (cost, negative payoff) of a resource depends on the number of players that allocate the resource, and the delay of a set of allocated resources corresponds to the sum of the delays of the resources in the set. A well known potential function argument of Rosenthal [11] shows that congestion games always possess Nash equilibria[1], i. e., allocations of resources from which no player wants to deviate unilaterally.

The existence of Nash equilibria gives a natural solution concept for congestion games. Unfortunately, this property does not hold anymore if we slightly extend the class of considered games towards congestion games with player-specific delay

* This work was supported in part by the EU within the 6th Framework Programme under contract 001907 (DELIS) and by DFG grant Vo889/2-1.

[1] In this paper, the term *Nash equilibrium* always refers to a pure equilibrium.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 50–61, 2006.
© Springer-Verlag Berlin Heidelberg 2006

functions, i. e., a variant of congestion games in which different players might have different delay functions, and weighted congestion games, i. e., a variant of congestion games in which the delay of a resource depends on a weighted number of players. For both of these classes one can easily construct examples of games that do not possess Nash equilibria (cf. Fotakis et al. [4] in the case of weighted network congestion games). In this paper, we study which conditions on the strategy spaces of individual players guarantee the existence of Nash equilibria. We only consider games with non-decreasing delay functions since otherwise one can construct examples of weighted and player-specific *singleton congestion games*, i. e., games in which the players' strategy spaces contain only sets of cardinality one, that do not possess Nash equilibria.

It is known, however, that there exist pure equilibria for both of these variants in the case of singleton congestion games with non-decreasing delay functions [10,2]. We extend these results and show that both player-specific and weighted congestion games admit pure equilibria in the case of *matroid congestion games*, i. e., if the strategy space of each player consists of the bases of a matroid on the set of resources. We also show that the matroid property is the maximal condition on the players' strategy spaces that guarantees Nash equilibria without taking into account how the strategy spaces of different players are interweaved. In the case of player-specific matroid congestion games, our analysis also yields a polynomial time algorithm for computing pure equilibria. Let us remark that the best response dynamics may cycle for player-specific singleton congestion games [10]. For weighted matroid congestion games we do not have an efficient algorithm for computing a Nash equilibrium, but we show that players playing "lazy best responses" converge to a Nash equilibrium.

Related Work. Milchtaich [10] considers player-specific singleton congestion games and shows that every such game possesses at least one Nash equilibrium. Additionally, he shows that players iteratively playing best responses in such games do not necessarily reach a Nash equilibrium, that is, the best response dynamics may cycle. However, he implicitly describes an algorithm for computing an equilibrium. Our work generalizes Milchtaich's analysis from singleton congestion games towards matroid congestion games. Gairing et al. [6] consider player-specific singleton congestion games with linear delay functions without offsets and show that the best response dynamics of these games do not cycle anymore. Milchtaich [10] also addresses the existence of Nash equilibria in congestion games which are both player-specific and weighted. In this case, a Nash equilibrium does not necessarily exist in singleton congestion games. However, Georgiou et al. [7] and Garing et al. [6] conjecture that these games possess Nash equilibria in the case of linear player-specific delay functions without offsets.

Even-Dar et al. [2] consider a load balancing scenario with weighted jobs. They show that in this scenario at least one Nash equilibrium always exists and that players iteratively playing best responses converge to such an equilibrium. A similar result can also be found in [10] and [3]. Our proof that every weighted matroid congestion game possesses at least one Nash equilibrium reworks the proof in [2]. Even-Dar et al. [2] also consider the convergence time in the case of unrelated, related, and identical machines, and different types of job weights. They show that players do not necessarily converge quickly in any of these scenarios. Fotakis et al. [4] consider weighted network congestion games in which the strategy space of each player corresponds to the set of all paths between possibly different sources and sinks in a network. First they show that a Nash equilibrium does not necessarily exist. However, they are able to show that in the case of l-layered networks with delays equal to the congestion every weighted network congestion game possesses at least one Nash equilibrium. This shows that if we consider more than the combinatorial structure of the strategy spaces of the players, then one can identify larger classes of weighted congestion games possessing Nash equilibria.

It is interesting to relate the results about the existence of Nash equilibria in player-specific and weighted matroid congestion games to our recent work about the convergence time of standard congestion games: In [1] we characterize the class of congestion games that admit polynomial time convergence to a Nash equilibrium. Motivated by the fact that in singleton congestion games players converge quickly [9], we show that if the strategy space of each player consists of the bases of a matroid on the set of resources, then players iteratively playing best responses reach a Nash equilibrium quickly. Furthermore, we show that the matroid property is a necessary and sufficient condition on the players' strategy spaces for guaranteeing polynomial time convergence to a Nash equilibrium if one does not take into account the global structure of the game.

Formal Definition of Congestion Games. A *congestion game* Γ is a tuple $(\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i\in\mathcal{N}}, (d_r)_{r\in\mathcal{R}})$ where $\mathcal{N} = \{1, \ldots, n\}$ denotes the set of players, $\mathcal{R} = \{1, \ldots, m\}$ the set of resources, $\Sigma_i \subseteq 2^{\mathcal{R}}$ the strategy space of player i, and $d_r : \mathbb{N} \to \mathbb{N}$ a delay function associated with resource r. We call a congestion game *symmetric* if all players share the same set of strategies, otherwise we call it *asymmetric*. We denote by $S = (S_1, \ldots, S_n)$ the *state of the game* where player i plays strategy $S_i \in \Sigma_i$. Furthermore, we denote by $S \oplus S'_i$ the state $S' = (S_1, \ldots, S_{i-1}, S'_i, S_{i+1}, \ldots, S_n)$, i. e., the state S except that player i plays strategy S'_i instead of S_i. For a state S, we define the *congestion* $n_r(S)$ on resource r by $n_r(S) = |\{i \mid r \in S_i\}|$, that is, $n_r(S)$ is the number of players sharing resource r in state S. Players act selfishly and like to play a strategy $S_i \in \Sigma_i$ minimizing their individual delay. The delay $\delta_i(S)$ of player i in state S is given by $\delta_i(S) = \sum_{r\in S_i} d_r(n_r(S))$. Given a state S, we call a strategy S^*_i a *best response* of player i to S if, for all $S'_i \in \Sigma_i$, $\delta_i(S \oplus S^*_i) \leq \delta_i(S \oplus S'_i)$. Furthermore, we call a state S a *Nash equilibrium* if no player can decrease her delay by changing her strategy, i. e., for all $i \in \mathcal{N}$ and for all $S'_i \in \Sigma_i$, $\delta_i(S) \leq \delta_i(S \oplus S'_i)$. Rosenthal [11] shows that every congestion game possesses at least one Nash equilibrium by considering the potential function $\phi : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{N}$ with $\phi(S) = \sum_{r\in\mathcal{R}} \sum_{i=1}^{n_r(S)} d_r(i)$.

There are two well known extensions of congestion games, namely *player-specific congestion games* and *weighted congestion games*. In a player-specific congestion game every player i has its own delay function $d^i_r : \mathbb{N} \to \mathbb{N}$ for every resource $r \in \mathcal{R}$. Given a state S, the delay of player i is defined as $\delta_i(S) = \sum_{r \in S_i} d_r^i(n_r(S))$. In a weighted congestion game every player $i \in \mathcal{N}$ has a weight $\omega_i \in \mathbb{N}$. Given a state S, we define the congestion on resource r by $n_r(S) = \sum_{i: r \in S_i} \omega_i$, that is, $n_r(S)$ is the weight of all players sharing resource r in state S.

Matroids and Matroid Congestion Games. We now introduce *matroid congestion games*. Before we give a formal definition of such games we shortly introduce matroids. For a detailed discussion we refer the reader to [12].

Definition 1. *A tuple $\mathcal{M} = (\mathcal{R}, \mathcal{I})$ is a matroid if $\mathcal{R} = \{1, \ldots, m\}$ is a finite set of resources and $\mathcal{I}$ is a nonempty family of subsets of $\mathcal{R}$ such that, if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$, and, if $I, J \in \mathcal{I}$ and $|J| < |I|$, then there exists an $i \in I \setminus J$ with $J \cup \{i\} \in \mathcal{I}$.*

Let $I \subseteq \mathcal{R}$. If $I \in \mathcal{I}$, then we call I an *independent set*, otherwise we call it *dependent*. It is well known that all maximal independent sets of $\mathcal{I}$ have the same cardinality. The *rank* $rk(\mathcal{M})$ of the matroid is the cardinality of the maximal independent sets. A maximal independent set B is called a *basis* of $\mathcal{M}$. In the case of a weight function $w : \mathcal{R} \to \mathbb{N}$, we call a matroid *weighted*, and seek to find a basis of minimum weight, where the weight of an independent set I is given by $w(I) = \sum_{r \in I} w(r)$. It is well known that such a basis can be found by a greedy algorithm. Now we are ready to define matroid congestion games.

Definition 2. *We call a congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$ a* matroid congestion game *if for every player $i \in \mathcal{N}$, $\mathcal{M}_i := (\mathcal{R}, \mathcal{I}_i)$ with $\mathcal{I}_i = \{I \subseteq S \mid S \in \Sigma_i\}$ is a matroid and Σ_i is the set of bases of M_i. Additionally, we denote by $rk(\Gamma) = \max_{i \in \mathcal{N}} rk(\mathcal{M}_i)$ the* rank of a matroid congestion game *Γ.*

The obvious application of matroid congestion games are network design problems in which players compete for the edges of a graph in order to build a spanning tree [13]. There are quite a few more interesting applications as even simple matroid structures like uniform matroids, that are rather uninteresting from an optimization point of view, lead to rich combinatorial structures when various players with possibly different strategy spaces are involved. Illustrative examples based on uniform matroids are market sharing games with uniform market costs [8] and scheduling games in which each player has to injectively allocate a given set of tasks (services) to a given set of machines (servers).

Let us remark that, in the case of matroid congestion games, the assumption that all delays are positive is not a restriction. Since all strategies have the same size, one can easily shift all delays by the same value in order to obtain positive delays without changing the better and best response dynamics.

2 Player-Specific Matroid Congestion Games

In this section, we consider player-specific matroid congestion games with non-decreasing player-specific delay functions and prove that every such game possesses at least one Nash equilibrium. Moreover, the proof we present implicitly describes an efficient algorithm to compute an equilibrium.

Theorem 3. *Every player-specific matroid congestion game Γ with non-decreasing delay functions possesses at least one Nash equilibrium.*

Proof. Recall that since the strategy space of player i corresponds to the set of bases of a matroid $\mathcal{M}_i$, all strategies of player i have the same size $rk(\mathcal{M}_i)$. In the following, we represent a strategy of player i by $rk(\mathcal{M}_i)$ tokens that the player places on the resources she allocates. Suppose that we reduce the number of tokens of some of the players, that is, player i has $k_i \leq rk(\mathcal{M}_i)$ tokens that she places on the resources of an independent set of cardinality k_i. Observe that the independent sets of cardinality k_i form the bases of a matroid $\mathcal{M}'_i$ whose independent sets correspond to those independent sets of $\mathcal{M}_i$ with cardinality at most k_i. Hence, a game in which some of the players have a reduced number of tokens is also a matroid congestion game.

We prove the theorem by induction on the total number of tokens $\tau = \sum_{i\in\mathcal{N}} rk(\mathcal{M}_i)$ that the players are allowed to place, that is, we prove the existence of Nash equilibria for a sequence of games $\Gamma_0, \Gamma_1, \ldots \Gamma_\tau$, where $\Gamma_{\ell+1}$ is obtained from Γ_ℓ by giving one more token to one of the players. Γ_0 is the game in which each player has only the empty strategy. Obviously, Γ_0 has only one state and this state is a Nash equilibrium.

As induction hypothesis assume that player i has placed $k_i \geq 0$ tokens, for $1 \leq i \leq n$, and this placement corresponds to a Nash equilibrium of the player-specific matroid congestion game $\Gamma_\ell = (\mathcal{N}, \mathcal{R}, (\Sigma_i^{k_i})_{i\in\mathcal{N}}, (d_r^i)_{i\in\mathcal{N}, r\in\mathcal{R}})$ with $\ell = \sum_{i\in\mathcal{N}} k_i$, in which the set of strategies $\Sigma_i^{k_i}$ coincides with the set of independent sets of size k_i of the matroid $\mathcal{M}_i$.

Now assume that some player i_0 has to place an additional token t_0. We show how to compute a Nash equilibrium for the game $\Gamma_{\ell+1}$ obtained from a Nash equilibrium of Γ_ℓ by changing i_0's strategy space to the set of independent sets of size $k_{i_0}+1$. Due to the greedy property of matroids, there exists a resource r_0 such that placing the token t_0 on r_0 gives an independent set of size $k_{i_0}+1$ with minimum delay among all independent sets of the same size. Thus, assuming that the tokens of the other players are fixed, an optimal strategy for player i_0 is to place t_0 on r_0 and leave all other tokens unchanged. However, as the congestion on r_0 is increased by one, other players might want to move their tokens from r_0 in order to obtain a better independent set. We now use matroid properties to show that a Nash equilibrium of $\Gamma_{\ell+1}$ can be reached with only $n \cdot m \cdot rk(\Gamma)$ moves of tokens.

Lemma 4. *Let $\mathcal{M}$ be a weighted matroid and B_{opt} be a basis of $\mathcal{M}$ with minimum weight. If the weight of a single resource $r_{opt} \in B_{opt}$ is increased such that B_{opt} is no longer of minimum weight, then, in order to obtain a minimum weight basis again, it suffices to exchange r_{opt} with a resource r^* of minimum weight such that $B_{opt} \cup \{r^*\} \setminus \{r_{opt}\}$ is a basis.*

Proof. In order to prove the lemma we use the following property of a matroid $\mathcal{M} = (\mathcal{R}, \mathcal{I})$. Let $I, J \in \mathcal{I}$ with $|I| = |J|$ be independent sets. Consider the bipartite graph $G(I \Delta J) = (V, E)$ with $V = (I \setminus J) \cup (J \setminus I)$ and $E = \{\{i, j\} \mid$

$i \in I \setminus J, j \in J \setminus I : I \cup \{j\} \setminus \{i\} \in \mathcal{I}\}$. It is well known that $G(I\Delta J)$ contains a perfect matching (cf. Lemma 39.12(a) from [12]).

Let B'_{opt} be a minimum weight basis w. r. t. the increased weight of r_{opt}. Let P be a perfect matching of the graph $G(B_{opt}\Delta B'_{opt})$ and denote by e the edge from P that contains r_{opt}. For every edge $\{r, r'\} \in P \setminus \{e\}$, it holds $w(r) \leq w(r')$ as, otherwise, if $w(r) > w(r')$, the basis $B_{opt} \cup \{r'\} \setminus \{r\}$ would have smaller weight than B_{opt}.

Now denote by r'_{opt} the resource that is matched with r_{opt}, i. e., the resource such that $e = \{r_{opt}, r'_{opt}\} \in P$. As $w(r) \leq w(r')$ for every $\{r, r'\} \in P \setminus \{e\}$, the weight of $B_{opt} \setminus \{r_{opt}\}$ is bounded from above by the weight of $B'_{opt} \setminus \{r'_{opt}\}$. By the definition of the matching P, $B_{opt} \cup \{r'_{opt}\} \setminus \{r_{opt}\}$ is a basis. By our arguments above, the weight of this basis is bounded from above by the weight of B'_{opt} . Hence, this basis is optimal w. r. t. the increased weight of r_{opt}. □

After placing token t_0 of player i_0 on resource r_0, resource r_0 has one additional token in comparison to the initial Nash equilibrium S of the game Γ_ℓ. Since we assume non-decreasing delay functions, only the players with a token on r_0 might now have an incentive to change their strategies. Let i_1 be one of these players. It follows from Lemma 4 that i_1 has a best response in which she moves a token t_1 from resource r_0 to another resource that we call r_1. Now r_1 is the only resource with one additional token in comparison to S. Suppose we have not yet reached a Nash equilibrium. Only those players with a token on r_1 might have an incentive to change their strategies. Again applying Lemma 4, we can identify a player i_2 that has a best response in which she moves a token t_2 from r_1 to a resource r_2, which then is the only resource with one additional token.

The token migration process described above can be continued in the same way until it reaches a Nash equilibrium of the game $\Gamma_{\ell+1}$. The correctness of the process is ensured by the following invariant.

Invariant 1. *For every $j \geq 0$, after player i_j moves token t_j onto resource r_j,*

a) only players with a token on r_j might violate the Nash equilibrium condition,
b) the Nash equilibrium condition of all players would be satisfied if one ignores the additional token on r_j, that is, if each player calculates the delay on r_j as if there would be one token less on this resource.

The invariant follows by induction on j: For player i_j the invariant is satisfied as this player plays a best response according to Lemma 4. Thus she satisfies the Nash equilibrium condition even without virtually reducing the congestion on r_j. For all other players, the validity of the invariant for j follows directly from the validity of the invariant for $j - 1$ as these players do not move their tokens.

Thus, in order to show the existence of a Nash equilibrium for $\Gamma_{\ell+1}$, it suffices to show that the token migration process is finite. Consider an arbitrary token t of any player i. For a resource r, let $D_i(r)$ denote the delay of i on r if r has one more token than in the initial state S. Observe, whenever t is moved by the migration process from a resource r to a resource r' then $D_i(r) > D_i(r')$. Hence, the token t can visit each resource at most once during the token migration process. As there are at most $n \cdot rk(\Gamma)$ tokens, the migration process terminates after at most $n \cdot m \cdot rk(\Gamma)$ steps in a Nash equilibrium of $\Gamma_{\ell+1}$. □

The proof of Theorem 3 implicitly describes an efficient algorithm to compute a Nash equilibrium with at most $n^2 \cdot m \cdot rk^2(\Gamma)$ moves of tokens.

Corollary 5. *There exists a polynomial time algorithm to compute a Nash equilibrium of a player-specific matroid congestion game with non-decreasing player-specific delay functions.*

3 Weighted Matroid Congestion Games

In this section we consider weighted matroid congestion games with non-decreasing delay functions and show that every such game possesses at least one Nash equilibrium. Moreover, we show that players find such an equilibrium if they iteratively play *"lazy best responses"*. Formally, given a state S we call a best response S_i^* of player i *lazy* if it can be decomposed into a sequence of strategies $S_i = S_i^0, S_i^1, \ldots, S_i^k = S_i^*$ with $|S_i^{j+1} \setminus S_i^j| = 1$ and $\gamma_i(S \oplus S_i^{j+1}) < \gamma_i(S \oplus S_i^j)$ for $0 \leq j < k$. The existence of such a best response is guaranteed since given a weighted matroid $\mathcal{M} = (\mathcal{R}, \mathcal{I})$, a basis $B \in \mathcal{I}$ is an optimal basis of $\mathcal{M}$ if and only if there exists no basis $B^* \in \mathcal{I}$ with $|B \setminus B^*| = 1$ and $w(B^*) < w(B)$ (cf. Lemma 39.12(b) from [12]). In particular, a best response which exchanges the least number of resources compared to the current strategy S_i is a lazy best response.

Theorem 6. *Every weighted matroid congestion game Γ with non-decreasing delay functions possesses at least one Nash equilibrium which is reached after a finite number of lazy best responses.*

Proof. Let S be a state of Γ. With each resource r, we associate a pair $z_r(S) = (d_r(n_r(S)), n_r(S))$ consisting of the delay and the congestion of r in state S. For two resources r and r' and states S and S', let $z_r(S) \geq z_{r'}(S')$ iff $d_r(n_r(S)) > d_{r'}(n_{r'}(S'))$ or $d_r(n_r(S)) = d_{r'}(n_{r'}(S'))$ and $n_r(S) \geq n_{r'}(S')$. Let $z_r(S) > z_{r'}(S')$ iff $z_r(S) \geq z_{r'}(S')$ and $z_r(S) \neq z_{r'}(S')$. Let $\bar{z}(S)$ denote a vector containing the pairs $z_r(S)$ of all resources $r \in \mathcal{R}$ in non-increasing order, that is, $\bar{z}_j(S) \geq \bar{z}_{j+1}(S)$, where $\bar{z}_j(S)$ denotes the j-th component of $\bar{z}$, for $1 \leq j < |\mathcal{R}|$.

We denote by $\leq_{\text{lex}}$ the lexicographic order among the vectors $\bar{z}(S)$, i. e., $\bar{z}(S_1) \leq_{\text{lex}} \bar{z}(S_2)$ if there exists an index l such that $\bar{z}_k(S_1) = \bar{z}_k(S_2)$, for all $k \leq l$, and $\bar{z}_l(S_1) \leq \bar{z}_l(S_2)$. Additionally, we define $\bar{z}(S_1) <_{\text{lex}} \bar{z}(S_2)$ if $\bar{z}(S_1) \leq_{\text{lex}} \bar{z}(S_2)$ and $\bar{z}(S_1) \neq \bar{z}(S_2)$.

Now given a state S, let player i play a lazy best response S_i^*. Since S_i^* is a lazy best response, there exists a sequence of strategies $S_i = S_i^0, \ldots, S_i^k = S_i^*$ such that, for every $0 \leq j < k$, $|S_i^{j+1} \setminus S_i^j| = 1$ and

$$\gamma_i(S) = \gamma_i(S \oplus S_i^0) > \gamma_i(S \oplus S_i^1) > \ldots > \gamma_i(S \oplus S_i^k) = \gamma_i(S \oplus S_i^*) \ .$$

We claim that $\bar{z}(S \oplus S_i^{j+1}) <_{\text{lex}} \bar{z}(S \oplus S_i^j)$, for every $0 \leq j < k$. Let r_j be the unique resource in S_i^j that is not contained in S_i^{j+1} and let r_j^* be the resource that is contained in S_i^{j+1} but not in S_i^j. Since the delay decreases strictly with the exchange, we have

$$d_{r_j}(n_{r_j}(S \oplus S_i^j)) > d_{r_j^*}(n_{r_j^*}(S \oplus S_i^{j+1})) \ .$$

Additionally, since we assume non-decreasing delay functions,

$$d_{r_j}(n_{r_j}(S \oplus S_i^j)) \geq d_{r_j}(n_{r_j}(S \oplus S_i^{j+1})) = d_{r_j}(n_{r_j}(S \oplus S_i^j) - \omega_i) \ .$$

Furthermore, $n_{r_j}(S \oplus S_i^j) > n_{r_j}(S \oplus S_i^{j+1})$. Combining these inequalities implies $z_{r_j}(S \oplus S_i^j) > z_{r_j}(S \oplus S_i^{j+1})$ and $z_{r_j}(S \oplus S_i^j) > z_{r_j^*}(S \oplus S_i^{j+1})$. Combined with the observation that $z_{r_j}(S \oplus S_i^j) > z_{r_j^*}(S \oplus S_i^j)$, this yields $\bar{z}(S \oplus S_i^j) >_{\text{lex}} \bar{z}(S \oplus S_i^{j+1})$, that is, the lexicographic order decreases with every exchange and, hence, with every lazy best response. This concludes the proof of the theorem. □

In the full version of this paper we show that playing lazy best responses is a necessary assumption in order to obtain convergence to a Nash equilibrium, that is, we present a weighted matroid congestion game in which the best response dynamic cycles if players are not restricted to lazy best responses. The delay functions in this congestion game are non-decreasing but not strictly increasing. We leave open the questions whether players playing arbitrary best responses converge to a Nash equilibrium if each delay function is strictly increasing and whether there is an efficient algorithm for computing a Nash equilibrium in weighted matroid congestion games in general. To the best of our knowledge the only positive result is known in the case of weighted singleton matroid congestion games with identical resources, i. e., all resources have identical, non-decreasing delay functions. In this case, Gairing et al. [5] show how to compute a Nash equilibrium in polynomial time. If additionally the players are symmetric, Even-Dar et al. [2] show that if one assigns the players in non-increasing order of their weights to the resources, then the resulting assignment is a Nash equilibrium.

Finally, we like to comment on the convergence time. Theorem 6 implies that players iteratively playing lazy best responses reach a Nash equilibrium after at most $\min\left\{\left(\sum_{i=1}^n \omega_i\right)^m, \binom{m}{rk(\Gamma)}^n\right\}$ strategy changes. The first term is an upper bound on the maximal number of different vectors $\bar{z}(S)$ and the second one bounds the number of different states of a matroid congestion game. Even-Dar et al. [2] establish an exponential lower bound in the case of weighted singleton congestion games with symmetric players and identical resources. However, they use exponentially large weights to show this. In the full version of this paper we present an infinite family of weighted singleton congestion games possessing superpolynomially long best response sequences although every player has either weight one or two and all delays are polynomially bounded in the number of players and resources. This immediately implies that players do not necessarily reach a Nash equilibrium in pseudopolynomial time.

4 Non-matroid Strategy Spaces

In this section, we show that the matroid property is the maximal property on the individual players' strategy spaces that guarantees the existence of Nash equilibria in player-specific and weighted congestion games with non-decreasing (player-specific) delay functions. For this, let Σ be a set system over a set $\mathcal{R}$ of resources. We call Σ *inclusion-free* if for every $X \in \Sigma$, no proper superset $Y \supset X$ belongs to Σ. Moreover, we call Σ a *non-matroid set system* if the tuple $(\mathcal{R}, \{X \subseteq S \mid S \in \Sigma\})$ is not a matroid. In [1] we show that every inclusion-free, non-matroid set system possesses the $(1,2)$-exchange property. Here we need a variant of this property with positive (instead of non-negative) delays.

Definition 7 (**$(1,2)$-exchange property**). *Let Σ be an inclusion-free set system over a set of resources $\mathcal{R}$. We say that Σ satisfies the* $(1,2)$-exchange property *if we can identify three distinct resources $a, b, c \in \mathcal{R}$ with the property that for any given $k \in \mathbb{N}$ with $k > |\mathcal{R}|$, we can choose a delay $d(r) \in \{1, k + |\mathcal{R}|\}$ for every $r \in \mathcal{R} \setminus \{a, b, c\}$ such that for every choice of the delays of a, b, and c with $|\mathcal{R}| \le d(a), d(b), d(c) \le k$, the following property is satisfied: If $d(a) + |\mathcal{R}| \le d(b) + d(c)$, then for every set $S \in \Sigma$ with minimum delay, $a \in S$ and $b, c \notin S$. If $d(a) \ge d(b) + d(c) + |\mathcal{R}|$, then for every set $S \in \Sigma$ with minimum delay, $a \notin S$ and $b, c \in S$.*

Lemma 8. *Let Σ be an inclusion-free set system over a set of resources $\mathcal{R}$. Furthermore, let $\mathcal{I} = \{X \subseteq S \mid S \in \Sigma\}$, and assume that $(\mathcal{R}, \mathcal{I})$ is not a matroid, i. e., that Σ is not the set of bases of some matroid. Then Σ possesses the* $(1,2)$-*exchange property.*

Proof. Since $(\mathcal{R}, \mathcal{I})$ is not a matroid, there exist two sets $X, Y \in \Sigma$ and a resource $x \in X \setminus Y$ such that for every $y \in Y \setminus X$, the set $X \setminus \{x\} \cup \{y\}$ is not contained in Σ (cf. Theorem 39.6 from [12]).

Let X and Y be such sets and let $x \in X$ be such a resource. Consider all subsets Y' of the set $X \cup Y \setminus \{x\}$ with $Y' \in \Sigma$. Every such set Y' can be written as $Y' = X \setminus \{x = x_1, \ldots, x_l\} \cup \{y_1, \ldots, y_{l'}\}$ with $x_i \in X \setminus Y$ and $y_i \in Y \setminus X$ and $l + l' > 2$. This is true since l as well as l' are both larger than 0 as Σ is inclusion-free. Furthermore l and l' cannot both equal 1 as otherwise we obtain a contradiction to the choice of X, Y, and x. Among all these sets Y', let $Y_{\min}$ denote one set for which l' is minimal. Observe that we can replace Y by $Y_{\min}$ without changing the aforementioned properties of X, Y, and x. Hence, in the following, we assume that $Y = Y_{\min}$, that is, we assume that $Y \setminus X = Y' \setminus X$ for all of the aforementioned sets Y'.

We claim that we can always identify resources $a, b, c \in X \cup Y$ such that either $a \in X \setminus Y$ and $b, c \in Y \setminus X$ or $a \in Y \setminus X$ and $b, c \in X \setminus Y$ with the property that for every $Z \subseteq X \cup Y$ with $Z \in \Sigma$, if $a \notin Z$, then $b, c \in Z$. In order to see this, we distinguish between the cases $l' = 1$ and $l' \ge 2$:

1. Let $Y \setminus X = \{y_1\}$ and hence $X \setminus Y = \{x = x_1, \ldots, x_l\}$ with $l \ge 2$. Then we set $a = y_1$, $b = x_1$, and $c = x_2$. Consider a set $Z \subseteq X \cup Y$ with $Z \in \Sigma$ and $a \notin Z$. Then $Z = X$ since Σ is inclusion-free, and hence $b, c \in Z$.

2. Let $Y \setminus X = \{y_1, \ldots, y_{l'}\}$ with $l' \geq 2$. Then we set $a = x$, $b = y_1$, and $c = y_2$. Consider a set $Z \subseteq X \cup Y$ with $Z \in \Sigma$ and $a \notin Z$. Since we assumed that $Y = Y_{\min}$, it must be $b, c \in Z$ as otherwise $Z \setminus X \neq Y \setminus X$.

Now we define delays for the resources in $\mathcal{R} \setminus \{a, b, c\}$ such that the properties in Definition 7 are satisfied. Let $k \in \mathbb{N}$ be chosen as in Definition 7, that is, $d(a), d(b), d(c) \in \{|\mathcal{R}|, \ldots, k\}$. We set $d(r) = k + |\mathcal{R}|$ for every resource $r \notin X \cup Y$ and $d(r) = 1$ for every resource $r \in (X \cup Y) \setminus \{a, b, c\}$. First of all, observe that in the first case the delay of Y equals $d(a) + |Y| - 1 < k + |\mathcal{R}|$ and that in the second case the delay of X equals $d(a) + |X| - 1 < k + |\mathcal{R}|$. Hence, a set $Z \in \Sigma$ that contains a resource $r \notin X \cup Y$ can never have minimum delay as its delay is at least $k + |\mathcal{R}|$. Thus, only sets $Z \in \Sigma$ with $Z \subseteq X \cup Y$ can have minimum delay. Since for such sets, $a \notin Z$ implies $b, c \in Z$, we know that every set with minimum delay must contain a or it must contain b and c.

Consider the case $d(a) + |\mathcal{R}| \leq d(b) + d(c)$ and assume for contradiction that there exists an optimal set Z^* with $a \notin Z^*$. Due to the choice of a, b, and c, the set Z^* must then contain b and c. Hence $d(Z^*) \geq d(b) + d(c)$. Furthermore, again due to the choice of a, b, and c, there exists a set $Z' \subseteq X \cup Y$ with $a \in Z'$ and $b, c \notin Z'$. The delay of Z' is $d(Z') = d(a) + |Z'| - 1 < d(a) + |\mathcal{R}| \leq d(b) + d(c) \leq d(Z^*)$, contradicting the assumption that Z^* has minimum delay. Hence every optimal set Z^* must contain a. If Z^* additionally contains b or c, then its delay is at least $d(a) + |\mathcal{R}| > d(Z')$. Hence, in the case $d(a) + |\mathcal{R}| \leq d(b) + d(c)$ every optimal set Z^* contains a but it does not contain b and c.

Consider the case $d(a) \geq d(b) + d(c) + |\mathcal{R}|$ and assume for contradiction that there exists an optimal set Z^* with $b \notin Z^*$ or $c \notin Z^*$. Then Z^* must contain a and hence its delay is at least $d(a)$. Due to the choice of a, b, and c, there exists a set $Z' \subseteq X \cup Y$ with $a \notin Z'$ and $b, c \in Z'$. The delay of Z' is $d(Z') = d(b) + d(c) + |Z'| - 2 < d(b) + d(c) + |\mathcal{R}| \leq d(a) \leq d(Z^*)$, contradicting the assumption that Z^* has minimum delay. Hence every optimal set Z^* must contain b and c. If Z^* additionally contains a, then its delay is at least $d(b) + d(c) + |\mathcal{R}| > d(Z')$. Hence, in the case $d(a) \geq d(b) + d(c) + |\mathcal{R}|$ every optimal set Z^* contains b and c but it does not contain a. □

Theorem 9. *For every inclusion-free, non-matroid set system Σ over a set of resources $\mathcal{R}$ there exists a weighted congestion game Γ with two players whose strategy spaces are isomorphic to Σ that does not possess a Nash equilibrium. The delay functions in Γ are positive and non-decreasing.*

Proof. Given an inclusion-free, non-matroid set system we describe how to construct a weighted congestion game with the properties stated in the theorem. We will first describe how the strategy spaces are defined and then how the delay functions are chosen.

Let Σ_1 and Σ_2 be two set systems over sets of resources $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. In the following we assume that both sets are isomorphic to Σ and that Σ_i is the strategy space of player i, for $i = 1, 2$. Due to the $(1, 2)$-exchange property we can, for every player i, identify three distinct resources $a_i, b_i, c_i \in \mathcal{R}_i$ with the properties as in Definition 7. Since we have not made any assumption on the global structure of the resources, we can arbitrarily decide which resources from $\mathcal{R}_1$ and $\mathcal{R}_2$ coincide. The resources $\mathcal{R}_i \setminus \{a_i, b_i, c_i\}$ are exclusively used by player i. Hence, we can assume that their delays are chosen such that the $(1,2)$-exchange property is satisfied. Thus, to simplify matters we can assume that

$$\Sigma_1 = \{\underbrace{\{a_1\}}_{S_1^1}, \underbrace{\{b_1, c_1\}}_{S_1^2}\} \text{ and } \Sigma_2 = \{\underbrace{\{a_2\}}_{S_2^1}, \underbrace{\{b_2, c_2\}}_{S_2^2}\} .$$

In the following, we assume that $a_1 = b_2$, $b_1 = a_2$ and $c_1 = c_2$. Thus we can rewrite the strategy spaces as follows: $\Sigma_1 = \{\{x\}, \{y, z\}\}$ and $\Sigma_2 = \{\{y\}, \{x, z\}\}$.

We set $\omega_1 = 2$ and $\omega_2 = 1$ and define the following non-decreasing delay functions for the resources x, y and z, where $m = |\mathcal{R}|$:

	$n_r = 1$	$n_r = 2$	$n_r = 3$
$d_x(n_x)$	m	$20 \cdot m$	$21 \cdot m$
$d_y(n_y)$	$5 \cdot m$	$12 \cdot m$	$15 \cdot m$
$d_z(n_z)$	$3 \cdot m$	$4 \cdot m$	$10 \cdot m$

One can easily verify that $|\delta_i(S \oplus S_i^1) - \delta_i(S \oplus S_i^2)| \geq m$, for $i = 1, 2$, regardless of the choice of the other player. Hence, for every player, one of the inequalities in Definition 7 is always satisfied. This game does not possess a Nash equilibrium since player 1 prefers to play strategy S_1^2 if player 2 plays strategy S_2^1, and S_1^1 if player 2 plays strategy S_2^2. Additionally, player 2 prefers to play strategy S_2^2 if player 1 plays strategy S_1^2, and S_2^1 if player 1 plays strategy S_1^1. □

Theorem 10. *For every inclusion-free, non-matroid set system Σ over a set of resources $\mathcal{R}$ there exists a player-specific congestion game Γ with two players whose strategy spaces are isomorphic to Σ that does not possess a Nash equilibrium. The delay functions in Γ are positive and non-decreasing.*

Proof. The proof is similar to the proof of Theorem 9. In particular, the construction of the strategy spaces of the players is identical. The player-specific delay functions are obtained from the delay functions in the proof of Theorem 9 as follows: For the first player $d_r^1(n_r) = d_r(n_r + 1)$, for every resource $r \in \{x, y, z\}$ and every congestion $n_r \in \{1, 2\}$. For the second player $d_r^2(1) = d_r(1)$ and $d_r^2(2) = d_r(3)$, for every resource $r \in \{x, y, z\}$. □

Summarizing, every inclusion-free non-matroid set system can be used to construct a player-specific or weighted congestion game with positive delay functions that does not posses a Nash equilibrium. Observe that this result also holds if the system is not inclusion-free but the *pruned set system*, i. e., the set system obtained after removing all supersets, is not the set of bases of a matroid because supersets cannot occur in a Nash equilibrium in the case of positive delay functions. Correspondingly, our results presented in Theorems 3 and 6 show that a player-specific or weighted congestion game in which all players' strategy spaces correspond to the bases of a matroid after pruning the supersets possesses a Nash equilibrium with respect to the pruned and, hence, also with respect to the original strategy spaces as supersets are weakly dominated by subsets in the case of non-negative delay functions. Thus, the matroid property (applied to the pruned strategy spaces) is necessary and sufficient to show the existence of Nash equilibria.

Corollary 11. *The matroid property is the maximal property on the pruned strategy spaces of the individual players that guarantees the existence of Nash equilibria in weighted and player-specific congestion games with non-negative, non-decreasing delay functions.*

References

1. Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. In *Proc. 47th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, 2006, (to appear).
2. Eyal Even-Dar, Alexander Kesselman, and Yishay Mansour. Convergence time to nash equilibria. In *Proc. 30th Int. EATCS Coll. on Automata, Languages and Programming (ICALP)*, pages 502–513, 2003.
3. Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proc. 36th Ann. ACM Symp. on Theory of Comput. (STOC)*, pages 604–612, 2004.
4. Dimitris Fotakis, Spyros C. Kontogiannis, and Paul G. Spirakis. Selfish unsplittable flows. *Theor. Comput. Sci.*, 348(2-3):226–239, 2005.
5. Martin Gairing, Thomas Lücking, Marios Mavronicolas, and Burkhard Monien. Computing nash equilibria for scheduling on restricted parallel links. In *Proc. 36th Ann. ACM Symp. on Theory of Comput. (STOC)*, pages 613–622, 2004.
6. Martin Gairing, Burkhard Monien, and Karsten Tiemann. Routing (un-)splittable flow in games with player-specific linear latency functions. In *Proc. 33rd Int. EATCS Coll. on Automata, Languages and Programming (ICALP)*, 2006. to appear.
7. Chryssis Georgiou, Theophanis Pavlides, and Anna Philippou. Uncertainty in selfish routing. In *Proc. 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.
8. Michel X. Goemans, Erran L. Li, Vahab S. Mirrokni, and Marina Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *MobiHoc*, pages 55–66, 2004.
9. Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. Fast and compact: A simple class of congestion games. In *20th National Conference on Artificial Intelligence (AAAI)*, 2005.
10. Igal Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1):111–124, 1996.
11. Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
12. Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Volume B, Matroids, Trees, Stable Sets, Chapters 39-69. Springer, 2003.
13. Renato Werneck, João Setubal, and Arlindo da Conceicão. Finding minimum congestion spanning trees. *J. Exp. Algorithmics*, 5:11, 2000.

On the Complexity of Pure-Strategy Nash Equilibria in Congestion and Local-Effect Games

—Extended Abstract—

Juliane Dunkel and Andreas S. Schulz

Operations Research Center, Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA
{juliane, schulz}@mit.edu

Abstract. Congestion games are a fundamental class of noncooperative games possessing pure-strategy Nash equilibria. In the network version, each player wants to route one unit of flow on a path from her origin to her destination at minimum cost, and the cost of using an arc only depends on the total number of players using that arc. A natural extension is to allow for players sending different amounts of flow, which results in so-called weighted congestion games. While examples have been exhibited showing that pure-strategy Nash equilibria need not exist, we prove that it actually is strongly NP-hard to determine whether a given weighted network congestion game has a pure-strategy Nash equilibrium. This is true regardless of whether flow is unsplittable (has to be routed on a single path for each player) or not.

A related family of games are local-effect games, where the disutility of a player taking a particular action depends on the number of players taking the same action and on the number of players choosing related actions. We show that the problem of deciding whether a bidirectional local-effect game has a pure-strategy Nash equilibrium is NP-complete, and that the problem of finding a pure-strategy Nash equilibrium in a bidirectional local-effect game with linear local-effect functions (for which the existence of a pure-strategy Nash equilibrium is guaranteed) is PLS-complete. The latter proof uses a tight PLS-reduction, which implies the existence of instances and initial states for which any sequence of selfish improvement steps needs exponential time to reach a pure-strategy Nash equilibrium.

1 Introduction

Game theory in general and the concept of Nash equilibrium in particular have lately come under increased scrutiny by theoretical computer scientists. Computing a mixed Nash equilibrium is a case in point. Goldberg and Papadimitriou (2006) showed only recently that finding a mixed Nash equilibrium in a game of any constant number of players is reducible to solving a 4-player game. Daskalakis, Goldberg, and Papadimitriou (2006) showed in turn that the latter problem is PPAD-complete. Subsequently, Chen and Deng (2005) and Daskalakis

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 62–73, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and Papadimitriou (2005) proved that computing mixed Nash equilibria in games with three players is PPAD-complete as well. Eventually, Chen and Deng (2006) established the same result for the two-player case.

While Nash (1951) showed that mixed Nash equilibria do exist in any finite noncooperative game, it is well known that pure-strategy Nash equilibria are in general not guaranteed to exist. It is therefore natural to ask which games have pure-strategy Nash equilibria and, if applicable, how difficult is it to find one. In this article, we study these questions for certain classes of weighted congestion and local-effect games.

Congestion games were introduced by Rosenthal (1973), who showed that they are guaranteed to possess pure-strategy Nash equilibria. In a congestion game, a player's strategy consists of a subset of resources, and her disutility only depends on the number of players choosing the same resources. An important subclass of congestion games can be represented by means of networks. Each player wants to route one unit of flow from her origin to her destination, at minimal cost. The network arcs are the resources, and a player's set of pure strategies consists of the sets of arcs corresponding to paths connecting her origin-destination pair. Fabrikant, Papadimitriou, and Talwar (2004) studied the computational complexity of finding pure-strategy Nash equilibria in congestion games. For symmetric network congestion games, where all players have the same origin-destination pair, they presented a polynomial-time algorithm for computing a pure-strategy Nash equilibrium. On the other hand, they proved that this problem is PLS-complete for symmetric congestion games as well as for asymmetric network congestion games. A simpler proof of the latter result was given by Ackermann, Röglin, and Vöcking (2006a), who also showed that this result still holds if the cost functions are affine-linear.

In (unweighted) network congestion games, each player routes exactly one unit of flow along a single path. In weighted congestion games, players can have different amounts of flow. Depending on whether players are allowed to split their flows or not, a player's strategy consists of a set of paths with corresponding integer flow values between her origin-destination pair, or of a single path.

Fotakis, Kontogiannis, and Spirakis (2005) studied weighted network congestion games with unsplittable flows. They constructed simple examples of symmetric instances that do not possess a pure-strategy Nash equilibrium. On the other hand, they proved that for the special case of affine cost functions, a pure-strategy Nash equilibrium is always guaranteed to exist. Awerbuch, Azar, and Epstein (2005) derived a tight bound of $(\sqrt{5}+3)/2$ on the pure price of anarchy for this special case. They also considered the case when the cost functions are polynomials of fixed degree greater than 1. However, Goemans, Mirrokni, and Vetta (2005) showed that a pure-strategy Nash equilibrium need not exist for instances with cost functions that are polynomials of degree at most 2. Milchtaich (1996) had earlier shown that weighted congestion games with player-specific disutility functions on networks consisting of parallel arcs only do not always have a pure-strategy Nash equilibrium.

In this article, we show that the problem of deciding whether a weighted network congestion game with simple, non-linear cost functions possesses a pure-strategy Nash equilibrium is strongly NP-hard, regardless of whether we consider splittable or unsplittable flows. In the unsplittable case, the problem remains NP-hard even if all players have the same origin and the same destination. The same is true for weighted congestion games with affine player-specific cost functions in networks consisting of parallel arcs only.

Leyton-Brown and Tennenholtz (2003) introduced local-effect games as a tool to model situations in which the use of one resource can affect the cost of other resources. Local-effect games are in general not guaranteed to possess pure-strategy Nash equilibria. However, Leyton-Brown and Tennenholtz showed that so-called bidirectional local-effect games with linear local-effect functions belong to the class of exact potential games, and therefore always have pure-strategy Nash equilibria. The question of whether there exists a polynomial-time algorithm for finding a pure-strategy Nash equilibrium for these games was left open.

We prove that computing a pure-strategy Nash equilibrium is PLS-complete. Because the proof uses a tight PLS-reduction, our result implies the existence of instances of these games that have exponentially long shortest improvement paths. It also implicates that the problem of finding a pure-strategy Nash equilibrium that is reachable from a given strategy state via selfish improvement steps is PSPACE-hard. In addition, we show that, given an initial strategy profile for a bidirectional local-effect game with linear local-effect functions and a positive integer k (unarily encoded), it is NP-complete to decide whether there is a sequence of at most k selfish steps that transforms the initial state into a pure-strategy Nash equilibrium. We also prove that the problem of deciding whether a bidirectional local-effect game with general, strictly increasing local-effect functions has a pure-strategy Nash equilibrium is NP-complete.

For bidirectional local-effect games with linear local-effect functions (for which a pure-strategy Nash equilibrium is guaranteed to exist), we also study the pure price of stability w.r.t. the social objective that is given by the sum of the costs of all players. In the case of linear cost functions, in which the worst-possible ratio of the social cost of a pure-strategy Nash equilibrium to that of a social optimum (i.e., the pure price of anarchy) is unbounded, we obtain a bound of 2 on the pure price of stability. Thus, there always exists a pure-strategy Nash equilibrium whose cost is at most twice that of a socially optimal solution. For the case of quadratic cost functions and linear local-effect functions we derive a bound of 3 on the pure price of stability.

Before we present the details of our results on weighted congestion games and local-effect games in Sections 2 and 3, respectively, let us end this introduction by briefly discussing additional related work on the computational complexity of pure-stratgey Nash equilibria. Gottlob, Greco, and Scarcello (2005) considered restrictions of strategic games intended to capture certain aspects of bounded rationality. Among other results, they proved that even in the setting where each player's payoff function depends on the actions of at most three other players and where each player is allowed to play at most three actions, the problem of determining whether a strategic game has a pure-strategy Nash equilibrium is NP-complete. This result was strengthened by Fischer, Holzer, and Katzenbeisser (2006) who showed that this problem remains hard even if each player has only two actions to choose from and her payoff depends on the actions of at most two other players. Àlvarez, Gabarró, and Serna (2005) studied how various representations of a strategic game influence the computational complexity of deciding the existence of a pure-strategy Nash equilibrium. They showed that this problem is NP-complete when the number of players is large and the number of strategies for each player is constant, while the problem is $\sum_2^p$-complete when the number of players is constant and the size of the sets of strategies is exponential (with respect to the length of the strategies). Schoenebeck and Vadhan (2006) analyzed the computational complexity of deciding whether a pure-strategy Nash equilibrium exists in graph games and circuit games. Brandt, Fischer, and Holzer (2006) studied the impact of various notions of symmetry in strategic games on the computational complexity of finding pure-strategy Nash equilibria. Expanding on a line of research started by Ieong et al. (2005), who considered singleton congestion games, Ackermann, Röglin, and Vöcking (2006a) proved that the lengths of all best-response sequences are polynomially bounded in the number of players and resources in congestion games where the strategy space of each player consists of the bases of a matroid over the set of resources. This especially implies that pure-strategy Nash equilibria for congestion games with the matroid property can be computed in polynomial time, even in the case of player-specific costs (Ackermann, Röglin, and Vöcking 2006b). In the latter paper, Ackermann et al. also showed the existence of pure-strategy Nash equilibria in weighted congestion games with the same matroid property.

Due to space limitations, proofs are only sketched or omitted completely from this extended abstract. Most details can be found in Dunkel (2005). A journal version is forthcoming.

2 Weighted Congestion Games

An *unweighted congestion game* is a tuple $\langle N, E, (S_i)_{i\in N}, (f_e)_{e\in E}\rangle$, where $N = \{1, 2, \dots, n\}$ is the set of players, and E is a set of resources. For each player $i \in N$, her set S_i of available strategies is a collection of subsets of the resources; that is, $S_i \subseteq 2^E$. A cost function $f_e : \mathbb{N} \to \mathbb{R}_+$ is associated with each resource $e \in E$. Given a strategy profile $s = (s_1, s_2, \dots, s_n) \in S = S_1 \times S_2 \times \cdots \times S_n$, the cost (disutility) of player i is $c_i(s) = \sum_{e\in s_i} f_e\big(n_e(s)\big)$, where $n_e(s)$ denotes the number of players using resource e in s. In other words, in a congestion game each player chooses a subset of resources that are available to her, and the cost to a player is the sum of the costs of the resources used by her, where the cost of a resource only depends on the total number of players using this resource.

A *network congestion game* is a congestion game where the arcs of an underlying directed network represent the resources. Each player $i \in N$ has an origin-destination pair (a_i, b_i), where a_i and b_i are nodes of the network, and the set S_i of pure strategies available to player i is the set of directed (simple) paths from a_i

to b_i. A symmetric network congestion game is also called a *single-commodity* network congestion game because all players have the same origin-destination pair.

In a *weighted network congestion game* $\langle N, E, (w_i)_{i\in N}, (S_i)_{i\in N}, (f_e)_{e\in E}\rangle$, each player $i \in N$ has a positive integer weight w_i, which constitutes the amount of flow that player i wants to ship from a_i to b_i. In the case of unsplittable flows, the cost of player i adopting strategy s_i in a strategy profile $s = (s_1, s_2, \dots, s_n) \in S$ is given by $c_i(s) = \sum_{e\in s_i} f_e\big(\theta_e(s)\big)$, where $\theta_e(s) = \sum_{i:e\in s_i} w_i$ denotes the total flow on arc e in s. In integer-splittable network congestion games, a player with weight greater than one can choose a subset of paths on which to route her flow simultaneously; that is, player i's strategy consists of the specification of the a_i-b_i-paths used and the (integer) amounts of flow routed on them, which sum up to w_i.

In terms of the input size of a weighted network congestion game, we assume that the cost functions are explicitly specified; that is, for each $0 \le x \le \sum_{i\in N} w_i$ and each arc e, the value $f_e(x)$ is given in binary encoding.

While every unweighted congestion game possesses a pure-strategy Nash equilibrium (Rosenthal 1973), this is not true for weighted congestion games; see, e.g., Fig. 1 in Fotakis, Kontogiannis, and Spirakis (2005). We can actually turn their instance into a gadget to derive the following result.

Theorem 1. *The problem of deciding whether a weighted symmetric network congestion game with unsplittable flows possesses a pure-strategy Nash equilibrium is strongly NP-complete.*

The proof is by a reduction from 3-PARTITION, and it is omitted from this extended abstract. While the NP-hardness of the corresponding decision problem for weighted network congestion games with player-specific payoff functions follows immediately, we can actually strengthen this result.

Theorem 2. *The problem of deciding whether a weighted network congestion game with parallel arcs and affine player-specific disutility functions possesses a pure-strategy Nash equilibrium is strongly NP-complete.*

For network congestion games with integer-splittable flows, we obtain the following result.

Theorem 3. *The problem of deciding whether a weighted network congestion game with integer-splittable flows possesses a pure-strategy Nash equilibrium is strongly NP-hard. Hardness even holds if there is only one player with weight* 2, *and all other players have unit weights.*

Proof. Consider an instance of MONOTONE3SAT with set of variables $X = \{x_1, x_2, \dots, x_n\}$ and set of clauses $C = \{c_1, c_2, \dots, c_m\}$. We construct a game that has one player p_x for every variable $x \in X$ with weight $w_x = 1$, origin x and destination $\bar{x}$. Moreover, each clause $c \in C$ gives rise to a player p_c with weight $w_c = 1$, origin c, and destination $\bar{c}$. There are three more players p_1, p_2, and p_3 with weights $w_1 = 1, w_2 = 2, w_3 = 1$ and origin-destination pairs $(s, t_1), (s, t_2), (s, t_3)$, respectively. For every variable $x \in X$ there are two disjoint paths P_x^1, P_x^0 from x to $\bar{x}$ in the network. Path P_x^0 consists of $2\,|\{c \in C \mid x \in c\}| + 1$ arcs and P_x^1 has $2\,|\{c \in C \mid \bar{x} \in c\}| + 1$ arcs with cost functions as shown in Fig. 1. For each pair $(c, \bar{c})$, we construct two disjoint paths P_c^1, P_c^0 from c to $\bar{c}$. Path P_c^1 consists of only two arcs. The paths P_c^0 will have seven arcs each and are constructed for $c = c_j$ in the order $j = 1, 2, \ldots, m$ as follows. For a positive clause $c = c_j = (x_{j_1} \vee x_{j_2} \vee x_{j_3})$ with $j_1 < j_2 < j_3$, path P_c^0 starts with the arc connecting c to the first inner node v_1 on path $P_{x_{j_1}}^1$ that has only two incident arcs so far. The second arc is the unique arc (v_1, v_2) of path $P_{x_{j_1}}^1$ that has v_1 as its start vertex. The third arc connects v_2 to the first inner node v_3 on path $P_{x_{j_2}}^1$ that has only two incident arcs so far. The fourth arc is the only arc (v_3, v_4) on $P_{x_{j_2}}^1$ with start vertex v_3. From v_4, there is an arc to the first inner node v_5 on $P_{x_{j_3}}^1$ that has only two incident arcs so far, followed by (v_5, v_6) of $P_{x_{j_3}}^1$. The last arc of P_c^0 connects v_6 to $\bar{c}$ (see Fig. 1). For a negative clause $c = c_j = (\bar{x}_{j_1} \vee \bar{x}_{j_2} \vee \bar{x}_{j_3})$ we proceed in the same way, except that we choose the inner nodes v_i from the upper variable paths $P_{x_{j_1}}^0, P_{x_{j_2}}^0, P_{x_{j_3}}^0$. The strategy set of player p_x is $\{P_x^1, P_x^0\}$. We will interpret it as setting the

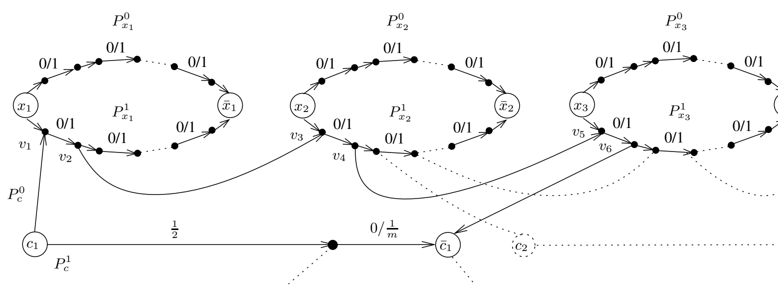


Fig. 1. Part of the constructed network corresponding to a positive clause $c_1 = (x_1 \vee x_2 \vee x_3)$. The notation a/b defines a cost per unit flow of value a for load 1 and b for load 2. Arcs without specified values have zero cost.

variable x to *true* (*false*) if p_x sends her unit of flow over P_x^1 (P_x^0). Note that player p_c can only choose between the paths P_c^1 and P_c^0. This is due to the order in which the paths $P_{c_j}^0$ are constructed. Depending on whether player p_c sends her unit of flow over path P_c^1 or P_c^0, the clause c will be satisfied or not.

The second part of the network consists of all origin-destination pairs and paths for players p_1, p_2, p_3 (see Fig. 2). Player p_1 can choose between paths $U_1 = \{(s, t_2), (t_2, t_1)\}$ and $L_1 = \{(s, t_1)\}$. Player p_2 is the only player who can split her flow; that is, she can route her two units either both over path $U_2 = \{(s, t_2)\}$, both over path $L_2 = \{(s, t_1), (t_1, t_2\}$, or one unit on the upper and the other unit on the lower path; i.e., $S_2 = \{L_2, U_2, LU_2\}$. Finally, player p_3 has three possible paths to choose from. The upper path U_3 shares an arc with each clause path P_c^1 and has some additional arcs to connect these. The paths $M_3 = \{(s, t_2), (t_2, t_3)\}$

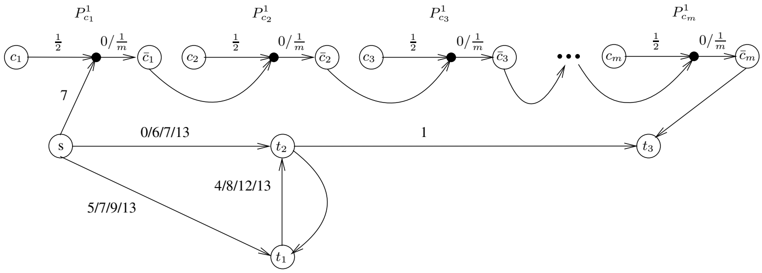


Fig. 2. Part of the constructed network that is used by players p_1, p_2, and p_3. A single number a on an arc defines a constant cost a per unit flow for this arc.

and $L_3 = \{(s, t_1), (t_1, t_2), (t_2, t_3)\}$ have only arcs with the paths of p_1 and p_2 in common. The cost functions are defined in Fig. 2.

Given a satisfying truth assignment, we define a strategy state s of the game by setting the strategy of player p_x to be P_x^1 for a true variable x, and P_x^0 otherwise. Each player p_c plays P_c^1. Furthermore, $s_1 = L_1$, $s_2 = U_2$, and $s_3 = M_3$. It is easy to show that no player can decrease her cost by unilaterally switching to another strategy; i.e., the defined strategy configuration is a pure-strategy Nash equilibrium.

For the other direction, we first observe that any pure-strategy Nash equilibrium s of the game has the following properties: (a) player p_3 does not use path U_3, (b) for the cost of player p_3 we have $c_3(s) \geq 8$, and (c) each player p_c routes her unit flow over path P_c^1. Property (a) follows from the fact that the sub-game shown in Fig. 3 with players p_1 and p_2 only does not have a pure-strategy Nash equilibrium. Property (a) implies (b), and property (c) can be proved by contradiction assuming (a) and (b). We claim that the truth assignment that

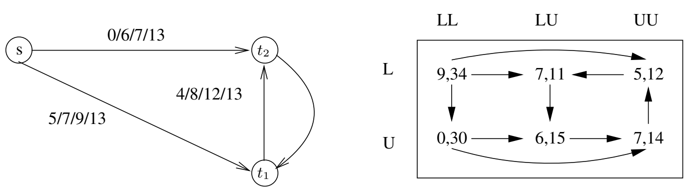


Fig. 3. Sub-game with two players without pure-strategy Nash equilibrium (Papadimitriou 2003)

sets a variable x to *true* if the corresponding player uses P_x^1, and x to *false* otherwise, satisfies all clauses. Suppose for a positive clause $c = (x_1 \vee x_2 \vee x_3)$ that all variables are *false*; i.e., $s_{x_i} = P_{x_i}^0$ for $i = 1, 2, 3$. By property (c), player p_c uses P_c^1. Because of (a), her current cost is $c_c(s) = \frac{1}{2}$. Choosing path P_c^0 instead would decrease the cost to zero, which contradicts the assumption of s being a Nash equilibrium. A similar argument holds for a negative clause. □

3 Local-Effect Games

A *local-effect game* is a tuple $\langle N, A, \mathcal{F} \rangle$ where $N = \{1, 2, \ldots, n\}$ is the set of players, A is a common set of actions (strategies) available to each player, and $\mathcal{F}$ is a set of cost functions. For each pair of actions $a, a' \in A$, the function $F_{a',a} : \mathbb{Z}_+ \to \mathbb{R}_+$ expresses the impact of action a' on the cost of action a, which depends only on the number of players that choose action a'. For $a, a' \in A$ with $a \neq a'$, $F_{a',a}$ is called a *local-effect function*, and it is assumed that $F_{a',a}(0) = 0$. Moreover, the local-effect function $F_{a',a}$ is either strictly increasing or identical zero. If $F_{a',a}$ is not identical zero, then this is also the case for $F_{a,a'}$. In other words, if action a has an effect on action a', then the converse is also true. For a given strategy state $s = (s_1, s_2, \ldots, s_n) \in A^n$, n_a denotes the number of players playing action a in s. The cost to a player $i \in N$ for playing action s_i in strategy state s is given by $c_i(s) = F_{s_i,s_i}(n_{s_i}) + \sum_{a \in A, a \neq s_i} F_{a,s_i}(n_a)$. If the local-effect functions $F_{a',a}$ are zero for all $a \neq a'$, the local-effect game is equivalent to a symmetric network congestion game with only parallel arcs.

A local-effect game is called a *bidirectional* local-effect game if for all $a, a' \in A$, $a \neq a'$, and for all $x \in \mathbb{Z}_+$, $F_{a',a}(x) = F_{a,a'}(x)$. Leyton-Brown and Tennenholtz (2003) gave a characterization of local-effect games that have an exact potential function and which are therefore guaranteed to possess pure-strategy Nash equilibria. One of these subclasses are bidirectional local-effect games with linear local-effect functions. However, without linear local-effect functions, deciding the existence is hard.

3.1 Computational Complexity

Theorem 4. *The problem of deciding whether a bidirectional local-effect game has a pure-strategy Nash equilibrium is NP-complete.*

The proof will be given in the full version of this paper. The next result implies that computing a pure-strategy Nash equilibrium for a bidirectional local-effect game with linear local-effect functions is as least as hard as finding a local optimum for several combinatorial optimization problems with efficiently searchable neighborhoods.

Theorem 5. *The problem of computing a pure-strategy Nash equilibrium for a bidirectional local-effect game with linear local-effect functions is PLS-complete.*

Proof. We reduce from PosNae3Flip (Schäffer and Yannakakis 1991): Given not-all-equal clauses with at most three literals, (x_1, x_2, x_3) or (x_1, x_2), where x_i is either a variable or a constant (0 or 1), and a weight for each clause, find a truth assignment such that the total weight of satisfied clauses cannot be improved by flipping a single variable.

For simplicity, we assume that we are given an instance of PosNae3Flip with set $C = C_2 \dot\cup C_3$ of clauses containing two or three variables but no constants, a positive integer weight w_c for each clause $c \in C$, and set of variables $\{x_1, \ldots, x_n\}$. We construct a bidirectional local-effect game with linear local-effect functions as follows: There are n players with common action set A that contains two actions a_i and $\overline{a}_i$ for each variable x_i, $i = 1, 2, \ldots, n$. Let $M = 2n \sum_{c \in C} w_c + 1$. For each action $a \in A$, $F_{a,a}(x) = 0$ if $x \leq 1$, and $F_{a,a}(x) = M$ otherwise. If $C_i = \{c \in C \mid x_i \in c\}$ denotes the subset of clauses containing variable x_i, the local-effect functions are given for $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$, by

$$F_{a_i,a_j}(x) = F_{\overline{a}_i,\overline{a}_j}(x) = \left(2 \sum_{c \in C_2 \cap C_i \cap C_j} w_c + \sum_{c \in C_3 \cap C_i \cap C_j} w_c \right) x \ .$$

However, the local-effect functions F_{a_i,a_j} and $F_{\overline{a}_i,\overline{a}_j}$ are zero if there is no clause containing both x_i and x_j. Furthermore, $F_{a_i,\overline{a}_i}(x) = F_{\overline{a}_i,a_i}(x) = M\,x$ for all $i \in \{1, 2, \ldots, n\}$. All local-effect functions not defined so far are identical zero. For any solution $s = (s_1, s_2, \ldots, s_n)$, $s_i \in A$, of the game, we define the corresponding truth assignment to the variables x_i of the PosNae3Flip instance by $x_i = 1$ if $|\{j \mid s_j = a_i\}| \geq 1$, and $x_i = 0$ otherwise.

Now we show that for any pure-strategy Nash equilibrium $s = (s_1, s_2, \ldots, s_n)$ of the game, the corresponding truth assignment is indeed a local optimum of the PosNae3Flip instance. The proof is demonstrated only for the case of flipping a positive variable $x_i = 1$ to $x'_i = 0$. First, observe that for all $i \in \{1, 2, \ldots, n\}$

$$\big|\{j \mid s_j = a_i\}\big| + \big|\{j \mid s_j = \overline{a}_i\}\big| = 1 \ , \tag{1}$$

since otherwise, because of the choice of M, there is always a player who can decrease her cost by choosing another action.

Let X and X' denote the truth assignments before and after flipping variable x_i. Let the set of clauses that contain variable x_i and are satisfied by truth assignment X, X' be C_i^X, $C_i^{X'}$, respectively. Further, let $C_i^{X \setminus X'}$ ($C_i^{X' \setminus X}$) be the set of clauses containing x_i that are satisfied by truth assignment X (X'), but not by X' (X). Then the difference in the total weight of satisfied clauses by X' and X can be written as

$$\Delta W = \sum_{c \in C_2 \cap C_i^{X' \setminus X}} w_c + \sum_{c \in C_3 \cap C_i^{X' \setminus X}} w_c - \sum_{c \in C_2 \cap C_i^{X \setminus X'}} w_c - \sum_{c \in C_3 \cap C_i^{X \setminus X'}} w_c \ . \tag{2}$$

For a clause $c = (x_i, x_j) \in C_i^{X' \setminus X}$, it follows because of $x_i = 1$ that $x_j = 1$. Then, by definition of X and by (1), $n_{a_j} = 1$ and $n_{\overline{a}_j} = 0$. If $c = (x_i, x_j) \in C_i^{X \setminus X'}$, we have $x_j = 0$, $n_{a_j} = 0$, and $n_{\overline{a}_j} = 1$. Similarly, for a three-variable clause $c = (x_i, x_j, x_k) \in C_i^{X' \setminus X}$, $x_i = 1$ implies $x_j = x_k = 1$, $n_{a_j} = n_{a_k} = 1$, and $n_{\overline{a}_j} = n_{\overline{a}_k} = 0$. If $c = (x_i, x_j, x_k) \in C_i^{X \setminus X'}$, then $x_j = x_k = 0$, $n_{a_j} = n_{a_k} = 0$, and $n_{\overline{a}_j} = n_{\overline{a}_k} = 1$. Thus we can rewrite (2) as

$$\begin{aligned} \Delta W = &\sum_{j=1, j \neq i}^{n} \left[\left(\sum_{c \in C_2 \cap C_i^{X' \setminus X} \cap C_j} w_c \right) n_{a_j} - \left(\sum_{c \in C_2 \cap C_i^{X \setminus X'} \cap C_j} w_c \right) n_{\overline{a}_j} \right] \\ &+ \frac{1}{2} \sum_{j=1, j \neq i}^{n} \left[\left(\sum_{c \in C_3 \cap C_i^{X' \setminus X} \cap C_j} w_c \right) n_{a_j} - \left(\sum_{c \in C_3 \cap C_i^{X \setminus X'} \cap C_j} w_c \right) n_{\overline{a}_j} \right] \ . \end{aligned} \tag{3}$$

By the above observations on the numbers n_{a_j} and $n_{\overline{a}_j}$ we have

$$\sum_{j=1,j\neq i}^{n} \sum_{c\in C_2\cap C_i^{X\setminus X'}\cap C_j} w_c n_{a_j} = -\sum_{j=1,j\neq i}^{n} \sum_{c\in C_2\cap C_i^{X'\setminus X}\cap C_j} w_c n_{\overline{a}_j} = 0 \ , \qquad (4)$$

$$\sum_{j=1,j\neq i}^{n} \sum_{c\in C_3\cap C_i^{X\setminus X'}\cap C_j} w_c n_{a_j} = -\sum_{j=1,j\neq i}^{n} \sum_{c\in C_3\cap C_i^{X'\setminus X}\cap C_j} w_c n_{\overline{a}_j} = 0 \ . \qquad (5)$$

Now consider clauses $c = (x_i, x_j, x_k) \in (C_3 \cap C_i) \setminus (C_i^{X\setminus X'} \cup C_i^{X'\setminus X})$. Since the case of clause c not being satisfied by both X and X' cannot happen, we have $c \in C_i^X \cap C_i^{X'}$. Then, either $x_j = 1, x_k = 0$ or $x_j = 0, x_k = 1$, and therefore $n_{a_j} = 1, n_{a_k} = 0$ or $n_{a_j} = 0, n_{a_k} = 1$. By (1), we have in both cases $n_{a_j} + n_{a_k} = n_{\overline{a}_j} + n_{\overline{a}_k} = 1$; i.e., $w_c(n_{a_j} + n_{a_k}) - w_c(n_{\overline{a}_j} + n_{\overline{a}_k}) = 0$. Thus

$$\sum_{j=1,j\neq i}^{n} \left(\sum_{c\in C_3\cap C_i^X\cap C_i^{X'}\cap C_j} w_c \right) n_{a_j} - \sum_{j=1,j\neq i}^{n} \left(\sum_{c\in C_3\cap C_i^X\cap C_i^{X'}\cap C_j} w_c \right) n_{\overline{a}_j} = 0 \ . \qquad (6)$$

Adding the terms in (4), (5), and (6) to (3), we obtain

$$\begin{aligned} \Delta W = &\sum_{j=1,j\neq i}^{n} \left[\left(\sum_{c\in C_2\cap C_i\cap C_j} w_c \right) n_{a_j} - \left(\sum_{c\in C_2\cap C_i\cap C_j} w_c \right) n_{\overline{a}_j} \right] \\ &+ \frac{1}{2} \sum_{j=1,j\neq i}^{n} \left[\left(\sum_{c\in C_3\cap C_i\cap C_j} w_c \right) n_{a_j} - \left(\sum_{c\in C_3\cap C_i\cap C_j} w_c \right) n_{\overline{a}_j} \right] \leq 0 \ . \end{aligned}$$

Here, the last inequality follows from the fact that the player i with action $s_i = a_i$ cannot decrease her cost by switching to action $\overline{a}_i$. The described construction is indeed a PLS-reduction. □

Since the reduction actually is a *tight* PLS-reduction, we obtain the following results.

Corollary 6. *There are instances of bidirectional local-effect games with linear local-effect functions that have exponentially long shortest improvement paths.*

Corollary 7. *For a bidirectional local-effect game with linear local-effect functions, the problem of finding a pure-strategy Nash equilibrium that is reachable from a given strategy state via selfish improvement steps is PSPACE-complete.*

The following result underlines that finding a pure Nash equilibrium for bidirectional local-effect games with linear local-effect functions is indeed hard.

Theorem 8. *Given an instance of a bidirectional local-effect games with linear local-effect functions, a pure-strategy profile s_0, and an integer $k > 0$ (unarily encoded), it is NP-complete to decide whether there exists a sequence of at most k selfish steps that transforms s_0 to a pure-strategy Nash equilibrium.*

3.2 Pure Price of Stability for Bidirectional Local-Effect Games

We derive bounds on the pure-price of stability for games with linear local-effect functions where the social objective is the sum of the costs of all players.

Theorem 9. *The pure price of stability for bidirectional local-effect games with only linear cost functions is bounded by* 2*.*

The proof is based on a technique suggested by Anshelevich et al. (2004) using the potential function introduced by Leyton-Brown and Tennenholtz (2003). By the same technique, we can derive the following bound for the case of quadratic cost-functions and linear local-effect functions.

Theorem 10. *The pure price of stability for bidirectional local-effect games with* $F_{a,a}(x) = m_a\, x^2 + q_a\, x$, $q_a \geq 0$ *for all* $a \in A$ *and linear local-effect functions is bounded by* 3*.*

References

Ackermann, H., H. Röglin, and B. Vöcking (2006a). On the impact of combinatorial structure on congestion games. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, Berkeley, CA, to appear.

Ackermann, H., H. Röglin, and B. Vöcking (2006b). Pure Nash equilibria in player-specific and weighted congestion games. This volume.

Àlvarez, C., J. Gabarró, and M. Serna (2005). Pure Nash equilibria in games with a large number of actions. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, Gdansk, Poland. *Lecture Notes in Computer Science 3618*, 95–106.

Anshelevich, E., A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden (2004). The price of stability for network design with fair cost allocation. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, Rome, Italy, pp. 295–304.

Awerbuch, B., Y. Azar, and A. Epstein (2005). The price of routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, Baltimore, MD, pp. 57–66.

Brandt, F., F. Fischer, and M. Holzer (2006). Symmetries and the complexity of pure Nash equilibrium. *Electronic Colloquium on Computational Complexity TR06-091.*

Chen, X. and X. Deng (2005). 3-Nash is PPAD-complete. *Electronic Colloquium on Computational Complexity TR05-134.*

Chen, X. and X. Deng (2006). Settling the complexity of 2-player Nash-equilibrium. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, Berkeley, CA, to appear.

Daskalakis, C., P. Goldberg, and C. Papadimitriou (2006). The complexity of computing a Nash equilibrium. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, Seattle, WA, pp. 71–78.

Daskalakis, C. and C. Papadimitriou (2005). Three-player games are hard. *Electronic Colloquium on Computational Complexity TR05-139.*

Dunkel, J. (2005). The Complexity of Pure-Strategy Nash Equilibria in Non-Cooperative Games. Diplomarbeit, Institute of Mathematics, Technische Universität Berlin, Germany, July 2005.

Fabrikant, A., C. Papadimitriou, and K. Talwar (2004). The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, Chicago, IL, pp. 604–612.

Fischer, F., M. Holzer, and S. Katzenbeisser (2006). The influence of neighbourhood and choice on the complexity of finding pure Nash equilibria. *Information Processing Letters 99*, 239–245.

Fotakis, D., S. Kontogiannis, and P. Spirakis (2005). Selfish unsplittable flows. *Theoretical Computer Science 348*, 226–239.

Goemans, M., V. Mirrokni, and A. Vetta (2005). Sink equilibria and convergence. *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, Pittsburgh, PA, pp. 142–154.

Goldberg, P. and C. Papadimitriou (2006). Reducibility among equilibrium problems. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, Seattle, WA, pp. 61–70.

Gottlob, G., G. Greco, and F. Scarcello (2005). Pure Nash equilibria: Hard and easy games. *Journal of Artificial Intelligence Research 24*, 357–406.

Ieong, S., R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun (2005). Fast and compact: A simple class of congestion games. In *Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference*, Pittsburgh, PA, 489–494.

Leyton-Brown, K. and M. Tennenholtz (2003). Local-effect games. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 772–780.

Milchtaich, I. (1996). Congestion games with player-specific payoff functions. *Games and Economic Behavior 13*, 111–124.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics 54*, 268–295.

Rosenthal, R. (1973). A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory 2*, 65–67.

Schäffer, A. and M. Yannakakis (1991). Simple local search problems that are hard to solve. *SIAM Journal on Computing 20*, 56–87.

Schoenebeck, G. and S. Vadhan (2006). The computational complexity of Nash equilibria in concisely represented games. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, Ann Arbor, MI, pp. 270–279.

Strong and Correlated Strong Equilibria in Monotone Congestion Games

Ola Rozenfeld and Moshe Tennenholtz

Technion–Israel Institute of Technology
Haifa 32000, Israel

Abstract. The study of congestion games is central to the interplay between computer science and game theory. However, most work in this context does not deal with possible deviations by coalitions of players, a significant issue one may wish to consider. In order to deal with this issue we study the existence of strong and correlated strong equilibria in monotone congestion games. Our study of strong equilibrium deals with monotone-increasing congestion games, complementing the results obtained by Holzman and Law-Yone on monotone-decreasing congestion games. We then present a study of correlated-strong equilibrium for both decreasing and increasing monotone congestion games.

Keywords: Congestion Games, Strong Equilibrium.

1 Introduction and Overview of Results

A *congestion game* (Rosenthal, [7]) is defined as follows: A finite set of players[1], $N = \{1, ..., n\}$; A finite non-empty set of facilities, M; For each player $i \in N$ a non-empty set $A_i \subseteq 2^M$, which is the set of actions available to player i (an action is a subset of the facilities). We denote by A the set of all possible action profiles ($A = \prod_{i \in N} A_i$). With every facility $m \in M$ and integer number $1 \leq k \leq n$ a real number $v_m(k)$ is associated, having the following interpretation: $v_m(k)$ is the utility to each user of m if the total number of users of m is k. Let $a \in A$; the ($|M|$ dimensional) congestion vector corresponding to a is $\sigma(a) = (\sigma_m(a))_{m \in M}$ where $\sigma_m(a) = |\{i | m \in a_i\}|$. The utility function of player i, $u_i : A \to \mathrm{R}$ is defined as follows: $u_i(a) = \sum_{m \in a_i} v_m(\sigma_m(a))$. It is assumed that all players try to maximize their utility. Therefore, equilibrium analysis is typically used for the study of these settings.

Congestion games have become a central topic of study in the interplay between computer science and game theory (see e.g. [1,9,8,6]). Congestion games possess some interesting properties. In particular, Rosenthal [7] showed that every congestion game possesses a pure strategy Nash equilibrium. In this paper we would like to explore the possibility of replacing Nash equilibrium with stronger solution concepts.

[1] We will use the terms player and agent interchangeably.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 74–86, 2006.
© Springer-Verlag Berlin Heidelberg 2006

One particular weakness of Nash equilibrium is its vulnerability to deviations by coalitions of players. This issue is addressed in the solution concept known as strong equilibrium (Aumann, [2]): Let us denote the projection of $a \in A$ on the set of players $S \subseteq N$ (resp. on $N \setminus S$) by a_S (resp. by a_{-S}). We say that a profile of actions $a^* \in A$ is a *strong equilibrium* (SE) if for no non-empty coalition $S \subseteq N$ there is a choice of actions $a_i \in A_i, i \in S$ such that $\forall i \in S \;\; u_i(a_S, a^*_{-S}) > u_i(a^*)$.

Such profiles are indeed much more stable than simple Nash equilibria, and therefore their existence is a very desirable property; however, simple examples show that congestion games in general need not possess a strong equilibrium (in fact, the well-known Prisoner's Dilemma may be obtained as a congestion game).

The above definition applies to the case where the players may use only pure strategies. A natural extension of Aumann's definition of strong equilibrium to settings where mixed strategies are available is to apply the original definition to the mixed extension of the original game. Formally, we say that a profile of actions $a^* \in \prod_{i \in N} \Delta(A_i)$ is a *mixed strong equilibrium* (MSE) if for no non-empty coalition $S \subseteq N$ there is a choice of actions $a_i \in \Delta(A_i), i \in S$ such that $\forall i \in S \;\; U_i(a_S, a^*_{-S}) > U_i(a^*)$. Here, by $\Delta(A_i)$ we mean the set of all probability distributions overA_i, and U_i denotes the expected utility of player i.

There are two important things to note when considering the definition of MSE. First, notice that unlike the extension of Nash equilibrium to mixed strategies, this definition yields a stronger solution concept even when applied to pure strategy profiles; i.e., a pure profile of actions may be a strong equilibrium, but not a mixed strong equilibrium. A second point to notice is that in the definition of MSE we assume that the players cannot use *correlated* mixed strategies, i.e. choose their actions using a joint probability distribution. However, in many settings this assumption is too restrictive: if we assume that a coalition of players has the means to choose a coordinated profile of actions, it is natural to assume that they have means of communication that would also allow them to coordinate their actions using joint coin flips. The above leads us to the following definition: we say that $a^* \in \Delta(A)$ is a *correlated strong equilibrium* (CSE) if for no non-empty coalition $S \subseteq N$ there is a choice of actions $a_S \in \Delta(\prod_{i \in S} A_i)$, such that $\forall i \in S \;\; U_i(a_S, a^*_{-S}) > U_i(a^*)$. This definition is strictly stronger than the previous one: every CSE is also an MSE, but not vice versa.[2]

The aim of this article is to explore the conditions for existence of strong and correlated strong equilibria within two most interesting and central subclasses of congestion games:

We call a congestion game *monotone-increasing* (or simply *increasing*) if $\forall m \in M, 1 \leq k < n \;\; v_m(k) \leq v_m(k+1)$. These games model settings where congestion

[2] Notice that although we allow a^* to be a correlated profile, CSE doesn't extend the notion of correlated Nash equilibrium [3] to the context of deviations by coalitions: our solution concept is weaker, since we assume that the deviators cannot see the "signals" that result from the current realization. However, in the scope of this article, generalizing Aumann's definition would yield the same results.

has a positive effect on the players, e.g. settings in which the cost of using a facility is shared between its users.

We call a congestion game *monotone-decreasing* (or simply *decreasing*) if $\forall m \in M, 1 \leq k < n\ v_m(k) \geq v_m(k+1)$. These games model settings where congestion has a negative effect on the players, e.g. routing games, where cost represents latency.

Ron Holzman and Nissan Law-Yone [5,4] explored the conditions for existence of *strong equilibria* in *monotone decreasing* congestion games. They start by observing that a strong equilibrium always exists in the case where all strategies are singletons. Following that, they explore the *structural properties* of the strategy sets that are necessary and sufficient to guarantee the existence of strong equilibria. These structural properties may, for example, refer to the underlying graph structure in route selection games.

In this paper we first explore the conditions for existence of strong equilibria in *monotone increasing* congestion games. Then, we extend the study of both the decreasing and increasing settings to the solution concept of correlated strong equilibrium. Our contributions can therefore be described by the following table:

	Pure deviations	Correlated deviations
Increasing	*This work*	*This work*
Decreasing	*Holzman & Law-Yone*	*This work*

Main results

Throughout this paper, when we refer to strong equilibrium, we present the results of Holzman and Law-Yone [5] for the decreasing setting alongside our results for the increasing setting. This is done for the sake of viewing the complete picture and ease of comparing between the two settings.

In section 2 we explore the case of singleton strategies, i.e. resource selection games where each player should choose a single resource from a set of resources available to him. In the decreasing setting Holzman and Law-Yone observe that every Nash equilibrium of the game is, in fact, a strong equilibrium. For the increasing setting, we present an efficient algorithm for constructing a strong equilibrium; however, unlike in the decreasing setting, we show that not every Nash equilibrium of the game is strong.

In section 3 we develop a notation, *congestion game forms*, that allows us to speak about the underlying structure of congestion games; using this notation we will be able to formalize statements such as "a certain structural property is necessary and sufficient for the existence of SE in all games with that underlying structure". We define two substructures, which we call *d-bad configuration* and *i-bad configuration* and prove some simple properties of strategy spaces that avoid bad configurations. These properties will serve as a technical tool in some of our proofs.

Section 4 explores the conditions for existence of strong equilibrium. In the decreasing setting, [5] shows that a SE always exists if and only if d-bad configurations are avoided. In the increasing setting we show that a SE always exists if and only if i-bad configurations are avoided, in which case the equilibrium can be efficiently computed. As we will show, our results imply that avoiding i-bad configurations makes the games essentially isomorphic to the case of singleton strategies.

Section 5 deals with the concept of correlated strong equilibrium. We show that a CSE might not be achievable even in simple (two players, two strategies) examples of the decreasing setting. In the increasing setting, though, we show that all our results regarding SE still hold with CSE, namely that a CSE always exists if and only if i-bad configurations are avoided, in which case it can be efficiently computed. Moreover, we show that in this case every SE of the game is also a CSE (a claim which doesn't hold if i-bad configurations are not avoided).

Together, our results provide full characterization for the connection between the underlying game structure and the existence of SE and CSE for both the decreasing and the increasing cases.

2 SE: The Case of Singleton Strategies

Here we investigate the case in which only singleton strategies are allowed, i.e. resource selection games where each player should choose a facility from among a set of facilities available to him.[3]

First, recall the result for the decreasing case:

Theorem 1. *[5] Let G be a monotone decreasing congestion game in which all strategies are singletons. Then G possesses a strong equilibrium; moreover, every Nash equilibrium of G is SE.*

We now address the existence of SE in monotone-increasing congestion games:

Theorem 2. *Let G be a monotone increasing congestion game in which all strategies are singletons. Then G possesses a strong equilibrium; moreover, a SE can be efficiently computed.*

Proof (sketch): Consider the following algorithm for computing a strong equilibrium: at each step, we assign a facility to a non empty subset of the remaining players, in the following way: for each facility $m \in M$, we compute $v_m(k)$, where k is the maximal number of the remaining players that can choose $\{m\}$ as their strategy. We choose m for which such $v_m(k)$ is maximal, and assign $\{m\}$ to all the players that can choose it. We continue in the same fashion until all players are assigned a facility.

[3] In particular, this classical setting can model simple route selection games. In a simple route selection game each player has to select a link for reaching from source to target in a graph consisting of several parallel links. In general, each player may have a different subset of the links that he may use.

We claim that the resulting strategy profile is a SE. We prove by induction on the steps of the algorithm, that no player can belong to a deviating coalition in which his payoff strictly increases: in the first step of the algorithm, this is obvious because each assigned player gets the highest possible payoff in the game (due to monotonicity); at subsequent steps, we use the induction hypothesis and assume that all players from the previous steps don't belong to the deviating coalition, i.e. all of them use the facilities they were assigned; but this means that the game is effectively reduced to the remaining players and the remaining facilities, so the same reasoning applies: due to monotonicity, each assigned player gets the highest possible payoff in the (new) game. Regarding the complexity of the algorithm, it is trivial to verify that the most straightforward implementation runs in $O(m^2n^2)$; it is also a simple exercise to construct an implementation that runs in $O(mn)$. □

Unlike in the decreasing setting, not every Nash equilibrium (NE) of the decreasing game is a SE. Consider, for example, an instance with two facilities $\{m_1, m_2\}$ and two players, where the cost of a facility is shared equally between its users. The cost of m_1 is 2, and the cost of m_2 is 1. Both facilities are available to both players. Then, the profile (m_1, m_1) is a NE, since each player cannot decrease his cost of 1 by deviating alone; but it is not a SE, since if both players deviate to m_2, their cost decreases to 0.5.

We will now illustrate why our proof of Thm. 2 wouldn't hold in the general case (where the strategies don't have to be singletons). The situation is best illustrated by an example. Fig.1 presents a graph of an instance of a network

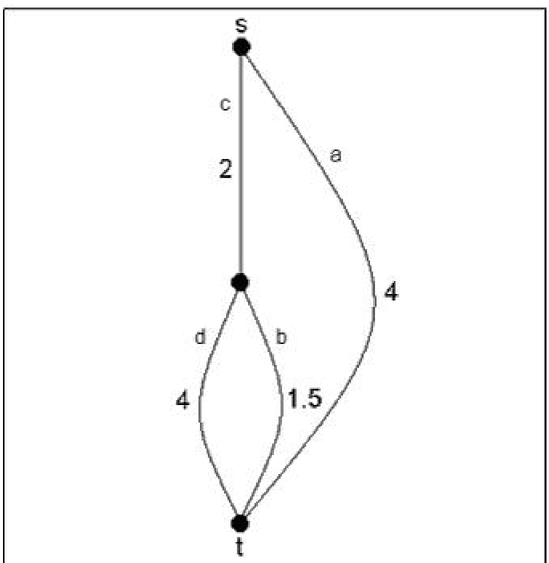


Fig. 1. SE doesn't exist

design game in the increasing setting: there are two agents, who both need to construct a path from s to t, using the edges available in the graph. The construction cost of each edge (the number near the edge) is shared equally between the agents. Each agent wants to minimize his construction cost; however, agent 1 cannot use edge a, and agent 2 cannot use edge b.

Our algorithm assigns $\{c, d\}$ to 1 and $\{c, d\}$ to 2, with a payoff of 3 each. This however is not an SE; in fact there is no SE in this game; to see this observe that playing $\{c, b\}$ is dominant for agent 1, and given that playing $\{a\}$ is dominant for agent 2, which leads to a payoff of (4,3.5), which is smaller than (3,3). Therefore, a SE does not exist in this instance.

3 Congestion Game Forms, Bad Configurations and Tree Representations

In this section we extend upon the definitions and notations introduced in [5] in order to provide some basic tools that will be useful for our characterization results.

A *congestion game form* is a tuple $F = (M, N, A)$ where M is the set of facilities, $N = \{1, ..., n\}$ is the set of players, and $A \subseteq 2^M$. A congestion game $G = (M, N, \{A_i\}, \{v_m(k)\})$ is said to be *derived* from F if $A = \bigcup_{i \in N} A_i$. Given a congestion game form F, one can derive from it a whole family of (monotone, increasing or decreasing) congestion games by assigning (monotone, increasing or decreasing) utility levels to the facilities and assigning specific strategy sets to the players. The congestion game form represents the underlying structure of the strategy spaces; for example, in the network design setting, it is the game graph. We say that a congestion game form F is *strongly consistent* if every monotone congestion game derived from F possesses a SE (we will always specify which setting, increasing or decreasing, is under discussion). We say that a congestion game form F is *strong-Nash equivalent* if in every monotone congestion game derived from F every NE is a SE. Similarly, we say that a congestion game form F is *correlated strongly consistent* if every monotone congestion game derived from F possesses a CSE; F is *correlated-strong equivalent* if in every monotone congestion game derived from F every SE is a CSE.

In this terminology, the results of section 2 state: if $F = (M, N, A)$ is a congestion game form in the decreasing setting in which A contains only singletons, then F is strong-Nash equivalent; if $F = (M, N, A)$ is a congestion game form in the increasing setting in which A contains only singletons, then F is strongly consistent.

We are interested in a property of A which is both necessary and sufficient for F to be strongly consistent.

Let $A \subseteq 2^M$. A *d-bad configuration* in A is a tuple $(x, y; X, Y, Z)$ where:

$$x, y \in M$$
$$X, Y, Z \in A$$

and the following relations hold:

$$\begin{array}{ll} x \in X & y \notin X \\ x \notin Y & y \in Y \\ x \in Z & y \in Z \end{array}$$

Thus, two facilities x, y give rise to a d-bad configuration if there is a strategy that uses both x and y, there is a strategy that uses x without y, and there is a strategy that uses y without x. We call $A \subseteq 2^M$ *d-good* if it does not contain a d-bad configuration.

An *i-bad configuration* in A is a tuple $(x, y; X, Y, Z)$ where:

$$\begin{array}{l} x, y \in M \\ X, Y, Z \in A \end{array}$$

and the following relations hold:

$$\begin{array}{ll} x \in X & y \notin X \\ x \notin Y & \\ x \in Z & y \in Z \end{array}$$

Thus, two facilities x, y give rise to an i-bad configuration if there is a strategy that uses both x and y, there is a strategy that uses x without y, and there is a strategy that avoids x (with, or without using y). In Fig.1, for example, the edges c, d give rise to a i-bad configuration. We call $A \subseteq 2^M$ *i-good* if it does not contain an i-bad configuration. In particular, a d-bad configuration is an i-bad configuration, so A is i-good implies that A is d-good.

By an *M-tree*, we shall mean the following:

- a tree with a root r
- a labeling of the nodes of the tree (except r) by elements of M; not all elements of M must appear, but each can appear at most once
- a designated subset D of the nodes, which contains all terminal nodes (and possibly other nodes as well).

An example of an M-tree appears in Fig. 2.

Given an M-tree T, we associate with it a set A of strategies on M, as follows: to each node in D there corresponds a strategy in A consisting of the labels which appear on the path from r to that node. For instance, if T is the M-tree depicted in Fig. 2, then $A = \{\{a, b\}, \{a, b, c\}, \{a, d\}, \{a, e, f\}, \{a, e, g\}, \{h\}, \{h, i\}, \{h, j, k\}\}$. If $r \in D$, it means that $\emptyset \in A$. If A is obtained from T in this way, we say that T is a *tree-representation* of A.

Lemma 1. *[5] Let A be a nonempty set of strategies on M. Then A is d-good if and only if it has a tree-representation.*

Given a congestion game form $F = (M, N, A)$, a tree representation of A gives us a convenient method of reasoning about equilibria, since in this case any congestion game derived from F is isomorphic to a tree-game: a game where

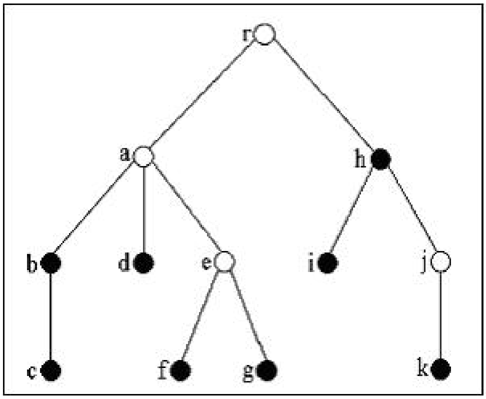


Fig. 2. An M-tree. The labels appear to the left of the nodes; the nodes in D are blackened.

given an M-tree, players must build a path from r to one of the nodes in D, and the strategies of each player can be represented by the subset of D that he is allowed to use.

Given a tree representation of A, a non-leaf node v is called *split* if $v \in D$ or v has more than one child (intuition: a path from r that reaches v has more than one way to be extended to a path leading to a node in D). A tree representation of A is called *simple* if no path from r to a node in D contains more than one split node. The general case of a simple tree representation is depicted in Fig. 3.

We can now prove:

Lemma 2. *Let A be a nonempty set of strategies on M. Then A is i-good if and only if it has a simple tree-representation.*

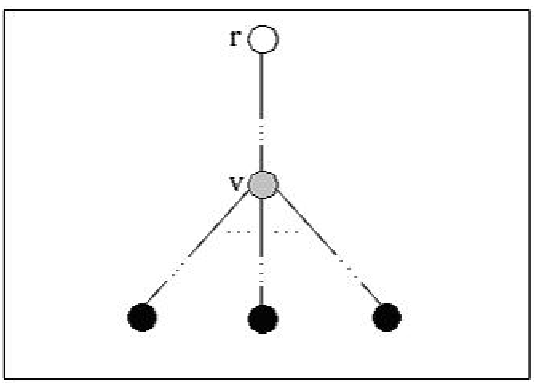


Fig. 3. The general case of a simple M-tree. The grayed node v can belong to D and can be outside of D; The dots represent chains of nodes (could be empty), where no intermediate node belongs to D.

Proof: Suppose A has a simple tree representation, and suppose, for contradiction, that A also has an i-bad configuration $(x, y; X, Y, Z)$. Since $x, y \in Z$, both x and y appear on the path from r to a node in D that corresponds to Z; also, x must occur above y on this path, since a path that corresponds to X contains x, but not y. This means that a split node v must exist between x and y on the path of Z; but since the path corresponding to Y doesn't include x, this means another split node v' must exist above x as well. So, the path of Z contains two split nodes – contradiction.

Suppose now that A is i-good. Then, it is also d-good, so by Lemma 1 A has a tree representation. Suppose, for contradiction, that this tree representation is not simple; i.e. there exists a path (corresponding to some strategy Z in A) with two split nodes, x and x'. W.l.o.g., suppose x' is above x. Since x is split, it has a child, y. Since x' is split and is above x, there exists a path (corresponding to some strategy Y) that doesn't contain x. Since x is split, there exists a path (corresponding to some strategy X) that contains x, but not y. Thus, $(x, y; X, Y, Z)$ is an i-bad configuration – contradiction. □

4 Structural Characterization of Existence of SE

Recall the following:

Theorem 3. *[5] Consider the monotone decreasing setting, and let F be a congestion game form with $n \geq 2$. Then, F is strongly consistent if and only if A is d-good.*

We now show:

Theorem 4. *Consider the monotone increasing setting, and let $F = (M, N, A)$ be a congestion game form, with $n \geq 2$. Then, F is strongly consistent if and only if A is i-good; moreover, if A is i-good, a SE can be efficiently computed.*

Proof: Let $F = (M, N, A)$ be a congestion game form, and suppose A is i-good. As we know from Lemma 2, A has a simple tree representation. In the general case, a simple M-tree has the form depicted in Fig. 3: a single chain descending from r to a single split node v, from which descend several chains to nodes in D. Each such chain (including the one from r to v) might be empty. What it means in terms of strategies in A, is that: $\exists C \subseteq M$ s.t. $\forall S_1 \neq S_2 \in A : S_1 \cap S_2 = C$; i.e. except one common subset of facilities that all players have to choose, their allowed strategies are either equal or disjoint. We claim that this case is strategically isomorphic to the case of singleton strategies. First, since all users must choose all the facilities in C, these facilities don't influence the game and can be removed. Then, A becomes pair wise disjoint collection of subsets of facilities; therefore, each such subset $S \in A$ can be replaced by a single new facility m_S, with $v_{m_S}(k) = \sum\limits_{m \in S} v_m(k)$ for every k. Now, we have an equivalent game with only singleton strategies allowed; as we know from Thm. 2, such game has a strong equilibrium which can be efficiently computed.

Now suppose $F = (M, N, A)$ is a congestion game form where $n \geq 2$ and A contains an i-bad configuration $(x, y; X, Y, Z)$. We must show that F is not strongly consistent; i.e. there exists a monotone increasing congestion game G derived from F which doesn't possess a SE. To construct such game, we must specify the exact strategy spaces $A_1, ..., A_n$ so that $A = \bigcup_{i \in N} A_i$, and specify monotone increasing $v_m(k)$ for each $m \in M$. We can express A as a union of four disjoint sets $A = A_X \cup A_Y \cup A_Z \cup A_\emptyset$, where:

$$\begin{aligned} A_X &= \{S \in A | S \cap \{x, y\} = \{x\}\}, \\ A_Y &= \{S \in A | S \cap \{x, y\} = \{y\}\}, \\ A_Z &= \{S \in A | S \cap \{x, y\} = \{x, y\}\}, \\ A_\emptyset &= \{S \in A | S \cap \{x, y\} = \emptyset\} \end{aligned}$$

From the i-bad configuration definition, we know that $A_X,\ A_Z$ and $A_Y \cup A_\emptyset$ are not empty (since $X \in A_X,\ Z \in A_Z$ and $Y \in A_Y \cup A_\emptyset$). We consider two distinct cases:

1. $A_\emptyset = \emptyset$. In this case, G is specified as follows:

$$A_1 = A_X \cup A_Z,\ A_2 = A_Y \cup A_Z,\ A_3 = ... = A_n = A_Z$$

$$v_m(k) = \begin{cases} -3,\ m \in \{x, y\},\ k < n \\ -1,\ m \in \{x, y\},\ k = n \\ 0,\ m \notin \{x, y\} \end{cases}$$

Since both facilities x,y have negative utility no matter how many players choose them, it is a strictly dominant strategy for players 1,2 to choose a subset that contains only one facility among x,y. Therefore, in any NE of the game (pure or mixed) player 1 will choose a strategy in A_X and player 2 will choose a strategy in A_Y, so both will gain -3. However, if both players deviate to a strategy in A_Z, both will gain -2. Therefore, any NE of the game is not strong, i.e. SE does not exist.

2. $A_\emptyset \neq \emptyset$. In this case, G is specified as follows:

$$A_1 = A_X \cup A_Z,\ A_2 = A_Y \cup A_Z \cup A_\emptyset,\ A_3 = ... = A_n = A_\emptyset$$

$$v_m(k) = \begin{cases} 2,\ m = x,\ k < 2 \\ 4,\ m = x,\ k \geq 2 \\ \text{-}5,\ m = y,\ k < 2 \\ \text{-}1,\ m = y,\ k \geq 2 \\ 0,\ m \neq x, y \end{cases}$$

Since the facility y always yields a negative utility, it is strictly dominant for player 1 to choose a strategy in A_X. Therefore, in any NE player 2 will choose a strategy in $A_\emptyset$; so, in any NE (pure or mixed) they will gain 2 and 0 respectively. But then, if both players deviate to a strategy in A_Z, they will gain 3; So in this case too a SE does not exist, which completes our proof. □

The above results suggest that in the monotone increasing setting there are (in a sense) strictly less games which possess SE than in the monotone decreasing setting (unless we consider the symmetric case). In the setting where congestion has a negative effect, the whole class of "tree games" is guaranteed to have a SE; in the increasing setting, where congestion has a positive effect on the players, SE is guaranteed to exist only in a strict subset of the corresponding structures. As shown in the proof of Thm. 4, this set of structures is strategically isomorphic to the singletons setting. This result is (perhaps) a bit surprising, since it contradicts the intuition – the players "help" each other instead of "harming" each other, but despite of that the setting is less stable, in the sense that there are less strong equilibria. Nevertheless, as we will later see, the decreasing case is not more stable than the increasing case when we consider CSE.

5 Structural Characterization of Existence of CSE

When we attempt to replace the notion of SE with the much stronger notion of CSE, many of the previous results no longer hold. It is easy to see that in the monotone decreasing setting even the following simple example with two players in a symmetric singleton strategies game doesn't possess a CSE. Consider two facilities $\{m_1, m_2\}$ with $v_1(m_1) = -2$, $v_2(m_1) = -4$, $v_1(m_2) = -5$, $v_2(m_2) = -10$. Both facilities are available to both players. Here, playing m_1 is a strictly dominant strategy for both players; however, (m_1, m_1) is not a CSE, since a deviation to the correlated profile $\{\frac{1}{2}(m_1, m_2), \frac{1}{2}(m_2, m_1)\}$ strictly increases the payoff of both players (each player will suffer a cost of 3.5 instead of a cost of 4). Therefore, a CSE doesn't exist in this example (which is a variant of the Prisoner's Dilemma). In fact, we can generalize this example to the following statement:

Proposition 1. *Consider the monotone decreasing setting, and let* $F = (M, N, A)$ *be a congestion game form with* $n \geq 2$ *and* $|A| \geq 2$. *Then,* F *is not correlated strongly consistent.*

Proof (sketch): The proof is in the same spirit as the proof of Theorem 4. It is therefore omitted due to lack of space.
In the increasing setting, however, we see that our results still hold; moreover, we can prove the following strong claim:

Theorem 5. *Consider the monotone increasing setting, and let* $F = (M, N, A)$ *be a congestion game form. Suppose* A *is i-good. Then* F *is correlated-strong equivalent.*

Proof: From our previous observations we know that if A is i-good, we can assume w.l.o.g. that A has only singleton strategies. So we must show that any SE of a monotone increasing congestion game where all strategies are singletons is also a CSE. Suppose, for contradiction, that $a^* \in A$ is a SE of a monotone increasing congestion game with singleton strategies, and it is not a CSE. Therefore, there exists a non-empty coalition $S \subseteq N$ and a correlated mixed strategy $a_S \in \Delta(\prod_{i \in S} A_i)$ such that $\forall i \in S \;\; U_i(a_S, a^*_{-S}) > U_i(a^*)$. Let

i be a player in S with maximal utility in a^*: $\forall j \in S \quad u_i(a^*) \geq u_j(a^*)$. Since $U_i(a_S, a^*_{-S}) > U_i(a^*)$, there must be a realization $b_S \in \prod_{j\in S} A_j$ of a_S such that $u_i(b_S, a^*_{-S}) > u_i(a^*)$. Since the game contains only singleton strategies, $u_i(b_S, a^*_{-S}) = v_m(\sigma_m(b_S, a^*_{-S}))$ for a resource m such that $b_i = \{m\}$. Let $T = \{j \in S | b_j = b_i\}$. T is non-empty, since $i \in T$. From the definition of T and since $T \subseteq S$ it holds that $\sigma_m(b_S, a^*_{-S}) \leq \sigma_m(b_T, a^*_{-T})$; therefore, since the game is monotone-increasing, $u_i(b_T, a^*_{-T}) \geq u_i(b_S, a^*_{-S}) > u_i(a^*)$. Since $\forall j \in T, u_j(b_T, a^*_{-T}) = u_i(b_T, a^*_{-T})$, we have that $\forall j \in T, u_j(b_T, a^*_{-T}) = u_i(b_T, a^*_{-T}) \geq u_i(b_S, a^*_{-S}) > u_i(a^*) \geq u_j(a^*)$, which contradicts our assumption that a^* is a SE. $\square$

This brings us to the following result:

Theorem 6. *Consider the monotone increasing setting, and let $F = (M, N, A)$ be a congestion game form, with $n \geq 2$. Then, F is correlated strongly consistent if and only if A is i-good; moreover, if A is i-good, a CSE can be efficiently computed.*

Proof: $\Leftarrow$ Follows from Thms. 4 and 5.

$\Rightarrow$ The proof is similar to the proof of this direction in Thm. 4, observing that the counter examples given there are solved via elimination of strictly dominated strategies, and therefore don't posses a CSE. $\square$

Notice that while the set of congestion game forms that are strongly consistent in the increasing case is a strict subset of the set of congestion game forms that are strongly consistent in the decreasing case, we get inclusion in the other direction when considering correlated-strong consistency.

6 Further Work

One interesting question is whether further common restrictions, e.g. linearity, on the utility functions may have significant effects on the existence of SE and CSE. A related aspect has to do with restrictions on the utility functions to be only positive or only negative. Our initial study suggests that using such assumptions (in addition to monotonicity) one can slightly expand the set of situations where SE and/or CSE exist, but only in a very esoteric manner. Other aspects of SE and CSE, such as uniqueness and Pareto-optimality are also under consideration.

References

1. E. Anshelevich, Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. 45th IEEE Symp. on Foundations of Computer Science*, pages 59–73, 2004.
2. R.J. Aumann. Acceptable points in general cooperative n-person games. In A.W. Tucker and R.D. Luce, editors, *Contribution to the Theory of Games, Vol. IV, Annals of Mathematics Studies, 40*, pages 287–324. 1959.

3. R.J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.
4. R. Holzman and N. Law-Yone. Network structure and strong equilibrium in route selection games. *Mathematical Social Sciences*, 46:193–205, 1997.
5. R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, 21:85–101, 1997.
6. C.H. Papadimitriou. Algorithms, Games, and the Internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–753, 2001.
7. R.W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
8. T. Roughgarden. Selfish routing with atomic players. In *Proc. The 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Vancouver, Canada*, pages 973–974, 2005.
9. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, April 2002.

The Equilibrium Existence Problem in Finite Network Congestion Games

Igal Milchtaich

Department of Economics, Bar-Ilan University,
Ramat Gan 52900, Israel
milchti@mail.biu.ac.il

Abstract. An open problem is presented regarding the existence of pure strategy Nash equilibrium (PNE) in network congestion games with a finite number of non-identical players, in which the strategy set of each player is the collection of all paths in a given network that link the player's origin and destination vertices, and congestion increases the costs of edges. A network congestion game in which the players differ only in their origin–destination pairs is a potential game, which implies that, regardless of the exact functional form of the cost functions, it has a PNE. A PNE does not necessarily exist if (i) the dependence of the cost of each edge on the number of users is player- as well as edge-specific or (ii) the (possibly, edge-specific) cost is the same for all players but it is a function (not of the number but) of the total weight of the players using the edge, with each player i having a different weight w_i. In a parallel two-terminal network, in which the origin and the destination are the only vertices different edges have in common, a PNE always exists even if the players differ in either their cost functions or weights, but not in both. However, for general two-terminal networks this is not so. The problem is to characterize the class of all two-terminal network topologies for which the existence of a PNE is guaranteed even with player-specific costs, and the corresponding class for player-specific weights. Some progress in solving this problem is reported.

Keywords: Congestion games, network topology, heterogeneous users, existence of equilibrium.

1 Introduction

1.1 Background

The theoretical study of congestion in networks began in the 1950's, at which time it was concerned mostly with transportation networks. The traffic flow was postulated to be at a so-called Wardrop equilibrium [30], in which the travel time on all used routes is equal, and less than or equal to that of a single vehicle on any unused route. An important milestone was the publication of Beckmann et al.'s book [3], which (under certain simplifying assumptions) presented the equilibrium as the optimal solution of a certain convex programming problem. In these authors' setting, users are

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 87 – 98, 2006.
© Springer-Verlag Berlin Heidelberg 2006

nonatomic in the sense that the effect of any single user on the others is negligible. Congestion games with a finite number of players, each with a non-negligible effect on the others, were first presented by Rosenthal [24]. He constructed what is now called an exact potential function on the space of strategy profiles and showed that every maximum point of the potential is a pure strategy Nash equilibrium (PNE) in the game. This is because, whenever a single player changes strategy, the change in that player's payoff is equal to the change in the potential function. Monderer and Shapley [19] showed that Rosenthal's games are in fact the only finite games for which an exact potential exists. Thus, any (finite) potential game can be presented as a congestion game, in which there is a finite set of common facilities and the strategy space of each player consists of subsets of facilities. The payoff from using each facility j depends only on the number of players whose chosen subset includes j. A special case of this is a network congestion game, in which the facilities correspond to the edges of a graph; the strategy space of each player is the collection of all directed paths, or routes, connecting two distinguished vertices, the player's origin and destination vertices; and the cost, or disutility, of using each edge is determined as a nondecreasing function by the flow on the edge. In Rosenthal's setting, players may differ only in their origin or destination vertices. If they are (i) differently effected by congestion, that is, have different cost functions, or (ii) have different weights, or congestion impacts, then the game is generally not a potential game and hence not a congestion game in Rosenthal's sense. For example, with player-specific costs, best-response cycles can occur if there are at least three players and at least three edges in the network [1,15]. Such cycles cannot occur in a potential game. Nevertheless, a network congestion game with either player-specific costs or weights, but not both, is guaranteed to have a PNE in the important special case of a parallel two-terminal network, i.e., one in which all players have the same origin–destination pair (in other words, a single-commodity network), which are the only vertices any two edges have in common [15]. In the case of player-specific weights, the result holds even if the weights are also edge-specific ("unrelated machines" [6]), and more generally, if the cost of each edge is an arbitrary nondecreasing function of the *set* of players using it [7]. A PNE does not necessarily exist, even in a parallel network, if the players have both player-specific costs and weights or if they are *positively* affected by congestion and the effects are player-specific [12,15,16].

The topological restriction on the network cannot be dispensed with. Libman and Orda [14] (see also [8]) gave an example of a two-terminal network with six edges for which there is a network congestion game with two players, one with twice the *weight* of the other, which does not have a PNE. They raised as an interesting subject for further research the problem of identifying non-parallel networks in which this is not possible, adding that series-parallel networks can be especially interesting. Konishi [11] gave an example of a different two-terminal network for which there is a three-player network congestion game with player-specific *costs* that does not have a PNE. He noted the similarity between the topological conditions for the existence of PNE and those for the *uniqueness* of the equilibrium in network congestion games with a *continuum* of non-identical players. (For such nonatomic games, existence of equilibrium is not an issue, since it is guaranteed by very weak assumptions on the cost functions [29].) Specifically, a parallel network is a sufficient condition in both cases.

The *equilibrium existence problem* that these authors point to is the identification of all two-terminal networks with the *topological existence property*: for any nondecreasing cost functions, with player-specific costs or weights (but not both), at least one PNE exists. This problem is substantially different from that of identifying classes of *cost functions* for which a PNE exists for all network topologies. An example of such a class is linear (more precisely, affine) functions. Regardless of the network topology, if all the players have identical, linear cost functions, a PNE always exists even with player-specific weights [8]. A similar distinction between the influences of the network topology and of the functional form of the cost functions applies also to the properties of efficiency and uniqueness, which are described below.

Efficiency of the equilibrium in a network congestion game has more than one possible meaning. It may refer to Pareto efficiency, that is, the impossibility of altering the players' route choices in a way that benefits them all, or to some aggregate measure of performance, such as the total cost or the cost of the worst route. In the latter case, the ratio between the chosen measure of performance at the worst Nash equilibrium and that at the social optimum is called the coordination ratio [13]. In nonatomic network congestion games with identical players, this ratio can be arbitrarily large for general cost functions, but it is bounded for certain families of functions, e.g., linear ones [28]. The least upper bound, dubbed the price of anarchy [22], is virtually independent of the network topology [25]. By contrast, the Pareto efficiency of the equilibria in nonatomic congestion games strongly depends on the topology. For a two-terminal network G, the equilibria are *always* Pareto efficient if and only if G has *linearly independent routes*, meaning that each route has an edge that is not in any other route [18]. In a sense, equilibria that are not Pareto efficient may occur in only three known two-terminal "forbidden" networks, which are the minimal ones *without* linearly independent routes. These results hold both with identical players and with player-specific cost functions.

For network congestion games with an arbitrary but finite number of players, who have identical cost functions but possibly different weights, the network topology is still irrelevant for the cost of anarchy if the players may split their flow among multiple routes [26]. However, if the flow is unsplittable and only pure strategies are allowed [27], the (so-called *pure*) cost of anarchy for linear cost functions apparently does depend on the network topology [2]. It also depends on whether or not the weights are player-specific [4]. (In the weighted case, the pure cost of anarchy only refers to games in which a PNE exists.) The topological conditions for Pareto efficiency of the equilibrium in the unsplittable, pure-strategy case were found by Holzman and Law-yone [10]. These conditions are very similar to those applying to nonatomic network congestion games if the players are identical. However, if the players have different cost functions, there are virtually no topological conditions that guarantee Pareto efficiency: Pareto inefficient (and non-unique) equilibria occur in all two-terminal networks with at least two routes.

The problem of the *topological uniqueness* of the equilibrium is relevant for nonatomic network congestion games in which different players may have different cost functions. (With identical players, the equilibrium is always essentially unique. With a finite number of non-identical players and unsplittable flow, it is virtually impossible to guarantee uniqueness.) The class of all two-terminal networks for which uniqueness is guaranteed is defined by five simple kinds of networks, called the *nearly parallel* networks [17]. The complementary class of all two-terminal networks for which player-specific costs can result in multiple equilibria consists of all the networks in which one of four known “forbidden” networks is embedded. These results can be extended to network congestion games with finitely many players and splittable flow [23].

1.2 Results

This paper presents some partial results pertaining to the equilibrium existence problem, which is to identify the topological conditions guaranteeing the existence of at least one PNE in every network congestion game with player-specific costs or weights. The class of two-terminal network topologies for which the existence of a PNE is guaranteed is extended in a nontrivial manner beyond parallel networks. On the other hand, several new topologies are presented for which a PNE does not always exist. These results narrow the search for the problem’s solution.

2 The Model

A *two-terminal network* (*network*, for short) G is defined in this work as a directed graph together with a distinguished pair of distinct vertices, the *origin* o and *destination* d, such that each vertex and each edge belong to at least one (directed) path $r = e_1 e_2 \cdots e_m$ linking o and d. Such a path is called a *route*. By definition, the terminal vertex of each edge e_j in a path except for the last one coincides with the initial vertex of the next edge, and all the vertices (and necessarily all the edges) are distinct [5]. This implies that loops are not allowed in G.[1] However, multiple edges are allowed.

For a given network G, a (finite) *network congestion game* is an n-player game, with $n \geq 1$, in which the strategy set of each player is the *route set* $\mathcal{R}$ of G, which consists of all the routes in the network. A *strategy profile* specifies a particular choice of route for each player. Players may differ from each other in their weight or cost functions.[2] The *weight* $w_i > 0$ of a player i is a measure of i’s congestion impact. For an edge e in G, the total weight of the players whose routes include e, denoted by f_e, is the *flow* (or load) on e. The flow affects the *cost* of traversing e, which, for each player i, is given by a nonnegative, nondecreasing *cost function* $c_e^i : [0, \infty) \to [0, \infty)$. Thus, if the flow on e is f_e, its cost for i is $c_e^i(f_e)$. If the players have identical cost functions, this notation may be simplified to $c_e(f_e)$. If they all have the same weight, it may be assumed without loss of generality that the weight is 1. The cost of each route in the network for a player is the sum of the costs of its edges. The player’s payoff in the game is the negative of this cost.

[1] Not allowing loops and other edges that do not belong to any route essentially involves no loss of generality, since such edges either cannot possibly be used or can only make a user’s way unnecessarily long.

[2] *Multi-commodity networks*, in which players may also have different origin–destination pairs, and hence different strategy sets, are not considered in this paper.

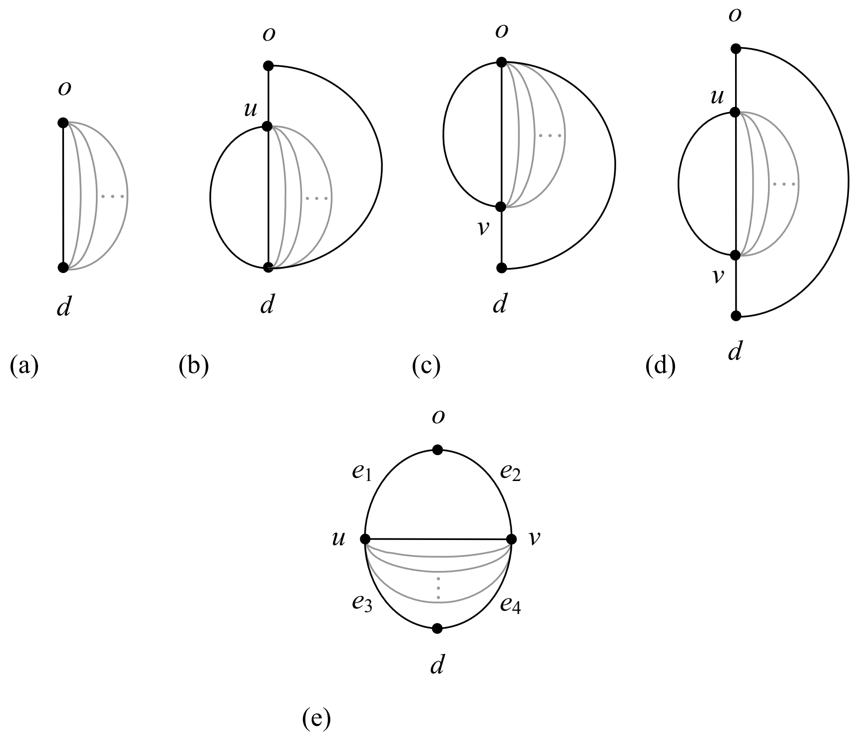


Fig. 1. Two-terminal networks with the topological existence property. In each network, the possible routes are the paths linking the origin o and destination d. Gray curves indicate optional edges. The directions of all the edges are unambiguous, except of those joining u and v in (e), which are assumed to be directed from u to v.

The following game theoretic terminology, which is not all standard, is used in this paper. A strategy profile is a *pure-strategy Nash equilibrium* (PNE) if none of the players can increase his payoff by unilaterally shifting to some other strategy. In other words, in a PNE each player plays a *best response* to the other players' strategies. The *superposition* of a finite number m of games with the same set of players is the game in which each of these players has to choose a strategy in each of the m games and his payoff is the sum of those in the m games [20]. Thus, the games are played simultaneously, but independently. Clearly, a strategy profile in the superposition of m games is a PNE if and only if it induces a PNE in each of these games. Two games Γ and Γ' with the same set of players and the same strategy set are *similar* if each player's payoff function in Γ is obtained from that in Γ' by adding to (or subtracting from) the latter a payoff function that depends only on the strategies of the other players. Similarity implies that the gain or loss for a player from unilaterally shifting from one strategy to another is the same in both games. Hence, it also implies that the games are *best-response equivalent*, i.e., a player's strategy is a best response to the others' strategies in one game if and only if this is so in the other game. Therefore, similar games have identical sets of PNEs.

3 Existence of PNE

If the players in a network congestion game differ in both their weights and cost functions, a PNE does not necessarily exist even in the case of a *parallel network* (Fig. 1(a)), which consists of one or more edges connected in parallel [15]. Therefore, for the notion of topological existence to be non-trivial, it is necessary to restrict attention to games in which players differ in only one of these respects. Doing so leads to the following positive result.

Theorem 1 [15]**.** *If G is a parallel network, then every network congestion game with player-specific costs or weights (but not both) has at least one PNE.*

For later reference, we note that this result can be slightly extended. Suppose that for each edge e there is pair of cost functions c_e and d_e (not necessarily different from zero) such that, for all players i and $x \geq w_i$,

$$c_e^i(x) = c_e(x) + d_e(x - w_i)\ . \tag{1}$$

Because of the second term in (1), if the players have different weights, they differ also in their cost functions. That term represents the effect of the other players using edge e on i; unlike the first term, it does not involve self-effect. Theorem 1 remains true if games with cost functions as in (1) are allowed. Such games will be referred to as network congestion games with player-specific weights *in the wide sense*. The existence of a PNE in this case can be proved by using the following algorithm (called greedy best response [9]). Players enter the game one after the other, ordered according to their weights from the highest to the lowest. Each player i chooses a route that is a best response to the route choices of the preceding players. It is not difficult to see that i's route remains a best response also after each of the remaining players i' enters the game, because $w_{i'} \leq w_i$ and the cost functions are nondecreasing. Therefore, the players' route choices constitute a PNE.

This constructive proof is specific to parallel networks; it cannot be extended in a straightforward manner to other network topologies. The same is true for all the other known proofs of Theorem 1, both for the case of player-specific costs and for player-specific weights [6,14,15]. In this respect, these proofs differ from that for the existence of PNE in network congestion games with identical players, for which the topology is irrelevant. Implicitly or explicitly, the latter proof uses the fact that every network congestion game Γ with identical players is similar (see the definition of similarity in Section 2) to some game Γ' in which the players have identical payoff functions, i.e., their payoffs are always the same [19,21,24]. This argument does not extend to network congestion games with player-specific costs. Even for parallel networks, such games are generally not similar, or even best-response equivalent, to games with identical payoffs. Indeed, best-response cycles may occur [15].

Nevertheless, Theorem 1 can be extended to other network topologies. An immediate extension is to allow the connection of several parallel networks in series. In this case, by Theorem 1, the "restriction" of every network congestion game with player-specific costs or weights to any of the constituent parallel networks has a PNE. As the following lemma shows, this implies that the game itself has a PNE, since it is the superposition of these "restricted" games (see Section 2).

Lemma 1. *If a network G can be obtained by connecting a finite number m of networks in series, then every network congestion game Γ is the superposition of m network congestion games, each of which is obtained by considering the edges in only one constituent network. If each of these games has a PNE, then so does Γ.*

Proof. This follows immediately from the fact that each route in G is the concatenation of m paths, each of which is a route in one constituent network, and conversely, every such concatenation constitutes a route in G, whose cost for each player is the sum of the costs of its m parts. ■

Less obviously, Theorem 1 can be extended to networks that are not even series-parallel, such as the Wheatstone network (Fig. 1(e)). This extension is based on the following result.

Lemma 2. *For each of the networks G in Fig. 1 there is a parallel network $\tilde{G}$ such that, for every network congestion game Γ for G with player-specific costs or player-specific weights in the wide sense, there is a network congestion game $\tilde{\Gamma}$ for $\tilde{G}$ with the same property that is similar to Γ.*

Proof. Suppose, first, that G is as in Fig. 1(e). Let $\tilde{G}$ be the parallel network obtained from G by *contracting* e_1 and e_4, that is, replacing each of these edges and its two end vertices with a single vertex [5]. There is a natural one-to-one correspondence between the route sets of G and $\tilde{G}$, which allows network congestion games for these two networks to be viewed as having the same strategy set. For a given network congestion game Γ for G, with weights (w_i) and cost functions (c_e^i), let $\tilde{\Gamma}$ be the game for $\tilde{G}$ with the same weights (w_i) and the cost functions $(\tilde{c}_e^i)$ defined as follows: If $e = e_2$, then $\tilde{c}_e^i(x) = c_{e_2}^i(x) - c_{e_1}^i(w_i + w - x) + c$ for all i, where $w = \sum_i w_i$ is the players' total weight and c is an arbitrary large constant (which serves to make the cost nonnegative). If $e = e_3$, then $\tilde{c}_e^i(x) = c_{e_3}^i(x) - c_{e_4}^i(w_i + w - x) + c$. Finally, if $e \neq e_2, e_3$, then $\tilde{c}_e^i(x) = c_e^i(x) + c$. If Γ is a game with player-specific costs but identical weights, then $\tilde{\Gamma}$ clearly has the same property. The same is true if Γ is a game with player-specific weights in the wide sense, since (1) implies that $\tilde{c}_{e_2}^i(x)$, for example, can be written as $c_{e_2}(x) - d_{e_1}(w - x) + c/2 + d_{e_2}(x - w_i) - c_{e_1}(w - (x - w_i)) + c/2$.

It remains to show that the games Γ and $\tilde{\Gamma}$ are similar. That is, for every choice of routes by the players and every player i, the difference between the cost in Γ and that in $\tilde{\Gamma}$ depends only on the routes of the other players. If i's route does not include e_2 or e_3, this difference is

$$c_{e_1}^i(w_i + w'_{-i}) + c_{e_4}^i(w_i + w''_{-i}) - c \,, \tag{2}$$

where w'_{-i} is the total weight of the players other than i whose route does not include e_2, and w''_{-i} is the corresponding weight for e_3. The same expression gives the difference between the costs in Γ and $\tilde{\Gamma}$ also if i's route includes either e_2 or e_3. Thus, the difference is independent of i's route, as had to be shown.

The above argument can easily be adapted for each of the other networks in Fig. 1. For networks G as in Fig. 1(b) and (c), $\tilde{G}$ is obtained by contracting only one edge. The network in (d) can be reduced to either of the previous two by moving one of the edges incident with the terminal vertices so that it becomes adjacent with the other such edge. Clearly, this rearrangement of edges does not affect the cost of any route.

Alternatively, the validity of the conclusion of the lemma for each of the other networks in Fig. 1 can easily be deduced from that for (e). ■

The assertion of Lemma 2 cannot be strengthened to *identity*, or isomorphism, between Γ and $\tilde{\Gamma}$. If the costs in Γ are player-specific, it may be qualitatively different from all network congestion games with the same property for parallel networks. For example, whereas games of the latter kind are always sequentially solvable [16], there are examples showing that Γ does not necessarily have this property. However, for present purposes, similarity is more than sufficient, since it implies that every PNE in $\tilde{\Gamma}$ is also a PNE in Γ. By Theorem 1 and the remark following it, at least one such PNE exists. Together with Lemma 1, this gives the following.

Theorem 2. *If G is one of the networks in Fig. 1 or can be obtained by connecting several of these networks in series, then every network congestion game with player-specific costs or weights (but not both) has a PNE.*

It is not known whether Theorem 2 can be extended to include also networks similar to those in Fig. 1(e) but with the reverse directions for some of the edges joining u and v. These networks and those in Fig. 1 are the directed versions of the *nearly parallel* networks [17], which are essentially the only two-terminal networks for which uniqueness of the equilibrium in nonatomic network congestion games with player-specific costs is guaranteed. This adds weight to Konishi's [11] observation that the conditions for topological existence (for a finite number of players with different cost functions) are similar to the conditions for topological uniqueness (for a continuum of such players). However, Theorem 2 leaves open the question of whether for every two-terminal network that is not nearly parallel there is a network congestion game with player-specific costs that does not have a PNE. Some results concerning this question are presented below.

4 Non-existence of PNE

The network in Fig. 2(d), which is obtained by connecting the Wheatstone network (Fig. 1(e)) in parallel with a single edge, differs substantially from the former in that network congestion games with player-specific costs or weights do not necessarily have a PNE. An example showing this for the case of different weights was given by Libman and Orda [14], and another one by Fotakis et al. [8]. These two examples are very similar to each other and to the next one; the different examples differ only in the cost functions.

Example 1. Two players simultaneously choose routes in the network in Fig. 2(d). The players have different weights, $w_1 = 1$ and $w_2 = 2$, but the same cost functions, given by $c_{e_1}(x) = 4x + 16$, $c_{e_2}(x) = 45$, $c_{e_3}(x) = 48$, $c_{e_4}(x) = x^3 - 9x^2 + 28x$, $c_{e_5}(x) = 16x$ and $c_{e_6}(x) = 65x$. For player 2, using e_5 is never optimal, since its cost is at least 32 whereas the difference between the costs of e_2 and e_1 is always less than that. Using e_6 is also never optimal for 2, since its cost is at least 130, which is always greater than $c_{e_1} + c_{e_3}$. This leaves player 2 with only two routes to choose from, and implies that 1 is the only player who may use e_5. The cost of that edge for player 1 is therefore 16, which is always less than the difference between the costs of e_2 and e_1, as well as the difference between the costs of e_3 and e_4. Therefore, using e_2 or e_3 is never optimal for player 1, which leaves him with only two possible routes, $r_1 = e_6$ and $r_2 = e_1\, e_5\, e_4$. If player 1's route is r_1 or r_2, the best-response route for 2 is $r_3 = e_1\, e_3$ or $r_4 = e_2\, e_4$, respectively. However, if player 2 uses r_3 or r_4, the best response for 1 is r_2 or r_1, respectively. Therefore, a PNE does not exist. Note that this would be true also if the constant functions c_{e_2} and c_{e_3} were replaced by sufficiently slowly increasing linear ones. However, if (the nonlinear) c_{e_4} were replaced by a linear function, a PNE *would* exist [8].

Essentially the same example shows that, in the network in Fig. 2(d), existence of a PNE is not guaranteed also with player-specific costs. This network is simpler than (i.e., it is a subnetwork of) the one used in Konishi's [11] example.

Example 2. This example is similar to the previous one, expect that the players differ not in their weights, which are given by $w_1 = w_2 = 1$, but in their cost functions, which are derived from those in Example 1 in the following manner: For each edge e, $c_e^1(1) = c_e(1)$, $c_e^2(1) = c_e(2)$, and $c_e^1(2) = c_e^2(2) = c_e(3)$. Clearly, the two-player game thus defined is identical to that in Example 1, and hence it does not have a PNE.

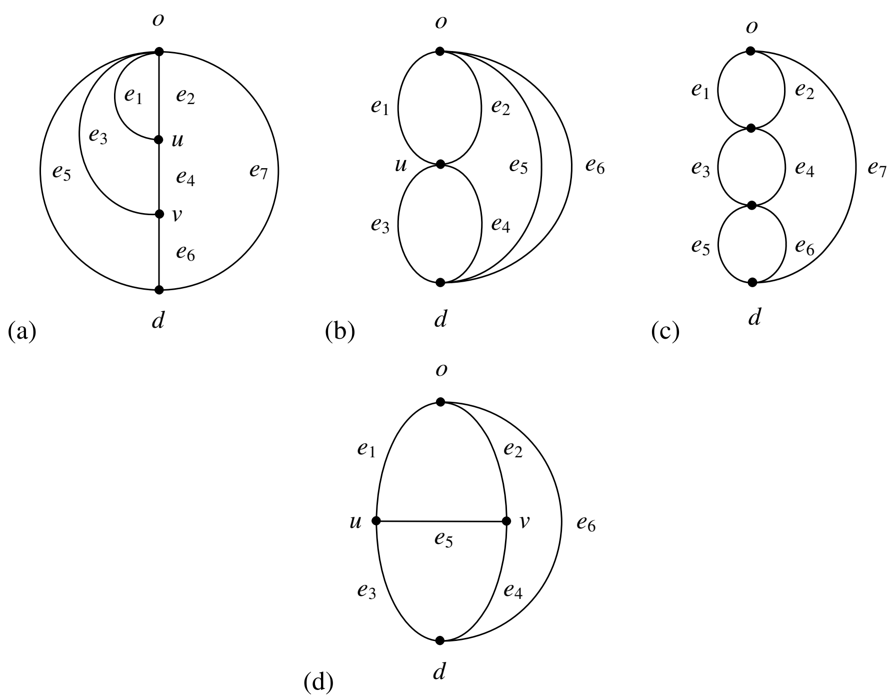


Fig. 2. Networks without the topological existence property. For each network, there is a network congestion game with player-specific cost functions or weights that does not have a pure-strategy Nash equilibrium. The edge joining u and v in (d) is directed from u to v.

Network congestion games without a PNE exist also for certain series-parallel networks. It is not known, however, whether these networks are the same for the cases of player-specific costs and player-specific weights. For the networks in Fig. 2(b) and (c), a PNE does not necessarily exist if the players have different cost functions. The next example concerns the former.

Example 3. Three players, all with weight 1, simultaneously choose routes in the network in Fig. 2(b). The cost of each edge for each player is given in Table 1. Effectively, each player has only one possible short route s (of length 1) and one long route l (of length 2). Player 1's l shares an edge (e_4) with 2's l, and his s coincides with 3's s. For player 1, the cost of s is less or greater than that of l if player 3 takes his l or s, respectively. Similarly, for player 2, s is preferable to l or the other way around if player 1 takes his l or s, respectively; and for 3, s is preferable to l or the other way around if player 2 takes his l or s, respectively. Clearly, this implies that a strategy profile in which everyone's route is optimal does not exist.

The network in the next example is not only series-parallel but is even ("extension-parallel" [10], or) a network with linearly independent routes [18].

Table 1. Cost functions for Example 3. For each player, the cost of each edge as a function of the flow on it is shown. Blank cells indicate prohibitively high costs.

	e_1	e_2	e_3	e_4	e_5	e_6
Player 1		$3x$		$3x$		$5x$
Player 2	x			$3x$	$6x$	
Player 3	x		x			$x/3+2$

Example 4. Three players, with weights $w_1 = 1$, $w_2 = 2$ and $w_3 = 4$, choose routes in the network in Fig. 2(a). The players' identical cost functions are given in Table 2. For player 3, there are effectively only two possible routes, $e_2\, e_4\, e_6$ and e_7. If the player chooses the former, then (regardless of what 1 does) player 2's best response is e_5, to which 1's best response is $e_1\, e_4\, e_6$. It is then better for player 3 to switch from $e_2\, e_4\, e_6$ (whose cost is 14) to e_7. However, if he chooses e_7, then (regardless of what 1 does) player 2's best response is $e_3\, e_6$, to which 1's best response is e_5. It is then better for player 3 to switch from e_7 to $e_2\, e_4\, e_6$ (whose cost is 12½). This proves that a PNE does not exist.

Table 2. Cost functions for Example 4. For each value of the flow on an edge, its cost (for all players) is shown. Blank cells indicate prohibitively high costs.

Flow	e_1	e_2	e_3	e_4	e_5	e_6	e_7
1	1	6	5	⅛	1	1	13
2		6	5½	¼	10	2	13
3		6	6	⅜	11	3	13
4		6		½		4	13
5				3		5	
6						6	
7						7	

These examples establish the possible non-existence of PNE also in many other networks, namely, those in which one or more of the four networks in Fig. 2 is embedded. For example, adding edges to any of the four networks would not make any difference, since extra edges can be effectively eliminated by assigning a very high cost to them. "Embedding" is used here in a somewhat generic sense. There are at least two different meanings for this term that may be relevant in the present context [17,18]. Very roughly, they correspond to the notions of a minor and topological minor of a graph [5].

Many two-terminal networks other than those mentioned above exist. Solving the equilibrium existence problem entails placing each of them either in the class of networks for which the existence of a PNE is guaranteed or in the class of those for which a network congestion game without a PNE exists. Whether this partition is the same for games with player-specific cost functions and for player-specific weights is not known.

References

1. Anantharam, V.: On the Nash Dynamics of Congestion Games With Player-Specific Utility. Proceedings of the 43rd IEEE Conference on Decision and Control (2004) 4673–4678
2. Awerbuch, B., Azar, Y., Epstein, A.: The Price of Routing Unsplittable Flow. Proceedings of the 37th Annual ACM Symposium on Theory of Computing (2005) 57–66
3. Beckmann, M., McGuire, C. B., Winsten, C. B.: Studies in the Economics of Transportation. Yale University Press, New Haven, CT (1956)
4. Christodoulou, G., Koutsoupias, E.: The Price of Anarchy of Finite Congestion Games. Proceedings of the 37th Annual ACM Symposium on Theory of Computing (2005) 67–73
5. Diestel, R.: Graph Theory. 3rd edn. Graduate Texts in Mathematics, Vol. 173. Springer-Verlag, Berlin Heidelberg New York (2005)
6. Even-Dar, E., Kesselman, A., Mansour, Y.: Convergence Time to Nash Equilibria. In: Baeten, J. C. M., Lenstra, J. K., Parrow, J., Woeginger, G. J. (eds.): Automata, Languages and Programming. Lecture Notes in Computer Science, Vol. 2719. Springer-Verlag, Berlin Heidelberg New York (2003) 502–513
7. Fabrikant, A., Papadimitriou, C., Talwar, K.: The Complexity of Pure Nash Equilibria. Proceedings of the 36th Annual ACM Symposium on Theory of Computing (2004) 604–612
8. Fotakis, D., Kontogiannis, S., Spirakis, P.: Selfish Unsplittable Flows. In: Diaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) Automata, Languages and Programming. Lecture Notes in Computer Science, Vol. 3142. Springer-Verlag, Berlin Heidelberg New York (2004) 593–605
9. Fotakis, D., Kontogiannis, S., Spirakis, P.: Symmetry in Network Congestion Games: Pure Equilibria and Anarchy Cost. In: Erlebach, T., Persiano, G. (eds.): Approximation and Online Algorithms. Lecture Notes in Computer Science, Vol. 3879. Springer-Verlag, Berlin Heidelberg New York (2006) 161–175
10. Holzman, R., Law-yone (Lev-tov), N.: Network Structure and Strong Equilibrium in Route Selection Games. Math. Social Sci. 46 (2003) 193–205
11. Konishi, H.: Uniqueness of User Equilibrium in Transportation Networks With Heterogeneous Commuters. Transportation Sci. 38 (2004) 315–330
12. Konishi, H., Le Breton, M., Weber, S.: Pure Strategy Nash Equilibrium in a Group Formation Game With Positive Externalities. Games Econom. Behav. 21 (1997) 161–182

13. Koutsoupias, E., Papadimitriou, C.: Worst-Case Equilibria. In: Meinel, C., Tison, S. (eds.): Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science. Lecture Notes in Computer Science, Vol. 1563. Springer-Verlag, Berlin Heidelberg New York (1999) 404.
14. Libman, L., Orda, A.: Atomic Resource Sharing in Noncooperative Networks. Telecommunication Sys. 17 (2001) 385–409
15. Milchtaich, I.: Congestion Games With Player-Specific Payoff Functions. Games Econom. Behav. 13 (1996) 111–124
16. Milchtaich, I.: Crowding Games are Sequentially Solvable. Internat. J. Game Theory 27 (1998) 501–509
17. Milchtaich, I.: Topological Conditions for Uniqueness of Equilibrium in Networks. Math. Oper. Res. 30 (2005) 225–244
18. Milchtaich, I.: Network Topology and the Efficiency of Equilibrium. Games Econom. Behav. 57 (2006), 321–346
19. Monderer, D., Shapley, L. S.: Potential Games. Games Econom. Behav. 14 (1996) 124–143
20. Morgenstern, O., von Neumann, J.: Theory of Games and Economic Behavior. 3rd edn. Princeton University Press, Princeton, NJ (1953)
21. Morris, S., Ui, T.: Best Response Equivalence. Games Econom. Behav. 49 (2004) 260–287
22. Papadimitriou, C. H.: Algorithms, Games, and the Internet. Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (2001) 749–753
23. Richman, O., Shimkin, N.: Topological Uniqueness of the Nash Equilibrium for Atomic Selfish Routing. Forthcoming in Math. Oper. Res.
24. Rosenthal, R. W.: A Class of Games Possessing Pure-Strategy Nash Equilibrium. Internat. J. Game Theory 2 (1973) 65–67
25. Roughgarden, T.: The Price of Anarchy is Independent of the Network Topology. J. Comput. System Sci. 67 (2003) 341–364
26. Roughgarden, T.: Selfish Routing With Atomic Players. Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (2005) 1184–1185
27. Roughgarden, T., Tardos, É.: How Bad is Selfish Routing? J. ACM 49 (2002) 236–259
28. Roughgarden, T., Tardos, É.: Bounding the Inefficiency of Equilibria in Nonatomic Congestion Games. Games Econom. Behav. 47 (2004) 389–403
29. Schmeidler, D.: Equilibrium Points of Nonatomic Games. J. Stat. Phys. 7 (1970) 295–300
30. Wardrop, J. G.: Some Theoretical Aspects of Road Traffic Research. Proceedings of the Institute of Civil Engineers, Part II 1 (1952) 325–378

First-Passage Percolation on a Width-2 Strip and the Path Cost in a VCG Auction

Abraham Flaxman[1], David Gamarnik[2], and Gregory B. Sorkin[3]

[1] Department of Mathematical Sciences, Carnegie Mellon University
Pittsburgh, PA
abie@cmu.edu
[2] MIT Sloan School of Management
Cambridge, MA
gamarnik@mit.edu
[3] Department of Mathematical Sciences, IBM T.J. Watson Research Center
Yorktown Heights, NY
sorkin@watson.ibm.com

Abstract. We study both the time constant for first-passage percolation, and the Vickery-Clarke-Groves (VCG) payment for the shortest path, on a width-2 strip with random edge costs. These statistics attempt to describe two seemingly unrelated phenomena, arising in physics and economics respectively: the first-passage percolation time predicts how long it takes for a fluid to spread through a random medium, while the VCG payment for the shortest path is the cost of maximizing social welfare among selfish agents. However, our analyses of the two are quite similar, and require solving (slightly different) recursive distributional equations. Using Harris chains, we can characterize distributions, not just expectations.

1 Introduction

The general topic of this paper is the random structure produced when a fixed graph is assigned edge costs independently at random. We will focus on a particular fixed graph, the n-long width-2 strip (defined below), and study some aspects of a minimum-cost path. In particular, we will consider the time constant for first-passage percolation, and the Vickery-Clarke-Groves (VCG) payment. These statistics attempt to describe two seemingly unrelated phenomena arising in physics and economics, respectively. However, our analyses of the two are quite similar.

First-passage percolation: First-passage percolation is a model of the time it takes a fluid to spread through a random medium [BH57, HW65, Kes87]. Mathematically, it is described by the shortest edge-weighted paths from an origin to every other point in a graph. For our purposes, the "time constant" is the limiting ratio of this length to the unweighted shortest path length n, as n tends to infinity. Previous research has derived upper and lower bounds for the time constant of first-passage percolation on the grid [SW78, Jan81, AP02] and on the random graph $G_{n,p}$ [HHM01]. For the easier case of the width-2 strip, we provide a method of exactly calculating the time constant for any discrete edge-length distribution; the method can also be used to provide arbitrarily

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 99–111, 2006.
© Springer-Verlag Berlin Heidelberg 2006

good bounds for any well-behaved continuous distribution, as we illustrate for the uniform distribution on $[0, 1]$. Our method is similar in spirit to to the Objective Method (or Local Weak Convergence) [Ald01, GNSar, AS04], and is also based on constructing a certain recursive distributional equation. The model in the present paper is considerably simpler due to the structure of the width-2 strip, which makes the underlying recursive distributional equation simply a Markov chain.

Because it is a Markov chain, the analysis for *discrete* edge-length distributions is straightforward: for a Bernoulli edge-length distribution $\mathrm{Be}(p)$ the incremental cost $\gamma(n)$ to go from stage $n-1$ to n has a unique stationary distribution with a simple, closed-form expression, and its expectation is the time constant in question. When the edge-length distribution is continuous (uniform, for example), replacing it with a rounded-down (respectively, rounded-up) discretized equivalent gives a lower (resp., upper) bound on the time constant, but no information about the incremental cost $\gamma(n)$. A subtly different approach gives stochastic lower and upper bound bounds on the incremental cost, and, separately, an analysis via *Harris chains* shows it to have a unique stationary distribution. The Harris-chain approach is well known in probability theory, but is worthy of greater attention in tangential fields.

VCG Payment: The Vickery-Clarke-Groves (VCG) mechanism applies to a setting in economics where each edge of a graph is controlled by a different selfish agent, and each agent has some private value describing the cost of using her edge [Vic61, Cla71, Gro73]. Anyone interested in buying a path in such a network is faced with the problem that an agent will lie about her edge cost if such a lie will yield her a higher payment. The VCG mechanism provides a solution to this problem in which payments to agents are structured to yield a cheapest path (maximizing social welfare) and so that each agent finds it in her best interest to reveal her true edge cost. The VCG mechanism was first applied to the shortest-path problem explicitly in [NR99].

Unfortunately, the VCG mechanism may pay more than the cost of the shortest path, and the overpayment can be large. The VCG overpayment can be large even in the case where the second-best path has cost close to that of the best path. See [AT02] for a detailed study of the worst-case behavior of the overpayment. Additional investigations of shortest paths in this setting appear in [MPS03, ESS04, CR04, Elk05].

It is possible that the worst-case bounds on the cost of the VCG mechanism are overly pessimistic. To investigate this, we compare the cost of the VCG mechanism with the shortest-path cost in the average-case setting (for the width-2 strip with random edge costs). Other average-case studies for completely different graphs appear in [MPS03, CR04, KN05], and real-world measurements appear in [FPSS02].

Generalizations: We rely on no special properties of the uniform distribution; the methods we use to analyze this edge-length distribution could equally well be applied to any well-behaved, bounded distribution.

For the $2 \times n$ strip, we show that it is not important whether edges parallel to the long direction must be traversed left-to-right or whether they can be traversed in either direction. Even for the $3 \times n$ strip, however, the distinction is important. For any fixed $m \geq 3$, our methods apply to the $m \times n$ strip in the left-to-right model (with more complicated recursive equations replacing (1) and (2)), but not to the undirected model.

2 The Model

Departing slightly from the usual convention, let $[n]$ denote $\{0, 1, \ldots, n-1\}$. Define the *infinite width-2 strip* to be the infinite graph whose vertex set is $[2] \times \mathbb{Z}$, and whose edges join vertices at Hamming distance 1, i.e., edges join (j, i) and (j', i') where $(|j - j'|, |i - i'|)$ is either $(0, 1)$ or $(1, 0)$. The *half-infinite strip* is the subgraph induced by $[2] \times \mathbb{Z}^{0,+}$, and an n*-long strip* is the (finite) subgraph induced by $[2] \times [n+1]$.

Let each edge e have a non-negative real weight $w(e)$. For each vertex v let $P(v)$ be the "shortest" (minimum-weight) path from $(0, 0)$ to v, and let $\ell(v)$ be the weight of this path. We consider two models: the "general-path" (GP) model where $P_{\mathrm{GP}}(v)$ may be any path from $(0, 0)$ to v, and the "left-right" (LR) model where $P_{\mathrm{LR}}(v)$ is restricted to be a left-to-right path. That is, $P_{\mathrm{LR}}(v)$ is the shortest path to v which does not traverse any edge from right to left, or, still more precisely, which contains no successive pair of vertices $(j, i), (j, i-1)$.

Suppose that the edge weights are drawn independently from some given distribution, such as $\mathrm{Be}(p)$ (the Bernoulli distribution with parameter p, where $X = 1$ with probability p and $X = 0$ w.p. $1-p$) or $U[0, 1]$ (the uniform distribution over the interval $[0, 1]$). Our first-passage percolation problem is simply to determine, for each of three types of strips, for a given distribution, and under the general-path or left-right model, the existence and value of the limiting time constant or "rate" of percolation,

$$\lim_{n \to \infty} \frac{\mathbb{E}\ell(0, n)}{n}.$$

We will also show that $\ell(0, n)/n$ almost surely converges to this value, and that the same statements hold for $\ell(1, n)$, with the same rate. Note that for all our purposes it suffices to determine path lengths up to an additive constant.

For convenience, for $a \leq b$, define $\mathrm{trunc}(x;\ a,\ b) := \max\{\min\{x, b\}, a\}$. Thus, $\mathrm{trunc}(x;\ a,\ b)$ is the "truncation" of x to the interval $[a, b]$: x if $a \leq x \leq b$; a if $x < a$; and b if $x > b$.

3 Shortest Paths

The following lemma shows that, up to an additive error of at most 2, distances to $(0, n)$ or to $(1, n)$, under any of the three graph models and the two distance models, are all equivalent.

Lemma 1. *Let G denote the infinite width-2 strip with an arbitrary, fixed set of edge weights in the range $[0, 1]$ (resp., random i.i.d. non-negative weights with expectation ≤ 1). Let H be the half-infinite restriction of G, and, for any $n \geq 0$, let K be the n-long restriction. Then, for any $j \in [2]$ and $i \in [n+1]$, the distances (resp., expected distances) $\ell_{\mathrm{LR}}(j, i)$ and $\ell_{\mathrm{GP}}(j, i)$, measured in the three graphs G, H, and K, span a range of at most 2.*

Proof. We will argue only the case of fixed edge weights; the random case proceeds identically. The cheapest GP path in G from $(0, 0)$ to whichever of $(0, i)$ and $(1, i)$ is cheaper is at most as expensive as any of the paths under consideration, because this

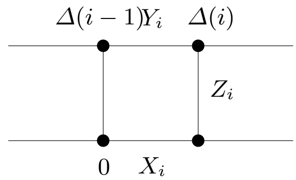


Fig. 1. Moving from Δ_{i-1} to Δ_i

path is the least constrained; denote this path $P^G_{\mathrm{GP}}(i)$. Fixing $j = 0$ (the $j = 1$ case is treated identically), the most constrained problem version is to find the cheapest LR path in K from $(0, 0)$ to $(0, i)$; the resulting path $P^K_{\mathrm{LR}}(0, i)$ is the most expensive one under consideration. By the nature of the width-2 strip, the restriction of $P^G_{\mathrm{GP}}(i)$ to K, unioned with the edges $\{(0, 0), (1, 0)\}$ and $\{(1, i), (0, i)\}$, is or includes a LR path in K from $(0, 0)$ to $(0, i)$. Thus $\ell^G_{\mathrm{GP}}(i) \leq \ell^K_{\mathrm{LR}}(0, i) \leq \ell^G_{\mathrm{GP}}(i) + 2$, and all the other lengths must also lie in this range. □

Because of Lemma 1, we will henceforth consider only LR paths, on the half-infinite strip H, to points $(0, n)$ and $(1, n)$. For convenience, we will write $\ell^H_{\mathrm{LR}}(1, i)$ simply as $\ell(1, i)$ and $\ell^H_{\mathrm{LR}}(0, i)$ as $\ell(0, i)$ or just $\ell(i)$. Define

$$\Delta(i) = \ell(1, i) - \ell(0, i).$$

For any $i > 0$, let X_i be the cost of the edge $\{(0, i - 1), (0, i)\}$ and Y_i the cost of $\{(1, i - 1), (1, i)\}$, and for any $i \geq 0$ let Z_i be the cost of $\{(0, i), (1, i)\}$. (See Figure 1 for visual reference.)

Observe that for $i > 0$,

$$\gamma(i) := \ell(i) - \ell(i - 1) = \min\{X_i, \Delta(i - 1) + Y_i + Z_i\} \tag{1}$$

$$\Delta(i) = \mathrm{trunc}(\Delta(i - 1) + Y_i - X_i;\ -Z_i,\ Z_i). \tag{2}$$

Since $\Delta(i - 1)$ depends only on values of X, Y, and Z with indices $i - 1$ and smaller, the four random variables $\Delta(i - 1)$, X_i, Y_i, and Z_i are mutually independent.

4 The Bernoulli Case

Suppose that all the random variables X_i, Y_i, and Z_i are i.i.d. with distribution $\mathrm{Be}(p)$, i.e., each is 1 with probability p and 0 w.p. $1 - p$.

A “stationary distribution” for equation (2) is a distribution for $\Delta(i - 1)$ giving rise to $\Delta(i)$ with the same distribution (though typically not independent).

Lemma 2. *When the edge weights are i.i.d. with distribution* $\mathrm{Be}(p)$, $0 \leq p < 1$, $\Delta(i)$ *is a Markov chain on* $\{-1, 0, 1\}$ *with a unique stationary distribution, namely* $\Delta = 1$ *w.p.* q*;* $\Delta = -1$ *w.p.* q*; and* $\Delta = 0$ *w.p.* $1 - 2q$*, where* $q = \frac{p^2}{1+3p^2}$.

Proof. All values in question are integral, and each $\Delta(i) \leq 1$, since $(1, i+1)$ may at worst be reached via $(0, i+1)$ at an additional cost of at most 1. Symmetrically, each $\Delta(i) \geq -1$. By the independence of $\Delta(i-1)$ from (X_i, Y_i, Z_i), $\Delta(i)$ is a Markov chain on the state space $\{-1, 0, 1\}$.

By definition, the stationary distribution of the Markov chain is independent of its initial state, so we may assume that $\Delta(0) = 0$. In this case, the initial state is symmetric, and so is the transition rule, so the distribution of $\Delta(i)$ is symmetric for every i.

From (2), if $Z_i = 0$ (which occurs w.p. $1-p$) then $\Delta(i) = 0$. Otherwise we have the following table of possibilities, their probabilities (including the probability p that $Z_i = 1$, and defining $q := \mathbb{P}[\Delta(i-1) = 1] = \mathbb{P}[\Delta(i-1) = -1]$), and the corresponding values of $\Delta(i)$:

$\Delta(i-1)$	X_i	Y_i	$\mathbb{P}$	$\Delta(i)$
1	0	0	$pq \cdot (1-p)^2$	1
1	0	1	$pq \cdot p(1-p)$	1
1	1	0	$pq \cdot p(1-p)$	0
1	1	1	$pq \cdot p^2$	1
0	0	0	$p(1-2q) \cdot (1-p)^2$	0
0	0	1	$p(1-2q) \cdot p(1-p)$	1
0	1	0	$p(1-2q) \cdot p(1-p)$	-1
0	1	1	$p(1-2q) \cdot p^2$	0
-1	0	0	$pq \cdot (1-p)^2$	-1
-1	0	1	$pq \cdot p(1-p)$	0
-1	1	0	$pq \cdot p(1-p)$	-1
-1	1	1	$pq \cdot p^2$	-1

If $\Delta(i-1) = 1$ and $\Delta(i) = 1$ are both to have probability q, we must have

$$q = pq \cdot (1-p)^2 + pq \cdot p(1-p) + pq \cdot p^2 + p(1-2q) \cdot p(1-p),$$

whose solution is $q = p^2/(1+3p^2)$. Thus if Δ is to be stationary, we must have, for this value of q, $\Delta = 1$ w.p. q; by symmetry $\Delta = -1$ w.p. q; and thus $\Delta = 0$ w.p. $1-2q$.

The Markov chain's transition matrix, which corresponds to the table above (plus the 12 omitted cases when $Z_i = 0$), is easily seen to be ergodic and aperiodic as long as $0 < p < 1$, and thus has a unique stationary distribution. When $p = 0$, $\Delta_i = 0$, deterministically, for all $i \geq 0$, which still happens to fit the same form. (When $p = 1$, $\Delta_i = 1$ deterministically: the sole exception.) □

Lemma 3. *When the edge weights are i.i.d. random variables with distribution* $\mathrm{Be}(p)$, $0 < p < 1$, $\gamma(i) = \ell(i) - \ell(i-1)$ *is a Markov chain on* $\{-1, 0, 1\}$ *with a unique stationary distribution: it is* -1 *w.p.* $p^2(1-p)^2/(3p^2+1)$*;* 1 *w.p.* $2p^2(1+p^2)/(3p^2+1)$*; and* 0 *with the remaining probability, giving* $\mathbb{E}(\gamma(i)) = p^2(1+p)^2/(3p^2+1)$.

Proof. That $\gamma(i)$ is a Markov chain, and is ergodic and aperiodic, follows as in the proof of the preceding lemma. Since $\gamma(i)$ depends on four independent random values all of whose distributions are known, calculating it is straightforward. Instead of presenting a table as above we divide it into a few cases. It is -1 iff $\Delta(i) = -1$, $Y_i = 0$, and

$Z_i = 0$ (the value of X_i is irrelevant), which occurs w.p. $q(1-p)^2$. It is 1 iff $X_i = 1$ and $\Delta(i) + Y_i + Z_i \geq 1$, the latter of which is satisfied if $\Delta(i) = -1$ and $Y_i = Z_i = 1$, if $\Delta(i) = 0$ and (Y_i, Z_i) is anything but $(0,0)$, or if $\Delta(i) = 1$, giving total probability $p\left[qp^2 + (1-2q)(1-(1-p)^2) + q\right]$. The rest of the calculation is routine. □

Theorem 4. *When the edge weights are i.i.d.* $\mathrm{Be}(p)$ *random variables, for any* p *with* $0 < p < 1$, *we have* $\lim_{n\to\infty} \frac{\mathbb{E}\ell(n)}{n} = \lim_{n\to\infty} \mathbb{E}\gamma(n) = p^2(1+p)^2/(3p^2+1)$, *and almost surely,* $\lim_{n\to\infty} \frac{\ell(n)}{n}$ *exists and has the same value.*

Proof. We have established that $\gamma(i)$ is an ergodic Markov chain with the unique stationary distribution described in Lemma 3. The ergodicity implies that almost surely

$$\lim_{n\to\infty} \frac{\ell(n)}{n} = \lim_{n\to\infty} \sum_{1\leq i\leq n} \frac{\ell(i) - \ell(i-1)}{n} = \lim_{n\to\infty} \sum_{1\leq i\leq n} \frac{\gamma(i)}{n} = \lim_{n\to\infty} \mathbb{E}(\gamma(n))$$
$$= p^2(1+p)^2/(3p^2+1).$$

Since the values $\gamma(n)$ are bounded almost surely (in fact surely, by unity, in absolute value), the almost sure convergence implies the convergence in expectation. □

5 Uniform Case: Expectation, Distribution, and Stationarity

What if X_i, Y_i, and Z_i have uniform distribution, $U[0,1]$? As in the previous sections, $\gamma(i)$ and $\Delta(i)$ are again Markov chains, but now with continuous state space. To avoid working with a continuous state space we will discretize it. First, for any (large) integer k, define $\underline{U}_k$ (resp., $\overline{U}_k$) to be the uniform distribution on the set $\{0, 1/k, \ldots, (k-1)/k\}$ (resp., $\{1/k, \ldots, (k-1)/k, 1\}$). Note that rounding a random variable $X \sim U[0,1]$ down and up to multiples of $1/k$ gives variables $\underline{X} \sim \underline{U}_k$ and $\overline{X} \sim \overline{U}_k$.

It is a simple observation that rounding all values X, Y and Z down or up gives (respectively) lower and upper bounds on any length $\ell(i)$. This allows bounds $\mathbb{E}\underline{\ell}_k(i) \leq \mathbb{E}\ell(i) \leq \mathbb{E}\overline{\ell}_k(i)$ to be computed much as in the Bernoulli case, via a finite Markov chain. By analogy with Theorem 4 and its proof (see full paper for details of the approach summarized in this paragraph), the first-passage percolation time constant can then be bounded by $\lim_{n\to\infty} \mathbb{E}[\underline{\ell}_k(n)/n] \leq \mathbb{E}[\gamma(n)] \leq \lim_{n\to\infty} \mathbb{E}[\overline{\ell}_k(n)/n]$. However, it is not true, for example, that $\gamma(i) \geq \underline{\ell}_k(i) - \underline{\ell}_k(i-1)$, and this natural approach thus characterizes γ's expectation but fails to say anything about its *distribution*. The distribution of γ may be of interest in itself, and that of Δ (which is essentially equivalent under (1)) is essential for computing quantities such as the expected VCG overpayment in the uniform model (paralleling its computation in the Bernoulli model in Section 6).

A different way of obtaining a finite Markov chain *does* provide us with random variables $\underline{\Delta}(n) \leq \Delta(n) \leq \overline{\Delta}(n)$, where $\underline{\Delta}(n)$ and $\overline{\Delta}(n)$ are given by finite Markov chains, allowing us to characterizes the distribution of $\Delta(n)$ and thereby giving access to any quantity of interest. Conceptually this method is quite different from the "make everything shorter / longer)" approach of the previous paragraph, but it is no harder: we simply derive what we want from the recurrence (2).

Letting $W = Y - X$, from (2),

$$\begin{aligned}\Delta' &= \operatorname{trunc}(\Delta + (Y - X);\ -Z,\ Z) \\ &= \operatorname{trunc}(\Delta + W;\ -Z,\ Z) \\ &\geq \operatorname{trunc}(\underline{\Delta} + \underline{W};\ -\overline{Z},\ \underline{Z}),\end{aligned}$$

where $\underline{Z}$, $\overline{Z}$, *etc.* are any lower and upper bounds on their respective quantities. Specifically, taking $\underline{Z}$, $\overline{Z}$, and $\underline{W}$ to be the rounded-down and rounded-up discretizations of the respective variables, for any $\underline{\Delta} \leq \Delta$, and for convenience defining $\epsilon = 1/k$, we have

$$\begin{aligned}\Delta' &\geq \operatorname{trunc}(\underline{\Delta} + \underline{W};\ -\underline{Z} - \epsilon,\ \underline{Z}) \\ &= \underline{W} + \operatorname{trunc}(\underline{\Delta};\ -\underline{Z} - \underline{W} - \epsilon,\ \underline{Z} - \underline{W}).\end{aligned}$$

Thus,

$$\underline{\Delta}' := \underline{W} + \operatorname{trunc}(\underline{\Delta};\ -\underline{Z} - \underline{W} - \epsilon,\ \underline{Z} - \underline{W}) \tag{3}$$

ensures $\Delta' \geq \underline{\Delta}'$, and is thus a recursive formula for lower bounds. Similarly,

$$\overline{\Delta}' := \underline{W} + \epsilon + \operatorname{trunc}(\overline{\Delta};\ -\underline{Z} - \underline{W} - \epsilon,\ \underline{Z} - \underline{W}) \tag{4}$$

defines a recursion for upper bounds. As initial conditions we set $\underline{\Delta}(0) = -1$ and $\overline{\Delta}(1) = 1$ (deterministically), ensuring $\underline{\Delta}(0) \leq \Delta(0) \leq \overline{\Delta}(0)$, whereupon following equation (3) to define $\underline{\Delta}(n) = \underline{\Delta}'$ from $\underline{\Delta}(n-1) = \underline{\Delta}$ and likewise for equation (4) ensures that for all n, $\underline{\Delta}(n) \leq \Delta(n) \leq \overline{\Delta}(n)$. From (1), trivially,

$$\gamma(i) \geq \min\{\underline{X}_i, \underline{\Delta}(i-1) + \underline{Y} + \underline{Z}_i\} \tag{5}$$

$$\gamma(i) \leq \min\{\overline{X}_i, \overline{\Delta}(i-1) + \overline{Y} + \overline{Z}_i\} \tag{6}$$

Theorem 6 will show that $\gamma(n)$ itself has a unique stationary distribution. Meanwhile, for any fixed k, the Markov chains for $\underline{\Delta}(n)$ and $\overline{\Delta}(n)$ are both well-behaved finite Markov chains, with stationary distributions we will call $\underline{\Delta}$ and $\overline{\Delta}$. Substituting $\underline{\Delta}$ and $\overline{\Delta}$ into (5) and (6) defines corresponding random variables $\underline{\gamma}$ and $\overline{\gamma}$, which are then stochastic lower and upper bounds on $\gamma(n)$. Distribution functions for $\underline{\gamma}$ and $\overline{\gamma}$ are plotted in Figure 2. By construction, the two curves never cross; the bounds are sufficiently good that they are largely visually indistinguishable. Of course, $\mathbb{E}\underline{\gamma} \leq \mathbb{E}\gamma \leq \mathbb{E}\overline{\gamma}$, and with $k = 150$ we obtain $0.4215 < \mathbb{E}\gamma < 0.4292$. Computational aspects are discussed in the full paper.

This method allows us to get arbitrarily good estimates of the distribution of $\Delta(n)$, for n large (and thus, by Theorem 6, of the stationary distribution of Δ). It suffices to show that, for k large, the stationary random variables $\underline{\Delta}$ and $\overline{\Delta}$ are arbitrarily near to one another: $d(\underline{\Delta}, \overline{\Delta}) = O(1/k)$, where we define the distance between continuous random variables X and Y as the area between their CDFs (cumulative density functions). (For any coupling of two variables X and Y, $\mathbb{E}[|X - Y|] \geq d(X, Y)$, with an optimal coupling giving equality.) Recall that $\epsilon = 1/k$.

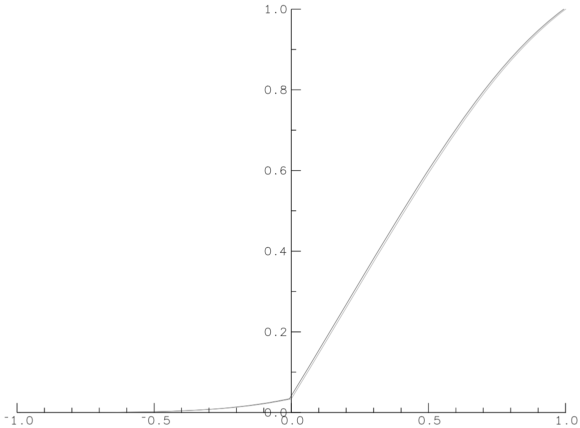


Fig. 2. Distribution functions for the stationary distributions of $\underline{\gamma}$ and $\overline{\gamma}$ given by $k = 150$. For any $n > 10$, the true incremental-length distribution $\gamma(n)$ lies between the two (incidentally proving that $0.4215 < \mathbb{E}\gamma < 0.4292$).

Theorem 5. *The stationary random variables* $\underline{\Delta}$ *and* $\overline{\Delta}$ *for equations* (3) *and* (4) *satisfy* $d(\underline{\Delta}, \overline{\Delta}) = O(\epsilon)$.

The proof is given in the full paper.

Arbitrarily good bounds on the stationary distribution of $\gamma(n)$, for n large, follow as a corollary. From (1), a random variable $\underline{\gamma}_i := \min\{\underline{X}_i, \underline{\Delta}(i-1)\}$ provides a lower bound $\underline{\gamma}_i \preceq \gamma_i$. Similarly, an upper bound is given by $\overline{\gamma}_i := \min\{\overline{X}_i, \overline{\Delta}(i-1)\} \leq \min\{\underline{X}_i + \epsilon, \overline{\Delta}(i-1)\}$. In the coupling, the random variables' values always satisfy $0 \leq \overline{\gamma}_i - \underline{\gamma}_i \leq \epsilon + (\overline{\Delta}(i-1) - \underline{\Delta}(i-1))$. Taking expectations over the stationary distributions we know to exist (these are finite-state Markov chains, with $\overline{\Delta}$, $\underline{\Delta}$, $\overline{\gamma}$, and $\underline{\gamma}$ all discrete random variables) gives $d(\underline{\gamma}, \overline{\gamma}) = \mathbb{E}(\overline{\gamma} - \underline{\gamma}) \leq \epsilon + \mathbb{E}(\overline{\Delta} - \underline{\Delta}) = O(\epsilon)$.

Finally, we show that Δ has a unique well-defined stationary distribution; from (1) it is then immediate that γ does as well.

Theorem 6. *The continuous Markov chain* $\Delta(i)$ *defined by* (2) *has a unique stationary distribution.*

Proof. Per the remarks after Definition 7, any recurrent Harris chain possesses a unique stationary distribution, and Lemma 8 shows that $\Delta(i)$ is a recurrent Harris chain. □

Definition 7. *A discrete time Markov chain* $\Phi(t)$ *with state space* Ω *is defined to be a recurrent Harris chain if there exist two sets* $A, B \subset \Omega$ *satisfying the following properties:*

1. $\Phi(t) \in A$ *infinitely often w.p. 1.*
2. *There exists a non-zero measure* ν *with support contained in* B *such that for every* $x \in A$ *and* $C \subset B$, $\mathbb{P}(\Phi(t+1) \in C \mid \Phi(t) = x) \geq \nu(C)$.

(See [Dur96] Section 5.6 pages 325-326 for a Harris chain, and page 329 for recurrent Harris.) It is known (see Durrett [Dur96]) that the recurrent Harris chain possesses a unique stationary distribution. Our next goal is to show that our chain $\Delta(i)$ is indeed recurrent Harris.

Lemma 8. *$\Delta(i)$ is a recurrent Harris chain, with $A = [-0.1, 1], B = [0, 0.4]$, and ν the uniform probability distribution on B multiplied by* 0.2.

Proof. To show that the chain is a recurrent Harris chain, we observe that when $\Delta(i) \in A$, that is $\Delta(i) \geq -0.1$, if in addition $W_{i+1} \geq 0.5$ and $Z_{i+1} \leq 0.4$, then $\Delta(i+1) = \mathrm{trunc}(\Delta(i) + W_{i+1};\ -Z_{i+1},\ Z_{i+1}) = Z_{i+1}$. Let $V_{i+1} = \mathbf{1}\{Z_{i+1} \leq 0.4\}$. Note that, conditioned on $V_{i+1} = 1$, Z_{i+1} is distributed uniformly on $[0, 0.4]$. Let $p = \mathbb{P}(W_{i+1} \geq 0.5, V_{i+1} = 1) = 0.2$. Then for every $C \subset B$ and $x \in A$ we have

$$\begin{aligned}
&\mathbb{P}(\Delta(i+1) \in C \mid \Delta(i) = x) \\
&\qquad\qquad \geq \mathbb{P}(W_{i+1} \geq 0.5, V_{i+1} = 1) \cdot \mathbb{P}(Z_{i+1} \in C \mid W_{i+1} \geq 0.5, V_{i+1} = 1) \\
&\qquad\qquad = p\mathbb{P}(Z_{i+1} \in C \mid V_{i+1} = 1) = p\mu(C) = \nu(C),
\end{aligned}$$

where μ is the uniform measure on B and we define ν by $\nu(C) = p\mu(C)$. Therefore, $\Delta(i)$ satisfies condition (2) of Definition 7. We now prove condition (1), that w.p. 1 the set A is visited infinitely often. This is a simple corollary of the fact that if $Z_i \leq 0.1$ then $\Delta(i) = \mathrm{trunc}(\Delta(i-1) + W_i;\ -Z_i,\ Z_i) \geq -0.1$, that is $\Delta(i) \in A$. Clearly this happens infinitely often w.p. 1. □

6 An Auction Model

Suppose that in the half-infinite width-2 strip, each edge is provided by an independent agent who incurs a cost for supplying it (or for allowing us to drive over it, transmit data over it, or whatever). In this setup, agents have an incentive to lie: their true cost is not the cost they will sensibly tell us. A popular way to deal with potentially dishonest agents is to assume that each agent will act independently to maximize her own utility, and to design a mechanism where this behavior will result in every agent acting truthfully. The VCG mechanism finds an outcome that maximizes social welfare in a truthful fashion. For buying an (s, t)-path, the VCG mechanism is the following: An "auctioneer" finds a cheapest path, and, for each edge on that path, pays the corresponding agent the difference between the cost of a cheapest path avoiding the edge and the cost of a cheapest path if the edge cost were 0. (The mechanism is truthful because by inflating her cost, an agent does not affect the amount she gets paid, until the point when she inflates the price so much that her edge is no longer in a shortest path and she gets paid nothing.)

Unfortunately, the VCG mechanism may result in the auctioneer paying much more than the cost of the shortest path. The simplest example comes from a source and sink connected by two parallel edges, one with cost 1 and the other with cost $c > 1$. The shortest path is the edge with cost 1, and the payment made to it is $c - 0 = c$; the ratio between this VCG cost and the simple shortest-path cost of 1 is unbounded if c is much larger than 1. In fact, even in the case where the second-best path has cost close to that

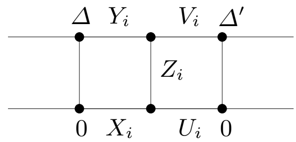


Fig. 3. The VCG cost at step i, working in from the left and the right, and assuming both sides are in stationarity

of the best one, the VCG overpayment can be large; see [AT02] for a detailed study of the worst-case behavior of this overpayment. An example from [AT02] consists of two disjoint (s,t)-paths, with costs C and $C(1+\epsilon)$, and with the cheaper path containing k edges; the total payment is $C(1+k\epsilon)$.

It is natural to wonder how the cost of the VCG mechanism compares with the shortest-path cost in the average-case setting. We will study the cost on the width-2 strip with random edge weights. (Average-case studies on other distributions of networks appear in [MPS03, CR04, KN05].)

Theorem 9. *When the edge weights are i.i.d.* $\mathrm{Be}(p)$ *random variables, with* $0 < p < 1$, *the VCG path cost satisfies*

$$\lim_{n\to\infty} \frac{1}{n}\mathbb{E}(\ell_{\mathrm{VCG}}(n)) = \frac{p\,(2+5\,p+4\,p^2+8\,p^3+11\,p^4-3\,p^6+p^8)}{(1+3\,p^2)^2}. \tag{7}$$

Proof. With reference to Figure 1, we compute the contribution of the ith triple of edges (X_i, Y_i, Z_i) to the expected VCG cost. Let $\omega(n)$ be any function tending to infinity much slower than n itself, i.e., with $1 \ll \omega(n) \ll n$. Note that any edge's contribution to the VCG cost is at most 3: we can circumvent any horizontal edge with a "loop" of 3 edges around it, each edge costing at most 1, and we can bypass any vertical edge at worst by going one more step to the right and traversing the next vertical edge, for a cost of at most 2. Thus the contribution of the first and last $\omega(n)$ edges to the limit is at most $6\omega(n)/n$, which tends to 0.

Now, for any i, a shortest path between $(0,0)$ and $(0,n)$ may be found by taking the shortest paths from $(0,0)$ to both $(0,i)$ and $(1,i)$, and also the shortest paths from $(0,n)$ to both $(0,i+2)$ and $(1,i+2)$, and finding the cheapest total way of joining one of the first paths to one of the second. The first two paths depend only on variables with indices less than i, and without loss of generality (up to an additive constant) we may consider their two costs to be 0 and Δ. Likewise, the second two paths depend only on variables with indices $i+2$ or more, and their costs may be given as 0 and Δ'. For $\omega(n) < i < n-\omega(n)$, Δ and Δ' are independent random variables drawn from a distribution asymptotically equal to the stationary distribution. Thus, with reference to Figure 3, we consider the payments we must make for the edges X_i, Y_i, and Z_i, when Δ and Δ' are i.i.d. random variables drawn from the stationary distribution, and X_i, Y_i, Z_i, U_i, and V_i are i.i.d. $\mathrm{Be}(p)$ random variables. Since, over all i, such groups (X_i, Y_i, Z_i) cover each edge exactly once (except for the single edge Z_0), the total of the expected payments for one such group is precisely the limiting expectation called for in (7).

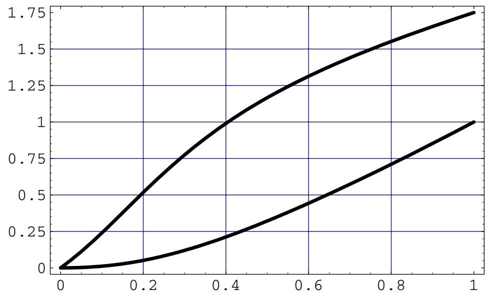


Fig. 4. VCG and usual shortest-path rates

This is a straightforward calculation. Dropping the subscripts for convenience, let $A = X + U$ be the cost of the path using X, U; $B = X + Z + V + \Delta'$ that of the path using X, Z, V; $C = \Delta + Y + V + \Delta'$ that using Y, V; and $D = \Delta + Y + Z + U$ that using Y, Z, U. If we break ties in favor of lower letters (A in favor of B in favor of C in favor of D), the payment to X is

$$C(X) = \mathbf{1}\{\min(A, B) \le \min(C, D)\} \cdot [\min(C, D) - (\min(A, B) - X)],$$

that is, it is 0 unless the edge X is used, and then it is the cost of the cheapest path avoiding X less the cost of the cheapest path if X were 0, which in this case is the cheapest path using X, minus X. Similarly, the payment to Y is

$$C(Y) = \mathbf{1}\{\min(C, D) < \min(A, B)\} \cdot [\min(A, B) - (\min(C, D) - Y)].$$

The payment to Z follows similarly, with slightly more complicated tie-breaking:

$$\begin{aligned} C(Z) = \mathbf{1}\{&(B < A) \vee (B \le \min(C, D)) \\ &\vee (D < \min(A, B, C))\} \cdot [\min(A, C) - (\min(B, D) - Z)]. \end{aligned}$$

Where the stationary probabilities for Δ and Δ' are written as $\mathbb{P}_\Delta(\cdot)$, and the Bernoulli probabilities as $\mathrm{Be}(1) = p$ and $\mathrm{Be}(0) = 1 - p$, the expected total payments for X, Y, and Z is

$$\sum_{\substack{X,Y,Z,\\ U,V,\Delta,\Delta'}} \mathrm{Be}(X)\,\mathrm{Be}(Y)\,\mathrm{Be}(Z)\,\mathrm{Be}(U)\,\mathrm{Be}(V)\mathbb{P}_\Delta(\Delta)\mathbb{P}_\Delta(\Delta') \cdot [C(X) + C(Y) + C(Z)],$$

the sum taken over the $2^5 3^2$ possible values of the variables. This is a small finite sum of an explicit expression, and is calculated (by Mathematica) to be the value shown in expression (7). $\square$

A plot of the VCG cost rate $\lim_{n\to\infty} \mathbb{E}\ell_{\mathrm{VCG}}(n)/n$, along with the corresponding shortest-path cost rate, is given in Figure 4.

References

[Ald01] David J. Aldous, *The $\zeta(2)$ limit in the random assignment problem*, Random Structures Algorithms **18** (2001), no. 4, 381–418.

[AP02] Sven Erick Alm and Robert Parviainen, *Lower and upper bounds for the time constant of first-passage percolation*, Comb. Probab. Comput. **11** (2002), no. 5, 433–445.

[AS04] David J. Aldous and J. Michael Steele, *The objective method: probabilistic combinatorial optimization and local weak convergence*, Probability on discrete structures, Encyclopaedia Math. Sci., vol. 110, Springer, Berlin, 2004, pp. 1–72.

[AT02] Aaron Archer and Éva Tardos, *Frugal path mechanisms*, Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms (San Francisco, California), January 06-08 2002, pp. 991–999.

[BH57] S. R. Broadbent and J. M. Hammersley, *Percolation processes i. crystals and mazes*, Proceedings of the Cambridge Philosophical Society **53** (1957), 629–641.

[Cla71] E. Clarke, *Multipart pricing of public goods*, Public Choice **8** (1971), 17–33.

[CR04] Artur Czumaj and Amir Ronen, *On the expected payment of mechanisms for task allocation (extended abstract)*, Proceedings of the fifth ACM Conference on Electronic Commerce, 2004, pp. 252–253.

[Dur96] Richard Durrett, *Probability: theory and examples*, second ed., Duxbury Press, Belmont, CA, 1996.

[Elk05] Edith Elkind, *True costs of cheap labor are hard to measure: edge deletion and VCG payments in graphs*, Proceedings of the sixth ACM Conference on Electronic Commerce, 2005, pp. 108–117.

[ESS04] Edith Elkind, Amit Sahai, and Ken Steiglitz, *Frugality in path auctions*, Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, 2004, pp. 694–702.

[FPSS02] Joan Feigenbaum, Christos Papadimitriou, Rahul Sami, and Scott Shenker, *A BGP-based mechanism for lowest-cost routing*, PODC '02: Proceedings of the twenty-first annual symposium on Principles of distributed computing (New York, NY, USA), ACM Press, 2002, pp. 173–182.

[GNSar] D. Gamarnik, T. Nowicki, and G. Swirscsz, *Maximum weight independent sets and matchings in sparse random graphs. Exact results using the local weak convergence method*, Random Structures and Algorithms (To appear).

[Gro73] T. Groves, *Incentives in teams*, Econometrica **41** (1973), no. 4, 617–631.

[HHM01] Remco Van Der Hofstad, Gerard Hooghiemstra, and Piet Van Mieghem, *First-passage percolation on the random graph*, Probab. Eng. Inf. Sci. **15** (2001), no. 2, 225–237.

[HW65] J. M. Hammersley and J. D. Welsh, *First-passage percolation, subadditive processes, stochastic networks, and generalized renewal theory*, Bernoulli-Bayes-Laplace Anniversary Volume, Springer-Verlag, 1965, pp. 61–110.

[Jan81] S. Janson, *An upper bound for the velocity of first-passage percolation*, Journal of Applied Probability **18** (1981), 256–262.

[Kes87] H. Kesten, *Percolation theory and first-passage percolation*, Ann. Probab. (1987), no. 15, 1231–1271.

[KN05] David Karger and Evdokia Nikolova, *Brief announcement: on the expected overpayment of VCG mechanisms in large networks*, PODC '05: Proceedings of the 24th annual ACM symposium on Principles of distributed computing (New York, NY, USA), ACM Press, 2005, pp. 126–126.

[MPS03] Milena Mihail, Christos Papadimitriou, and Amin Saberi, *On certain connectivity properties of the internet topology*, FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 2003, p. 28.

[NR99] Noam Nisan and Amir Ronen, *Algorithmic mechanism design (extended abstract)*, Proceedings of the thirty-first annual ACM symposium on Theory of computing (Atlanta, Georgia, United States), 1999, pp. 129–140.

[SW78] R.T. Smythe and John C. Wierman, *First-passage percolation on the square lattice*, Lecture Notes in Mathematics, vol. 671, Springer-Verlag, 1978.

[Vic61] W. Vickery, *Counterspeculation, auctions, and competitive sealed tenders*, Journal of Finance **16** (1961), 8–37.

Optimal Cost-Sharing Mechanisms for Steiner Forest Problems

Shuchi Chawla[1], Tim Roughgarden[2], and Mukund Sundararajan[2]

[1] University of Wisconsin-Madison, Department of Computer Science, 210 W. Dayton Street, Madison, WI 53706, USA
[2] Stanford University, Department of Computer Science, 353 Serra Mall, Stanford, CA 94305, USA

Abstract. Könemann, Leonardi, and Schäfer [14] gave a 2-budget-balanced and groupstrategyproof mechanism for Steiner forest cost-sharing problems. We prove that this mechanism also achieves an $O(\log^2 k)$-approximation of the social cost, where k is the number of players. As a consequence, the KLS mechanism has the smallest-possible worst-case efficiency loss, up to constant factors, among all $O(1)$-budget-balanced Moulin mechanisms for such cost functions. We also extend our results to a more general network design problem.

1 Introduction

We study the design and analysis of cost-sharing mechanisms for fundamental network design problems. A *cost-sharing mechanism* is a protocol that collects bids for a service or good from potential users (players), chooses a subset of players to receive the service and a feasible way of servicing them, and determines prices to charge the chosen players. The mechanism incurs a subset-dependent cost $C(S)$ defined by a known cost function C. In this paper, we are interested in problems where players seek connectivity between a group of vertices, and where the cost $C(S)$ corresponds to the cost of providing such connectivity to the players in the set S.

A cost-sharing mechanism can be viewed as an auction in which any number of players can "win", but in which the cost incurred by the auctioneer varies with the set of winners. The canonical problem of auctioning off a single good can be viewed as the special case in which the cost $C(S)$ is 0 if $|S| \leq 1$ and is $+\infty$ otherwise.

With more general cost functions, designing a mechanism requires choosing between several desirable but incompatible properties. As is standard, we insist on *incentive-compatibility*, meaning that players are motivated to bid their true private value v_i for receiving the service. We also require *budget-balance*, meaning that the mechanism recovers the incurred cost with the prices charged to the chosen players. Finally, we are interested in the social objective function of *efficiency*. Efficiency states that a set S should be chosen that trades off the cost $C(S)$ incurred and the valuations of the players in S in an optimal way.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 112–123, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Unfortunately, these three properties cannot be simultaneously achieved, even in very simple settings [8,21].

This impossibility result motivated two distinct approaches to designing cost-sharing mechanisms. The first approach, taken by *VCG mechanisms* (see e.g. [17,19]), ignores budget-balance. These mechanisms are optimally efficient and incentive compatible. They are typically not approximately budget-balanced for any reasonable approximation factor (see e.g. [5]).

The second approach, adopted in this paper, is to insist on incentive-compatibility and budget-balance, while regarding efficiency as a secondary objective. Moulin [18] introduced a class of mechanisms of this type. Researchers have developed approximately budget-balanced Moulin mechanisms for a number of different combinatorial optimization problems, including fixed-tree multicast [2,5,6]; submodular cost-sharing [18,19]; Steiner tree [11,12,13]; Steiner forest [14,15]; facility location [16,20]; and rent-or-buy network design [9,20]. The approach of Moulin [18] is the only known general technique for designing budget-balanced mechanisms with non-trivial costs.

Since Moulin mechanisms prioritize budget-balance over efficiency, nearly all previous papers that design Moulin mechanisms do not address the efficiency of the proposed mechanisms. Nevertheless, very recent work [23] shows that it is possible to discuss and compare the efficiency of Moulin mechanisms. Specifically, Roughgarden and Sundararajan [23] measured efficiency via the *social cost*, where the social cost of a set S is defined as the sum of the incurred service cost and the excluded valuations: $C(S) + \sum_{i \notin S} v_i$. This objective function is similar to the "prize-collecting" objectives that are commonly studied in approximation algorithms (see e.g. [7]). We call a mechanism *α-approximate* if it always outputs a solution with social cost at most an α factor times that of an optimal solution.

Roughgarden and Sundararajan [23] developed a framework to quantify the extent to which a Moulin mechanism minimizes the social cost. They applied this framework in [22,23] to well-known mechanisms for submodular [19], facility location [20], Steiner tree [11], and single-sink rent-or-buy [9] cost functions. In particular, all of these mechanisms are both $O(\text{polylog}(k))$-approximate, where k is the number of players, and are optimal (up to constant factors) among all Moulin mechanisms for the corresponding sets of cost functions.

A consequence of the main result in [23] is that only β-budget-balanced Moulin mechanisms can be β-approximate. The problem of designing Moulin mechanisms with good (say, polylogarithmic) budget-balance is itself highly non-trivial, and is provably impossible for several natural classes of cost functions [10]. Prior work [22,23] resolved the approximate efficiency of all known Moulin mechanisms with good budget-balance save one: the elegant 2-budget-balanced Moulin mechanism for Steiner forest cost-sharing problems due to Könemann, Leonardi, and Schäfer [14]. We call this the *KLS mechanism*.

There are two reasons that analyzing the approximate efficiency of the KLS mechanism is technically challenging. First, Steiner forest cost functions, defined formally in Section 2, seem more complex than those treated in previous works [22,23]. Indeed, the problem of designing an $O(1)$-budget balanced

Moulin mechanism for such functions was a well-known open question for several years prior to the invention of the KLS mechanism. Second, the KLS mechanism itself possesses properties that make the analytic framework in [23] difficult to apply. In fact, prior to the present work, no non-trivial bound on the approximate efficiency of the KLS mechanism was known even when the input is restricted to Steiner tree cost functions. On the other hand, the Steiner tree cost-sharing mechanism due to Jain and Vazirani [11] is known to be $O(\log^2 k)$-approximate [23]. The analysis of the Jain-Vazirani mechanism does not obviously carry over to the KLS mechanism (specialized to Steiner tree cost functions) since the latter mechanism, intuitively, charges players higher prices and therefore more aggressively discards them from the set S of winners. See Section 3.1 for a more technical discussion of this point.

In this paper, we overcome these difficulties and prove a tight upper bound on the approximate efficiency of the KLS mechanism. Specifically, in Section 3 we prove that the mechanism is $O(\log^2 k)$-approximate, where k is the number of players. Previous work [22] shows that, even for the special case of Steiner tree cost functions, every $O(1)$-budget-balanced Moulin mechanism is $\Omega(\log^2 k)$-approximate. Thus the KLS mechanism has the smallest-possible worst-case efficiency loss, up to constant factors, among all such mechanisms for Steiner forest cost functions. We also extend our results to a more general network design problem (Section 4).

2 Preliminaries

Cost-Sharing Mechanisms. We consider cost functions C that assign a cost $C(S)$ to every subset S of a universe U of players and are defined implicitly via instances of network design problems. We also assume that every player $i \in U$ has a private, nonnegative *valuation* v_i for service. A *generalized Steiner tree (GST)* cost function is defined by a graph $G = (V, E)$, where each edge $e \in E$ possesses a nonnegative cost c_e, and by a set U of players, where each player $i \in U$ is identified with a subset $A_i \subseteq V$ of vertices called *terminals*. For a subset $S \subseteq U$ of players, the cost $C(S)$ is defined as the minimum cost of a subgraph of G that, for each $i \in S$, connects all of the vertices in A_i. A *Steiner forest (SF)* cost function is a special case of a GST function in which every group A_i contains only two terminals, a *source* s_i and a *sink* t_i.

A *cost-sharing mechanism* collects a nonnegative bid b_i from each player $i \in U$, selects a set $S \subseteq U$ of players, and charges every player i a price p_i. The mechanisms we consider also produce a feasible solution to the network design problem induced by the served set S, which has cost $C'(S)$ that in general is larger than the optimal cost $C(S)$. (Of course, evaluating $C(S)$ exactly is NP-hard.) We only allow mechanisms that are "individually rational" in the sense that $p_i = 0$ for players $i \notin S$ and $p_i \leq b_i$ for players $i \in S$. We also require that all prices are nonnegative ("no positive transfers"). As is standard, we assume that players have *quasilinear* utilities, meaning that each player i aims to maximize $u_i(S, p_i) = v_i x_i - p_i$, where $x_i = 1$ if $i \in S$ and $x_i = 0$ if $i \notin S$.

Our incentive-compatibility constraint is the well-known strategyproofness condition, which intuitively requires that a player cannot gain from misreporting its bid. Formally, a mechanism is *strategyproof (SP)* if for every player i, every bid vector b with $b_i = v_i$, and every bid vector b' with $b_j = b'_j$ for all $j \neq i$, $u_i(S, p_i) \geq u_i(S', p'_i)$, where (S, p) and (S', p') denote the outputs of the mechanism for the bid vectors b and b', respectively.

For a parameter $\beta \geq 1$, a mechanism is *β-budget balanced* if $C(S)/\beta \leq \sum_{i \in S} p_i \leq C(S)$ for every outcome (S, p) of the mechanism. For a mechanism that outputs a feasible solution with cost $C'(S) \geq C(S)$, we require the stronger condition that $C'(S)/\beta \leq \sum_{i \in S} p_i \leq C(S)$. In particular, this requirement implies that the feasible solution produced by the mechanism has cost at most a β factor times that of optimal.

As discussed in the Introduction, we measure efficiency using the objective of *social cost minimization*. A cost-sharing mechanism is *α-approximate* if, assuming truthful bids, it always produces a solution with social cost at most an α factor times that of an optimal solution. Here, the social cost incurred by the mechanism is defined as the service cost $C'(S)$ of the feasible solution it produces for the network design instance corresponding to S, plus the sum $\sum_{i \notin S} v_i$ of the excluded valuations. Such a mechanism has two sources of inefficiency: first, it might choose a suboptimal set S of players to serve; second, it might produce a suboptimal solution to the network design instance induced by S.

Moulin Mechanisms and Cross-Monotonic Cost-Sharing Methods. Next we review *Moulin mechanisms*, a class of cost-sharing mechanisms that, for many cost functions, are SP, approximately budget-balanced, and approximately efficient. Such mechanisms are based on *cost sharing methods*, defined next.

A cost-sharing method χ is a function that assigns a non-negative *cost share* $\chi(i, S)$ for every subset $S \subseteq U$ of players and every player $i \in S$. A cost-sharing method is *β-budget balanced* for a cost function C and a parameter $\beta \geq 1$ if it always recovers β fraction of the cost: $C(S)/\beta \leq \sum_{i \in S} \chi(i, S) \leq C(S)$. We consider cost-sharing methods that, given a set S, produce both the cost shares $\chi(i, S)$ for all $i \in S$ and also a feasible solution for the network design problem induced by S. As above, we use the stronger condition $C'(S)/\beta \leq \sum_{i \in S} \chi(i, S) \leq C(S)$ for such methods, where $C'(S)$ is the cost of the produced feasible solution. A cost-sharing method is *cross-monotonic* if adding players to a set S only decreases the cost shares of players: for all $S \subseteq X \subseteq U$ and $i \in S$, $\chi(i, S) \geq \chi(i, X)$.

A cost-sharing method χ for C defines the following *Moulin mechanism* M_χ for C. First, collect a bid b_i for each player i. Initialize the set S to all of U and invoke the cost-sharing method χ to define a feasible solution to the network design problem induced by S and a price $p_i = \chi(i, S)$ for each player i. If $p_i \leq b_i$ for all $i \in S$, then halt, output the set S, the corresponding network design solution, and charge prices p. If $p_i > b_i$ for some player $i \in S$, then remove an arbitrary such player from the set S and iterate. A Moulin mechanism based on a cross-monotonic cost-sharing method thus simulates an iterative ascending auction, with the method χ suggesting prices for the remaining players at each iteration. Note that if χ produces a feasible solution in polynomial time, then so does M_χ. Also, M_χ clearly inherits the budget-balance factor of χ. Finally, Moulin [18] proved the following.

Theorem 1 ([18]). *If χ is a cross-monotonic cost-sharing method, then the corresponding Moulin mechanism M_χ is strategyproof.*

Theorem 1 reduces the problem of designing an SP[3], β-budget-balanced mechanism to that of designing a cross-monotonic, β-budget-balanced cost-sharing method.

Summability and Approximate Efficiency. Roughgarden and Sundararajan [23] showed that the approximate efficiency of a Moulin mechanism is completely controlled by its budget-balance and one additional parameter of its underlying cost-sharing method. We define this parameter next.

Definition 1 ([23]). *Let C and χ be a cost function and a cost-sharing method, respectively, defined on a common universe U of players. The method χ is α-*summable for C *if*

$$\sum_{\ell=1}^{|S|} \chi(i_\ell, S_\ell) \leq \alpha \cdot C(S)$$

for every ordering σ of U and every set $S \subseteq U$, where S_ℓ and i_ℓ denote the set of the first ℓ players of S and the ℓth player of S (with respect to σ), respectively.

We next summarize the main result in [23].

Theorem 2 ([23]). *Let U be a universe of players and C a nondecreasing cost function on U with $C(\emptyset) = 0$. Let M be a Moulin mechanism for C with underlying cost-sharing method χ. Let $\alpha \geq 0$ and $\beta \geq 1$ be the smallest numbers such that χ is α-summable and β-budget-balanced. Then the mechanism M is $(\alpha + \beta)$-approximate and no better than max$\{\alpha, \beta\}$-approximate.*

In particular, an $O(1)$-budget-balanced Moulin mechanism is $\Theta(\alpha)$-approximate if and only if the underlying cost-sharing method is $\Theta(\alpha)$-summable.

The KLS Cost-Sharing Method. Könemann, Leonardi and Schäfer [14] devised cross-monotonic, 2-budget-balanced cost-sharing methods for all Steiner forest cost functions. The cost-sharing method is based on a variant of the primal-dual method. By Theorem 1, this yields 2-budget-balanced and GSP mechanisms for all such functions. Due to space constraints, we refer the reader to [14] for a description of the KLS cost-sharing method; its details are important primarily for Sections 3.3 and 4.

[3] Moulin mechanisms satisfy a stronger notion of incentive compatibility called group-strategyproofness (GSP), which is a form of collusion resistance. Almost all known GSP cost-sharing mechanisms are Moulin mechanisms (see [10,18,19]).

3 The Efficiency of the KLS Mechanism

We now analyze the efficiency of the KLS mechanism. Our main result is the following.

Theorem 3. *For every Steiner forest cost function with k players, the KLS cost-sharing method is $O(\log^2 k)$-summable.*

Since the KLS cost-sharing method is 2-budget-balanced, Theorems 2 and 3 immediately give a guarantee on the approximate efficiency of the KLS mechanism.

Corollary 1. *For every Steiner forest cost function with k players, the KLS mechanism is $O(\log^2 k)$-approximate.*

Since every $O(1)$-budget-balanced Moulin mechanism for Steiner tree cost functions is $\Omega(\log^2 k)$-approximate [22], the KLS mechanism is an optimal mechanism of this type (up to constant factors).

3.1 Overview of the Proof of Theorem 3

This section provides an overview of our analysis. By the definition of summability (Definition 1), proving Theorem 3 requires analyzing the following procedure. Given an arbitrary Steiner forest instance and an arbitrary ordering of the players (source-sink pairs), we consider adding the players to the instance one-by-one, according to the given ordering. Each time we add a new player, we recompute the KLS cost shares using the KLS primal-dual algorithm and consider the cost share of the most recently added player. The key question is: by how much can the sum of these successive cost shares exceed the cost of servicing all of the players?

Our analysis proceeds in two steps. The first step is motivated by the difficulty in directly bounding the above successive cost shares in a general network. The idea of this step is to replace the given network by a forest with cost at most an $O(\log k)$ times that of an optimal Steiner forest. In addition, to facilitate our charging argument in the second step, we require that each tree of this forest be an ultrametric—i.e. all root-leaf paths have equal length. While this goal is reminiscent of probabilistic tree embeddings (see e.g. [3,4]), we cannot apply such an embedding as a black box. The reason is that our charging argument requires structure beyond the low distortion guarantee—it also needs the distances in the ultrametric to be tightly coupled with the dual growth process used to define the KLS cost-shares.

In the second step, we demonstrate how to charge the k successive KLS cost shares to the ultrametrics constructed in the first step. Loosely speaking, we show how subtrees in each ultrametric correspond to active components during the execution of the primal-dual algorithm that defines the KLS cost shares. Our charging scheme charges each point of each ultrametric $O(\log k)$ times, proving an $O(\log^2 k)$ bound on the summability of the KLS cost-sharing method.

While portions of this argument are similar to that used in [23] to upper bound the summability of the Jain-Vazirani Steiner tree cost-sharing method [11], the refined ultrametric structure and the charging argument in this paper are new. One reason we require the ultrametric structure is that the primal-dual algorithm underlying the KLS mechanism determines cost shares using fixed "death times", rather than via the component structure in the dual growth process. While crucial for cross-monotonicity, this property can cause a terminal to accumulate a cost share beyond the point at which it is connected to its mate, and it is not obvious how to bound this additional accumulation. In fact, we can exhibit an example with k players for which the summability of the KLS method is an $\Omega(\log k)$ factor times larger than that of the Jain-Vazirani method. Nonetheless, we prove in this section that the KLS method is always $O(\log^2 k)$-summable, matching the (tight) worst-case bound for the Jain-Vazirani method.

3.2 Building the Forest

In the first step of our proof of Theorem 3, we define a procedure with the following properties. The procedure takes as input a Steiner forest instance $G = (V, E)$ with edge costs and an (adversarial) ordering σ of the source-sink pairs $(s_1, t_1), \ldots, (s_k, t_k)$. It constructs a forest F, defined on the terminals, that has cost $O(\log k)$ times that of a minimum-cost Steiner forest, as well as other desirable structure. While the following description will be algorithmic, we emphasize that this construction is purely for the purposes of analyzing the summability of the KLS cost-sharing method.

Consider an optimal solution to the given Steiner forest instance. Our forest F will have one tree for each connected component of this optimal solution. We will construct these trees independently of each other, so we can restrict our description to a single component T^* of the optimal Steiner forest. Let A^* denote the terminals spanned by T^*. The vertex set of the tree T that we construct will contain all the terminals in A^* as well as some auxiliary vertices.

We now describe the construction of T. The ordering $\sigma = (s_1, t_1), \ldots, (s_k, t_k)$ on source-sink pairs induces an ordering $s_1, t_1, s_2, t_2, \ldots, s_k, t_k$ on the terminals and also an ordering of A^*. We construct T by adding terminals in A^* to it in this order. When a terminal is considered, we attach it to the existing tree and endow it with a *radius*. The *ball* of a terminal x with radius r is defined as the terminals of A^* at distance at most r from x in the given graph G. We begin with the first terminal (say x_1) of A^*, which is given an infinite radius. For technical reasons, we introduce an auxiliary root x_0 and create an edge e_0 between x_0 and x_1 of length D_{max}, where D_{max} is half the largest distance in G between two terminals of A^*. We call this edge e_0 the *backbone edge*.

Now consider some subsequent terminal x. Among all of the previously added terminals whose ball contains x, we define the terminal y with the minimum radius to be the *parent* of x and write $p(x) = y$. If y has finite radius—i.e., is not the first terminal of A^* with respect to σ—then we define x's radius r_x to be half of its parent's radius. Otherwise, we define the radius r_x to be half of the shortest-path distance between x and y in G. To attach x to the tree T, consider the path from y to x_0 in T. We connect x to the point along this path at a distance r_x from y, possibly creating a new internal node. The backbone edge and the definition of D_{max} ensure that this is always possible. Call this point $v(x)$. The length of the edge between $v(x)$ and x is defined to be r_x. We assign this new node $v(x)$ a label with value y; this label plays a role in our subsequent proofs.

We next prove several facts about this construction. We begin with an easy lemma.

Lemma 1. *The backbone edge e_0 has length at most $c(T^*)$, where T^* is the component of the optimal solution that spans A^*.*

Using this lemma and arguments similar to those in [23] for a related tree construction, we can bound the sum of edge costs in T^*.

Lemma 2. *The sum of the costs of the edges in T is $O(\log k) \cdot c(T^*)$.*

We next study distances between terminals in the tree T. We begin by noting that our construction does indeed produce an ultrametric.

Lemma 3. *The tree T is an ultrametric, with all root-leaf paths having length D_{max}. Moreover, the leaves of T are in bijective correspondence with the terminals A^*.*

Lemma 3 follows from an easy induction. In particular, when a new terminal x is added to the tree T, the distance from $v(x)$ to x equals the distance from $v(x)$ to $p(x)$.

For every two terminals x, y in A^*, let $d_T(x, y)$ and $d_G(x, y)$ denote the distances between x and y in the tree T and in the graph G, respectively. The next lemma follows immediately from the construction.

Lemma 4. *For every terminal $x \in A^*$ with parent $p(x)$, $d_T(x, p(x)) = 2r_x = r_{p(x)} \geq d_G(x, p(x))$.*

We next extend Lemma 4 to every pair $x, y \in A^*$ of terminals x, y. The idea is to consider a walk W_{xy} between x and y in T and relate the length of this walk to both $d_T(x, y)$ and $d_G(x, y)$.

Precisely, fix $x, y \in A^*$ and consider the (unique) path P_{xy} between x and y in the tree T. The length of this path is $d_T(x, y)$. To construct the walk W_{xy}, consider the sequence S_{xy} of vertices that the path P_{xy} visits; apart from x and y, all of these are internal nodes of T. Obtain a sequence S'_{xy} of terminals from S_{xy} by replacing the internal nodes of S_{xy} by their label values (terminals) and then removing duplicates. Obtain the walk W_{xy} in T by visiting the terminal nodes in S'_{xy} in order, along the unique paths in T that connect consecutive nodes. The walk W_{xy} contains P_{xy} as a subgraph, and can be decomposed into P_{xy} and a set of circuits, each of which starts and ends at an internal node of P_{xy}, visiting the terminal node corresponding to the label of the internal node along the way.

Let ℓ_{xy} denote the length of this walk. We prove the following three lemmas, with the third an immediate consequence of the first two.

Lemma 5. *For every pair $x, y \in A^*$ of terminals in T, $\ell_{x,y} \geq d_G(x, y)$.*

Lemma 6. *For every pair $x, y \in A^*$ of terminals in T, $d_T(x, y) \geq \ell_{x,y}/5$.*

Lemma 7. *For every pair $x, y \in A^*$ of terminals in T, $d_T(x, y) \geq d_G(x, y)/5$.*

Lemma 5 follows from Lemma 4 and the fact that consecutive nodes in S'_{xy} share a parent-child relationship (details omitted). The idea of the proof of Lemma 6 is as follows. Consider one circuit of W_{xy} rooted at the internal point $v(z)$ and visiting the terminal $p(z)$. The length of the circuit is at most $2r_z$, while the length of the segment of P_{xy} immediately preceding $v(z)$ is at least $r_z/2$ (r_z minus the radius of any of $z's$ children). Therefore, we can charge the length of every circuit to 4 times the section of P_{xy} that immediately precedes it in the walk W_{xy}.

For technical reasons, we multiply all of the edge costs of T by 10, yielding the tree T'. The following is just a restatement of Lemmas 2, 3, and 7.

Lemma 8. *T' satisfies the following properties:*

(a) The cost of T' is $O(\log k) \cdot c(T^*)$.
(b) T' is an ultrametric, with the terminals of A^* appearing only as leaves.
(c) For every pair $x, y \in A^*$ in T', $d_G(x, y) \leq \frac{1}{2} d_{T'}(x, y)$.

3.3 The Charging Argument

We are now ready to bound the summability of the KLS cost-share. Our charging argument will proceed independently for each ultrametric constructed in Sections 3.2; for most of this section, we will fix one such ultrametric T, spanning a set A^* of terminals.

Let x_ℓ and A_ℓ denote the ℓth terminal and the first ℓ terminals of A^*, respectively, with respect to the ordering induced by σ.

We aim to charge the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$ of a terminal $x_\ell \in A^*$ to points of the tree T. (A technical detail: since matched pairs of terminals appear consecutively in the ordering induced by σ, the set A_ℓ contains only matched pairs of terminals, plus possibly an orphaned source s_i. In either case, $\chi^{KLS}(x_\ell, A_\ell)$ denotes the KLS cost share assigned to the terminal x_ℓ in the Steiner forest instance induced by all of the players with at least one terminal in the set A_ℓ.)

The charging proceeds as follows. Let P_ℓ be the unique path in T from x_ℓ to x_0, and consider the primal-dual algorithm that assigns the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$. At each moment in time τ up to the death time of x_ℓ, the terminal's cost share increases at a positive rate, equal to the inverse of the number of active terminals in x_ℓ's component at time τ. For each such time τ, we charge this (marginal) increment in x_ℓ's cost-share to the point $g_\ell(\tau)$ which is at distance τ from x_ℓ along the path P_ℓ.

Since every leaf-root path of T has length at least D_{max} (Lemma 8(b))—half of the largest distance between two terminals of A^*—and since D_{max} is at least the death time of every terminal of A^*, this procedure fully charges the sum $\sum_\ell \chi^{KLS}(x_\ell, A_\ell)$ of the KLS cost shares to T.

We now claim that for every point g of the tree T, the sum of the (marginal) charges to g by the terminals of A^* is only $O(\log k)$. Fix a point g of T. Only terminals in the subtree of T rooted at g charge part of their cost share to g. By the ultrametric property (Lemma 8(b)), all of these terminals are equidistant from the point g in T; let this common distance be τ_g. Such a terminal charges part of its cost share to g if and only if its death time is at least τ_g; let B denote these terminals.

Using Lemma 8(c) we now show that, for a terminal $x_\ell \in B$, at time τ_g in the run of the primal-dual algorithm that defines the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$, the component containing x_ℓ also contains all of the terminals of $B \cap A_\ell$.

Lemma 9. *If $x, x' \in B$, then $d_G(x, x') \leq \tau_g$.*

Lemma 10. *Suppose $x_\ell \in B$ and $x \in A_\ell \cap B$. Then at time τ_g in the run of the primal-dual algorithm that defines the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$, the terminal x is active and lies in the same component as x_ℓ.*

Since the KLS cost-sharing method splits the increase in value of an active dual variable equally among the active terminals contained in the corresponding component, Lemma 10 implies that the marginal charge to the point g by the terminal $x_\ell \in B$ is at most $1/|B \cap A_\ell|$. Summing over the contributions of the terminals in B, we obtain the following.

Lemma 11. *For every point g of T, the total marginal charge to g is at most $\mathcal{H}_{|B|}$, where $\mathcal{H}_j = \sum_{i \leq j} 1/i$ denotes the jth Harmonic number.*

Theorem 3 now follows easily from Lemmas 8(a) and 11.

4 A Generalized Steiner Tree Mechanism

We now briefly consider an extension of the Steiner forest problem, deferring a detailed discussion to the full version. We consider a problem in which each player i controls a group A_i of terminals, and is interested in connecting all of these terminals together. This problem is also called the *generalized Steiner tree (GST)* problem [1]. Let k and n denote the number of terminal groups (players) and terminals, respectively. The Steiner forest problem is the special case where each group contains only two terminals.

Consider the following naive reduction to the Steiner forest problem. For each group of terminals corresponding to a player, nominate one of these terminals as a leader. Create terminal pairs by pairing each terminal in the group with the leader. Invoke the KLS cost-sharing method on these terminal pairs, and define the cost share of a player to be the sum of the cost shares assigned to its corresponding terminal pairs. Cross-monotonicity and 2-budget-balance are straightforward to establish, and Theorem 3 then implies that there is an $O(\log^2 n)$-approximate Moulin mechanism for all GST cost functions.

We can improve the above approximation factor by changing the way that the KLS primal-dual algorithm splits the increase in active dual variables between the active terminals. Specifically, if we modify the algorithm to split each such increase equally between the *players* that have at least one active terminal in the corresponding dual variable, rather than equally among the terminals themselves, then we obtain the following theorem.

Theorem 4. *Every GST cost function with k players and n terminals admits a 2-budget-balanced, $O(\log n \log k)$-approximate Moulin mechanism.*

Using techniques from [22], we can show that the bound in Theorem 4 is the best possible for a $O(1)$-budget-balanced Moulin mechanism for GST cost functions.

References

1. A. Agrawal, P. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.
2. A. Archer, J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47(1):36–71, 2004.
3. Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.
4. J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing (STOC)*, 2003.
5. J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 304:215–236, 2003.
6. J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.
7. M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.
8. J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.
9. A. Gupta, A. Srinivasan, and É. Tardos. Cost-sharing mechanisms for network design. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, volume 3122 of *Lecture Notes in Computer Science*, pages 139–150, 2004.
10. N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost-sharing schemes. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 602–611, 2005.
11. K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 364–372, 2001.
12. K. Jain and V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 313–321, 2002.

13. K. Kent and D. Skorin-Kapov. Population monotonic cost allocation on mst's. In *Operational Research Proceedings KOI*, pages 43–48, 1996.
14. J. Könemann, S. Leonardi, and G. Schäfer. A group-strategyproof mechanism for Steiner forests. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 612–619, 2005.
15. J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam. From primal-dual to cost shares and back: A stronger LP relaxation for the steiner forest problem. In *Proceedings of the 32nd Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1051–1063, 2005.
16. S. Leonardi and G. Schäfer. Cross-monotonic cost-sharing methods for connected facility location. In *Proceedings of the Fifth ACM Conference on Electronic Commerce (EC)*, pages 242–243, 2004.
17. A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
18. H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16:279–320, 1999.
19. H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001.
20. M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 584–593, 2003.
21. K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.
22. T. Roughgarden and M. Sundararajan. Approximately efficient cost-sharing mechanisms. Submitted, 2006.
23. T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *Proceedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 79–88, 2006.

Mechanisms to Induce Random Choice*

Antoniy Ganchev[1], Lata Narayanan[1], and Sunil Shende[2]

[1] Department of Computer Science and Software Engineering,
Concordia University,
Montreal, Canada H3G 1M8
a_ganche@cs.concordia.ca, lata@cs.concordia.ca
[2] Department of Computer Science,
Rutgers University,
Camden, NJ 08102, USA
shende@crab.rutgers.edu

Abstract. Media access protocols in wireless networks require each contending node to wait for a *backoff time* chosen randomly from a fixed range, before attempting to transmit on a shared channel. However, nodes acting in their own selfish interest may not follow the protocol. In this paper, we use a *mechanism design approach* to study how nodes might be induced to adhere to the protocol. In particular, a static version of the problem is modeled as a strategic game (the protocol) played by non-cooperating, rational players (the nodes). We present a game which exhibits a *unique* mixed-strategy Nash equilibrium that corresponds to nodes choosing backoff times randomly from a given range of values, according to any *apriori* given distribution. We extend this result to the situation when each player can choose a backoff value from a *different range*, provided there are at least two players choosing from the *largest* range. In contrast, we show that if there are exactly two players with different backoff ranges, then it becomes impossible to design a strategic game with a unique such Nash equilibrium. Finally, we show an impossibility result under certain natural limitations on the network authority.

1 Introduction

A number of recent papers have tried to address the problem of selfishness of autonomous agents using the tools of algorithmic mechanism design. In this paper we are interested in the media access problem in a wireless network. In the IEEE 802.11 protocol, for instance, all nodes wishing to access a common link must follow an algorithm whereby each one chooses a random *backoff value* in a specified range. After waiting for the amount of time indicated by the backoff value, the node attempts to transmit. The node with the smallest backoff value, if it is unique, gains access to the medium. However, if two or more nodes attempt to transmit at the same time, a collision results. In the event of a collision, all colliding nodes double the range from which the backoff value was chosen,

* Research supported by NSERC, Canada.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 124–135, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and retry. Nodes that did not collide keep the originally chosen backoff value, appropriately decremented, for the next round. Since nodes in a wireless network are autonomous agents, we cannot be sure that they will follow the protocol as specified. In particular, some nodes may try to cheat by always choosing a small backoff value, and getting an unfair share of access to the medium. If two or more nodes cheat simultaneously in this manner, then repeated collisions among cheating nodes may reduce the network throughput to zero, effectively making the network inoperative.

The problem of enforcing cooperation in a network has been studied recently, especially for network layer protocols [1]. Punishment-based approaches work by trying to isolate the misbehaving node [2,3,4]. In contrast, incentive-based or pricing-based approaches attempt to give some incentive to participating nodes to cooperate with the protocol [5,6]. For the media access problem, Kyasanur and Vaidya propose a modification to the 802.11 protocol, which supposes the presence of some trusted nodes [7]. In the modified protocol, instead of the sender choosing the backoff value, the receiver selects a random backoff value and sends it to the sender.

Game theory provides useful tools to study the behavior of selfish agents [8]. The problem of media access in a wireless network has been modeled as a game by Cagalj et.al. [9], where the authors show that non-cooperative behavior by more than one cheater can lead to network collapse. The equilibria of a game modeling an Aloha network with selfish users are analyzed by MacKenzie and Wicker [10].

In this paper, we take a *mechanism design* approach to the problem. Our goal is to construct a strategic game with actions and utility functions that automatically induces honest (*i.e.* protocol-compliant) behavior among players that are merely choosing actions selfishly to maximize their respective utility functions. Since each player may have its own valuation of any given outcome, the utility function will include not only the agent's intrinsic valuation of the outcome, but also an incentive or payment that the mechanism will pay to the player to elicit honest play. Nisan and Ronen [11] introduced the term *algorithmic mechanism design* for their framework of studying algorithms that assume that the participants all act according to their own self-interest. Their model is specific to optimization problems, and much of the work that followed (for example, [12]) has focused on the same class of problems, and the mechanisms designed are the so-called VCG-mechanisms, in which truth-telling is shown to be a dominant strategy for every player. In the wireless network setting, Anderegg and Eidenbenz propose a routing protocol for ad hoc networks, called Ad hoc-VCG, which implements a VCG mechanism that is guaranteed to find the minimum energy path in the network [13]. As far as we know, there has been no work that uses a mechanism design approach to the wireless media access problem.

The wireless media access problem, in its full generality, corresponds naturally to a *dynamic* game since nodes can (and do) modify their actions in response to the outcome of previous rounds. In this paper, we investigate games that try to model *a single round* of the media access problem. To wit, we assume that the k

nodes (the players) are competing for access to the shared wireless medium using a backoff protocol where the jth node should choose a backoff value *uniformly at random* from a range given $[1, n_j]$ (the contention window). Our goal is to invent utility functions such that there is a *unique* mixed strategy Nash equilibrium which corresponds to each player faithfully following the protocol, *viz.* choosing a backoff value uniformly at random. We stress here that it is not difficult to construct a game with a mixed strategy Nash equilibrium that corresponds to the uniform distribution (or indeed, any other distribution); the challenge lies in ensuring that this equilibrium is *unique* and not just one among many possible equilibria.

In fact, we consider a more general situation. Suppose that we are given an *apriori* distribution profile α^* that is desired, *viz.* we wish to invent a strategic game that *realizes* exactly this distribution as its unique mixed-strategy Nash equilibrium. We show that this is possible under certain circumstances: when the players have the same number of actions, and also when players have different numbers of actions but for the largest number of actions, there exist at least two players with that many actions. On the negative side, we prove that if there are exactly two players that have different number of actions, then constructing a game that realizes a given, full-support profile as its unique Nash equilibrium, is impossible.

2 Preliminaries

For any fixed positive integer m, let $[1, m]$ denote the set of integers $\{1, 2, \ldots, m\}$. We shall be concerned with a two such sets that arise in our strategic games:

- A finite ordered set of $k \geq 2$ players, $P = [1, k]$.
- A finite set of $n_j \geq 1$ possible actions (or pure strategies), $A_j = [1, n_j]$, for each player $j \in P$.

We use the terminology **profile** for an ordered tuple that is typically indexed by an index set such as P. Following standard game-theoretic notation, an *outcome* of the game is represented by an *action profile*, $s = (i_j)_{j \in P}$, with the interpretation that every player $j \in P$ performs the corresponding action $i_j \in A_j$ in the outcome.

A *utility function* is a function $u : \ A^k \to \mathrm{I\!R}$ that associates the real value $u(s)$ with the action profile s. A game specifies the utility function profile, $(u_j)_{j \in P}$ that is interpreted as follows: for any action profile s, the corresponding utility profile for the players is simply $(u_j(s))_{j \in P}$. We assume rational players that play independently (without any collusion) and seek to maximize their respective utilities, i.e. all else being equal, a player will prefer an action $a \in A_j$ over some other action $b \in A_j$ if the corresponding utility is strictly higher.

A *mixed strategy* α_j for a player $j \in P$ is a *discrete probability distribution* over its action set A_j. One interpretation of such a mixed strategy is that the player chooses any action $a \in A$ independently with probability $\alpha_j(a)$. A *mixed-strategy Nash equilibrium* is a special distribution profile, $\alpha^* = (\alpha_j^*)_{j \in P}$, with the property that a player cannot increase its (expected) utility by unilaterally changing its own distribution in the profile. A mixed-strategy Nash equilibrium α^* can also be characterized as follows: for every player $j \in P$, it holds that any pair of distinct actions $a, b \in A_j$ in the *support* of its distribution α_j^* have *exactly the same expected utility*. This characterization is important and will be used extensively in the discussion to follow.

We say that $\alpha = (\alpha_j^*)_{j \in P}$ is a *full-support* distribution profile if for every player $j \in P$, the support of the distribution α_j is the entire action set A_j, *i.e.* $\alpha_j(i) > 0$ for all $j \in P, i \in A_j$. In this paper, we are always interested in full-support distribution profiles.

3 Designing Games with Identical Player Strategies

We first consider the situation where players have the same set of actions, $A = [1, n]$. We will use the index set $P = [1, k]$ for the players[1] Suppose that we have an *apriori* known full-support distribution profile $\alpha^* = (\alpha_j^*)_{j \in P}$. We are interested in the following general question: is it possible to design a strategic game among the players with a *unique* Nash equilibrium that is given by the profile α^*? In what follows, we will construct such a game.

For ease of description, we will treat each of the index sets P and A as being *circularly ordered*, i.e. with player $k+1$ being interpreted as player 1, with action $n+1$ being interpreted as action 1 etc. We design our game with the following property: *The utility function for player $j \in P$ depends only on its own actions and those of its predecessor, player $j-1$.* This property allows us to present the game using an abbreviated version of the usual strategic form of presentation.

In particular, we can represent the utility function u_j for each player $j \in P$ as a two dimensional matrix M_j with n rows and n columns. The interpretation of this matrix is as follows: $M_j(a, b)$ is the value of the utility function u_j when applied to *every action profile* s in which player $j-1$ performs action b and player j performs action a. Thus, one thinks of the rows of M_j as being indexed by the pure strategies of player j and the columns of M_j as being indexed by the pure strategies of player $j-1$.

It is convenient to describe the construction of any matrix M_j as being spread over two steps. Let I_n be the identity matrix with n rows and columns. Consider the following matrix, V, obtained from I_n by shifting down (circularly) the rows of I_n:

$$V_n := \begin{bmatrix} 0 & \dots & 0 & 1 \\ 1 & \dots & 0 & 0 \\ & \dots & & \\ 0 & \dots & 1 & 0 \end{bmatrix} \tag{1}$$

[1] In principle, players need not have exactly the same set of actions; what matters is that they have the *same number* of possible actions.

For every player $j \in P$, an intermediate matrix $\hat{M}_j$ is defined as follows:

$$\hat{M}_j = \begin{cases} V_n & \text{if } j = 1 \\ I_n & \text{otherwise} \end{cases} \tag{2}$$

Now, let $\alpha^* = (\alpha_j^*)_{j \in P}$ be the desired unique mixed strategy equilibrium profile. Then, the utility matrix M_j for player $j \in P$ is defined as follows. Recall that the columns of M_j correspond to actions of the previous player $j-1$ in the circular ordering of P. For any pair of actions $a, b \in A$, we have

$$M_j(a, b) = \hat{M}_j(a, b)/\alpha_{j-1}^*(b) \tag{3}$$

In words, we obtain M_j from $\hat{M}_j$ by *scaling* each entry by the reciprocal of player $(j-1)$'s column probability for that column. Note that the matrices are well-defined since we have assumed that α^* is a full-support distribution profile for *every* player and therefore, the probability in the denominator of (3)'s right-hand side is always non-zero. We will now establish that the game defined above has a *unique* Nash equilibrium where player j chooses *exactly* the corresponding desired mixed strategy α_j^*.

Consider any mixed strategy, $\alpha = (\alpha_j)_{j \in P}$, for the game. Under this mixed strategy, player j's expected payoff for an action a is easily shown via (3) to be:

$$U_\alpha^j(a) = \begin{cases} \alpha_k(a-1)/\alpha_k^*(a-1) & \text{if } j = 1 \\ \alpha_{j-1}(a)/\alpha_{j-1}^*(a) & \text{otherwise} \end{cases} \tag{4}$$

Specifically, for the case when $\alpha = \alpha^*$, the right hand side is identically equal to 1 for *all actions of all the players* thus establishing that every player has equal payoffs for all its pure actions under distribution profile α^*. Thus, α^* is indeed a full support, mixed strategy Nash equilibrium for the game. It remains to show that this equilibrium is unique. We start with a useful definition specific to our game.

Definition 1. *Let $\alpha = (\alpha_j)_{j \in P}$ be a mixed strategy profile that differs from α^*. We say that an action a is α-deficient for a given player j if $\alpha_j(a)/\alpha_j^*(a) < 1$.*

Lemma 1. *Suppose that the profile $\alpha = (\alpha_j)_{j \in P}$ is a mixed strategy Nash equilibrium for the game. Then, the following implications hold:*

1. *If action a is α-deficient for player k, then $\alpha_1(a+1) = 0$.*
2. *For $1 \leq j < k$, if action a is α-deficient for player j, then $\alpha_{j+1}(a) = 0$.*

Proof. Let α be a Nash equilibrium for the game. To prove the first implication (Lemma 1.1 above), assume that the antecedent is true. Both α_k and α_k^* being probability distributions, $\sum_{b \in A} \alpha_k(b) = 1 = \sum_{b \in A} \alpha_k^*(b)$ and from this it follows that if $\alpha_k(a)/\alpha_k^*(a) < 1$, then there must be another action $b \neq a$ for which $\alpha_k(b)/\alpha_k^*(b) > 1$.

Applying (4) above with $j = 1$, we conclude that under the mixed strategy α, player 1 will have a strictly larger payoff for playing the pure strategy $b+1$ as compared to playing the pure strategy $a+1$. Consequently, action $a+1$ cannot be in the support of the claimed optimal strategy α_1 for player 1, and hence, $\alpha_1(a+1) = 0$. An almost identical argument works for the second part of the lemma except that we use (4) for the case when $1 < j \leq k$. □

Theorem 1. *The game outlined above is a k-player, n-strategy game that has the unique mixed strategy Nash equilibrium given by the full support distribution profile* $(\alpha_j^*)_{j \in P}$.

Proof. We have already shown that α^* is a Nash equilibrium for the game. To establish uniqueness, we will show that assuming a different equilibrium profile, $\alpha \neq \alpha^*$, yields a contradiction. Lemma 1 provides the intuition for this: it asserts that if an action a is α-deficient for a player j then so is action a for player $j+1$, except in the case when $j = k$ and then it is action $a+1$ that is α-deficient for player 1.

More formally, let $\alpha \neq \alpha^*$. Then there must be some player j for whom there is an action a that is α-deficient. If player j is someone other than player k (*i.e.* where $1 \leq j < k$), then Lemma 1.2 implies that $\alpha_{j+1}(a) = 0$. In fact, we can apply Lemma 1.2 in succession $(k-j)$ times to deduce that:

$$\begin{aligned} \alpha_j(a)/\alpha_j^*(a) < 1 &\implies \alpha_{j+1}(a) = 0 \\ &\implies \alpha_{j+2}(a) = 0 \\ &\cdots \\ &\implies \alpha_k(a) = 0 \end{aligned}$$

In other words, if α differs from α^*, then player k must have a α-deficient action.

Without loss of generality, there must be an action a that is α-deficient for player k with the further property that action $a+1$ is *not* α-deficient for player k, *i.e.* with $\alpha_k(a+1)/\alpha_k^*(a+1) \geq 1$. Now, Lemma 1.1 applies, and we get $\alpha_1(a+1) = 0$. Thus, action $(a+1)$ is α-deficient for player 1 and by applying Lemma 1.2 in succession $(k-1)$ times, we conclude that $\alpha_k(a+1) = 0$. This contradicts our earlier assertion that action $a+1$ is not α-deficient for player k. Thus, contrary to assumption, α cannot differ from α^*; the game exhibits the unique Nash equilibrium profile α^*. □

We note that Theorem 1 can be easily specialized to the case that is relevant for the single round version of the medium access control problem, with all backoff values in the range $[1, n]$ and the desired distribution being the discrete uniform distribution for all players.

4 Designing Games with Non-identical Player Strategies

In this section, we consider games where the players do not have the same number of strategies available to them, but we still desire to achieve a target distribution profile with full-support component distributions. In this case, we first show that it is impossible to design a game with this feature, when there are only two players.

We assume, in particular, that player 1 has the set of strategies $A_1 = [1, m]$ and player 2 has the set of strategies $A_2 = [1, n]$ with $m > n$. As before, we are given an *apriori* full-support distribution profile (α_1^*, α_2^*).

Theorem 2. *Given players* 1, 2 *and the full-support distribution profile* (α_1^*, α_2^*) *as described above, there is no 2-player strategic game which can realize the given profile as its unique mixed-strategy Nash equilibrium.*

Proof. Suppose, to the contrary, that such a game can be realized with utility functions represented by the matrices M_1 and M_2 for players 1 and 2 respectively. We will assume, as before, that the rows are indexed by the corresponding player's actions, *i.e.* that $M_1(a, b)$ (respectively, $M_2(b, a)$) is the utility for player 1 (respectively, for player 2) when player 1's action is $a \in [1, m]$ and player 2's action is $b \in [1, n]$. Recall that $m > n$.

Since (α_1^*, α_2^*) is a mixed-strategy Nash equilibrium for the game, and since both the component distributions have full support, it follows that the expected utility of any distinct pair of pure strategies for player 1 (using matrix M_1) and for player 2 (using matrix M_2) are equal. More to the point, the distribution α_1^* satisfies the system of equations:

$$\sum_{j=1}^{m} q_j [M_2(1, j) - M_2(i, j)] = 0 \qquad \text{for } 2 \leq i \leq n$$

$$\sum_{j=1}^{m} q_j = 1 \tag{5}$$

and the distribution α_2^* satisfies the system of equations:

$$\sum_{i=1}^{n} p_i [M_1(1, i) - M_1(j, i)] = 0 \qquad \text{for } 2 \leq j \leq m$$

$$\sum_{i=1}^{n} p_i = 1 \tag{6}$$

Since the system of equations (5) has more variables than equations, it must either have no solutions or an infinite number of solutions. We know that it has at least one solution, *viz.* α_1^*. Hence, it must have infinitely many solutions and by convexity, at least one of these solutions must be a distribution with full support that differs from α_1^*. Let β be such a distribution. Then, it follows that *any* strategy by player 2 would be a best response to player 1 adopting the strategy β. Hence, α_2^* is a best response to β. Similarly, since α_2^* is a full support distribution that satisfies the system of equations (6), we also conclude that any strategy, including strategy β, would be a best response by player 1 to player 2 adopting strategy α_2^*.

Hence, the game has at least two mixed strategy Nash equilibria given by the profiles (α_1^*, α_2^*) and (β, α_2^*), a contradiction. $\square$

We note that Theorem 2 implies that the result of Theorem 1 only works when the target distribution profile has full support. We next show that under some circumstances, for $k \geq 3$ players, it is possible to create games that realize given full-support distribution profiles as unique Nash equilibria. The first result, a simple corollary of Theorem 1, is stated here without proof.

Theorem 3. *Consider a set of players P that can be partitioned into subsets $P_1, P_2, \ldots, P_m$ where $|P_j| > 1$ for all $1 \leq j \leq m$. If the set of strategies for all players in subset P_j is $[1, n_j]$, then for any given full-support, distribution profile α^*, there exists a game whose unique Nash equilibrium is the profile α^*.*

The conditions in the above theorem can be relaxed. We show that so long as there are at least *two* players with the (same) *largest* number of strategies, we can create a game corresponding to any *apriori* given, full-support distribution profile α^*. Our basic idea is to create utility matrices so that player j's utilities depend on the actions chosen by player $j-1$ and player k.

Consider the players arranged in non-decreasing order of the number of actions available to them. Let n_j be the number of actions available to player j. Then $n_{k-1} = n_k$; the last two players have the same number of actions. For the remaining players, we will make the simplifying assumption that for all $j \in [1, k-2]$, $n_j < n_{j+1}$. It will be obvious from the construction how this assumption can be relaxed. The game itself is specified over two stages starting with unscaled utility matrices which will subsequently be scaled appropriately in a second stage. For convenience, we will assume that $n_0 = 0$ henceforth.

In the first stage, we will represent the utilities for j as simple unscaled matrices with player j's actions represented by the rows. Recall that for any $n \geq 1$, I_n is the identity matrix with n rows and n columns. The matrices are described below:

- Player 1's utilities only depend on player k's actions; the utility matrix (unscaled) is:

$$\hat{M}_1 = [V_{n_1} \mid 0]$$

  where V_{n_1} is the identity matrix with its rows shifted down once (circularly). The 0 sub-matrix corresponds to actions $n_1 + 1, \ldots, n_k$ of player k.
- Player k's utilities only depend on player $(k-1)$'s actions; the utility matrix (unscaled) is:

$$\hat{M}_k = I_{n_{k-1}}$$

  Note that $n_{k-1} = n_k$.
- For every other player j (hence, $2 \leq j \leq (k-1)$), the utilities depend both on player k as well as the previous player $(j-1)$. We represent the utilities as separate $(n_j \times n_{j-1})$ matrices for each of the actions $1, \ldots, n_k$ of player k. The matrices are divided into three groups; each matrix has an *upper sub-matrix* consisting of the first n_{j-1} rows and a *lower sub-matrix* consisting of the remaining $(n_j - n_{j-1})$ rows:

- For player k's action $a \in [1, n_{j-2}] \cup [n_j + 1, n_k]$, the matrix is:

$$\hat{M}_j^a = \begin{bmatrix} I_{n_{j-1}} \\ \hline 0\,0\ldots 1 \\ \ldots \\ 0\,0\ldots 1 \end{bmatrix}$$

  Every row in the lower sub-matrix is identical, and equals $[0,\ 0, \ldots,\ 1]$, the last row of the identity sub-matrix.
- For player k's action $a \in [n_{j-2} + 1, n_{j-1}]$, the matrix is:

$$\hat{M}_j^a = \begin{bmatrix} I_{n_{j-1}} \\ \hline x_j\ x_j\ \ldots\ 1 + x_j \\ \ldots \\ x_j\ x_j\ \ldots\ 1 + x_j \end{bmatrix}$$

  Every row in the lower sub-matrix is identical, and equals $[x_j,\ x_j, \ldots,\ 1+ x_j]$ for a value $x_j > 0$ to be determined later.
- Let $y_j > 0$ be a value to be determined. For action $(n_{j-1} + i)$ of player k, with $i \in [1, n_j - n_{j-1}]$, the corresponding matrix is:

$$\hat{M}_j^{n_{j-1}+i} = \begin{bmatrix} I_{n_{j-1}} \\ \hline C_{i,j} \end{bmatrix}$$

  where $C_{i,j}$, the lower sub-matrix, has as its ith row, the row vector

$$[-(1 + y_j),\ -(1 + y_j), \ldots,\ -(1 + y_j),\ -y_j],$$

  and whose remaining rows are all identically equal to the row vector

$$[-y_j,\ -y_j, \ldots,\ -y_j,\ 1 - y_j]$$

As before, we obtain the actual utility matrices by scaling the above matrix entries by the reciprocals of the α^*-probabilities of the actions of the relevant players that influence any given entry. Thus, for instance, the scaled matrix M_j^a for any player $j \in [2, k - 1]$, is obtained from the corresponding unscaled matrix $\hat{M}_j^a$ by multiplying each entry in column b by

$$\frac{1}{\alpha_{j-1}^*(b)\alpha_k^*(a)}$$

and so on. The scaled utility matrices define our game. We first show that if x_j and y_j are chosen so that

$$x_j(n_{j-1} - n_{j-2}) = y_j(n_j - n_{j-1}) + 1 \tag{7}$$

holds true, then the profile α^* is indeed a Nash equilibrium for the game. Accordingly, fix a game with values of x_j and y_j that satisfy (7), *e.g.* we may choose $x_j = 1$ and $y_j = (n_{j-1} - n_{j-2} - 1)/(n_j - n_{j-1})$.

Now, if an equilibrium α that differs from α^* is assumed, then we can show first that some action for player k must be α-deficient. The intuition behind this is that the upper submatrix (the identity) in every player $j > 1$'s utilities forces this conclusion (akin to Lemma 1.2. Once we have established that some action for player k must be α-deficient, we can now use the special properties of the lower submatrices in the utility definitions to show a contradiction. We defer a detailed proof of the following result to the full paper.

Theorem 4. *Suppose we are given k players with action sets $[1, n_j]$ for every player j such that $n_1 < n_2 < \ldots < n_{k-1} = n_k$. Then given any full support distribution profile α^* for this collection of players and actions, there is a game whose unique Nash equilibrium is α^*.*

It is easy to see that some simple modifications to the construction described in this section can yield a game in which there could possibly be more than one player with action set $[1, n_j]$ for any j, and for which an *apriori* given distribution is the only possible Nash equilibrium.

5 Implementing the Mechanism

We briefly describe how the mechanism described in Section 3 can be used to compute payments to participants to induce cooperation in the media access protocol. Recall that actions belong to the set $[1, n]$. For any action profile s, let $\min(s)$ denote the minimum among all actions in the profile. The profile s is said to induce an collision if there are two or more players that have the same action $\min(s)$ in the profile s.

There can be many possibilities for the natural valuation accorded by players to an outcome of the game. We describe one possible simple valuation function here. We assume that the valuation functions are symmetric, for example, all players have the *same* natural valuation for getting access to the medium in a particular time step. Specifically, we assume that every player p places a different valuation on being able to transmit successfully at time slot i, on someone else getting access to the medium, and the event of a collision. For an action profile $s = (i_1, i_2, \ldots, i_k)$,

$$v_j(s) = \begin{cases} t_i, & \text{if } s \text{ does not induce a collision and } i_j = i = \min(s) \\ d, & \text{if } s \text{ does not induce a collision, and } i_j \neq \min(s) \\ e, & \text{if } s \text{ induces a collision.} \end{cases}$$

As noted above, many other valuation functions are possible. For instance, player j may have different valuations for profiles that cause collisions in different time steps. Every player tries to maximize its utility function as given by the utility matrices presented in Section 3. The utility function corresponding to an action profile s can be thought of as the sum of the natural valuation of the player for that profile and the payment that would be made by the mechanism. Hence, the payments made by the mechanism to a player p_j for a profile s can be calculated as $u_j(s) - v_j(s)$. If a fixed price is to be charged for participation in the media access protocol, the payment can be implemented as a discount in this price.

The implementation described above assumes that the network authority is able to assign a payment based on the action profile s, that is, it knows the actions chosen by all players. However, in practice, the network authority can at most expect to know the backoff value (*i.e.* strategy) chosen by the winning player, or the fact that there was a collision. In particular, this implies that the network authority cannot distinguish between any two profiles s and s' such that $\min(s) = \min(s')$ where both s and s' induce collisions, and thus, the utilities for the two different profiles to any particular player must be identical. Once again, different players may earn different utilities for the same profile. With the given constraints, it can shown quite easily that any such game has least one pure strategy Nash equilibrium, *viz.* the action profile $s' = \{1, 1, \ldots, 1\}$. As a result, we get:

Theorem 5. *For $k > 2$ and $n \geq k$, there is no k-player, n-action game where $u_i(s) = u_i(s')$ for any two profiles s and s' such that $\min(s) = \min(s')$, and both s and s' induce collisions, which can realize a given full-support distribution profile α^* as its unique mixed-strategy Nash equilibrium.*

6 Discussion

In this paper, we have introduced the idea of mechanisms to induce random choice, which can be used to ensure compliance with protocols in which independent agents are to make random choices. Motivated by the wireless media access problem, for any given full-support distribution profile, we defined a game that has a unique mixed strategy Nash equilibrium realizing that profile, so long as all players have the same number of strategies. We also showed that for more than two players with different numbers of strategies, under certain conditions, it is possible to design games realizing any fixed distribution profile. In contrast, we showed that for two players with different numbers of strategies, it is impossible to design a game with a unique mixed strategy Nash equilibrium corresponding to any distribution profile.

In practice, wireless media access more closely resembles a dynamic game; in the event of a collision, all colliding nodes double their values of contention window, and retry. Nodes that did not collide keep the originally chosen backoff value, appropriately decremented, for the next round. As a result of this, in any given round, nodes not only have different contention window values, but the backoff value is related to the history of previous rounds. Future work aims at analyzing this in the context of dynamic mechanism design.

Acknowledgments

The authors are grateful to an anonymous referee whose suggestions simplified many aspects of the games constructed in the paper.

References

1. Conti, M., Gregori, E., Maselli, G.: Cooperation issues in mobile ad hoc networks. In: ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04). (2004) 803–808
2. Buchegger, S., Le Boudec, J.Y.: Performance analysis of the CONFIDANT protocol. In: Proceedings of MOBIHOC. (2002) 226–236
3. Marti, S., Guili, T., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of ACM MOBICOM. (2000) 255–265
4. Michiardi, P., Molva, R.: CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Proceedings of Communications and Multimedia Security. (2002) 107–121
5. Buttyan, L., Hubaux, J.P.: Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks. Technical Report DSC/2001, Swiss Federal Institue of Technology (2001)
6. Zhong, S., Chen, J., Yang, Y.R.: Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In: Proceedings of IEEE INFOCOM 2003. Volume 3. (2003) 1987–1997
7. Kyasanur, P., Vaidya, N.: Selfish MAC layer misbehavior in wireless networks. IEEE Transactions on Mobile Computing **4** (2005) 502–516
8. Osbourne, M., Rubenstein, A.: A Course In Game Theory. The MIT Press (1996)
9. Cagalj, M., Ganeriwal, S., Aad, I., Hubaux, J.P.: On selfish behavior in CSMA/CA networks. In: Proceedings of IEEE INFOCOM 2005. Volume 4. (2005) 2513–2524
10. MacKenzie, A., Wicker, S.: Selfish users in Aloha: a game theoretic approach. In: Proceedings of IEEE Vehicular Technology Conference. (2001) 1354–1357
11. Nissan, N., Ronen, A.: Algorithmic mechanism design. In: STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing. (1999) 129–140
12. Feigenbaum, J., Papadimitrou, C., Sami, R., Shenker, S.: A BGP-based mechanism for lowest-cost routing. In: Proceedings of the 21st Symposium on Distributed Computing. (2002) 173–182
13. Anderegg, L., Eidenbenz, S.: Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In: MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking. (2003) 245–259

Bayesian Optimal No-Deficit Mechanism Design*

Shuchi Chawla[1], Jason D. Hartline[2], Uday Rajan[3], and R. Ravi[4]

[1] Microsoft Research, Silicon Valley Campus, Mountain View, CA
shuchi@cs.wisc.edu
[2] Microsoft Research, Silicon Valley Campus, Mountain View, CA
hartline@microsoft.com
[3] Ross School of Business, University of Michigan, Ann Arbor, MI
urajan@bus.umich.edu
[4] Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA
ravi@cmu.edu

Abstract. One of the most fundamental problems in mechanism design is that of designing the auction that gives the optimal profit to the auctioneer. For the case that the probability distributions on the valuations of the bidders are known and independent, Myerson [15] reduces the problem to that of maximizing the common welfare by considering the *virtual valuations* in place of the bidders' actual valuations. The Myerson auction maximizes the seller's profit over the class of all mechanisms that are truthful and individually rational for all the bidders; however, the mechanism does not satisfy ex post individual rationality for the seller. In other words, there are examples in which for certain sets of bidder valuations, the mechanism incurs a loss.

We consider the problem of merging the worst case *no-deficit* (or ex post seller individual rationality) condition with this average case *Bayesian expected profit maximization* problem. When restricting our attention to ex post incentive compatible mechanisms for this problem, we find that the Myerson mechanism is the optimal no-deficit mechanism for *supermodular* costs, that Myerson merged with a simple *thresholding mechanism* is optimal for *all-or-nothing* costs, and that neither mechanism is optimal for general *submodular* costs. Addressing the computational side of the problem, we note that for supermodular costs the Myerson mechanism is NP-hard to compute. Furthermore, we show that for all-or-nothing costs the optimal thresholding mechanism is NP-hard to compute. Finally, we consider relaxing the ex post incentive compatibility constraint and show that there is a Bayesian incentive compatible mechanism that achieves the same expected profit as Myerson, but never incurs a loss.

1 Introduction

Suppose a seller is able to provide a service at total cost C to any number of users. Suppose further that the seller has done market research to determine the

* This work was done at Carnegie Mellon University and sponsored in part by the ALADDIN project.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 136–148, 2006.
© Springer-Verlag Berlin Heidelberg 2006

probability distribution from which the user valuations for receiving the service are drawn. What selling mechanism should the seller then run to obtain the highest possible profit? In a seminal paper [15], Myerson essentially answers this question: If the seller aims to maximize their expected profit, they must first compute each user's *virtual valuation*, and then sell the item to all users with non-negative virtual valuation if the sum of their virtual valuations is above the cost C. The Myerson mechanism is optimal for expected revenue, in the class of all mechanisms that induce the users to participate and reveal their preferences truthfully. However, it turns out that in many natural scenarios, this mechanism has a deficit on some possible instances of the users' values. A seller that is averse to such a loss might prefer a different mechanism.

We consider the general problem of Bayesian optimal mechanism design for arbitrary single-parameter agent problems (see e.g., [12,2,1]) when the seller requires the mechanism to never produce a deficit. Here, the seller must pay a cost that is a function of the outcome that the seller chooses. A deficit would arise if the total payments of the agents does not cover the cost of the outcome produced. In a single-parameter agent problem each agent has a publicly known partitioning of possible outcomes into two sets, the *reject* set and the *accept* set. It is assumed that agent i has valuation zero for any outcome in the reject set and private valuation v_i for any outcome in the accept set. For auction-like problems, agent i's accept set is simply the set of allocations where agent i is allocated their desired good (or service) and the reject set is the set of allocations where i is not allocated their desired good. The truthtelling strategy for agent i would be to report to the mechanism their true private valuation, v_i.

We follow the standard economics approach to profit maximization and assume that the agents' private valuations come from a known probability distribution. Our goal then is to design the seller optimal mechanism given knowledge of this distribution. We assume that the agents valuations are independent but not necessarily identically distributed.[1]

Motivating Problems

This paper considers a number of motivating problems, all of which fit in this single-parameter agent framework. Consider the following examples:

Fixed cost excludible good. The seller must pay a fixed cost C if any items are sold and zero otherwise. A motivating example of such a good is a digital good with production cost C and zero marginal cost. This is a special case of the general *multicast pricing* problem considered in [5,6,13].

Fixed cost non-excludible good. There is a fixed cost, C, for providing the good or service to all users and no cost for serving nobody. However, the mechanism is not allowed to serve some users and not others (i.e., the cost for such allocations is infinite). We will sometimes refer to this as the *all-or-none* case. The classic example of a fixed cost non-excludible good is the bridge building problem where if the bridge is built then anyone can use it.

[1] It is NP-hard to compute the optimal auction when valuations are correlated [16].

Submodular costs. The additional (marginal) cost in providing the good to any users is a decreasing function of the set of users already being provided. The excludible and non-excludible fixed cost problem and the multicast pricing problem are special cases of the general submodular cost problem. Goods with concave production costs or increasing returns to scale fall in this category.

Combinatorial auction (single-parameter). Each agent desires a subset of a set of items. The cost function is such that allocations to agents with disjoint subsets have cost zero and all other allocations have infinite cost. See e.g., [12,1].

Supermodular costs. The additional cost in providing the good to any users is an increasing function of the set of users already being provided. The single-parameter combinatorial auction problem is a special case of a supermodular cost function.

Mechanism Design Solution Concepts

The fundamental difference between mechanism design and algorithm design is that the inputs to a mechanism are the private values of selfish agents that will attempt to submit bids that result in outcomes that maximize their own *utility*. We adopt the following solution concepts.

Ex post incentive compatibility. Otherwise known as *truthful* or *strategyproof* mechanisms, ex post incentive compatible mechanisms (via the revelation principle) are such that each agent, independent of the acts of any other agent, has a dominant strategy of stating their true valuation as their bid.

Bayesian incentive compatibility. Bayesian incentive compatible mechanisms are those where each agent has an optimal strategy of bidding their true valuation given that the other agents values come from a prior distribution and that all other agents bid their true values. Note that such a truthtelling strategy may not be optimal ex post, i.e., once the bids of other agents are known.

Overview of Results

The major focus of this paper, besides describing the Bayesian optimal no-deficit mechanism, is to study the complexity of computing it. Myerson's optimal mechanism solves the single-parameter agent optimal mechanism design problem for any cost function given that the seller only wants to maximize their expected profit and spurious deficits are acceptable. For submodular costs, via a general algorithm due to Iwata et al. [10], it is possible to compute this optimal mechanism. However, for the single parameter combinatorial auction (and, thus general supermodular costs) this computational problem is NP-hard [12]. Of course the usual questions arise here as to whether it is possible to approximate the optimal mechanism via a polynomial time computation. For this problem, it is relatively easy to see that Myerson's reduction from the efficient mechanism to the optimal mechanism via virtual valuations respects approximations. Given an incentive compatible mechanism that approximates efficiency, the Myerson approach can be used to obtain an incentive compatible mechanism that gives the same approximation factor against the optimal mechanism.

For the problem of designing the ex post incentive compatible optimal no-deficit mechanism we consider both the form that the optimal mechanism takes as well as the problem of computing it. Like above, the answer to these questions depends on types of cost functions we are considering. We show that for supermodular costs functions, the Myerson mechanism is indeed no-deficit. Of course, by the above discussion such a mechanism is hard to compute. For the submodular case, and even the special case of a fixed cost excludible good, we show that Myerson is not no-deficit (Section 4). We then consider the most natural way to try to obtain a no-deficit mechanism that achieves good expected profit: merging the Myerson mechanism which has optimal expected profit with a thresholding mechanism, e.g. Moulin and Shenker's [14] cost sharing mechanism, which has no-deficit. We show that even for the fixed cost excludible good problem when bidders are independent and identically distributed, such a mechanism is not optimal (Section 5). We further show the somewhat surprising result that even though in this case the problem is completely symmetrical, the optimal deterministic no-deficit mechanism is not. None-the-less, as these thresholding mechanisms are intuitively easy to understand, we ask two questions, first, when are thresholding mechanisms optimal, and second can we compute them. We show that these mechanisms are indeed optimal for all-or-nothing costs; yet computing the optimal thresholding mechanism on this special case is NP-hard.

We then consider relaxing our solution concept from ex post incentive compatibility to Bayesian incentive compatibility. We show that while the ex post incentive compatible payment rule of Myerson is not no-deficit on some realizations of the agents' valuations, there is a Bayesian incentive compatible payment rule for Myerson's mechanism that obtains the same expected profit as the original Myerson payment rule and guarantees that there is never a deficit. We leave the problem of computing this payment rule as an open question.

Related Work

This work is based heavily on results of Myerson [15] on optimal mechanism design and generalizations observed by Bulow and Roberts [3]. Cornelli re-derives these results for the special case of a fixed cost excludible good and considers the related problem of designing optimal non-direct revelation mechanisms (where the set of allowable bids is a subset of possible valuations of the bidders) [4]. Mehta and Vazirani [13] consider the related computational question of how to compute the optimal "take it or leave it" offers for each agent prior to seeing any bids, for the aforementioned multicast pricing special case of submodular costs.

Another branch of related work is that of worst-case profit maximizing mechanism design. There is much work in this area. (See, for example, [9].) As an example, for the trivial cost function, Goldberg et al. give an approximately optimal worst case auction [8,7]. Fiat et al. consider the fixed cost excludible good problem and more general multicast pricing problem. They give approximately optimal mechanisms under certain assumptions [6].

2 Notation and Preliminaries

Let $S = \{1, \ldots, n\}$ denote a set of n agents. We represent the outcome of the mechanism as an *allocation* $A \subset S$ of *accepted* agents. We assume there is a general cost function $c(A)$ over allocations. As noted in the introduction, this allows us to represent any (binary) single parameter agent problem.

A cost function is said to be *submodular* if for all allocations A_1 and A_2, $c(A_1) + c(A_2) \geq c(A_1 \cup A_2) + c(A_1 \cap A_2)$. Likewise, it is said to be *supermodular*, if for all allocations A_1 and A_2, $c(A_1) + c(A_2) \leq c(A_1 \cup A_2) + c(A_1 \cap A_2)$.

Each agent i has a valuation v_i for being accepted. We assume that v_i is drawn independently from distribution F_i and corresponding density function f_i. The joint distribution, $\mathbf{F}$, is the product $F_1 \times \cdots \times F_n$. Without loss of generality, we assume that v_i is in the range $[0, h]$ for all i. We define the *virtual valuation* of agent i to be $\phi_i(v_i) = v_i - \frac{1-F_i(v_i)}{f_i(v_i)}$. Where v_i is implicit, we will refer to ϕ_i as agent i's virtual valuation. We restrict our attention to distributions F_i for which ϕ_i is an increasing function of v_i. This is implied by the *monotone hazard rate* assumption which is standard in mechanism design.

Assume for mechanism $\mathcal{M}$ that the agents submit bids $\mathbf{b} = (b_1, \ldots, b_n)$. We denote the allocation served by $\mathcal{M}(\mathbf{b})$. When $\mathcal{M}$ is a randomized mechanism, $\mathcal{M}(\mathbf{b})$ is a random variable. For valuations $\mathbf{v}$ and allocation A, we define the *surplus* of this allocation to be $\mathcal{S}_{\mathbf{v}}(A) = \sum_{i \in A} v_i - c(A)$. The *virtual surplus* we denote by $\hat{\mathcal{S}}_{\mathbf{v}}(A) = \mathcal{S}_{\phi(\mathbf{v})}(A) = \sum_{i \in A} \phi_i(v_i) - c(A)$. For ex post IC mechanisms, we have $\mathbf{b} = \mathbf{v}$, so we sometimes use $\hat{\mathcal{S}}_{\mathbf{b}}$ to denote the virtual surplus.

Let $p_i(b_i)$ denote the payment charged by mechanism $\mathcal{M}$ to agent i when he bids b_i. Define $q_i(b_i)$ as the probability that agent i is allocated when bidding b_i. Notice that this payment and probability are dependent on the randomization in the other bids, $\mathbf{b}_{-i}$, and the randomization in the mechanism, $\mathcal{M}$. A mechanism is incentive compatible if this agent's utility is maximized when bidding their true valuation. I.e., $v_i \in \operatorname{argmax}_b[v_i q_i(\mathbf{b}_{-i}, b) - p_i(\mathbf{b}_{-i}, b)]$. A mechanism is (ex post) *incentive compatible* (IC) if this holds for all values of the other agents bids, $\mathbf{b}_{-i}$, and *Bayesian incentive compatible* (BIC) if it holds when the other agents bid their true values, so that $\mathbf{b}_{-i}$ is drawn from the prior distribution $\mathbf{F}_{-i} = F_1 \times \cdots \times F_{i-1} \times F_{i+1} \times \cdots \times F_n$. It is well known [11] that the allocation rule and the expected payment of each agent satisfies the following conditions[2].

Lemma 1. *For any ex post incentive compatible mechanism $\mathcal{M}$, for $\mathbf{b}_{-i}$ fixed, $q_i(b_i)$ is non-decreasing in b_i, and $p_i(b_i) = b_i q_i(\mathbf{b}) - \int_{b=0}^{b=b_i} q_i(\mathbf{b}_{-i}, b) db$.*

Lemma 2. *For any Bayesian incentive compatible mechanism $\mathcal{M}$, when $\mathbf{b}_{-i}$ are drawn from $\mathbf{F}_{-i}$ then $q_i(b_i) = \mathbf{E}_{\mathbf{b}_{-i}} q_i(\mathbf{b})$ is non-decreasing in b_i, and $p_i(b_i) = b_i q_i(b_i) - \int_{b=0}^{b=b_i} q_i(b) db$.*

The above lemmas imply that for describing an incentive compatible mechanism, it is sufficient to specify an allocation rule that is monotone in the bids of each

[2] In general, the expressions for p_i may contain a constant $p_i(0)$ term, but because we are interested in profit maximization, we assume that this term is zero.

agent. Notice that for a deterministic ex post IC mechanism, the price $p_i(b_i)$ is the minimum bid that i must bid in order to be served.

In addition to incentive compatibility, we also require our mechanisms to satisfy the *no-deficit* condition defined below.

No-Deficit Condition. A mechanism $\mathcal{M}$ is said to satisfy the no-deficit condition if and only if for all bid vectors $\mathbf{b}$, the profit of the mechanism is non-negative: $\sum_i p_i(\mathbf{b}) - c(\mathcal{M}(\mathbf{b})) \geq 0$.

3 The Myerson Mechanism

Notice that the Vickrey-Clarke-Groves (VCG) mechanism applied to our single parameter setting is the mechanism that chooses the allocation that maximizes the surplus (defined above). It is easy to see that this allocation rule is monotone and thus there exist prices that incentivize agents to bid their true values. Myerson reduced the problem of Bayesian profit maximization to that of maximizing surplus via the concept of virtual valuations. He shows that the Bayesian optimal mechanism is the one that maximizes the virtual surplus. His theorem generalizes directly to our single parameter agent setting as follows.

Lemma 3. *[15] The expected profit of any truthful mechanism is exactly equal to its expected virtual surplus.*

Theorem 1. *[15] Given agents with valuations drawn from distribution* $\mathbf{F} = F_1 \times \cdots \times F_n$ *with each* F_i *satisfying the monotone hazard rate condition, the ex post IC mechanism with the maximum expected profit selects the outcome to maximize the virtual surplus, i.e.,* $\mathcal{M}(\mathbf{b}) = \operatorname{argmax}_A \hat{\mathcal{S}}_{\mathbf{b}}(A)$*. The expected profit of this mechanism is given by* $\mathbf{E}_{\mathbf{b}}\hat{\mathcal{S}}_{\mathbf{b}}(\mathcal{M}(\mathbf{b}))$*.*

One view of this theorem is that to maximize profit, first compute virtual valuations assuming that the agents bid their valuations, and then run the VCG mechanism on these virtual valuations. Payments can be determined by the payments of VCG in this setting by applying each agent's inverse virtual valuation function to their VCG payment. We refer to the mechanism that maximizes the virtual surplus as the *Myerson* mechanism.

3.1 The Discrete-Valued Case

Although all the definitions given above assume that the buyers' bids are continuous variables, it is easy to formulate similar expressions when bids are discrete-valued. We give analogs for the discrete case below. These descriptions are standard and we leave the proofs to the reader.

For the ith bidder, let $x_{i,j}$ denote the jth value that v_i can take. Let the corresponding probability be given by $f_{i,j}$, and let $F_{i,j} = \sum_{k=0}^{k=j} f_{i,k}$ denote the cumulative probability. The jth virtual valuation of bidder i is given by $\phi_{i,j} = x_{i,j} - \frac{1-F_{i,j}}{f_{i,j}}(x_{i,j+1} - x_{i,j})$.

The price $p_i(\mathbf{b})$ charged by an incentive compatible mechanism $\mathcal{M}$, when bidder i reports $b_i = x_{i,j}$, is given by $p_i(b) = b_i q_i(\mathbf{b}) - \sum_{k=0}^{k=j-1} q_i(x_{i,k}, \mathbf{b}_{-i})(x_{i,k+1} - x_{i,k})$. The virtual surplus and the Myerson mechanism are defined as before.

3.2 Some Examples and the No-Deficit Constraint

Consider the following example. There are three bidders, each having a value drawn uniformly from the interval $[0, 1]$. The cost of serving any non-empty subset of them is $C = 2$. Then, the virtual valuation function of the ith bidder is given by $\phi_i(v_i) = 2v_i - 1$. Consider the case when all the three valuations are 1. Then the total virtual valuation is 3. Therefore, the Myerson mechanism serves all three of the bidders. The payment of bidder i is given by the minimum virtual valuation at which bidder i gets served. This is $C - \sum_{j \neq i} \phi_j(v_j) = 2 - 2 = 0$. Therefore, the payment of bidder i is ${\phi_i}^{-1}(0) = 0.5$. The revenue of the mechanism at the bid vector $(1, 1, 1)$ is therefore 1.5, whereas the cost of serving the three bidders is 2. The mechanism incurs a loss.

A slightly different example shows that the ratio between the worst-case loss of the Myerson mechanism and its expected profit can be unbounded. Consider an example with n identically distributed bidders, each with bid distribution uniform over $[1, 2]$. The cost of serving any subset of the bidders is $C = 2n - 2$. The reader is encouraged to verify that the worst-case loss of the Extended Myerson mechanism in this case is $n - 2$ (for the bid vector $(2, \cdots, 2)$), whereas the expected profit of the mechanism is less than 2.

4 The No-Deficit Constraint for Supermodular Functions

In this section we prove that the Myerson mechanism always satisfies the no-deficit constraint if the cost function is supermodular. We start with a few properties of the Myerson mechanism that will be useful in our analysis.

4.1 Strong Monotonicity of Allocations in the Myerson Mechanism

We show that if any bidder served by Myerson unilaterally increases her bid, then the allocation of the mechanism stays the same. Note that if a bidder being served by the mechanism raises her bid, truthfulness (and thus, monotonicity) implies that the bidder continues being served. The next lemma however says something stronger—when the bidder raises her bid, no other bidder gets added or removed from the set being served.

Lemma 4. *Given any two bid vectors* $\mathbf{b}$ *and* $\mathbf{b}'$ *with* $b_j = {b_j}'$ *for all* $j \neq i$*, and* $b_i < {b_i}'$*, if* $i \in \text{Myerson}(\mathbf{b})$*, then* $\text{Myerson}(\mathbf{b}') = \text{Myerson}(\mathbf{b})$.

Proof. For an allocation A, let $\Delta(A) = \hat{\mathcal{S}}_{\mathbf{b}'}(A) - \hat{\mathcal{S}}_{\mathbf{b}}(A)$. Then, for any allocation A containing i, $\Delta(A) = {b_i}' - b_i > 0$, whereas, for any other allocation, $\Delta(A) = 0$. If $i \in \text{Myerson}(\mathbf{b})$, then $\Delta(\text{Myerson}(\mathbf{b})) \geq \Delta(A)$ for any allocation A. Also, we have $\hat{\mathcal{S}}_{\mathbf{b}}(\text{Myerson}(\mathbf{b})) \geq \hat{\mathcal{S}}_{\mathbf{b}}(A)$ for all A, by definition. Therefore, $\hat{\mathcal{S}}_{\mathbf{b}'}(\text{Myerson}(\mathbf{b}')) \geq \hat{\mathcal{S}}_{\mathbf{b}'}(A)$ for all A, and $\text{Myerson}(\mathbf{b}') = \text{Myerson}(\mathbf{b})$. □

Corollary 1. *The payment for bidder i given allocation A with $i \in A$ and other bids $\mathbf{b}_{-i}$ is the minimum bid b_i such that* $\text{Myerson}(\mathbf{b}) = A$.

Proof. Note that the payment for bidder i is less than the minimum bid b_i with $\text{Myerson}(b) = A$, because $i \in A$. Suppose the payment is $p_i(b_i) = b_i{}'$ with $b_i{}' < b_i$. Then the allocation $A_2 = \text{Myerson}(b_i{}', \mathbf{b}_{-i})$ contains i. This contradicts the lemma above, because when i increases her bid from $b_i{}'$ to b_i, the allocation changes from A_2 to $A \neq A_2$. □

4.2 Myerson Satisfies No-Deficit

Now we are ready to prove the main theorem of this section:

Theorem 2. *Myerson satisfies no-deficit for supermodular costs.*

Proof. Note that for all bid vectors $\mathbf{b}$ with $\text{Myerson}(\mathbf{b}) = A$, and for all $i \in A$, we have $\hat{\mathcal{S}}_{\mathbf{b}}(A) \geq \hat{\mathcal{S}}_{\mathbf{b}}(A \setminus \{i\})$. Then by the definition of $\hat{\mathcal{S}}_{\mathbf{b}}(A)$ and using the monotone hazard rate condition, we have $b_i \geq \phi_i(b_i) \geq c(A) - c(A \setminus \{i\})$.

Now let $\min_i(A)$ be the minimum bid b_i of bidder i, with $i \in A$, such that for some bid vector $\mathbf{b}_{-i}$, A is served, that is,

$$\min_i(A) = \min\{b_i : \exists \mathbf{b}_{-i} \text{ with } \text{Myerson}(b_i, \mathbf{b}_{-i}) = A\}.$$

Then Corollary 1 implies that the payment of bidder i at any vector $\mathbf{b}$ with $\text{Myerson}(\mathbf{b}) = A$ is given by $p_i(\mathbf{b}) \geq \min_i(A)$, which is larger than $c(A) - c(A \setminus \{i\})$ by our observation above.

Now, taking a sum over all i, we get that the total payment collected is at least $\sum_i \min_i(A) \geq \sum_i [c(A) - c(A \setminus \{i\})]$. The net profit obtained is at least $\sum_i [c(A) - c(A \setminus \{i\})] - c(A)$. Note that supermodularity implies $c(A) - c(A \setminus \{i\}) \geq c(B) - c(B \setminus \{i\})$, for any set $B \subset A$ with $i \in B$. Without loss of generality, let $|A| = k$ and $A = \{1, \ldots, k\}$. Then, the net profit is at least

$$\begin{aligned}\sum\nolimits_i [c(A) - c(A \setminus \{i\})] - c(A) &\geq \sum\nolimits_i [c(\{1, \ldots, i\}) - c(\{1, \ldots, i-1\})] - c(A) \\ &= c(A) - c(\emptyset) - c(A) = 0.\end{aligned}$$

□

4.3 Computation of the Optimal Mechanism

Next we consider the problem of computing the Myerson mechanism for supermodular costs. In particular, we consider the problem of determining the winning allocation, given the bid vector, bid distributions and the cost function. Supermodularity of the cost function implies that in general the optimal allocation is NP-hard to approximate better than an $\Omega(n^{1-\epsilon})$ factor. However, in special cases, given an approximate truthful mechanism for welfare mechanism for the same cost function, we can design an approximate truthful mechanism for profit maximization in the Bayesian setting. We obtain the following results.

Theorem 3. *Given a polynomial-time truthful deterministic mechanism $\mathcal{M}$ that α-approximates the social optimum in a worst-case setting, there exists a polynomial-time truthful mechanism $\mathcal{M}'$ that α-approximates the expected profit in a Bayesian setting.*

Theorem 4. *There exists a polynomial time mechanism that is truthful in expectation and obtains a $(1+\epsilon)$-approximation to the single-parameter combinatorial auction in a Bayesian setting.*[3]

We note that Theorem 4 is a non-trivial extension of Theorem 3 to auctions satisfying truthfulness in expectation, as the inverse virtual valuation function is not generally linear and thus it affects the expected utility of the agents. See the full paper for details.

5 Submodular Costs, Threshold Mechanisms and All-or-None Costs

In this section we consider submodular cost functions. As shown in Section 3.2, in this case, the Myerson mechanism does not always satisfy the no-deficit constraint. Intuitively, when the Myerson mechanism serves a large set A of bidders, the marginal cost of serving a bidder $i \in A$, and therefore the price charged to i, is very small. A simple way of dealing with these low costs is to supplement the Myerson mechanism with reserve prices or *thresholds* for each bidder, below which the bidder is not served. Precisely, let τ denote a budget-balanced cost-sharing method, and $\tau_i(A)$ denote the cost-share assigned to bidder i in coalition A. Then, if a mechanism serves the set A only if the bids of all bidders in A are above their respective thresholds, then the mechanism obtains prices at least $\tau_i(A)$ from each bidder $i \in A$, and therefore, meets the cost of serving the set. Furthermore, if the mechanism picks a set A with the maximum virtual surplus over all sets satisfying the thresholds, then it also achieves good expected profit. We call such a mechanism a *threshold mechanism.* Note that the price charged to a bidder still depends on other bidders' bids and not just the threshold (and can therefore change as others' bids change, even when the allocation stays the same); the threshold only ensures that this price is never too low.

A natural question to ask is whether threshold mechanisms are optimal in the class of all truthful mechanisms satisfying no-deficit. Unfortunately, this is not the case, even when the cost function is symmetric and submodular, and all the bids are identically distributed. See an example in the full paper for details.

Although threshold mechanisms are not optimal for arbitrary submodular cost functions, we now show that they are indeed optimal for a special class of cost functions, that we call *all-or-none* costs. An all-or-none cost function is one in which the only allocations served are the empty allocation or the one containing all bidders. That is, for all allocations A with $A \neq \emptyset$ and $A \neq \mathcal{B}$, we have $c(A) = \infty$.

[3] This mechanism is based on a social welfare maximizing mechanism due to Archer et al. [1] and assumes that multiple units of each item are available.

Lemma 5. *Let $\mathcal{M}$ be any truthful mechanism for an all-or-none cost function c. Then, $p_i(\mathbf{b})$ is non-increasing in bids b_j with $j \neq i$.*

Proof. Suppose that there are bidders i and j such that $p_i(\mathbf{b})$ not non-increasing in b_j. That is, there are bid vectors $\mathbf{b}$ and $bids'$ with $b_k{}' = b_k$ for all $k \neq j$ and $b_j{}' > b_j$, such that $p_i(\mathbf{b}') > p_i(\mathbf{b})$. Note that $\mathcal{M}$ is truthful, and so p_i does not depend on b_i. So we choose $b_i = b_i{}' = \frac{p_i(\mathbf{b}')+p_i(\mathbf{b})}{2}$. Now, i is served at $\mathbf{b}$ but not at $\mathbf{b}'$. However, since c is an all-or-none cost function, we have $\mathcal{M}(\mathbf{b}) = S$ (all bidders) and $\mathcal{M}(\mathbf{b}') = \emptyset$. This means that the allocation given by $\mathcal{M}$ is not a non-increasing function of b_j. Lemma 1 then implies a contradiction to the truthfulness of $\mathcal{M}$. □

Theorem 5. *For any all-or-none cost function, there exists a threshold mechanism that is optimal among the class of all truthful no-deficit mechanisms.*

Proof. Let $\mathcal{M}$ be any optimal truthful mechanism satisfying no-deficit. We will define a threshold mechanism $\mathcal{M}'$ with profit at least as large as the profit of $\mathcal{M}$, thereby proving the theorem.

Let $\bar{\mathbf{b}}$ be the bid vector with $\bar{b}_i = h$, the highest bid, for every i. Let $\tau_i(S) = p_i(\bar{\mathbf{b}})$ for all i. Then, $\sum_i \tau_i(S) = \sum_i p_i(\bar{\mathbf{b}}) \geq c(S)$, because $\mathcal{M}$ satisfies no-deficit. Consider the threshold mechanism $\mathcal{M}'$ given by thresholds τ_i.

For any bid vector $\mathbf{b}$ with $\mathcal{M}(\mathbf{b}) = S$, we must have $\hat{\mathcal{S}}_{\mathbf{b}}(S) > 0$. Otherwise, the mechanism $\mathcal{M}''$ given by $\mathcal{M}''(\mathbf{b}) = \mathcal{B}$ if $\mathcal{M}(\mathbf{b}) = S$ and $\hat{\mathcal{S}}_{\mathbf{b}}(S) > 0$ achieves a higher profit than $\mathcal{M}$ and also satisfies the no-deficit condition. Note also, that for all $\mathbf{b}$ with $\mathcal{M}(\mathbf{b}) = S$ and all i, we have $b_i \geq p_i(\mathbf{b}) \geq p_i(\bar{\mathbf{b}}) = \tau_i(S)$. Here the second inequality follows from Lemma 5. These two conditions along with the definition of $\mathcal{M}'$ imply that $\mathcal{M}'(\mathbf{b}) = S$.

This means that for all $\mathbf{b}$ with $\mathcal{M}(\mathbf{b}) = S$, we have $\mathcal{M}'(\mathbf{b}) = S$. Furthermore, for all $\mathbf{b}$ with $\mathcal{M}(\mathbf{b}) = \emptyset$, we have $\hat{\mathcal{S}}_{\mathbf{b}}(\mathcal{M}'(\mathbf{b})) \geq 0 = \hat{\mathcal{S}}_{\mathbf{b}}(\mathcal{M}(\mathbf{b}))$. Therefore, we get $\hat{\mathcal{S}}_{\mathbf{b}}(\mathcal{M}(\mathbf{b})) \leq \hat{\mathcal{S}}_{\mathbf{b}}(\mathcal{M}'(\mathbf{b}))$, for all vectors $\mathbf{b}$. Lemma 3 now implies that $\mathcal{M}'$ has a larger expected profit than $\mathcal{M}$. □

5.1 The Hardness of Computing the Optimal Mechanism

Although threshold mechanisms are not always optimal, their simplicity is appealing and may make them practically useful. In this section we investigate the complexity of computing the optimal threshold mechanism. In particular, given bid distributions, and a cost function, we consider the decision problem of determining whether there is a threshold mechanism with total expected profit greater than some given value. Via a reduction from the knapsack problem, we show that even for a very simple input, in which every bidder has only two possible bids, and the cost function is an all-or-none function, it is NP-hard to compute the optimal threshold mechanism (which is also the optimal mechanism satisfying no-deficit in this case). See the full paper for details.

Theorem 6. *Computing the optimal no-deficit mechanism is NP-hard.*

6 Bayesian Incentive Compatible Mechanisms

We now consider relaxing ex post incentive compatibility to consider Bayesian incentive compatible (BIC) mechanisms. See the full paper for proofs of the following results.

Theorem 7. *The optimal BIC no-deficit mechanism gets the same expected profit as Myerson.*

The allocation procedure of this optimal BIC mechanism is precisely the allocation procedure of Myerson; the payment rule, however, is different. A BIC no-deficit payment rule can be derived by shifting payment from inputs where there is a deficit to ones where their is a surplus. These shifts can be done based on the joint density function, $\mathbf{F}$, so as to keep the expected payment of an agent, given their valuation, the same.

Although the proof of this theorem is constructive, it does not give a polynomial time procedure for computing the prices in general. Interestingly, when there are only two agents, there is a much simpler way of achieving optimality.

Lemma 6. *The optimal BIC no-deficit mechanism for two agents is to charge each agent the expected payment they must make conditioned on being allocated.*

Unfortunately, as the next lemma shows, this simple technique does not extend to more than two bidders.

Lemma 7. *The BIC mechanism for three or more agents that charges each agent the expected payment they must make conditioned on being allocated does not always satisfy the no-deficit constraint.*

Proof. Consider the following counter-example: there are three identical agents; each (independently) has a value of 2 with probability 0.9 and 11 with probability 0.1. The corresponding virtual valuations are 1 and 11 respectively. The costs of serving any one, any two, or all three of the agents are 10.99, 20 and 20 respectively. When all three bidders bid 11, Myerson serves all of them at a price of 2 each, incurring a deficit of 14. When two of the bidders bid 11, they are all served and each is charged a price equal to her bid. When only one bidder bids 11, the bidder is served at a price of 11. The expected payment of a bidder when bidding 11 can be computed to be 10.91. So when all the bidders bid 11, their combined expected payments are sufficient to cover the total cost of 20. On the other hand, the expected payment of a bidder on bidding 2 and losing is 0. Therefore, when the three bidders bid 11, 2, and 2, the sum of their expected payments is $10.91 < 10.99$, which is insufficient to cover the cost of serving the highest bidder. □

The counter-example in the above proof shows that other natural approaches fail as well and implies that the proof of Theorem 7 is necessarily not simple.

7 Conclusions

In this work we have explored the issue of merging the worst case no-deficit condition with the average case Bayesian optimality objective. We have found that for many interesting classes of problems it is not easy to describe the optimal solution nor is there a known algorithm for computing it. Particular questions of interest are:

1. Is there a concise description of the Bayesian optimal no-deficit ex post incentive compatible mechanism? In particular this question is interesting for submodular and general cost functions.
2. Is there a concise description of the payment rule of the Bayesian optimal no-deficit Bayesian incentive compatible mechanism? (Recall that the allocation rule is the same as Myerson's.)
3. Is there an algorithm that computes the Bayesian optimal no-deficit Bayesian incentive compatible mechanism for submodular costs? It is possible to compute the allocation so the open question is to compute the payments.
4. The BIC no-deficit mechanism constructed in our proof of Theorem 7 is only *ex interim* individually rational for the agents (i.e., they may have negative utility). This is standard for no-deficit mechanism design in economics. It is an open question as to whether there is a no-deficit mechanism that is also ex post individually rational.

Acknowledgments

We are grateful to Ilya Segal, Amitabh Sinha, and Rakesh Vohra for many helpful discussions.

References

1. A. Archer, C. Papadimitriou, K. Tawar, and E. Tardos. An Approximate Truthful Mechanism for Combinatorial Auctions with Single Parameter Agents. In *Proc. 14th Symp. on Discrete Alg.* ACM/SIAM, 2003.
2. A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science*, 2001.
3. J. Bulow and J. Roberts. The Simple Economics of Optimal Auctions. *The Journal of Political Economy*, 97:1060–90, 1989.
4. F. Cornelli. Optimal Selling Procedures with Fixed Costs. *Journal of Economic Theory*, 71:1–30, 1996.
5. J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the Cost of Multicast Transmissions. In *Proc. of 32nd Symposium Theory of Computing*, pages 218–226. ACM Press, New York, 2000.
6. A. Fiat, A. Goldberg, J. Hartline, and A. Karlin. Competitive Generalized Auctions. In *Proc. 34th ACM Symposium on the Theory of Computing.* ACM Press, New York, 2002.
7. A. V. Goldberg and J. D. Hartline. Competitiveness via Concensus. In *Proc. 14th Symp. on Discrete Alg.* ACM/SIAM, 2003.

8. A. V. Goldberg, J. D. Hartline, and A. Wright. Competitive Auctions and Digital Goods. In *Proc. 12th Symp. on Discrete Alg.*, pages 735–744. ACM/SIAM, 2001.
9. J. Hartline. Stanford cs364b: Topics in algorithmic game theory. Course Notes, 2006.
10. S. Iwata, S. Fujishige, and L. Fleischer. A Combinatorial, Strongly Polynomial-Time Algorithm for Minimizing Submodular Functions. *Journal of the ACM*, 48(4):761–777, 2001.
11. Vijay Krishna. *Auction theory.* Academic Press, San Diego, CA, USA, 2002.
12. D. Lehmann, L. I. O'Callaghan, and Y. Shoham. Truth Revelation in Approximately Efficient Combinatorial Auctions. In *Proc. of 1st ACM Conf. on E-Commerce*, pages 96–102. ACM Press, New York, 1999.
13. A. Mehta, S. Shenker, and V. Vazirani. Profit-Maximizing Multicast Pricing Via Approximate Fixed Points. In *Proc. of 4th ACM Conference on Electronic Commerce.* ACM Press, New York, 2003.
14. H. Moulin and S. Shenker. Strategyproof Sharing of Submodular Costs: Budget Balance Versus Efficiency. *Economic Theory*, 18:511–533, 2001.
15. R. Myerson. Optimal Auction Design. *Mathematics of Operations Research*, 6:58–73, 1981.
16. A. Ronen. On Approximating Optimal Auctions. In *Proc. of Third ACM Conference on Electronic Commerce.* ACM Press, New York, 2001.

Succinct Approximation of Trade-Off Curves*

Mihalis Yannakakis

Department of Computer Science
Columbia University
New York, NY 10027

When evaluating different solutions from a design space, it is often the case that more than one criteria come into play. The trade-off between the different criteria is captured by the so-called Pareto curve. The Pareto curve has typically an exponential number of points. However, it turns out that, under general conditions, there is a polynomially succinct curve that approximates the Pareto curve within any desired accuracy.

In the first part of the talk we address the question of when such an approximate Pareto curve can be computed in polynomial time. We discuss general conditions under which this is the case, and relate the multiobjective approximation to the single objective case. In the second part of the talk, we address the problem of computing efficiently a good approximation of the trade-off curve using as few solutions (points) as possible. If we are to select only a certain number of solutions, how shall we pick them so that they represent as accurately as possible the whole spectrum of possibilities? Can we find a solution that offers the best compromise between the various objectives?

(The talk is based on joint works with Christos Papadimitriou, Sergei Vassilvitskii and Ilias Diakonikolas.)

* Work partially supported by NSF Grant CCF-04-30946.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, p. 149, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Game-Theoretic Aspects of Designing Hyperlink Structures

Nicole Immorlica, Kamal Jain, and Mohammad Mahdian

Microsoft Research, One Microsoft Way, Redmond, WA
{nickle, kamalj, mahdian}@microsoft.com

Abstract. We study the problem of designing the hyperlink structure between the web pages of a web site in order to maximize the revenue generated from the traffic on the web site. We show this problem is equivalent to the well-studied setting of infinite horizon discounted Markov Decision Processes (MDPs). Thus existing results from that literature imply the existence of polynomial-time algorithms for finding the optimal hyperlink structure, as well as a linear program to describe the optimal structure. We use a similar linear program to address our problem (and, by extension all infinite horizon discounted MDPs) from the perspective of cooperative game theory: if each web page is controlled by an autonomous agent, is it possible to give the individuals and coalitions incentive to cooperate and build the optimal hyperlink design? We study this question in the settings of transferrable utility (TU) and non-transferrable utility (NTU) games. In the TU setting, we use linear programming duality to show that the core of the game is non-empty and that the optimal structure is in the core. For the NTU setting, we show that if we allow "mixed" strategies, the core of the game is non-empty, but there are examples that show that the core can be highly inefficient.

1 Introduction

As electronic commerce begins to dominate the business model of many companies, the design of an efficient and revenue-maximizing web site is of increasing importance. A major component of web site design is the selection of the hyperlink structure among the web pages. A web designer can be likened to a city planner, building hyperlink structure so as to steer traffic in a globally optimal manner. One consideration, which is of particular importance for web sites whose objective is to provide information for the users, is to facilitate the navigation through the contents of the web site. The other consideration, in particular for designing e-commerce web sites, is to present links on each page in order to direct a surfer through a path of high revenue. The latter objective is the focus of this paper.[1]

We provide a graph-theoretic model for this problem. Web pages generate varying amounts of revenue, perhaps through advertisements or product sales.

[1] As we will see in Section 5, our framework can be generalized to accommodate content-related constraints as well.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 150–161, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Additionally, web pages display hyperlinks to some other pages on the web site. Each possible hyperlink has a transition probability representing the probability that a surfer clicks on the hyperlink conditional on the other links on the page. The web designer now must select a subgraph which maximizes the expected revenue of a random walk. As we will show, the stated problem is in fact equivalent to infinite horizon discounted Markov Decision Processes (MDPs) (see [7]). Thus, the value iteration algorithm for MDPs [7] can be used to compute the optimal hyperlink structure efficiently.

In this paper, we use a linear-programming formulation for MPDs to give us insight into some game-theoretic aspects of the web design question (and of MDPs in general). Often, in large companies like Amazon or MSN, web pages are controlled by distinct (and sometimes even competing) profit centers, each responsible for their own profit and loss (P&L) account. It is not reasonable to assume that a particular profit center, or group of profit centers, will comply with the optimal web design at its own expense. Rather, it is necessary to divide the total revenue of the web site among the profit centers to ensure stability. We formulate our concern as a transferrable utility game and use insights from cooperative game theory and the LP formulation of the problem to compute an allocation scheme in the core of the game. This implies that there is always a way to divide revenue among profit centers such that the optimal web site design is stable, i.e. each group of profit centers receives a total revenue at least as large as the revenue they would be able to extract if they deviate as a coalition. We further study the non-transferrable utility game which is more suitable for situations where monetary transfer between agents managing different web pages is not possible, i.e., when each web page receives precisely the revenue it generates. We prove that in this case, if "mixed" strategies are allowed, the core is non-empty, i.e. there is a web site design where no profit center (or group of profit centers) can deviate and increase its revenue. However, the efficient web site design need not be in the core of the game, and furthermore, we show that there are examples where the revenue of the core is worse than the optimal solution by an arbitrary factor.

Our work bears some similarity to the long-standing tradition of network formation games in the economics literature (see [5] for a survey). This literature takes the standpoint that social networks play a key role in many economic settings, including labor markets [4], international free trade agreements [3], and peering and transit relations on the Internet [1]. As such, much effort has been invested in understanding economic incentives facing agents forming links in these social networks. A variety of value functions have been proposed to describe the effect of particular network structures on individuals, and our framework can be adopted to study these settings as well. However, for many of them, the computational questions remain open.

The rest of this paper is organized as follows. In Section 2, we define the model and its relationship to MDPs. In Section 3, we present a linear-programming formulation for describing the optimum (revenue-maximizing) web site design. In Section 4, we use the LP presented in Section 3 to discuss game-theoretic problems and prove that the cores of both the transferrable utility and non-transferrable utility games are non-empty. We conclude with the discussion of a multitude of interesting generalizations and open questions in Section 5.

2 Model

We model a web site as a directed graph $G = (N, E)$. Each node $i \in N$ is a web page. We denote the number of nodes by $n = |N|$. An edge ij exists from node i to node j if page i links to page j. We assume that this graph contains no self-loop, i.e., a web page does not link to itself.

A web surfer is represented by a random walk on this graph. For each page j, there is a probability p_j that the surfer starts surfing from page j. For each page i, set $S \subset N \setminus \{i\}$ of other pages, and page $j \in S$, there is a probability $p_{ij,S}$ that a surfer on page i follows a hyperlink to page j, assuming that the set of pages linked from page i is S. We assume that for all i and $S \subset N \setminus \{i\}$, $\sum_{j \in S} p_{ij,S} \leq 1 - \delta$ for some positive constant $\delta > 0$, i.e., in each step there is a non-zero probability that the surfer exits the web site.

We define a revenue for a random walk on the web site. The simplest way to do this is to assign a revenue r_j to each page j (this would correspond to the expected revenue that a surfer visiting page j would generate for the web site owner, perhaps from the advertisement on the page or by buying a product on the page), and define the expected revenue of a random walk as the sum, over all j, of r_j times the expected number of times that the random walk visits j. In this paper, we consider a more general model where the revenues are assigned to edges instead of vertices: for each hyperlink ij, there a value $r_{ij,S}$ representing the expected revenue generated for page j by a web surfer who has followed link ij when the links on page i were S. The total revenue is defined as the sum, over all edges ij in the graph, of $r_{ij,S}$ times the expected number of times the random walk traverses the edge ij. Notice that this is a strictly stronger model, since setting $r_{ij,S} = r_j$ for all i and S would be equivalent to assigning revenues to vertices (of course, we also need to add the value $\sum_j p_j r_j$ for the revenue of the first page the surfer visits). Assigning revenues to edges enables us to model situations where the conversion rate of a user depends on the web page she is coming from, and will be useful in modelling content-related constraints (as discussed in Section 5). Note that we defined the total revenue by multiplying $r_{ij,S}$'s by the expected *number of times* the random walk takes the corresponding edge, as opposed to the probability that the random walk takes this edge. This means that if the random walk visits a vertex twice, it will benefit the web site owner twice. This is a realistic assumption in many situations, e.g., where the revenue is generated from "per-impression" advertisements. For a discussion of alternative models, see Section 5.

2.1 The Case of No Externalities

The above model is strong enough to model situations where the probability that a surfer clicks on a link to page j placed on page i depends not only on i and j, but also on the set of other links on the page i. In economic terminology, this means that we can model *externalities* among the links placed on a page i. An interesting and important special case is the case of *no externalities*. In this case, each page has limited real-estate in which it can display links, and so each node i can have out-degree at most k_i (a parameter). For each $i, j \in N$, there is a probability p_{ij} that a surfer on page i follows a hyperlink to page j, if such a link exists. We assume that for all i, and for any set S of k_i pages, the sum $\sum_{j \in S} p_{ij} \leq 1 - \delta$, so these probabilities define a random walk with exit probability at least δ in each step.[2] See Section 5 for a discussion of other models where instead of (or in addition to) the limit k_i on the number of links, there is a cost associated with placing each link.

2.2 Equivalence to Markov Decision Processes

A Markov Decision Process (MDP) is a common construct used to describe scenarios with sequential decision-making processes. An MDP consists of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}(s)$ for each state $s \in \mathcal{S}$, and a revenue $r_{a,s}$ for each action/state pair.

In each iteration of an MDP, the system is in a state s, and an action $a \in \mathcal{A}(s)$ must be chosen. Actions induce a probability distribution over future states, and revenue $r_{a,s'}$ is interpreted as the revenue of taking action a given that the resulting state is s'. The goal is to chose an action for each state which maximizes the total (expected) revenue of the system over time. In infinite horizon discounted MDPs, the total revenue is calculated with respect to a discounting factor (λ), i.e. the (expected) revenue r in the t'th iteration contributes $\lambda^t r$ to the total (expected) revenue.

That the model introduced above is equivalent to infinite horizon discounted MPDs can be seen by equating the set of states $\mathcal{S}$ with the web pages N. The actions $\mathcal{A}(i)$ for a web page $i \in N$ are subsets S of other pages. By adding a "terminal" web page and links from each page to the terminal page with appropriate probabilities, we can ensure that the sum of the probabilities of the links leaving each page is precisely $1 - \delta$. Given this assumption, the induced probability distribution for taking the action S at state i of the MDP can be defined as $p_{ij,S}/(1 - \delta)$. For action $S \in \mathcal{A}(i)$, a revenue of $r_{ij,S}$ is generated given that the action resulted in future state j. The discounting factor λ is equal to $1 - \delta$.

Due to the above equivalence, one can easily adapt known algorithms for MDPs, such as the value iteration algorithm, to compute the optimal hyperlink design efficiently. Furthermore, it can be easily seen that all of the results of this paper can be applied to general infinite horizon discounted MDPs.

[2] Strictly speaking, in this model there is still externality among the links, since placing each link further limits the number of other links that can be placed on the page. However, this is the only form of externality allowed in this case.

3 Linear Programming Formulation

In this section, we present a linear program which describes the revenue-maximizing hyperlink structure. For simplicity of presentation, we describe the program in the case of no externalities.

The optimization question facing a web designer in our setting is to find a subgraph of the complete graph in which each node has degree at most k_i and the total revenue is maximized. This can be formulated as a mathematical program as follows. Let x_i be a variable representing the expected number of times a web surfer encounters node i and y_{ij} be an indicator variable for the existence of hyperlink ij. Thus, the expected number of times a web surfer traverses link ij is simply $x_i p_{ij} y_{ij}$. Relaxing the integrality constraint on y_{ij}, the problem then becomes

$$\begin{aligned} \max \quad & \sum_{i,j \in N} r_{ij} \cdot (x_i p_{ij} y_{ij}) && (1) \\ \text{s.t.} \quad & \forall\, j \in N:\ x_j \leq p_j + \sum_{i \in N} x_i p_{ij} y_{ij} && (2) \\ & \forall\, i \in N:\ \sum_{j \in N} y_{ij} \leq k_i && (3) \\ & \forall\, i,j \in N:\ 0 \leq y_{ij} \leq 1 \\ & \forall\, i \in N:\ x_i \geq 0. \end{aligned}$$

Constraint 2 encodes the "conservation of flow": the expected number of times x_j a surfer visits node j can not be more than the expected number of times p_j he starts surfing from j plus the expected number of times $\sum_{i \in N} x_i p_{ij} y_{ij}$ that he enters j from a neighboring node. Constraint 3 encodes the out-degree constraint on a node i.

This mathematical program can be transformed to a linear program by performing the change of variables $z_{ij} = x_i y_{ij}$. This gives us the program

$$\begin{aligned} \max \quad & \sum_{i,j \in N} r_{ij} p_{ij} z_{ij} && (4) \\ \text{s.t.} \quad & \forall\, j \in N:\ x_j \leq p_j + \sum_{i \in N} p_{ij} z_{ij} \\ & \forall\, i \in N:\ \sum_{j \in N} z_{ij} \leq k_i x_i \\ & \forall\, i,j \in N:\ z_{ij} \leq x_i \\ & \forall\, i \in N:\ x_i \geq 0 \\ & \forall\, i,j \in N:\ z_{ij} \geq 0 \end{aligned}$$

which is linear in the variables x_i and z_{ij}. In the next section, we show how to round an optimal fractional solution (x_i, z_{ij}) to LP 4 to a solution in which $z_{ij}/x_i \in \{0, 1\}$ for all $i, j \in N$. This shows that the above LP formulation exactly captures the hyperlink design problem, a fact that will be used in the next section to derive the game-theoretic results.

3.1 Rounding Technique

Consider an optimal fractional solution to LP 4. For all $i \in N$ such that $x_i > 0$ and all $j \in N$, define $y_{ij} = z_{ij}/x_i$. Notice if $y_{ij} \in \{0,1\}$ for all $i, j \in N$, then we can use these y_{ij} to define a feasible hyperlink structure with optimal revenue.

Otherwise, let $G = (N, E)$ be the graph where edge ij exists if $y_{ij} > 0$ and has transitional probability $p_{ij}y_{ij}$. Consider an arbitrary node $i_0 \in N$ with at least one fractional out-going edge, i.e. for at least one j, $0 < y_{i_0 j} < 1$. We "fix" this node without sacrificing any of the total revenue.

Lemma 1. *There is a graph G' with total expected revenue equal to G in which i_0 has exactly k_{i_0} integral out-links.*

Proof. In order to prove this claim, we will write the fractional out-links of i_0 in G as a convex combination of feasible integral out-links and show that one of these corresponding graphs has revenue at least that of G.

As G is an optimal fractional graph, we may assume that $\sum_j y_{i_0 j} = k_{i_0}$. Thus, the $\{y_{i_0 j}\}$ lie in the integral polytope described by $\sum_j y_{i_0 j} = k_{i_0}$ and $0 \leq y_{i_0 j} \leq 1$. Let $F_l \in \{0,1\}^{|N|}$ be the vertices of this polytope, and note that each F_l has exactly k_{i_0} non-zero coordinates. We represent the $\{y_{i_0 j}\}$ as a convex combination of these vertices $\sum_l \lambda_l F_l$ where $\sum_l \lambda_l = 1$ and $\lambda \geq 0$.

Consider the graph $G_l = (N, E_l)$ where i_0 only has links in F_l. In other words, $E_l = E - \{y_{i_0 j}\} + \{i_0 j : F_l(j) = 1\}$. Let R'_l be the expected revenue that a random walk in G_l starting at i_0 collects before returning to i_0. Furthermore, let p_l be the probability that a random walk in G_l starting at i_0 returns to i_0. Note $p_l < 1$ as there is an exit probability at each node. Thus, the total expected revenue R_l of a random walk starting from i_0 in G_l can be written as $R_l = R'_l + p_l R_l$, and so

$$R_l = \frac{R'_l}{1 - p_l}.$$

We would like to prove that for some l, the revenue R_l of G_l starting at i_0 is at least the revenue of G starting at i_0 . We can write the revenue R of G starting at i_0 in terms of R'_l as follows: by linearity of expectation, the expected revenue that a random walk in G starting at i_0 collects before returning to i_0 is simply $\sum_l \lambda_l R'_l$. Also, the probability of returning to i_0 is $\sum_l \lambda_l p_l$. Therefore, $R = \sum_l \lambda_l R'_l + \sum_l \lambda_l p_l R$, and so

$$R = \frac{\sum_l \lambda_l R'_l}{1 - \sum_l \lambda_l p_l}.$$

Using the fact that $\sum_l \lambda_l = 1$, we can rewrite R as

$$R = \frac{\sum_l \lambda_l R'_l}{\sum_l \lambda_l (1 - p_l)}$$

where we restrict the summation to the vertices F_l such that $\lambda_l > 0$. Using the fact that $(\sum_l a_l)/(\sum_l b_l) \le \max_l(a_l/b_l)$ for any two sequences of positive reals $\{a_l\}$ and $\{b_l\}$, we see for some l, the revenue of G_l starting at i_0 is at least the revenue of G starting at i_0. Note that the revenue of a random walk starting from a node $j \neq i_0$ is the same in G and G_l until it reaches i_0 as we only changed the out-going links of i_0. Therefore, we can conclude that the total revenue of G_l is at least that of G.

We can now proceed to "fix" iteratively all nodes i with fractional out-links to get an integral graph G with optimal revenue.

3.2 General Externalities Between Links

We remark that all the results of this section can be extended to the general case by using the following mathematical programming formulation. Let $y_{i,S}$ be an indicator variable for the event that page i chooses to link to pages in S. As before, x_i represents the expected number of times a surfer visits page i. By convention, we define $p_{ij,S} = 0$ for $j \notin S$.

$$\begin{aligned} \max \quad & \sum_{i,j \in N, S \subseteq N} r_{ij,S} \cdot (x_i p_{ij,S} y_{i,S}) && (5) \\ \text{s.t.} \quad & \forall\, j \in N: \ x_j \le p_j + \sum_{i \in N, S \subseteq N} x_i p_{ij,S} y_{i,S} \\ & \forall\, i \in N: \ \sum_{j \in N, S \subseteq N} y_{i,S} \le 1 \\ & \forall\, i,j \in N: \ 0 \le y_{i,S} \le 1 \\ & \forall\, i \in N: \ x_i \ge 0. \end{aligned}$$

4 The Cooperative Hyperlink Design Game

Cooperative game theory, defined by von Neumann and Mergenstern in 1944 [10], studies games in which the primitives are actions taken by coalitions of players (see [6] for background on cooperative game theory). The setting defined in Section 2 can be interpreted as a cooperative game where the nodes of the graph (i.e., the web pages) are the players. Thus, each web page is owned by a individual self-motivated agent such as a profit center within a company. This individual seeks hyperlinks that maximize his own revenue, but may cooperate with other web page owners in doing so and thereby capitalize on the induced externalities between web pages. For simplicity of presentation, we again describe our results in the case of no externalities between links, although all our results extend easily to the general case using program 5.

We consider both a transferrable and non-transferrable utility setting. In a transferrable utility setting, the value generated by a coalition may be distributed in an arbitrary manner among the members of the coalition whereas in our non-transferrable setting, each node in a coalition receives only the revenue it generates[3].

4.1 Cooperative Game with Transferrable Utility

In a transferrable utility game, the underlying assumption is that the revenue generated by a coalition may be shared among its members in any manner. A *transferrable utility* (TU) game is defined by a value function v which assigns to every possible coalition of players the value they can achieve. In our setting, the value $v(S)$ of a subset S of nodes is the value of the corresponding linear program 4 with variables restricted to the set S (i.e., the LP applied to the subgraph induced by the nodes in S). A solution of the game is a set of payoffs ξ_i, one for each player, such that $\sum_{i \in N} \xi_i = v(N)$.

We would like to define a notion which describes the stable solutions of the game. A standard such notion is that of the *core*, defined by Gillies, Shapley, and Shubik in a series of papers in the 1950s and 1960s. A solution is in the *core* of a coalitional game with transferrable utility if for all coalitions S, $\sum_{i \in S} \xi_i \geq v(S)$. Thus, the core is described by a set of linear inequalities.

Definition 1. *A set of payoffs ξ_i is in the* core *if $\sum_{i \in N} \xi_i = v(N)$ and for all $S \subset N$, $\sum_{i \in S} \xi_i \geq v(S)$.*

We prove that our game has a non-empty core. This claim can be proved using a famous theorem of Bondareva [2] and Shapley [9] which characterizes the games with non-empty cores. However, we provide a proof based on LP-duality to establish our algorithmic result for computing a solution in the core.

In order to write the dual of linear program 4, we assign variables α_j, β_i, and γ_{ij} corresponding to the first, second, and third inequality, respectively. The dual is then

$$
\begin{aligned}
\min \ & \sum_{i \in N} \alpha_i p_i \qquad\qquad (6)\\
\text{s.t.} \ & \forall\, j \in N: \ \alpha_j - k_j \beta_j - \sum_{i \in N} \gamma_{ij} \geq 0 \\
& \forall\, i,j \in N: \ -\alpha_j p_{ij} + \beta_i + \gamma_{ij} \geq r_{ij} p_{ij} \\
& \forall\, j \in N: \ \alpha_j \geq 0 \\
& \forall\, i \in N: \ \beta_i \geq 0 \\
& \forall\, i,j \in N: \ \gamma_{ij} \geq 0.
\end{aligned}
$$

We claim the payoffs $\xi_i = \alpha_i p_i$ are in the core. Clearly $\sum_{i \in N} \xi_i = \sum_{i \in N} \alpha_i p_i = v(N)$ by LP-duality. Also by LP-duality, to prove for all $S \subset N$, $\sum_{i \in S} \xi_i \geq v(S)$, we only need to show that the optimal solution $(\alpha_j, \beta_i, \gamma_{ij})$ to LP 6 is a feasible solution to LP 6 restricted to players in S. This follows easily as the inequalities of LP 6 restricted to the players in S are a subset of those in LP 6.

[3] The formal definition of non-transferrable games allows for more general payoff vectors.

We have thus proved that our game has a non-empty core, and we can find a solution in this core in polynomial time.

4.2 Cooperative Game with Non-transferrable Utility

Transferrable utility games assume that the players are able to distribute the total utility in *any* manner. In many settings, such an assumption is unreasonable. For example, in our setting, the performance of a profit center is often measured in terms of the amount of revenue it generates for the company, and there is no mechanism through which profit centers may share revenue prior to review. A non-transferrable utility game generalizes transferable utility games by studying situations such as these in which not all payoff vectors are feasible for a coalition.

A *non-transferrable utility* (NTU) game consists of a set N of players and for each coalition $S \subseteq N$ a set $\mathcal{V}(\mathcal{S}) \subset \Re^{|S|}$ of feasible payoff vectors for that coalition. The sets $\mathcal{V}(\mathcal{S})$ are assumed to satisfy some mild assumptions, namely: 1. $\mathcal{V}(\mathcal{S})$ is closed; 2. if $v \in \mathcal{V}(\mathcal{S})$, then for all $v' \in \Re^{|S|}$ with $v' \leq v$ (coordinate-wise), $v' \in \mathcal{V}(\mathcal{S})$; and 3. the set of vectors in $\mathcal{V}(\mathcal{S})$ in which each player receives at least the utility he can achieve individually is a nonempty, bounded set. Intuitively, a solution to an NTU game with payoffs $v \in \mathcal{V}(N)$ is stable (in the core) if no coalition S can withdraw and achieve a payoff vector $v' \in \mathcal{V}(S)$ such that each member of S improves his payoff. For notational convenience, we will use $v|_S$ denote the vector in $\Re^{|S|}$ whose coordinates are the coordinates of v restricted to the players in S. A vector $v \in \mathcal{V}(N)$ is in the *core* of the NTU game if there is no coalition S and vector $v' \in \mathcal{V}(S)$ such that $v' > v|_S$ (coordinate-wise). The following result of Scarf [8] states a condition under which an NTU game has a non-empty core.[4] Let λ_S be a *fractional partition* λ_S of players, i.e., a set of coefficients $0 \leq \lambda_S \leq 1$ of subsets of N such that for all players i, $\sum_{S:i\in S} \lambda_S = 1$. An NTU game is called *balanced* if, for every fractional partition λ_S, a vector $v \in \Re^{|N|}$ must be in $\mathcal{V}(N)$ if $v|_S \in \mathcal{V}(S)$ for all S with $\lambda_S > 0$.

Theorem 1. *(Scarf) A cooperative game with non-transferrable utility has a non-empty core if it is balanced.*

In our setting, the set $\mathcal{V}(S)$ consists of the payoff vectors v where v_i is (at most) the revenue of i in some hyperlink structure on S. More formally, $v \in \mathcal{V}(S)$ if and only if there is a (fractional) graph G on nodes S such that for each player $i \in S$, v_i is at most the expected revenue of i in G. Alternatively, we can state this condition using program 1: $v \in \mathcal{V}(S)$ if and only if there is a feasible solution (x_i, y_{ij}) to program 1 such that for each player $i \in S$, v_i is at most $\sum_j r_{ji} \cdot (x_j p_{ji} y_{ji})$ (the expected revenue of i). These sets $\mathcal{V}(S)$ clearly satisfy the assumptions stated above, and so our game is an NTU game. Here, we use Scarf's theorem to prove the following statement.

Theorem 2. *There is a* fractional *graph in the core of the web site game.*

[4] This is a generalization of the result of Bondareva [2] and Shapley [9] which states a condition under which a TU game has a non-empty core.

Proof. Consider any fractional partition λ_S and payoff vectors $v(S) \in \mathcal{V}(S)$. Let v be the vector whose i'th coordinate is the minimum over all S containing i of $v(S)|_i$. We prove that there is a fraction graph G whose corresponding payoff vector v' satisfies $v' \geq v$ (coordinate-wise) and thus the game is balanced. Our theorem then follows from Scarf's theorem.

Let $(x_{i,S}, z_{ij,S})$ be a feasible solution to LP 4 on set S such that $v(S) \leq \sum_{j \in S} r_{ji} p_{ji} z_{ji,S}$. Consider the solution (x_i, z_{ij}) to LP 4 where $x_i = \sum_{S:i \in S} \lambda_S x_{i,S}$ and $z_{ij} = \sum_{S:i,j \in S} \lambda_S z_{ij,S}$. As (x_i, z_{ij}) is a convex combination of feasible solutions, it is also feasible and thus the corresponding graph G is a feasible fractional graph. Furthermore, the revenue $v'|_i$ of a node i in G is at least $\sum_{j \in N} r_{ji} p_{ji} z_{ji} = \sum_{S:i \in S} \lambda_S \sum_{j \in S} r_{ji} p_{ji} z_{ji,S} \geq \min_S v(S)|_i \geq v|_i$.

Fractional graphs can be thought of as the result of mixed strategies in link selection. In other words, if we allow a node i to have fractional out-links of total weight at most k_i (or probabilistically select k_i links according to their fractional weight), then the core is non-empty. See section 5 for a discussion regarding computational issues and "pure" strategies.

We end with a comment regarding the efficiency of the graphs in the core. Whereas the efficient (that is, revenue-maximizing) graph is in the TU core, this may not be the case for the NTU core. In fact, the solutions in the NTU core may be arbitrarily inefficient. As an example, consider the game on three nodes a, b, and c. Suppose $p_a = 1$ so a surfer always enters the site at node a. The revenue of any link entering node a is 1, b is 1, and c is R for an arbitrarily large R. Each node is allowed (fractionally) one out-link. The transition probabilities are $p_{ab} = 1/2$, $p_{ba} = 1/2$, $p_{ac} = 1/2$, and all other transition probabilities are 0. It is easy to check that the only solutions in the NTU core of this game include integrally the set of links $\{ab, ba\}$. However, the revenue of any such graph is constant while the efficient graph $\{ba, ac\}$ has arbitrarily large revenue.

5 Discussion

Our model and results are quite general and can be accommodated to handle a large number of scenarios. We discuss some of them here, and mention a few open questions.

Content-related restrictions. In many web sites, the link structure might be subject to certain content-related restrictions. For example, perhaps MSNBC is required to link to MSN search regardless of the transition probability. Our setting is general enough to handle a wide variety of such restrictions by appropriately setting certain $p_{ij,S}$ to zero. In the above example, if a link ij is required to appear, we can set the probability $p_{i'j',S}$ to zero for all i', j', and S where $j \notin S$. Similarly, if a link ij is forbidden from appearing, we can force our solution to obey this restriction by setting the probability $p_{ij,S}$ of the link to zero for all S.

Costly links. In our model, the optimal hyperlink structure of a web site depends on the transition probabilities of a links which in turn depend on the set of other links on the page. In addition, one could imagine a model in which each link ij incurs an associated cost c_{ij}. This situation can easily be handled in all our results by appropriately adjusting the maximization objective.

Location-dependent probabilities. We can model situations where page i has k_i "slots" for placing links to other pages, and the probability that a link is clicked also depends on the slot in which the link is placed.

Accounting for revisits. Our model thus far ignored the history of a surfer in defining the transition probabilities. However, in some settings it is reasonable to assume that a surfer is less likely to return to a page he has already visited, especially in the recent past. The limiting case, when a surfer never returns to a page he has already visited, is NP-hard as can be seen easily by a reduction from the longest-path problem. Approximating this instance remains open, as does the computability of very interesting special cases of limited memory or simple history-dependent probability structures. In another model, we could assume that the transition probabilities remain the same regardless of the history, but the revenue structure changes, i.e., a surfer does not incur any extra revenue the second time he visits a page.

Handling different demographics. It is commonly acknowledged that different demographics have different surfing and purchasing patterns. A 21 year-old computer scientist from Seattle is more likely to navigate to the automotive section of Amazon.com than a 10 year-old school-girl from Wichita and is more likely to more spend money there than a 46 year-old farmer from Boise. One way to optimize a web site given such information is to dynamically update the link structure for each demographic, and indeed some of the larger web sites are starting to take this approach with a subset of their links. If dynamic links are an option for a web site, our results apply trivially by solving the problem separately for each demographic. However, static link structures are still the most prevalent style, and computing an optimal static link structure given demographic data and associated probabilities and revenues remains an interesting open problem.

The NTU game. As the proof of Scarf's theorem uses an exponential-time algorithm (or, alternatively, fixed-point theorems), our result regarding existence of the core in the NTU game is non-constructive and we do not know how to find a fractional graph in the core in polynomial time. Furthermore, we do not know how to find an *integral* graph in the core or even prove that one always exists (although we have not been able to find a counter-example). It might be possible to prove existence of (and perhaps even compute) an integral graph in the core using potential proofs similar to those for proving existence of pure Nash equilibria in non-cooperative games.

The PageRank objective. One of the most commonly used systems for sorting web pages in search engine results is PageRank. As search engines are the single most essential portal to the web for most surfers, "search engine optimization" (SEO, as the industry calls it) of a web site is crucial to its success, so crucial that the commodity of PageRank sells for nearly \$100 on eBay[5]. Informally, the PageRank of a web page is defined as the probability of that page in the stationary distribution of a random walk on the web. Although the internal hyperlink structure of a web site does not affect it's average (over all pages) PageRank, it does affect the maximum: to maximize the PageRank over all pages of a page in the web site, all pages should link to the page with highest entrance probability. This structure is trivial and unlikely to work given the search engine industry's spam detection efforts. However, one could try to maximize this objective with certain restrictions on the hyperlink structure that attempt to avoid detection like maximum in-degree. The NTU game also poses an interesting question.

Acknowledgments. We would like to thank Gary Flake for asking the question that motivated this work. We have benefitted from discussions with Christian Borgs, Jennifer Chayes, Uri Feige, and Gary Flake.

References

1. N. Badasyan and S. Chakrabarti. Private peering among internet backbone providers. mimeo: Virginia Tech., 2003.
2. O. N. Bondareva. Some applications of linear programming to cooperative games. *Problemy Kibernetiki*, 10:119–139, 1963.
3. T. Furusawa and H. Konishi. Free trade networks. mimeo: Yokohama National University and Boston College, 2002.
4. M. Granovetter. *Getting a job: a study of contacts and careers.* University of Chicago Press, 2 edition, 1995.
5. M. O. Jackson. A survey of models of network formation: Stability and efficiency. In G. Demange and M. Wooders, editors, *Group Formation in Economics; Networks, Clubs and Coalitions*, chapter 1, pages 11–57. Cambridge University Press, 2004.
6. M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.
7. M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley-Interscience, 1994.
8. H. E. Scarf. The core of an n-person game. *Econometrica*, 35:50–69, 1967.
9. L. S. Shapley. On balanced sets and cores. *Naval Research Logistics Quarterly*, 14:453–460, 1967.
10. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*, chapter 13, pages 257–275. John Wiley and Sons, 1944.

[5] Merchants selling PageRank are purportedly themselves high-ranked pages and are selling the placement of a text-link on their page with the sole intent of boosting the linked-to page's rank.

Competing for Customers in a Social Network: The Quasi-linear Case

Pradeep Dubey[1,2], Rahul Garg[3], and Bernard De Meyer[4,2]

[1] Center for Game Theory, Stony Brook University, USA
[2] Cowles Foundation, Yale University, USA
[3] IBM India Research Lab, New Delhi, India
[4] Cermsem, Univesité Paris 1, Paris, France

Abstract. There are many situations in which a customer's proclivity to buy the product of any firm depends not only on the classical attributes of the product such as its price and quality, but also on who else is buying the same product. We model these situations as games in which firms compete for customers located in a "social network". Nash Equilibrium (NE) in pure strategies exist and are unique. Indeed there are closed-form formulae for the NE in terms of the exogenous parameters of the model, which enables us to compute NE in polynomial time.

An important structural feature of NE is that, if there are no a priori biases between customers and firms, then there is a cut-off level above which high cost firms are blockaded at an NE, while the rest compete *uniformly* throughout the network.

We finally explore the relation between the connectivity of a customer and the money firms spend on him. This relation becomes particularly transparent when externalities are dominant: NE can be characterized in terms of the invariant measures on the recurrent classes of the Markov chain underlying the social network.

1 Introduction

Consider a situation in which firms compete for customers located in a "social network". Any customer i has, of course, a higher proclivity to buy from firm α, if α lowers its price relative to those quoted by its rivals. But another, quite independent, consideration also influences i's decision. He is keen to conform to his neighbors in the network. If the bulk of them purchase firm β's product, then he is tempted to do likewise, even though β may be charging a higher price than α. Customer i's behavior thus involves a delicate balance between the "externality" exerted by his neighbors and the more classical constituents of demand — the price and the intrinsic quality of the product itself. Such externalities arise naturally in several contexts (see, e.g., [1],[5],[6],[3],[8],[7]).

The externality in demand clearly has significant impact on the strategic interaction between the firms. Firm α may spend resources marketing its product to i, not because α cares about i per se as a client, but because i enjoys the position of a "hub" in the social network and so wields influence on other potential clients that are of value to α. This in turn might instigate rival firms to

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 162–173, 2006.
© Springer-Verlag Berlin Heidelberg 2006

spend further on i, since they wish to wean i away from an excessive tilt toward α; causing α to increase its outlay on i even more, unleashing yet another round of incremental expenditures on i.

The scenario invites us to model it as a non-cooperative game between the firms[1]. We take our cue from [1],[5] which explore the optimal marketing strategy of a *single* firm, based on the "network value" of the customers. Our innovation is to introduce competition between *several* firms in this setting. The model we present is more general than that of [1],[5], though inspired by it. As in [1],[5], the social network, specifying the field of influence of each customer, is taken to be exogenous. Rival firms choose how much money to spend on each customer. For any profile of firms' strategies, we show that the externality effect stabilizes over the social network and leads to unambiguous customer-purchases. A particular instance of our game arises when firms compete for advertisement space on different web-pages in the Internet (see Section 2.1).

Our main interest is in understanding the structure of the Nash Equilibria (NE) of the game between the firms. Will they end up as regional monopolies, operating in separate parts of the network? Or will they compete fiercely throughout? Which firms will enter the fray, and which will be blockaded? And how will the money spent on a customer depend on his connectivity in the social network?

For ease of presentation, the focus of this paper is on the quasi-linear[2] case (which includes the model in [1], by setting # firms = 1). We show that NE are unique and can be computed in polynomial time via closed-form expressions involving matrix inverses. It turns out that, provided that there are no a priori biases between firms and customers, any NE has a cut-off cost: all firms whose costs are above the cut-off are blockaded, and the rest enter the fray. Moreover there is no "regionalization" of firms in an NE: each active firm spends money on *every* customer-node of the social network. The money spent on node i is related to the connectivity of i, but the relation is somewhat subtle, though expressible in precise algebraic form. When externalities are dominant, however, this relation becomes more transparent: NE can be characterized in terms of the invariant measures on the recurrent classes of the Markov chain underlying the social network (see Section 4). In particular suppose that the graph representing the social network is undirected and connected, all the neighbors of any customer-node exert equal influence on him, and each company values all the nodes equally. Then, at the NE, the money spent by a company on a node is proportional to the degree of the node.

2 The Model

There is a finite set $\mathcal{A}$ of firms and $\mathcal{I}$ of customers. We shall define a strategic game Γ among the firms. The customers themselves are non-strategic in our model and described in behavioristic terms.

---

[1] Customers are not strategic in our model. As in [1],[5], they are described in behavioristic terms.

[2] For generalizations to the non-linear case, see Section 4.

Firm $\alpha \in \mathcal{A}$ can spend m_i^α dollars on customer $i \in \mathcal{I}$ by way of marketing its product to him. This could represent the discounts or special warranties offered by α to i (in effect lowering, for i, the fixed price that α has quoted for its product), or free add-ons of supplementary products, or simply the money spent on advertising to i, etc. The strategy set of firm α may thus be viewed as[3] $R_+^{\mathcal{I}}$, with elements $m^\alpha \equiv (m_i^\alpha)_{i \in \mathcal{I}}$.

Consider a profile of firms' strategies $m \equiv (m^\alpha)_{\alpha \in \mathcal{A}} \in R_+^{\mathcal{I} \times \mathcal{A}}$. The *proclivity* of customer i to buy from any particular firm α clearly depends on the profile m, i.e., not just the expenditure of α but also that of its rivals. We denote this proclivity by $p_i^\alpha(m)$. One can think of $p_i^\alpha(m)$ as the quantity of α's product purchased by i. Or, interpreting i to be a mass of customers such as those who visit a web page i, one can think of $p_i^\alpha(m)$ as the fraction of mass i that goes to α (or, equivalently, as the probability of i going to α). In either setting, we take $p_i(m) \equiv (p_i^\alpha(m))_{\alpha \in \mathcal{A}} \in [0,1]^{\mathcal{A}}$. (When $p_i^\alpha(m)$ is a quantity, there is a physical upper bound on customer i's capacity to consume which, w.l.o.g., is normalized to be 1).

The benefit to any particular firm α from its *clientele* $p^\alpha(m) \equiv (p_i^\alpha(m))_{i \in \mathcal{I}}$ is $\sum_{i \in \mathcal{I}} u_i^\alpha p_i^\alpha(m)$ and the cost of its expenditures m^α is $\sum_{i \in \mathcal{I}} c_i^\alpha m_i^\alpha$.

Thus α's *payoff* in the game is given by

$$\Pi^\alpha(m) = \sum_{i \in \mathcal{I}} u_i^\alpha p_i^\alpha(m) - \sum_{i \in \mathcal{I}} c_i^\alpha m_i^\alpha$$

It remains to define the map from m to $p(m)$.

Customer i's proclivity p_i^α to purchase from firm α is clearly positively correlated with α's expenditure m_i^α on i, and negatively correlated with the expenditures $m_i^{-\alpha} \equiv (m_i^\beta)_{\beta \in \mathcal{I} \setminus \{\alpha\}}$, of α's rivals.

In addition we suppose that there is a positive externality exerted on i by the choice of any neighbor j: increases in p_j^α may boost p_i^α. Negative cross-effects of p_j^β on p_i^α, for $\beta \neq \alpha$, can be incorporated under certain assumptions (which we make precise in [2]), but here we suppose that they are absent.

By way of an example of such an externality, think of firms' products as specialized software. Then if the users with whom i frequently interfaces (i.e., i's "neighbors") have opted for α's software, it will suit i to also purchase predominantly from α in order to more smoothly interact with them. Or else suppose the firms are in an industry focused on some fashion product. Denote by i's neighbors the members of i's peer group with whom i is eager to conform. Once again, p_i^α is positively correlated with p_j^α where j is a neighbor of i. Another typical instance comes from telephony: if most of the people, who i calls, subscribe to service provider α and if α-to-α calls have superior connectivity compared with α-to-β calls, then i may have incentive to subscribe to α even if α is costlier than β.

To define the map from m to $p(m)$, we must turn to the social network. It is represented by a directed, weighted graph $G = (\mathcal{I}, E, w)$. The nodes of G are

[3] Budget constraints on expenditures can be incorporated via cost functions (see Section 4).

identified with the set of customers $\mathcal{I}$. Each directed edge $(i, j) \in E \equiv \mathcal{I} \times \mathcal{I}$ has weights $(w_{ij}^{\alpha})_{\alpha \in \mathcal{A}}$, where $w_{ij}^{\alpha} \geq 0$ is a measure of the influence j has on i, with regard to purchases from α. Precisely, if $p^{\alpha} = (p_j^{\alpha})_{j \in \mathcal{I}}$ denotes the proclivities of purchases, then the externality impact of p^{α} on i is $\sum_{j \in \mathcal{I}} w_{ij}^{\alpha} p_j^{\alpha}$. We assume that $\sum_{j \in \mathcal{I}} w_{ij}^{\alpha} \leq 1$, for all $i \in \mathcal{I}$ and $\alpha \in \mathcal{A}$. (One may view $(\mathcal{I}, E^{\alpha}, w^{\alpha})$ as the social network relevant for firm α, with $E^{\alpha} = \{(i, j) \in E : w_{ij}^{\alpha} > 0\}$).

Let us now make explicit how firms' expenditures, in conjunction with the externality effect, determine purchases in the social network.

Fix a profile $m \equiv (m^{\beta})_{\beta \in \mathcal{A}} \equiv ((m_j^{\beta})_{j \in \mathcal{I}})_{\beta \in \mathcal{A}}$ of firms' strategies.

For any firm α and customer i, let $\gamma_i^{\alpha}(m_i) \in [0, 1]$ denote the proclivity with which i is *initially* impelled to buy from firm α on account of the direct "marketing impact", where (recall) $m_i \equiv (m_i^{\beta})_{\beta \in \mathcal{A}}$ gives the expenditures induced on i by m.

Denoting $(m_i^{\beta})_{\beta \in \mathcal{A} \setminus \{\alpha\}}$ by $m_i^{-\alpha}$, it stands to reason that the impact $\gamma_i^{\alpha}(m_i^{\alpha}, m_i^{-\alpha})$ be strictly increasing in m_i^{α} for any fixed $m_i^{-\alpha}$. We assume this and a little bit more: γ_i^{α} is also concave in m_i^{α} for fixed $m_i^{-\alpha}$, reflecting the diminishing returns to α of incremental dollars spent on i.

A canonical example we have in mind is $\gamma_i^{\alpha}(m_i) = m_i^{\alpha}/\overline{m}_i$ where $\overline{m}_i \equiv (\sum_{\beta \in \mathcal{I}} m_i^{\beta})$ (with $\gamma_i^{\alpha}(0) \equiv 0$). In short, i's probability of purchase from different firms is simply set proportional to the money they spend on him[4].

Customer i weights the two factors (i.e., the externality impact and the marketing impact) by θ_i^{α} and $1 - \theta_i^{\alpha}$, where $0 \leq \theta_i^{\alpha} < 1$. Thus, given a strategy profile m, the final steady-state proclivities of purchase $p(m) \equiv (p^{\alpha}(m))_{\alpha \in \mathcal{A}} \in [0, 1]^{\mathcal{I} \times \mathcal{A}}$, where $p^{\alpha} \equiv (p_j^{\alpha}(m))_{j \in \mathcal{I}}$, must satisfy.

$$p_i^{\alpha}(m) = (1 - \theta_i^{\alpha})\gamma_i^{\alpha}(m_i) + \theta_i^{\alpha} \sum_{j \in \mathcal{I}} w_{ij}^{\alpha} p_j^{\alpha}(m) \tag{1}$$

for all $\alpha \in \mathcal{A}$ and $i \in \mathcal{I}$.

Define the $|\mathcal{I}| \times |\mathcal{I}|$-matrices: $I \equiv$ identity, $\Theta^{\alpha} \equiv$ the diagonal matrix with $\Theta_{ii}^{\alpha} = \theta_i^{\alpha}$ and $W^{\alpha} \equiv$ the matrix with entries w_{ij}^{α}. Then equation (1) reads

$$p^{\alpha}(m) = (I - \Theta^{\alpha})\gamma^{\alpha}(m) + \Theta^{\alpha} W^{\alpha} p^{\alpha}(m).$$

Since $I - \Theta^{\alpha} W^{\alpha}$ is invertible (its row sums being less than 1), we obtain

$$p^{\alpha}(m) = (I - \Theta^{\alpha} W^{\alpha})^{-1}(I - \Theta^{\alpha})\gamma^{\alpha}(m).$$

This gives

$$\Pi^{\alpha}(m) = [u^{\alpha}]^{\top}(I - \Theta^{\alpha} W^{\alpha})^{-1}(I - \Theta^{\alpha})\gamma^{\alpha}(m) - [c^{\alpha}]^{\top} m^{\alpha} \tag{2}$$

[4] More generally, $\gamma_i^{\alpha}(m_i) = (m_i^{\alpha}/\overline{m}_i)(\overline{m}_i)^r$ where $0 \leq r < 1$. We may think of $(\overline{m}_i)^r$ as the "market penetration", which rises with the total money spent. (If $\gamma_i^{\alpha}(m_i)$ is to be a probability, one must amend $(\overline{m}_i)^r$ to $\max\{(\overline{m}_i^r), 1\}$ or a suitably smoothed version of this function.)

where $u^\alpha \equiv (u_j^\alpha)_{j\in\mathcal{I}} \in R_+^{\mathcal{I}}$ and $c^\alpha \equiv (c_j^\alpha)_{j\in\mathcal{I}} \in R_{++}^{\mathcal{I}}$ are column vectors and $\top$ stands for the transpose operation. Denote

$$v^\alpha \equiv [u^\alpha]^\top (I - \Theta^\alpha W^\alpha)^{-1}(I - \Theta^\alpha) \tag{3}$$

Then (2) may be rewritten:

$$\Pi^\alpha(m) = \sum_{i\in\mathcal{I}} (v_i^\alpha \gamma_i^\alpha(m_i) - c_i^\alpha m_i^\alpha) \tag{4}$$

Our key assumption on $\gamma_i^\alpha(m_i)$ is that it depends only on the variables m_i^α and $\overline{m}_i^{-\alpha} \equiv \sum_{\beta\in\mathcal{A}\setminus\{\alpha\}} m_i^\beta$, i.e., firm α is affected only by the aggregate[5] expenditure of its rivals.

Assume $\gamma_i^\alpha(m_i^\alpha, \overline{m}_i^{-\alpha})$ is continuous; and, furthermore, increasing and differentiable w.r.t. m_i^α whenever $\overline{m}_i \equiv \sum_{\beta\in\mathcal{A}} m_i^\beta = m_i^\alpha + \overline{m}_i^{-\alpha} > 0$. Let

$$\phi_i^\alpha(m_i^\alpha, \overline{m}_i^{-\alpha}) \equiv \frac{\partial}{\partial m_i^\alpha} \gamma_i^\alpha(m_i^\alpha, \overline{m}_i^{-\alpha})$$

and next define

$$\lambda_i^\alpha(r_i^\alpha, \overline{m}_i) \equiv \phi_i^\alpha(r_i^\alpha \overline{m}_i, (1 - r_i^\alpha)\overline{m}_i)$$

(Thus $r_i^\alpha \equiv m_i^\alpha / \overline{m}_i$.) We suppose that

$$\lambda_i^\alpha \text{ is strictly decreasing in } r_i^\alpha \text{ and in } \overline{m}_i \tag{5}$$

for fixed $\overline{m}_i$ and r_i^α respectively. This condition reflects the diminishing returns on incremental dollars spent by α; it also states that an incremental dollar of α counts for less when α's rivals have put in more money.

We also assume that

$$\lim_{\delta\to 0} \frac{\gamma_i^\alpha(\delta, 0)}{\delta} = \infty. \tag{6}$$

Note that both conditions (5) and (6) are satisfied by our canonical example and its variants in footnote 4.

Finally we assume that for each customer there exist at least two firms that value him:

$$\forall i \in \mathcal{I}, \exists \alpha, \alpha' \in \mathcal{A} \text{ such that } : \alpha \neq \alpha' \text{ and } u_i^\alpha > 0 \text{ and } u_i^{\alpha'} > 0. \tag{7}$$

This will create enough competition in an NE to ensure that positive money is bid on each client, enabling us to steer clear of possible discontinuity[6] of γ_i^α at 0.

[5] Aggregation is a form of anonymity that is common to many markets. It says, in essence, that if a firm pretends to be two entities and splits its expenditure between them, this has no effect on *other* firms. This form of "anonymity toward numbers" is tantamount to aggregation.

[6] As occurs in our canonical example.

2.1 An Example: Competition for Advertisement on the Web

Think of the web as a set $\mathcal{I}$ of pages, each of which corresponds to a distinct node of a graph. A directed arc (i,j) means that there is a link from page j to page i.

At the beginning of any period, two kind of "surfers" visit page i. There are those who transit to i from other pages j in the web. Furthermore, there are "fresh arrivals", entering the web for the first time, via page i at rate ψ_i.

At the end of the period, a fraction $(1-\theta_i)$ of the population on the page i exits the web, while the remaining fraction θ_i continues surfing (where $0 \leq \theta_i < 1$). The weight on (i,j), which we denote ω_{ij}, gives the probability that a representative surfer, who is on page j and who continues surfing, moves on to page i (or, alternatively, the fraction of surfers on page j who transit to page i). Thus $\sum_{i\in\mathcal{I}} \omega_{ij} = 1$ for all $j \in \mathcal{I}$.

Companies $\alpha \in \mathcal{A}$ compete for advertisement on the web pages. If they spend $m_i \equiv (m_i^\alpha)_{\alpha\in\mathcal{A}}$ dollars to place their ads on page i, they get "visibility" (time, space) on page i in proportion to the money spent. Thus the probability that a surfer views company α's ad on page i is $m_i^\alpha/\overline{m}_i = \gamma_i^\alpha(m_i^\alpha, \overline{m}_i)$.

The payoff of a company is the aggregate "eyeballs" of its advertisement obtained, in the long run (i.e., in the steady state).

To compute the payoff, let us first examine the population distribution of surfers across nodes in the unique steady state of the system.

Denote by ϕ_i denote the arrival rate of surfers (of both kinds) to page i. Then, in a steady state, we must have

$$\phi_i = \psi_i + \sum_{j\in\mathcal{I}} \omega_{ij}\theta_j\phi_j$$

for all $i \in \mathcal{I}$. In matrix notation, this is

$$\phi = \psi + \Omega\Theta\phi$$

where $\phi \equiv (\phi_i)_{i\in\mathcal{I}}$ and $\psi \equiv (\psi_i)_{i\in\mathcal{I}}$ are column vectors, Θ is the diagonal $\mathcal{I}\times\mathcal{I}$ matrix with entries $\theta_{ii} = \theta_i$, and Ω is the $\mathcal{I}\times\mathcal{I}$ matrix with entries ω_{ij}. Hence

$$\phi = (I - \Omega\Theta)^{-1}\psi$$

The total eyeballs (per period) obtained by company α is then

$$\sum_{i\in\mathcal{I}} \phi_i\gamma_i^\alpha(m)$$

which fits the format of (4).

More generally, suppose surfers have bounded recall of length k. Then firm α will only care about any surfer's eyeballs in the last k periods prior to the surfer's exit. When $k=1$, α's payoff is

$$\sum_{i\in\mathcal{I}} (1-\theta_i)\phi_i\gamma_i^\alpha(m)$$

The expression for v_i^α will become complicated when the recall $k > 1$ (more so, if discounting of past memory is incorporated). But the payoffs in all these cases still fit the format of (4).

Generalizing in a different direction, suppose that surfers at page i, who have spent t periods in the web, exit at rate θ_i^t for $t = 1, 2 \ldots$. Denote by Θ^t the diagonal matrix whose ii^{th} entry is θ_i^t. Then $\phi = (I + \Omega\Theta^1 + \Omega\Theta^2\Omega\Theta^1 + \ldots)\psi$, which is well-defined provided we assume $\theta_i^t \leq \Delta < 1$ for some Δ (for all t, i). This retains the format of (4) though the expression for v_i^α becomes even more complicated. One could also incorporate bounded recall in this setting, without departing from (4).

Notice that the "externality" in the above examples is reflected in the movement of traffic across pages in the web. Also notice that the games derived are *anonymous* i.e. $v_i^\alpha = v_i$ for all α. Such games will be singled out for special attention later.

2.2 Uniqueness of Nash Equilibrium

Recall that a strategy profile m is called a **Nash Equilibrium**[7] (NE) of the game Γ if

$$\Pi^\alpha(m) \geq \Pi^\alpha(\tilde{m}^\alpha, m^{-\alpha}) \ \ \forall\, \tilde{m}^\alpha \in R_+^{\mathcal{I}}$$

for all $\alpha \in \mathcal{A}$ (where $m^{-\alpha} \equiv (m^\beta)_{\beta \in \mathcal{I}\setminus\{\alpha\}}$).

Theorem 1. *Under hypotheses (5), (6), (7), there exists a unique Nash Equilibrium in the quasi-linear model.*

Proof: See [2].

2.3 Characterization of Nash Equilibrium

Theorem 2. *Consider our canonical case: $\gamma_i^\alpha(m_i) = m_i^\alpha/\overline{m}_i$ (other closed-form expressions for the γ_i^α will lead to analogous characterizations). Fix customer i and rank all the firms in $\mathcal{A} \equiv \{1, 2, \ldots, n\}$ in order of increasing $\kappa_i^\alpha \equiv c_i^\alpha/v_i^\alpha$ (see (3) for the definition of v_i^α). For convenience denote this order $\kappa_i^1 \leq \kappa_i^2 \leq \ldots \leq \kappa_i^n$. Let*

$$k_i = \max\left\{ l \in \{2, \ldots, n\} : (l-2)\kappa_i^l < \sum_{\alpha=1}^{l-1} \kappa_i^\alpha \right\} \tag{8}$$

In the unique NE, firms $1, \ldots, k_i$ will spend money on customer i as follows:

$$m_i^\alpha = \left(\frac{k_i - 1}{\sum_{\beta=1}^{k_i} \kappa_i^\beta}\right)\left(1 - \frac{(k_i-1)\kappa_i^\alpha}{\sum_{\beta=1}^{k_i} \kappa_i^\beta}\right) \tag{9}$$

Firms $k_i + 1, \ldots, n$ put no money on customer i.

[7] Throughout we confine attention to "pure" strategies.

Proof: See [2].

According to Theorem 2, companies α can be ranked, at each customer-node i, according to their "effective costs" κ_i^α. The money m_i^α, spent by α on i, is a strictly decreasing function of κ_i^α upto some threshold, after which it becomes zero.

Theorem 2 confirms the obvious intuition that $m_i^\alpha = 0$ if $v_i^\alpha = 0$ (i.e., $\kappa_i^\alpha = \infty$, recalling that $c_i^\alpha > 0$ by assumption). It also brings to light a different, and more important, feature of NE. First recall that, by (3), v_i^α may well be highly positive even though the direct value u_i^α of customer i to company α is zero. This is because v_i^α incorporates the *network value* of i, stemming from the possibility that i may be exerting a big externality on other customers whom α does directly value. Now, since κ_i^α falls with v_i^α, (9) reveals that α may be spending a huge m_i^α on i even when u_i^α is zero, *purely* on account of the network value of i.

2.4 Impact of the Social Network on Nash Equilibrium

To get a better feel for Theorem 2, it might help to consider some examples.

Suppose there are five customers $\{1, 2, \dots, 5\}$ and four firms $\{\alpha_1, \alpha_2, \beta_1, \beta_2\}$. The customers are arranged in a linear network, with i connected to $i+1$ via an undirected (i.e., directed both ways) edge, for $i = 1, 2, 3, 4$. Suppose each node is equally influenced by its neighbors in the purchase of any firm's product. Thus $(w_{11}^\gamma, w_{12}^\gamma, w_{13}^\gamma, w_{14}^\gamma, w_{15}^\gamma) = (0,\ 1,\ 0,\ 0,\ 0)$, $(w_{21}^\gamma, w_{22}^\gamma, w_{23}^\gamma, w_{24}^\gamma, w_{25}^\gamma) = (0.5,\ 0,\ 0.5,\ 0,\ 0)$ etc., for any company γ. Further suppose $\theta_i^\gamma = 0.1$ and $c_i^\gamma = 1$ for all γ and i. Finally let $u^{\alpha_1} = u^{\alpha_2} = (1,\ 1,\ 0,\ 0.1,\ 0.1)$ and $u^{\beta_1} = u^{\beta_2} = (0.1,\ 0.1,\ 0,\ 1,\ 1)$. Formula (3) yields $v^{\alpha_1} = v^{\alpha_2} = (0.950,\ 0.998,\ 0.055,\ 0.102,\ 0.095)$ and $v^{\beta_1} = v^{\beta_2} = (0.095,\ 0.102,\ 0.055,\ 0.998,\ 0.950)$ and hence $\kappa^{\alpha_1} = \kappa^{\alpha_2} = (1.053,\ 1.002,\ 18.182,\ 9.779,\ 10.514)$ and $\kappa^{\beta_1} = \kappa^{\beta_2} = (10.514,\ 9.779,\ 18.182,\ 1.002,\ 1.053)$. It follows from Theorem 2 that firms α_1 and α_2 will put no money on customers $4, 5$ and positive money on the rest; while firms β_1 and β_2 will put no money on customers $1, 2$ and positive money on the rest. In effect, there will "regionalization" of customers into α-territory $\{1, 2, 3\}$ and β-territory $\{3, 4, 5\}$. The only overlap is customer 3, who is of zero direct value u_3^γ to all firms γ and yet is being equally targeted by them, purely on account of his network value.

The situation dramatically changes when the game is anonymous i.e., $v_i^\alpha = v_i$ and $c_i^\alpha = c^\alpha$ for all α and i. (The first identity holds in particular — see (3) — when $w_{ij}^\alpha = w_{ij}$, $\theta_i^\alpha = \theta_i$, and $u_i^\alpha = u_i$, for all α, i and j, i.e., there are no a priori biases between firms and customers.) Our analysis in Section 2.3 immediately implies that we can rank the firms, independently of i, by their costs; say (after relabeling)

$$c^1 \le c^2 \le \dots \le c^n$$

At the Nash Equilibrium a subset of low-cost firms $\{1, \dots, k\}$ will be active (see (8), while all the higher-cost firms $\{k+1, \dots, n\}$ will be blockaded, where

$$k = \max \left\{ l \in \{2, \dots, n\} : (l-2)c^l < \sum_{\beta=1}^{l-1} c^\beta \right\}$$

Each active firm $\alpha \in \{1, \ldots, k\}$ will spend an amount $m_i^\alpha > 0$ on all the nodes $i \in \mathcal{I}$ that is proportional to v_i. Indeed, by (9), we have

$$m_i^\alpha = \frac{v_i(k-1)}{\sum_{\beta=1}^k c^\beta} \left(1 - \frac{(k-1)c^\alpha}{\sum_{\beta=1}^k c^\beta}\right)$$

which also shows that $\overline{m}^\alpha \geq \overline{m}^\beta$ if $\alpha < \beta$, i.e., lower cost firms spend more money than their higher-cost rivals. Finally, adding across α we obtain

$$\overline{m}_i = \frac{v_i(k-1)}{\sum_{\beta=1}^k c^\beta}$$

Thus there is no regionalization of customer territory at NE, with firms operating in disjoint pieces of the social network. Instead, firms that are not blockaded, compete *uniformly* throughout the social network.

3 When Externalities Become Dominant

3.1 A Markov Chain Perspective

It is often is too expensive for a firm α to provide meaningful subsidies m_i^α to each customer i. Indeed the marketing division of firm α is typically allocated a fixed budget M^α and, if there is a large population of customers, then the individual expenditures m_i^α must perforce be small. In this event, customers' behavior is predominantly driven by the externality effect of their neighbors. We can capture the situation in our model by supposing that all the θ_i^α are close to 1.

Thus we are led to inquire about the limit of the NE as the $\theta_i^\alpha \longrightarrow 1$ for all α and i. (In this scenario we will also obtain a more transparent relation between NE and the graphical structure of the social network.)

To this end — and even otherwise— it is useful to recast our model in probabilistic terms. Assume, for simplicity, that $\sum_{j \in \mathcal{I}} w_{ij}^\alpha = 1$ for all i and α. Let us consider a Markov chain with $\mathcal{I}$ as the state space and W^α as the transition matrix (i.e., w_{ij}^α is the probability of going from i to j.). Let i_t denote the (random) state of the chain at date $t = 0, 1, 2, \ldots$. Suppose that, upon arrival in state i_t, a choice $L_t \in \{Stop, Move\}$ is made with $Prob(L_t = Move) = \theta_{i_t}^\alpha$. Let T be the first time $L_t = Stop$ and consider the random variable $\gamma_{i_T}^\alpha(m)$. If $\phi^\alpha(i)$ denotes the conditional expectation $E[\gamma_{i_T}^\alpha(m)|i_0 = i]$, then clearly the I-dimensional vector ϕ^α, substituted for $p^\alpha(m)$, satisfies equation (1). Since this equation has a unique solution, it must be the case that $p^\alpha(m) = \phi^\alpha$.

Recall that each vector u^α is positive, and so we may write $u^\alpha = y^\alpha \xi^\alpha$, where $y^\alpha > 0$ is a scaler and ξ^α is a probability distribution on $\mathcal{I}$. The weighted sum $[u^\alpha]^\top p(m)$ is then equal to $y^\alpha \sum_{i \in \mathcal{I}} \xi_i^\alpha \phi^\alpha(i)$ which in turn can be expressed as $y^\alpha E[\gamma_{i_T}^\alpha(m)]$, provided we assume that the probability distribution of the initial state i_0 is ξ^α. Therefore the vector v^α / y^α is just the probability distribution of i_T initializing the Markov chain at ξ^α.

We want to analyze the asymptotics of v^α as the θ_i^α converge to 1 (since the unique NE of our games are determined by v^α). Let us first consider the simple case when $\theta_i^\alpha = \theta^\alpha$ for all i. Then the random time T becomes independent of the Markov chain and we get easily that $prob(T = t) = (1 - \theta^\alpha)(\theta^\alpha)^t$.

Therefore

$$\begin{aligned} v_i^\alpha / y^\alpha &= prob(i_T = i) \\ &= \textstyle\sum_{t=0}^\infty prob(T = t) prob(i_t = i | T = t) \\ &= \textstyle\sum_{t=0}^\infty prob(T = t) prob(i_t = i) \\ &= \textstyle\sum_{t=0}^\infty prob(T = t) E[\mathbb{1}_i(i_t)] \\ &= E[\textstyle\sum_{t=0}^\infty (1 - \theta^\alpha)(\theta^\alpha)^t \mathbb{1}_i(i_t)] \end{aligned}$$

where $\mathbb{1}_i$ is the indicator function of i: $\mathbb{1}_i(j) = 0$ if $j \neq i$ and $\mathbb{1}_i(i) = 1$.

Recall that a sequence $\{a_t\}_{t \in \mathbb{N}}$ of real numbers is said to

i) Abel -converge to a if $\lim_{\theta \to 1} \sum_{t=0}^\infty (1 - \theta)(\theta)^t a_t = a$.

ii) Cesaro-converge to a if $\lim_{N \to \infty} N^{-1} \sum_{t=0}^{N-1} a_t = a$.

The Frobenius theorem (see, e.g., line 11 on page 65 of [4]) states that a Cesaro-convergent sequence is Abel-convergent to the same limit. So, to analyse the limit behavior of v_i^α, it is sufficient to consider the Cesaro-convergence of $\{\mathbb{1}_i(i_t)\}_{t \in \mathbb{N}}$.

The finite state-set $\mathcal{I}$ of our Markov chain can be partitioned into recurrent classes $I_1^\alpha, \ldots, I_{k(\alpha)}^\alpha$ and a set of transient states I_0^α. Each recurrent class I_s^α is the support of a unique invariant probability measure μ_s^α.

If the Markov process starts within a recurrent class I_s^α (i.e., $i_0 \in I_s^\alpha$), then the ergodic theorem states that, for an arbitrary function f on $\mathcal{I}$, $N^{-1} \sum_{t=0}^{N-1} f(i_t)$ converges almost surely to $E_{\mu_s^\alpha}[f]$.

If it starts at a transient state $i \in I_0^\alpha$, then we may define the first time τ that it enters $\cup_{s \geq 1} I_s^\alpha$. Let S be the index of the recurrence class i_τ belongs to. The ergodic theorem also tells us in this case that $N^{-1} \sum_{t=0}^{N-1} f(i_t)$ converges almost surely to the random variable $E_{\mu_S^\alpha}[f]$.

Let us define $\hat{\mu}^{\alpha,i}$ as the expectation $E[\mu_S^\alpha]$, if $i \in I_0^\alpha$ and as μ_s^α if $i \in I_s^\alpha$ $(s \geq 1)$. Then we clearly get $E[N^{-1} \sum_{t=0}^{N-1} f(i_t) | i_0 = i] \longrightarrow E_{\hat{\mu}^{\alpha,i}}[f]$. Therefore, denoting $\hat{\mu}^\alpha \equiv \sum_{i \in \mathcal{I}} \xi_i^\alpha \hat{\mu}^{\alpha,i}$, the Frobenius theorem implies

Theorem 3. *As θ^α tends to 1, v_i^α converges to $y^\alpha E_{\hat{\mu}^\alpha}[\mathbb{1}_i] = y^\alpha \hat{\mu}_i^\alpha$.*

Corollary 1. *Suppose that the graph of the underlying social network is undirected and connected. Further suppose*

$$\theta_i^\alpha = \theta, \; w_{ik'} = w_{ik} \text{ and } \sum_{j \in \mathcal{I}} w_{ij} = 1$$

for all $\alpha \in \mathcal{A}, i \in \mathcal{I}$ and k, k' such that $w_{ik} > 0$ and $w_{ik'} > 0$ (i.e., all the nodes connected to i have the same influence on i). Finally suppose that u_i^α is invariant of i for all α (i.e., each company values all clients equally), w.lo.g. $u_i^\alpha = 1/|\mathcal{I}|$ for all α and i. Then as θ tends to 1, the money spent at NE by a company on any node is proportional to the degree[8] of the node.

[8] Recall that the degree of a node in an undirected graph is the number of edges incident on it.

To verify the corollary note that the invariant measure is (obviously) proportional to the degree. By Theorem 3, $v_i^\alpha = v_i$ converges to the degree of i as θ tends to 1. But, by Section 2.4, m_i^α is proportional to v_i.

Let us now deal with the general case where θ_i^α are not all the same. We will analyze the situation where θ_i^α is a function of a parameter θ going to 1 with the following hypotheses:

$$\lim_{\theta\to 1} \theta_i^\alpha(\theta) = 1, \text{ for all } i \tag{10}$$

$$\theta_i^\alpha(\theta) < 1, \text{ for all } i \text{ and } \theta < 1. \tag{11}$$

$$0 < \lim_{\theta\to 1} \frac{1-\theta_i^\alpha(\theta)}{1-\theta_1^\alpha(\theta)} = \delta_i^\alpha < \infty \tag{12}$$

For simplicity, we will also assume that $\mathcal{I} = I_1^\alpha$, i.e., there is just one recurrent class comprising all the vertices.

Theorem 4. *Under (10), (11), (12), v_i^α converges to $y^\alpha \frac{\delta_i^\alpha \mu_i^\alpha}{\sum_{j\in\mathcal{I}} \delta_j^\alpha \mu_j^\alpha}$ as θ tends to 1.*

Proof: See [2].

4 Generalizations

We have reported on some of the key results in [2]. But as shown in [2], much of the analysis can be extended to the case when externalities, utilities and costs are not necessarily linear but satisfy certain concavity/convexity conditions. In particular it can be shown that, if externalities form a "contraction", the strategic game between the firms is well-defined. Furthermore, under standard convexity hypothesis, NE continue to exist in pure strategies (see Theorem 1 of [2]). The important fixed-budget case

$$C^\alpha(m) = \begin{cases} 0 \text{ if } \sum_{i\in\mathcal{I}} m_i^\alpha \le M^\alpha \\ -\infty \text{ otherwise} \end{cases}$$

is admitted by us, as C^α is convex. (One may imagine here that the marketing division of each company α has been allocated a budget M^α to spend freely as it likes.)

It is no longer true that NE are unique (see the simple example in [2]). But if there is "enough competition" between firms, in the sense that each firm has "sufficiently many" rivals whose characteristics are "nearby", uniqueness of NE is restored. Uniqueness also holds if firms' valuations of clients are anonymous (i.e., there are no a priori biases between firms and clients), no matter how heterogenous the costs of the firms (for details see Section 5 of [2]).

Finally in [2], we also show that cross-effects (of p_j^β on p_i^α) can be incorporated, under some constraints, in our model without endangering the existence of NE.

Acknowledgements

We are grateful to Sachindra Joshi and Chandan Deep Nath for introducing us to the social network literature (in particular to the references [1,5]), and for stimulating initial discussions.

References

1. Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, San Francisco, USA, August 2001.
2. Pradeep Dubey, Rahul Garg, and Bernard De Meyer. Competing for customers in a social network. Stony Brook Working Paper (WP 06-1), August 2006. http://www.sunysb.edu/economics/research/working_paper/.
3. Matthew Jackson. The economics of social networks. In *Proceedings of the 9th World Congress of the Econometric Society (to appear), edited by Richard Blundell, Whitney Newey, and Torsten Persson*, Cambridge University Press, July 2005.
4. C. N. Moore. Summability of series. *The American Mathematical Monthly*, 39(2):62–71, February 1932.
5. Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–70, Edmonton, Alberta, Canada, July 2002.
6. J. Scott. *Social Network Analysis: A Handbook 2nd Ed.* Sage Publications, London, 2000.
7. Carl Shapiro and Hal R. Varian. *Information Rules: A Strategic Guide to the Network Economy.* Harvard Business School Press, November 1998.
8. Oz Shy. *The Economics of Network Industries.* Cambridge University Press, 2001.

Selfish Service Installation in Networks
(Extended Abstract)

Jean Cardinal[1] and Martin Hoefer[2,⋆]

[1] Computer Science Department, Université Libre de Bruxelles,
CP 212, 1050 Brussels, Belgium
jcardin@ulb.ac.be
[2] Department of Computer & Information Science, Konstanz University,
Box D 67, 78457 Konstanz, Germany
hoefer@inf.uni-konstanz.de

Abstract. We consider a scenario of distributed service installation in privately owned networks. Our model is a non-cooperative *vertex cover game* for k players. Each player owns a set of edges in a graph G and strives to cover each edge by an incident vertex. Vertices have costs and must be purchased to be available for the cover. Vertex costs can be shared arbitrarily by players. Once a vertex is bought, it can be used by any player to fulfill the covering requirement of her incident edges. Despite its simplicity, the model exhibits a surprisingly rich set of properties. We present a cumulative set of results including tight characterizations for prices of anarchy and stability, NP-hardness of equilibrium existence, and polynomial time solvability for important subclasses of the game. In addition, we consider the task of finding approximate Nash equilibria purchasing an approximation to the optimum social cost, in which each player can improve her contribution by selfish defection only by at most a certain factor. A variation of the primal-dual algorithm for minimum weighted vertex cover yields a guarantee of 2, which is shown to be tight.

1 Introduction

In this paper we consider a simple model for service installation in networks, e.g. highway or communication networks like the internet. Many networks including the internet are built and maintained by a number of different agents with relatively limited goals whereas others are centrally planned and operated – e.g. the system of interstate highways in some countries is centrally owned and planned whereas in other countries certain roads are owned privately. In particular, we consider a simple model in which network owners have to make a concrete investment to establish a service at a location in the network. Network connections are owned by different players, and each player strives to establish a service point at different locations along her connections. These service points could be resting facilities at highways or caching, buffering, or amplification technology in telecommunication networks. We investigate the question of how

⋆ Supported by DFG Research Training Group 1042 "Explorative Analysis and Visualization of Large Information Spaces".

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 174–185, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the quality and density of these service locations changes when networks are owned privately vs. owned by a central authority. A player owning a set of connections has an incentive to cover all her connections with service. The motivation for this might either be economical or lawfully enforced. If at a location a service point is already established, the incident connections are covered. This might alter the motivation for some players to invest. Formally we model this interaction with a non-cooperative game, which we call the *vertex cover game* and analyze using notions from algorithmic game theory.

Our game is similar in spirit to the one considered in [2] for network creation. We assume that a number of k non-cooperative players have to create service points in a network. The network is modeled as a graph $G = (V, E)$, in which edges represent roads or connections and vertices represent possible service point locations. Each player i owns a subset $E_i \subseteq E$ of edges and strives to establish a service point at at least one endpoint of each edge in E_i, but with minimum investment. For establishing a service point at a vertex v, a cost $c(v)$ has to be paid, which can be shared among different players. A strategy for a player is an assignment of payments to vertices in V, and once a vertex is bought – that is, when a total amount of $c(v)$ is offered by the players for a vertex v, this vertex can be used by all players to cover any of their incident edges – no matter whether they contribute to the cost or not. In this game both the problem of finding the optimum strategy for a player and the problem of finding a centralized optimum cover for all edges of all players are the classic optimization problem of minimum weighted vertex cover.

We investigate our non-cooperative game in terms of stable solutions, which are the pure strategy Nash equilibria of the game. We do not consider mixed strategy equilibria, because our environment requires a concrete investment rather a randomized action, which would be the result of a mixed strategy. We consider the *price of anarchy* [14, 16], which measures the ratio of the cost of the worst Nash equilibrium over the cost of a minimum cost cover satisfying all requirements of all players for a game. In addition, we investigate the *price of stability* [1], which measures the best Nash equilibrium in terms of the optimum cost instead of the worst equilibrium. As in general both of these ratios are in $\Theta(k)$, we investigate the question how to derive cheap covers and cost distributions that provide low incentives to selfishly defect. We present an efficient algorithm with small constant approximation ratios and provide tightness results. In addition, we show that determining existence of Nash equilibria in the vertex cover game is NP-hard.

1.1 Related Work

The vertex cover problem is a classic optimization problem in graph theory and has been studied for decades. Recently, distributed variants of the problem have attracted interest in the area of algorithmic game theory. Specifically, a cooperative vertex cover game was studied in a more general context by Immorlica et al. [11]. In this coalitional game, each edge is an agent and each coalition of players is associated with a certain cost value - the cost of a minimum cover. In [11] cross-monotonic cost sharing schemes were investigated. For each coalition of players covered they distribute the cost to players in a way that every player is better off if the coalition expands. The authors showed that no more than $O(n^{-\frac{1}{3}})$ of the cost can be charged to the agents with a cross-monotonic scheme.

Closely related to cooperative games is the study of cost sharing mechanisms. Here a central authority distributes service to players and strives for their cooperation. Starting with [6] cost sharing mechanisms have been considered for a game based on set cover. Every player corresponds to a single item and has a private utility (i.e. a willingness to pay) for being in the cover. The mechanism asks each player for her utility value. Based on this information it tries to pick a subset of items to be covered, to find a minimum cost cover for the subset and to distribute costs to covered item players such that no coalition can be covered at a smaller cost. A strategyproof mechanism allows no player to lower her cost by misreporting her utility value. The authors in [6] presented strategyproof mechanisms for set cover and facility location games. For set cover games [18, 15] recently considered different social desiderata like fairness aspects and model formulations with items or sets being agents.

Cooperative games and the mechanism design framework are used to capture situations with selfish service receivers who can either cooperate to an offered cost sharing or manipulate. Players may also be excluded from the game depending on their utility. A major goal has been to derive good cost sharing schemes that guarantee truthfulness or budget balance. Our game, however, is strategic and non-cooperative in nature and allows players a much richer set of actions. In our game each player is motivated to participate in the game. We investigate distributed uncoordinated service installation scenarios rather than a coordinated environment with a mechanism choosing customers, providing service and charging costs. Our study is, however, related to these developments – especially the singleton games, which we study in Section 5.

Our analysis uses concepts developed for non-cooperative games in the area of algorithmic game theory, in particular prices of anarchy and stability characterizing worst- and best-case Nash equilibria. The price of anarchy has been studied in a large and diverse number of games, e.g. in areas like routing and congestion [14, 17, 3], network creation [2, 8], or wireless ad-hoc networks [7, 9]. The price of stability [1] has been introduced more recently and studied for instance in network creation games [1, 10] or linear congestion games [5]. Characterizing selfish improvement possibilities and social cost of a strategy combination in terms of multiplicative factors has been recently introduced in the study of network creation games [2, 10].

1.2 Outline and Contributions

We study our vertex cover game with respect to quality of pure strategy exact and approximate Nash equilibria. Throughout the paper we denote a feasible cover by $\mathcal{C}$ and the centralized optimum cover by $\mathcal{C}^*$. All proofs omitted in this extended abstract will be given in the full version of this paper. Our contributions are as follows.

– Section 2 presents the model and some initial observations. In Section 3 we show that the price of anarchy in the vertex cover game is k, even when the underlying graph is a tree. There exist simple unweighted and weighted games for two players without Nash equilibria. They can be used to prove that the price of stability can be arbitrarily close to $k-1$. Determining existence of Nash equilibria for a given game is NP-hard, even for unweighted games or two players.

- In Section 4 we study a two-parameter optimization problem: Find covers that are cheap and allow low incentives for players to deviate. We formalize this notion as (x, y)-approximate Nash equilibria and propose a simple algorithm that finds (2,2)-approximate Nash equilibria for any vertex cover game. In addition to this algorithmic result, we show that in general there are games without a (x, y)-approximate equilibrium for $x < 2$. Recent progress on the complexity status of the minimum vertex cover problem can be used to reasonably conjecture that there can be no polynomial time algorithm with a better guarantee for the approximation ratio y as well. For planar games our argument extends to a lower bound of 1.5 on x, which can be increased close to 2 by forcing y to be close to 1 indicating a Pareto relationship between the ratios.
- Finally, in Section 5 we present games for which the price of stability is 1. For the class of *singleton games*, in which each player owns exactly one edge, we relate the results to recent work on mechanism design and cooperative game theory. For *bipartite games*, in which the graph is bipartite, our proof is based on the max-flow/min-cut technique for vertex cover. This provides new game-theoretic interpretations of classic results from graph theory and polynomial time algorithms to calculate cheap Nash equilibria.

2 The Model and Basic Results

The vertex cover game for k players is defined as follows. In an undirected graph $G = (V, E)$ with $n = |V|$ and $m = |E|$ each player i owns a set $E_i \subseteq E$ of edges. We denote by $G[E_i]$ the graph induced by the edges in E_i, and by $V(G[E_i])$ the set of vertices of $G[E_i]$. Each player strives to establish service at least one endpoint of each of her edge. For each vertex v there is a nonnegative cost $c(v)$ for establishing service at this vertex. A strategy for a player i is a function $p_i : V \to \mathbb{R}_0^+$ specifying an offer to costs of each vertex. The cost of a strategy p_i for player i is the sum of all money she offers to the vertices. Once the sum of offers of all players for vertex v exceeds its cost it is considered *bought*. Bought vertices can be used by all players to cover their incident edges. Each player strives to minimize her cost, but insists on covering her edges. A *payment scheme* is a vector $p = (p_1, \ldots, p_k)$ specifying a strategy for each player. A *Nash equilibrium* is a payment scheme such that no player i can unilaterally improve her payments by changing her strategy and still cover all her edges in E_i. A (x, y)-*approximate Nash equilibrium* is a payment scheme purchasing a cover $\mathcal{C}$ for which every player can improve her cost at most by a factor of x by switching to another strategy, and such that $c(\mathcal{C}) \leq yc(\mathcal{C}^*)$. We will refer to the factor y as the *approximation ratio*, and we term x as the *stability ratio*. The definitions of the approximation ratio and the stability ratio coincide for single-player games. Finally, we call a game *unweighted* if all vertices have equal costs, and *weighted* otherwise. We refer to games with a planar graph G as *planar games*.

The following observations can be used to simplify a game. Suppose an edge e is not included in any of the players edge sets. This edge is not considered by any player and has no influence on the game. Hence, in the following w.l.o.g. we will assume that $E = \bigcup_{i=1}^{k} E_i$.

For a player i assume the graph $G[E_i]$ induced by the players edge set E_i is not connected. The player has to cover edges in each component and her optimum strategy decomposes to cover both components independently at minimum cost. Hence, we can form an equivalent game in which the edges for each of the k_i components are owned by different subplayer $i_1, \ldots, i_{k_i}$. Then any approximate Nash equilibrium from this equivalent game can be translated to the original game, and eventually the stability ratio improves. Hence, for deriving approximate Nash equilibria we can assume that the edges of each player form only a single connected component.

Suppose an edge $e \in E$ is owned by a player i and a set of players J, i.e. $e \in E_i \cap (\bigcap_{j \in J} E_j)$. This is equivalent to one parallel edge for each player. Now consider a Nash equilibrium for an adjusted game in which there is only one edge e owned only by player i. In this equilibrium a player $j \in J$ has no better strategy to cover the edges in $E_j - e$. However, e is covered as well, potentially by a different player. If e is added to E_j again j has no incentive to deviate from her strategy as her covering requirement only increases. The Nash equilibrium for the adjusted game yields a Nash equilibrium in the original game. Hence, in the following we will assume that all edge sets E_i are mutually disjoint.

3 Quality and Existence of Nash Equilibria

In this section we consider the quality of pure Nash equilibria and the hardness of determining their existence. In general it is not possible to guarantee their existence, they can be hard to find or expensive. At first observe that the price of anarchy in the vertex cover game is k.

Theorem 1. *The price of anarchy in the vertex cover game is exactly k.*

Proof. Consider a star in which each vertex has cost 1 and each player owns a single edge. The centralized optimum cover $\mathcal{C}^*$ is the center vertex of cost 1. If each player purchases the vertex of degree 1 incident to her edge, we get a Nash equilibrium of cost k. Hence, the price of anarchy is at least k. On the other hand, k is a simple upper bound. If there is a Nash equilibrium $\mathcal{C}$ with $c(\mathcal{C}) > kc(\mathcal{C}^*)$, there is at least one player i that pays more than $c(\mathcal{C}^*)$. She could unilaterally improve by purchasing $\mathcal{C}^*$ all by herself. □

Note that the price of anarchy is k even for very simple games in which every player owns only one edge and G is a tree. Hence, we will in the following consider existence and quality of the best Nash equilibrium in a game.

Lemma 1. *There are planar games for two players without Nash equilibria.*

Proof. We consider the game for two players in Fig. 1(a) for an $\epsilon > 0$. For this game we examine four possible covers. A cover including all three vertices cannot be an equilibrium, because vertex u is not needed by any player to fulfill the covering requirement. Hence, any player contributing to the cost of u could feasibly improve by removing these payments. Suppose the cover representing an equilibrium includes v_1 and v_2. If player 1 contributes to v_1, she can remove these payments, because she only needs v_2

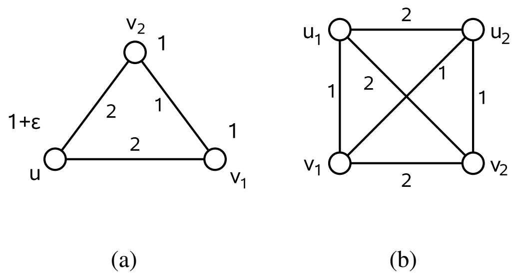


Fig. 1. Games for two players without Nash equilibria. (a) weighted game; (b) unweighted game. Edge type indicates player ownership. For the weighted game numbers at vertices indicate vertex costs.

to cover her edge. With the symmetric statement for v_2 we can see that in equilibrium player 1 could not pay anything. Player 2, however, cannot purchase both v_1 and v_2, because buying u offers a cheaper alternative to cover her edges. Finally, suppose u and v_1 are in the cover. In equilibrium player 1 will not pay anything for u. Player 2, however, cannot purchase u completely, because v_2 offers a cheaper alternative to cover the edge (u, v_2). With the symmetric observation for the cover of u and v_2, we see that there is no feasible cover that can be purchased by a Nash equilibrium. With similar arguments we can prove that the game on K_4 depicted in Fig. 1(b) has no pure Nash equilibria. This proves the lemma. □

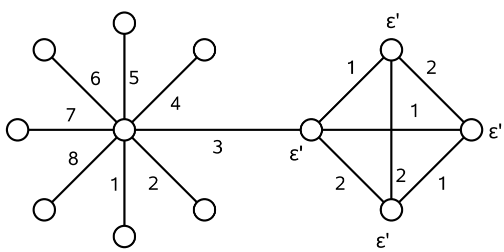


Fig. 2. A game with k=8, for which the cost of any Nash equilibrium is close to $(k-1)c(\mathcal{C}^*)$. Numbering of edges indicates player ownership. Indicated vertices have cost $\epsilon' \ll 1$, vertices without labels have cost 1.

Theorem 2. *For any $\epsilon > 0$ there is a weighted game in which the price of stability is at least $(k-1)-\epsilon$. There is an unweighted game in which the price of stability is $\frac{k+2}{4}$.*

Proof. Consider a game as depicted in Fig. 2. The centralized optimum cover includes the center vertex of the star and three vertices of the K_4-gadget yielding a total cost of $1 + 3\epsilon'$. If the center vertex of the star is in the cover and we assume to have a Nash equilibrium, no player can contribute anything to vertices of the K_4-gadget incident to edges of player 1 and 2. For this network structure, however, it is easy to note that players 1 and 2 cannot agree on a set of vertices covering their edges. Hence, to allow for a Nash equilibrium, the star center must not be picked which in turn requires all other adjacent star vertices to be in the cover. Under these conditions the best feasible cover includes the vertex that connects K_4 to the star yielding a cost of $k - 1 + 3\epsilon'$. Note that we can derive a Nash equilibrium purchasing this cover by assigning each player to purchase a star vertex - including the vertex that also belongs to K_4. Players 1 and 2 are assigned to purchase one of the additional K_4 vertices, respectively. With $\epsilon = \frac{3\epsilon'(k-2)}{1+3\epsilon'}$ the first part of the theorem follows. For the unweighted case we simply consider the game graph with all vertex costs equal to 1. A similar analysis delivers the stated bound and proves the second part of the theorem. □

Theorem 3. *It is NP-hard to determine whether (1) an unweighted vertex cover game or (2) a weighted vertex cover game for 2 players has a pure strategy Nash equilibrium, even if the graphs $G[E_i]$ are forests.*

4 Approximate Equilibria

In the previous section we saw that in general cheap pure Nash equilibria can be absent from the game. Hence, we study existence and algorithmic computation of solutions to a two-parameter optimization problem. Recall that (x, y)-approximate Nash equilibria are payment schemes that allow each player to reduce her payments by at most a factor of x and approximate $c(\mathcal{C}^*)$ to a factor of y.

Algorithm 1: (2,2)-approximate Nash equilibria

$p_i(v) \leftarrow 0$ for all players i and vertices v
$\gamma_i(e) \leftarrow 0$ for all players i and edges e
while there is an uncovered edge $e = (u, v) \in E$ **do**
  Let i be the player owning edge e, and let $\gamma_i(e) \leftarrow \min(c(u), c(v))$
  Increase payments: $p_i(u) \leftarrow p_i(u) + \gamma_i(e)$ and $p_i(v) \leftarrow p_i(v) + \gamma_i(e)$
  Add all purchased vertices to the cover
  Reduce vertex costs: $c(u) \leftarrow c(u) - \gamma_i(e)$ and $c(v) \leftarrow c(v) - \gamma_i(e)$

Theorem 4. *Algorithm 1 returns a (2,2)-approximate Nash equilibrium in $O(k(n+m))$ time.*

The algorithm is an adaption of the primal-dual algorithm for minimum vertex cover. It is also used to show that any socially optimum cover $\mathcal{C}^*$ can always be purchased by a $(2, 1)$-approximate Nash equilibrium.

Theorem 5. *For every game there is a (2,1)-approximate Nash equilibrium.*

For lower bounds on the ratios we note that any algorithm to find a (x, y)-approximate Nash equilibrium in the vertex cover game can be used as an approximation algorithm for minimum weighted vertex cover with approximation ratio $\min(x, y)$. The argument follows simply by considering a game with one player. This observation can be combined with recent conjectures on the complexity status of the minimum weighted vertex cover problem [13]. It suggests that if $P \neq NP$ and the unique games conjecture holds, there is no polynomial time algorithm delivering (x, y)-approximate Nash equilibria with $x < 2 - o(1)$ or $y < 2 - o(1)$. This bound applies only to polynomial time computability in general games. We now show that 2 is also a lower bound for the stability ratio, in a stronger sense.

Theorem 6. *For any $x < 2$ there is an unweighted game without (x, y)-approximate Nash equilibria for any $y \geq 1$.*

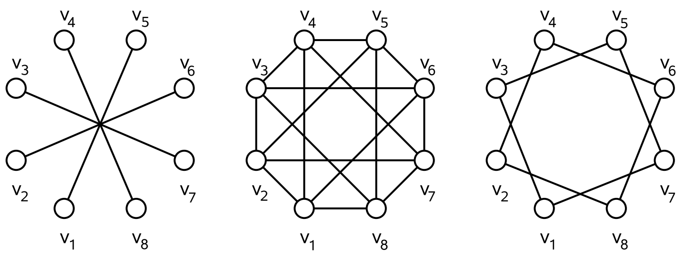


Fig. 3. From left to right the edges owned by the players in the first, second, and third classes of players for K_8. The first and second class consist of four players each, the third class of two players. Players in the first class own a single edge, while players in other classes own cycles of length 4.

Proof. The proof follows with a game on K_{4g} with $g \in \mathbb{N}$. We assume the vertices are numbered v_1 to v_{4g} and distribute the edges of the game to $2g^2 + g$ players in $g + 1$ classes as follows. In the first class there are $2g$ players. Every player i from this class owns only single edge (v_i, v_{2g+i}). Then, for each integer $j \in [1, g - 1]$ there is another class of $2g$ players. A player i in one of the classes owns a cycle of four edges (v_i, v_{i+j}), (v_{i+j}, v_{2g+i}), (v_{2g+i}, v_{2g+i+j}) and (v_{2g+i+j}, v_i). Finally, there are g players in the last class. Each player i in this class also owns a cycle of four edges (v_i, v_{g+i}), (v_{g+i}, v_{2g+i}), (v_{2g+i}, v_{3g+i}) and (v_{3g+i}, v_i). See Fig. 3 for $g = 2$ and the distribution of the 10 players into 3 classes on K_8.

Any feasible vertex cover of a complete graph is composed of either all or all but one vertices. For a cover of all $4g$ vertices we can simply drop the payments to one vertex. This reduces the payment for at least one player. In addition, it increases the cost of some of the deviations as the players must now purchase the uncovered vertex in total. The stability ratio of the resulting payment scheme can only decrease. Hence, the minimum stability ratio is obtained by purchasing $4g - 1$ vertices.

So w.l.o.g. consider a cover of $4g - 1$ vertices including all but vertex v_{4g}. Note that some player subgraphs do not include v_{4g}, and there are only two types of player subgraphs - a single edge or a cycle of length 4. First, consider a player subgraph that consists of a single edge and both endvertices are covered. If the player contributes

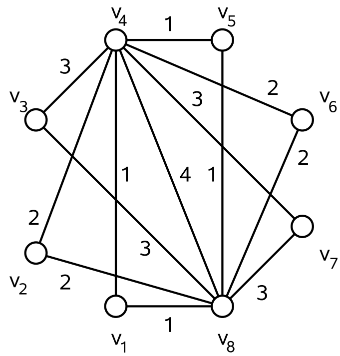


Fig. 4. Players that include v_8 in their subtree. Numbering of players as described in the text. Edge labels indicate player ownership.

to the cost of the incident vertices, she can drop the maximum of both contributions. Thus, if she contributes more than 0 to at least one of the vertices, her incentive to deviate is at least a factor of 2. Second, consider a player subgraph that consists of a cycle of length four. Label the four included vertices along a Euclidean tour with u_1, u_2, u_3 and u_4. Let the contributions of the player to u_j be x_j for $j = 1, 2, 3, 4$, resp. To optimally deviate from a given payment scheme, the player picks one of the possible minimum vertex covers $\{u_1, u_3\}$ or $\{u_2, u_4\}$ and removes all payments outside this cover. A factor of r bounding her incentives to deviate must thus obey the inequalities $\sum_{j=1}^{4} x_j \leq r(x_1 + x_3)$ and $\sum_{j=1}^{4} x_j \leq r(x_2 + x_4)$. In order to find the minimum r that is achievable we assume each player contributes only to vertices inside her subgraph. Summing the two inequalities yields $(2 - r) \sum_{j=1}^{4} x_j \leq \sum_{j=1}^{4} x_j$, so either her overall contribution is 0 or $r \geq 2$. Hence, to derive a payment scheme with stability ratio of less than 2, all $4g - 1$ vertices in the cover must be purchased by the $2g$ players whose subgraph includes v_{4g}.

For the rest of the proof we will concentrate on these $2g$ players. We will refer to player i, if she includes v_i in her subgraph, for $i = 1, \ldots, 2g - 1$. All these players own cycle subgraphs. The player that owns the edge (v_{2g}, v_{4g}) is labeled player $2g$. See Fig.4 for an example on K_8. We denote the contribution of player i to vertex v_j by p_{ij} for all $i = 1, \ldots, 2g$ and $j = 1, \ldots, 4g - 1$. Observe that for each player the set $\{v_{2g}, v_{4g}\}$ forms a feasible vertex cover. To achieve a stability ratio r, we must ensure that each player can only reduce her payments by a factor of at most r when switching to this cover. In the case of player $2g$ only $\{v_{2g}\}$ is needed, so we must ensure that she can reduce her payments by at most r when dropping all payments but $p_{2g,2g}$. As v_{4g} is not part of the purchased cover its cost of 1 must be purchased completely by a player that strives to use it in a deviation. This yields the following set of $2g$ inequalities: $\sum_{j=1}^{4g-1} p_{ij} \leq r(p_{i,2g} + 1)$, for $i = 1, \ldots, 2g - 1$ and $\sum_{j=1}^{4g-1} p_{2g,j} \leq rp_{2g,2g}$. We again strive to obtain the minimum ratio r that is possible. Note that in the minimum case no vertex gets overpaid, i.e. $\sum_{i=1}^{2g} p_{ij} = 1$ for all $j = 1, \ldots, 4g - 1$. Using this property in the sum of all the inequalities gives

$$4g-1=\sum_{j=1}^{4g-1}\sum_{i=1}^{2g}p_{ij}\le r\left(2g-1+\sum_{i=1}^{2g}p_{i,2g}\right)\le 2gr,$$

which finally yields $r \ge 2 - \frac{1}{2g}$. This proves that in the presented game no (x,y)-approximate Nash equilibrium with $x < 2 - \frac{1}{2g}$ exists. Thus, for every $\epsilon > 0$ we can pick $g \ge (2\epsilon)^{-1}$, which then yields a game without $(2-\epsilon, y)$-approximate Nash equilibria for any $y \ge 1$. □

It would be interesting to see, whether this lower bound is due to the integrality gap of vertex cover. Such a relation exists for approximate budget balanced core solutions in the cooperative game [12]. In a core solution each possible player coalition S contributes less than the cost of a minimum vertex cover for S. In our game, however, players make concrete strategic investments at the vertices, which alter the cost of the minimum cover for other players. In particular, our result is mainly due to the fact that the majority of players is sufficiently overcovered leaving only a small number of contributing players. This makes a relation to the integrality gap seem more complicated to establish.

Some classes of the vertex cover problem can be approximated to a better extent. For example, there is a PTAS for the vertex cover problem on planar graphs [4]. It is therefore natural to explore whether for planar games we can find covers with approximation and stability ratio arbitrarily close 1. The bad news is that in general there are also limits to the existence of cheap approximate Nash equilibria even on planar games. In particular, Theorem 6 provides a lower bound of 1.5 on the stability ratio for unweighted planar games. For weighted planar games there is an additional Pareto relationship between stability and approximation ratios that yields a stability ratio close to 2 for socially near-optimal covers.

Corollary 1. *There is a planar unweighted game without (x,y)-approximate Nash equilibria for any $x < 1.5$ and $y \ge 1$. For any $y < \frac{7}{6}$ there is a planar weighted game without (x,y)-approximate Nash equilibria for $x < 2/(2y-1)$.*

The better an algorithm is required to be in terms of social cost, the more it allows for selfish improvement by a factor close to 2. Note that all our lower bounds apply directly to any algorithm with or without polynomial running time.

5 Games with Cheap Nash Equilibria

In this section we present two classes of games that have cheap Nash equilibria: singleton games, in which each player owns only a single edge, and bipartite games, in which the graph is bipartite.

5.1 Singleton Games

An *exchange-minimal* vertex cover is a cover which cannot be improved by replacing a single vertex in the cover by a subset of its neighbors.

Lemma 2. *In singleton games every exchange-minimal vertex cover for G allows a distribution of vertex costs, such that no player can unilaterally improve her payments.*

Proof. Suppose we are given an exchange-minimal cover $\mathcal{C} \subset V$. For $v \in \mathcal{C}$ denote the neighbors outside the cover by $N_v(\mathcal{C}) = \{u \in V | (u,v) \in E, u \notin \mathcal{C}\}$. Suppose $c(N_v(\mathcal{C})) < c(v)$; then we can form a new cheaper feasible cover $\mathcal{C}'$ by replacing v with $N_v(\mathcal{C})$. This is a contradiction to $\mathcal{C}$ being exchange-minimal. Hence, for any $v \in \mathcal{C}$ it follows that $c(N_v(\mathcal{C})) \geq c(v)$.

This property allows a very simple algorithm to construct a Nash equilibrium from a given exchange-minimal cover $\mathcal{C}$. First initialize all payments of all players to 0. Then for each vertex $v \in \mathcal{C}$ iteratively consider all players owning an edge $e = (u,v)$ with $u \notin \mathcal{C}$. For player i set her contribution to $p_i(v) = \min(c(u), c(v) - \sum_{j \neq i} p_j(v))$. This leaves her no chance for improvement. In addition, by the previous argument every vertex $v \in \mathcal{C}$ gets paid for. □

Clearly, the centralized optimum cover $\mathcal{C}^*$ is an exchange-minimal cover, and hence there is a Nash equilibrium as cheap as $\mathcal{C}^*$. This proves that the price of stability in singleton games is 1. It does not prove, however, that a $(1,2)$-approximate Nash equilibrium can be found in polynomial time, since a 2-approximation algorithm for minimum vertex cover does not necessarily yield an exchange-minimal cover. We can devise an algorithm that starts from such an approximate cover and performs exchange operations to turn it into an exchange-minimal cover. In the weighted case, however, the number of exchange operations is not necessarily polynomial, and our algorithm could take exponential time. To circumvent this problem, we borrow a trick from Anshelevich et al. [2]. In the proposed algorithm each exchange operation guarantees a minimum improvement of the overall cost. The drawback is that we can only compute $(1+\epsilon, 2)$-approximate Nash equilibria, for any constant ϵ.

Theorem 7. *There is a polynomial time algorithm that finds $(1+\epsilon, 2)$-approximate Nash equilibria for weighted singleton games and $(1,2)$-approximate Nash equilibria for unweighted singleton games.*

Singleton games are similar in spirit to cooperative vertex cover games and mechanism design, as we assume that each edge is a single player. It is known that the core of the cooperative game contains only cost sharing functions that are at most 1/2 budget balanced. Our result states that once players have an intrinsic motivation to participate in the game and consider only selfish non-cooperative deviations, there is a cost-sharing function to distribute the full costs of an optimum cover. In this interpretation our game is close to a cooperative game that deals only with the global and singleton coalitions. Furthermore, our game is strategic, i.e. it specifies exactly to which vertex a player pays how much and in what way a player is motivated to reallocate her payments. This is a feature that is not considered in the cooperative framework.

5.2 Bipartite Games

Lemma 3. *In bipartite games there is an optimum vertex cover $\mathcal{C}^*$ for G which allows a distribution of vertex costs such that no player can unilaterally improve her payments.*

The proof relies on standard algorithmic techniques like maximum weight matching and max-flow/min-cut calculations. This allows to construct Nash equilibria with optimum social cost in polynomial time.

Theorem 8. *The price of stability in bipartite games is 1. Nash equilibria purchasing $\mathcal{C}^*$ can be found in polynomial time.*

References

1. E. Anshelevich, A. Dasgupta, J. Kleinberg, T. Roughgarden, É. Tardos, and T. Wexler. The price of stability for network design with fair cost allocation. In *Proc. 45th FOCS*, pages 295–304, 2004.
2. E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proc. 35th STOC*, pages 511–520, 2003.
3. B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proc. 37th STOC*, pages 331–337, 2005.
4. B. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
5. G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proc. 13th ESA*, pages 59–70, 2005.
6. N. Devanur, M. Mihail, and V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location problems. In *Proc. 4th EC*, pages 108–114, 2003.
7. S. Eidenbenz, A. Kumar, and S. Zust. Equilibria in topology control games for ad hoc networks. In *Proc. DIALM-POMC 2003*, pages 2–11, 2003.
8. A. Fabrikant, A. Luthera, E. Maneva, C. Papadimitriou, and S. Shenker. On a network creation game. In *Proc. 22nd PODC*, pages 347–351, 2003.
9. M. Halldórsson, J. Halpern, L. Li, and V. Mirrokni. On spectrum sharing games. In *Proc. 23rd PODC*, pages 107–114, 2004.
10. M. Hoefer and P. Krysta. Geometric network design with selfish agents. In *Proc. 11th COCOON*, pages 167–178, 2005.
11. N. Immorlica, M. Mahdian, and V. Mirrokni. Limitations of cross-monotonic cost sharing schemes. In *Proc. 16th SODA*, pages 602–611, 2005.
12. K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proc. 33rd STOC*, pages 364–372, 2001.
13. S. Khot and O. Regev. Vertex cover might be hard to approximate within 2-ϵ. In *Proc. 18th CCC*, page 379, 2003.
14. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. 16th STACS*, pages 404–413, 1999.
15. X. Li, Z. Sun, and W. Wang. Cost sharing and strategyproof mechanisms for set cover games. In *Proc. 22nd STACS*, pages 218–230, 2005.
16. C. Papadimitriou. Algorithms, games and the internet. In *Proc. 33rd STOC*, pages 749–753, 2001.
17. T. Roughgarden and É.Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.
18. Z. Sun, X. Li, W. Wang, and X. Chu. Mechanism design for set cover games when elements are agents. In *Proc. 1st AAIM*, 2005.

Games of Connectivity

Pradeep Dubey[1,2] and Rahul Garg[3]

[1] Center for Game Theory in Economics, Stony Brook University, USA
[2] Cowles Foundation, Yale University, USA
[3] IBM India Research Lab, New Delhi, India

Abstract. We consider a communications network in which users transmit beneficial information to each other at a cost. We pinpoint conditions under which the induced cooperative game is supermodular (convex). Our analysis is in a lattice-theoretic framework, which is at once simple and able to encompass a wide variety of seemingly disparate models.

Keywords: information lattice, multicast/unicast transmission, cooperative games, Shapley value, convex/supermodular games.

JEL Classification: C71, D82, L96.

1 Introduction

A cooperative game $w : 2^N \to R$ (with $w(\phi) = 0$) on the player set N describes what each coalition can obtain by itself. The core $\mathcal{C}(w)$ is the set of all payoffs[1] $x \in R^N$ such that $\sum_{i \in N} x_i = w(N)$ and $\sum_{i \in S} x_i \geq w(S)$ for all $S \subset N$. In short, the core consists of divisions of the maximal proceeds $w(N)$ in the game such that no coalition has incentive to break away and get more on its own.

On the other hand, the Shapley value $\Phi(w) \in R^N$ defines a "fair" allocation of $w(N)$ among the players (see [6]) for details.

The problem is that often these two concepts are at odds with each other: the Shapley value $\Phi(w)$ is not in the core $\mathcal{C}(w)$.

In a seminal paper [7], Shapley showed that if w is supermodular[2] (i.e., $w(S \cup T) + w(S \cap T) \geq w(S) + w(T)$ for all $S \subset N, T \subset N$) then $\Phi(w)$ is not only in $\mathcal{C}(w)$ but in fact is the "center of gravity" of $\mathcal{C}(w)$ (see [6] for the precise details). In such games the plausibility of the Shapley value as a solution concept is considerably bolstered because it is not only fair but also (coalitionally) stable.

In this paper we pinpoint conditions under which certain games of connectivity are supermodular. Players in our model are located at the vertices of a

[1] The component x_i represents what player i gets.

[2] Shapley called such games "convex" and pointed out that the "snowball effect" i.e., $w(T \cup \{i\}) - w(T) \geq w(S \cup \{i\}) - w(S)$ whenever $S \subset T \subset N$ and $i \notin T$, is equivalent to convexity. The snowball effect enables us to interpret supermodular (convex) games as those that exhibit increasing returns to cooperation: the marginal contribution of a player to a coalition goes up as the coalition is enhanced.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 186–197, 2006.
© Springer-Verlag Berlin Heidelberg 2006

communications network and can stand to gain a lot by sharing disparate bits of information that they initially hold. Indeed information is more amenable to sharing than standard commodities. Commodities are typically lost to the person who gives them away. Information in contrast has "the quality of mercy", blessing him that gives and him that takes, since the giver retains all his information even as he sends it out. Nevertheless it is not automatic that all information will be shared. This is because, though costlessly duplicable, information may be costly to transmit (e.g., on account of setup costs of links in the communications network). Any coalition must do a careful cost-benefit analysis, choosing that pattern of information transmission which minimizes the total net benefit to its members.

It should be pointed out that our model is inspired by a multicast transmission game presented in [5], though the focus there was on using the Shapley value (or else the marginal cost rule) to define a mechanism that is group-strategyproof and has other desirable properties. The approach in [2] and [1] is similar, in that cost-sharing schemes (such as the Shapley value), are invoked to construct non-cooperative games on networks. In contrast, we here analyze network games from a *purely cooperative* point of view.

An important feature of our approach is that we formulate information in terms of a *lattice*. This leads to a framework that is at once universal and simple. We can encompass a wide variety of seemingly different models, involving unicast and multicast modes of transmission, setup and variable costs in the communications network, and information that comes in various guises (from finite dimensional vectors, to partitions of a set, to layered encoding). The lattice framework makes for a remarkably transparent analysis in all cases.

The paper is organized as follows. In Section 2 we present some motivating examples, starting with the model in [5]. The abstract lattice-theoretic framework is presented in in Section 3. In Section 4 we establish our main result which states that games of connectivity are supermodular. Section 5 points out a monotonicity property of optimal transmissions. Finally, in Section 6, we show how to fit the examples into our lattice-theoretic framework; and we also examine the tightness of our assumptions and indicate some generalizations of the model.

2 Examples

We present a series of examples of information transmission in a network, all of which yield supermodular games, as we shall see in Sections 4 and 6.

2.1 Multicast Transmission

First let us recall the game presented in [5]. There is a finite tree Γ with a sender δ located at its root and and a distinct receiver at each leaf (terminal vertex). Any receiver α can get information from δ if α is connected to δ using the edges of Γ. The tree Γ is viewed as a digital network which carries a public broadcast by δ, and it is assumed that information flowing into any vertex of the tree can be costlessly duplicated and sent out (multicast) on any subset of the outgoing edges. But the edges of Γ do have setup costs associated to them. Offsetting these costs are benefits $B(\alpha)$ to α when he receives information from δ.

A cooperative game is induced on the player-set N of receivers in a natural manner. Any coalition $S \subset N$ can use an arbitrary subtree Γ' of Γ at the cost $C(\Gamma')$ of all the edges of Γ'. The benefit S derives from Γ' is $B(S, \Gamma') = \sum_\alpha B(\alpha)$, where the summation runs over all α in S which are connected to δ via Γ'. Thus the "worth" $w(S)$ of coalition S (i.e., the most S can guarantee to itself) is obtained by maximizing the net benefit $B(S, \Gamma') - C(\Gamma')$ over all possible subtrees Γ'.

There can be several senders located at different vertices of the tree, each with its own distinctive information to transmit. Moreover not all senders need be "dummies" as in [5]. Some of them could be bona fide players in the game with the power to withhold their information. One could also imagine them to have different transmission trees, possibly with significant overlap.

In spite of these complications, the game remains supermodular and so the Shapley value continues to be centrally located in the core (but its computation may no longer be as felicitous as in [5]).

2.2 Unicast Transmission

Imagine a set of users connected to each other through a hierarchical network (as in telephony). Again suppose they are located on the leaves of a tree Γ with other vertices acting as relays. But the communication is private rather than public, and the users transmit information to each other on a one-to-one basis.

The user at leaf α can choose the amount of information $\tau_{\alpha\beta} \in [0, m], m > 0$, to be sent to β. The total benefit derived at β is $\sum_\alpha B_{\alpha\beta}(\tau_{\alpha\beta})$, where $B_{\alpha\beta}$ is an arbitrary non-decreasing function. As before, it costs to use the tree. Each edge now has not only a setup cost, but also an arbitrary non-decreasing variable cost for every α−to-β flow on it. (The variable costs here add across flows, but the setup cost is invariant of them.)

This unicast scenario also gives rise to a cooperative game in an obvious way. Any coalition S chooses $\tau = \{\tau_{\alpha\beta} : \alpha \in S, \beta \in S\}$, and a subforest of Γ to carry τ, so as to maximize the net benefit.

It turns out that this game is also supermodular.

2.3 Transmission of Layered Information

We turn to a situation where information is encoded or organized in layers (e.g., as in a video transport system, see [8]). To be precise, suppose layer L_i consists of " information bricks" numbered by integers $m_{i-1} + 1, m_{i-1} + 2, \ldots, m_i$. The bricks in $L = \cup_{i=1}^k L_i$ are, however, distributed arbitrarily among the n players located at the vertices of a communication tree Γ, with no duplication. So, denoting by Σ_α the set of bricks held at vertex α, we have $\Sigma_\alpha \cap \Sigma_\beta = \phi$ if $\alpha \neq \beta$. Players wish to receive bricks in order to build a "knowledge pyramid", but they cannot construct layer L_i unless all previous layers $L_1, L_2, \ldots, L_{i-1}$ are in place. Of course, since these bricks are not standard commodities but signify information, no player loses any of his own bricks by sending them to others. The player at vertex α may transmit any subset $Q_e \subset \Sigma_\alpha$ on any edge e emanating from α. Then for any edge e' that follows from e, he can send $Q_{e'} \subset Q_e$, and so on. In short he can contemplate multicast transmission on Γ with α as the root.

There is a set-up cost for every edge e as earlier, and additional flow costs $C_{e,\alpha}(x)$ for $x \in \Sigma_\alpha$.

Benefits accrue as follows. Denoting by $Q_{\beta\alpha} \subset \Sigma_\beta$ the subset of bricks that α receives from β, the benefit to α is $f_\alpha(n)$, where

$$n = \max\{j : L_i \subset \Sigma_\alpha \cup (\cup_\beta Q_{\beta\alpha}) \forall i \leq j\}$$

and $f(n)$ is an arbitrary non-decreasing function.

The idea here, as was said, is that information is organized in pyramidical form. Information of layer L_i is not usable unless all layers $L_1, L_2, \ldots, L_i$ are complete.

The cooperative game, arising in this setup, is once again supermodular.

2.4 Transmission of Information Partitions

As before, Γ is a tree with players located at its vertices. Let $Q = \{1, 2, \ldots, k\}$ be the set of states of nature, and let $\{Q_\alpha : \alpha \in V\}$ be a partition of Q. (Here V denotes the set of vertices of Γ and Q_α is understood to be the empty set if no player is located at α.) Further let P_α be a partition of Q_α. The interpretation is that $\{P_\alpha, Q \backslash Q_\alpha\}$ is the private information initially held by the player at vertex α. Notice that private information is disjoint across players, i.e., each player is in the dark about states that other players can distinguish.

For simplicity every player α has a state-contingent endowment $(a_1(\alpha), \ldots, a_k(\alpha))$ of a single non-tradeable resource (such as his skill), to be used as input in his individual production. He must, of course, use the same input in states that he cannot distinguish. But since expected profit of any player depends on his state-contingent vector of inputs, there are inherent gains from sharing information. The precise model is as follows.

Each player can transmit its information partition (or any coarsening thereof) to other vertices prior to the production stage. If the player at vertex α winds up with the partition P of Q, his profit (via production) is

$$\max f_\alpha(x_1, x_2, \ldots, x_k)$$

$$\begin{aligned} \text{Subject to: } x_i &\leq a_i(\alpha) \\ x_i &\geq 0 \\ \text{and } i \sim_P j \Rightarrow x_i &= x_j \end{aligned}$$

where $i \sim_P j$ means that i and j are in the same cell of the partition P. We assume that the production function f_α is supermodular on R^k_+, i.e., (assuming differentiability):

$$\frac{\partial}{\partial x_i} \frac{\partial f_\alpha}{\partial x_j} \geq 0$$

for all i, j and α. In other words the inputs $x_1, x_2, \ldots, x_k$ are weakly complementary: if α increases his input in some state, this does not diminish his marginal productivity in any state.

When a coalition S forms, its members can transmit information to each other through any subforest of Γ after paying the setup costs, and then they can pool their profits.

This, too, induces a game that is supermodular.

2.5 General Network with Controlled Edges

Let G be an arbitrary undirected graph with edge set E and vertex set V. For each vertex $\alpha \in V$, let $\Gamma(\alpha) \subset G$ be a tree rooted at α on which α is constrained to transmit its information. Further suppose that edges of G are subject to the control of coalitions.

Thus when a coalition S forms, each $\alpha \in S$ has access to only those edges in $\Gamma(\alpha)$ whose controllers are contained in S.

In this setup, players who are neither senders nor receivers of information, may nevertheless have a vital role to play in the game on account of their control of edges (such as cable operators or monopoly network providers).

All of our preceding examples can be embedded in this larger framework. The games induced will still be supermodular.

3 The Abstract Model

We build an abstract lattice-theoretic model of information and its transmission, which unifies the above (and more) examples and makes for a particularly transparent analysis.

3.1 The Communications Network

Let $G = (V, E)$ be a graph where V is a finite set of vertices and E is a set of undirected edges.

For every $\alpha \in V$ there is a tree $\Gamma(\alpha) \equiv (V(\alpha), E(\alpha)) \subset G$, rooted at α, that can be used by α to transmit its information to other vertices.

3.2 Information

Information is modeled as a lattice $\mathcal{L}$ with $\geq$ denoting the partial order and $\vee, \wedge$ the join and the meet operators[3]. We assume that $0 \equiv \wedge\{x : x \in \mathcal{L}\}$ exists in $\mathcal{L}$ and that that $\wedge$ distributes over $\vee$, i.e.,

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

[3] Recall (see e.g. [3]) that for and x and y in $\mathcal{L}$, there exists a greatest lower bound w.r.t. $\geq$ (denoted $x \wedge y$) and a least upper bound (denoted $x \vee y$).

for all $x, y, z \in \mathcal{L}$. This property holds in a variety of contexts and is well-known (see [3]).

The canonical examples we have in mind is that $\mathcal{L}$ is the power set of a finite set with $\geq$ corresponding to the set-theoretic notion of $\supset$; or that $\mathcal{L}$ is the set of all partitions of a finite set with $\geq$ corresponding to refinement; or that $\mathcal{L}$ is a closed interval of the real line with $\geq$ corresponding to the standard order; or that $\mathcal{L}$ is the product lattice of finitely many such lattices. In all of these cases 0 exists in $\mathcal{L}$ and the distributive property holds.

Any vertex $\alpha \in V$ can transmit information from a sub-lattice $\mathcal{L}(\alpha)$ of $\mathcal{L}$. A key assumption we make is that the information held at different vertices is disjoint, i.e.,

$$x \in \mathcal{L}(\alpha), y \in \mathcal{L}(\beta), \alpha \neq \beta \Rightarrow x \wedge y = 0$$

We also assume that each vertex can opt to send no information, i.e., $0 \in \mathcal{L}(\alpha)$ for all $\alpha \in V$.

3.3 Location of Players and Public Facilities

Let $N = \{1, 2, \ldots, n\}$ be the set of players. There is an additional dummy player, labeled $n+1$, used to model public facilities available to all players in N. Denote $\tilde{N} = N \cup \{n+1\}$.

Each vertex is occupied by a player[4] as specified by a *location map*

$$\eta : V \to \tilde{N}$$

where $\eta(\alpha)$ denotes the player (possibly, dummy) at vertex α. Let $V(S)$ represent the set of all the vertices occupied by players in $S \cup \{n+1\}$ i.e.,

$$V(S) = \{\alpha \in V : \eta(\alpha) \in S \cup \{n+1\}\}$$

3.4 Control of Edges

Edges are controlled by coalitions of players in accordance with a *control map*

$$\kappa : E \to 2^N$$

where $\kappa(e)$ denotes the coalition that controls[5] the use of edge e. (If $\kappa(e) = \phi$, then e is accessible to everyone.)

3.5 The Transmission of Information

Each vertex α can transmit information $x \in \mathcal{L}(\alpha)$ to other vertices on its tree $\Gamma(\alpha) \equiv (V(\alpha), E(\alpha))$. Concatenating across vertices, the total transmission may be viewed as a map $\tau : E \times V \to \mathcal{L}$ with the interpretation that $\tau(e, \alpha)$ is the

[4] The case where several players occupy a vertex is included in our set-up (see remark 3 in Section 6).

[5] A natural case: if $e = (\alpha, \beta)$, then $\kappa(e) = (\eta(\alpha) \cup \eta(\beta)) \cap N$.

information transmitted by the vertex α on the edge e. Some natural conditions must be imposed on this map τ. Any vertex α can send information only out of $\mathcal{L}(\alpha)$ i.e.,

$$\tau(e, \alpha) \in \mathcal{L}(\alpha) \tag{1}$$

for all $\alpha \in V$ and $e \in E(\alpha)$. Moreover, no vertex α can send any (except null) information on edges outside its tree i.e.,

$$\tau(e, \alpha) = 0 \text{ if } e \notin E(\alpha) \tag{2}$$

for all $\alpha \in V$ and $e \in E$. Finally, the join of all the information of α that flows out of a vertex must be no more than the information of α that arrives at it, i.e.,

$$\tau(e, \alpha) \geq \vee\{\tau(e', \alpha) : e' \in F(e, \alpha)\} \tag{3}$$

for all $\alpha \in V$ and $e \in E(\alpha)$, where $F(e, \alpha)$ denotes the set of immediate offspring edges of e in the tree $\Gamma(\alpha)$.

Let $\mathcal{T}$ denote the set of all possible transmissions, i.e.,

$$\mathcal{T} = \{\tau : E \times V \to \mathcal{L} : \tau \text{ satisfies (1), (2) and (3)}\}$$

The set $\mathcal{T}$ itself forms a lattice under the natural definitions: $\tau \geq \tau'$ if $\tau(e, \alpha) \geq \tau'(e, \alpha)$ for all e, α; $(\tau \vee \tau')(e, \alpha) = \tau(e, \alpha) \vee \tau'(e, \alpha)$ for all e, α; $(\tau \wedge \tau')(e, \alpha) = \tau(e, \alpha) \wedge \tau'(e, \alpha)$ for all e, α.

For any coalition $S \subset N$, define the subset $\mathcal{T}(S) \subset \mathcal{T}$ of *transmissions feasible for S* as follows:

$$\mathcal{T}(S) = \{\tau \in \mathcal{T} : \text{ for any } e \text{ and } \alpha,\ \tau(e, \alpha) > 0 \Rightarrow \kappa(e) \subset S \text{ and } \alpha \in S \cup \{n+1\}\}$$

In other words, only members of S or public vertices can transmit information in $\mathcal{T}(S)$; and only the edges under the control of S may be used.

3.6 The Reception of Information

A transmission $\tau \in \mathcal{T}$ induces a reception $\sigma(\tau, \alpha) \in \mathcal{L}$ at every vertex $\alpha \in V$ as follows:

$$\sigma(\tau, \alpha) = (x^*(\alpha)) \vee (\vee\{\tau(e(\beta, \alpha), \beta) : \beta \in V \backslash \{\alpha\} \text{ and } \alpha \in \Gamma(\beta)\})$$

where $e(\beta, \alpha)$ is the edge coming into α from β in $\Gamma(\beta)$ and $x^*(\alpha) \equiv \vee\{x : x \in \mathcal{L}(\alpha)\}$.

Here $x^*(\alpha)$ represents the maximum information in $\mathcal{L}(\alpha)$. Since α can costlessly receive its own information, and since information is valuable, we suppose that α always "sends" $x^*(\alpha)$ to itself. The total reception at α is obtained by joining $x^*(\alpha)$ with the bits of information $\tau(e(\beta, \alpha), \beta)$ sent to α by other vertices β.

3.7 The Cost of a Transmission

The cost of transmitting information (originating at different vertices) on any edge is given by[6] $c_e : \mathcal{L}^V \to R_+$, where $c_e((x(\alpha))_{\alpha \in V}) \equiv$ the cost of the flow $(x(\alpha))_{\alpha \in V}$ on e. We postulate that c_e is *submodular* on $\mathcal{L}^V$, i.e.,

$$c_e(x \vee y) + c_e(x \wedge y) \leq c_e(x) + c_e(y)$$

for all $e \in E$ and $x, y \in \mathcal{L}^V$. Such costs can arise in several ways. For instance, suppose there is a set-up cost $f(e)$ for e, and a further set-up cost $f(e, \alpha)$ for every vertex α that uses e, i.e.,

$$c_e((x(\alpha))_{\alpha \in V}) = \begin{cases} 0, \text{ if } x(\alpha) = 0 \text{ for all } \alpha \\ f(e) + \sum_{x:x(\alpha)>0} f(e, \alpha), \text{ otherwise} \end{cases}$$

It is evident that this cost function is submodular, and that it remains so if we add variable costs $\sum_{\alpha \in V} g_\alpha(x(\alpha))$ provided each $g_\alpha : \mathcal{L} \to R_+$ is itself submodular (i.e., evinces economy of scale).

The *cost of transmission* $\tau \in \mathcal{T}$ is the sum of the costs incurred on all the edges, i.e.,

$$C(\tau) = \sum_{e \in E} c_e((\tau(e, \alpha))_{\alpha \in V})$$

It is easy to verify that C is submodular on $\mathcal{T}$, i.e.,

$$C(\tau) + C(\tau') \geq C(\tau \vee \tau') + C(\tau \wedge \tau') \tag{4}$$

3.8 The Benefit from a Transmission

For every vertex $\beta \in V$, there is a benefit function $B_\beta : \mathcal{L} \to R_+$, where $B_\beta(x)$ represents the benefit to β from receiving information $x \in \mathcal{L}$. We assume that B_β is supermodular and non-decreasing for all $\beta \in V$ i.e.,

$$B_\beta(x \vee y) + B_\beta(x \wedge y) \geq B_\beta(x) + B_\beta(y)$$

and

$$x \geq y \Rightarrow B_\beta(x) \geq B_\beta(y)$$

The benefit to a coalition $S \subset N$ from transmission $\tau \in \mathcal{T}$ is given by

$$B(S, \tau) = \sum_{\beta \in V(S)} B_\beta(\sigma(\tau, \beta))$$

It is again easy to verify that B is supermodular on $\mathcal{T}$ (with S fixed). But the supermodularity of B and the submodularity of C do not immediately lead to the supermodularity of the game w defined in the next section.

[6] Note that $\mathcal{L}^V$ is a finite product of $\mathcal{L}$ with itself (V times) and is a product lattice.

4 The Connectivity Game

We consider the cooperative game that arises from the communications network. A non-empty coalition $S \subset N$ can choose any $\tau \in \mathcal{T}(S)$ to transmit information between its members or to receive information from public vertices. The coalition obtains total benefit $B(S, \tau)$ but at a cost $C(\tau)$. The maximum net benefit that S can guarantee is therefore given by

$$w(S) = \max_{\tau \in \mathcal{T}(S)} B(S, \tau) - C(\tau)$$

(with $w(\phi)$ understood to be 0). We call w the *connectivity game.*

Recall that a game $w : 2^N \to R$ is called *supermodular* (or, as in [7], convex) if w is supermodular on the lattice 2^N, i.e.,

$$w(S \cup T) + w(S \cap T) \geq w(S) + w(T)$$

for all $S \subset N$ and $T \subset N$. Our main result is:

Theorem 1. *The connectivity game w is supermodular.*

For the proof see [4].

5 The Growing Transmissions Property

It is worth noting that optimal transmissions grow with the coalitions in the sense made precise by Theorem 2 below.

Theorem 2. *Let $S \subset T \subset N$ and let $\tau_1 \in \mathcal{T}(S)$ be an optimal transmission for S. Then there exists an optimal transmission $\tau \in \mathcal{T}(T)$ for T such that $\tau \geq \tau_1$.*

For the proof see [4].

6 Remarks

Remark 1 (Embedding the examples). We briefly indicate how to fit our examples (from Section 2) into the abstract model.

For Section 2.1, take $\Gamma(\alpha) = \Gamma$ rooted at α, $\kappa(e) = \phi$ for all e, $\mathcal{L}(\delta) = \{0, 1\}$, $\mathcal{L}(\alpha) = \{0\}$ for all $\alpha \neq \delta$, $\mathcal{L}$ = the cross product of all these lattices, $B_\delta = 0$, $B_\alpha(0) = 0$ and $B_\alpha(1) = B(\alpha)$ for all $\alpha \neq \delta$. Finally the cost of an edge is its setup cost if there is a non-zero transmission on it and zero otherwise.

For Section 2.2, let $\mathcal{L}(\alpha) = [0, m]^V$, each of whose elements specifies the information sent by α to all the other vertices. The lattice operations $\vee$ and $\wedge$ are obtained by taking component-wise maximum and minimum. $\mathcal{L}$ as usual is the cross product of all the $\mathcal{L}(\alpha)$. The cost functions are obvious. The rest of the construction is as before. (Notice that despite the fact that the components of the benefit and cost functions have no supermodularity or concavity assumptions on them, the benefit/cost functions are supermodular/submodular in our lattice framework. This follows from the fact that they are additive over their components and that super or sub-modularity is no constraint on a function of one variable.)

For the example in Section 2.3, take $\mathcal{L}(\alpha)$ to be the totally ordered set $\{0\} \cup \Sigma_\alpha$, and $\mathcal{L}$ to be the cross product. We leave it to the reader to verify that the benefit function is supermodular.

Finally, for the example in Section 2.4, take $\mathcal{L}(\alpha)$ to be the lattice of all partitions of Q which are coarser than $\{P_\alpha, Q \backslash Q_\alpha\}$. The supermodularity of the benefit functions follows from that of $f_\alpha, \alpha \in V$.

Remark 2 (Acyclicity). Cycles in the transmissions network $\Gamma(\alpha)$ can cause our result to breakdown. Consider the network in Figure 1 in which players 1, 2, 3, 4, each have access to the whole graph, with costs as shown and with $\epsilon < 1$.

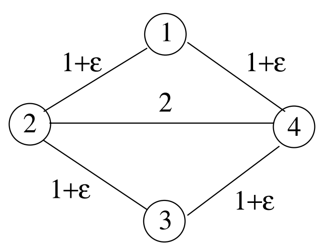


Fig. 1. Cycles in the communications network

Further suppose that 1, 2, 3 each derive benefit $B > 2(1+\epsilon)$ from being connected to 4. Then it is clear that

$$\begin{aligned} w(2,4) &= B - 2 \\ w(2,3,4) &= 2B - 2(1+\epsilon) \\ w(1,2,4) &= 2B - 2(1+\epsilon) \\ w(1,2,3,4) &= 3B - 3(1+\epsilon) \end{aligned}$$

But then

$$w(1,2,3,4) + w(2,4) = 4B - 5 - 3\epsilon \leq 4B - 4 - 2\epsilon = w(1,2,4) + w(2,3,4)$$

showing that w is not supermodular.

Remark 3 (Multiple players at a vertex). Our model allows for many players to be located at the same vertex α. Indeed, by creating a new vertex for each player present at α, and joining these with zero-cost edges to α, we create an expanded graph which fits our model (see Figure 2).

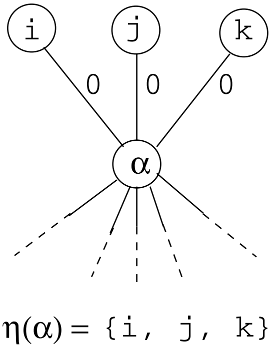


Fig. 2. Modeling multiple players at a vertex

Remark 4 (Control of vertices). Our model also permits coalitions to control vertices by the graph expansion shown in Figure 3. Every edge incident at α is intercepted with a zero-cost edge controlled by the coalition controlling α.

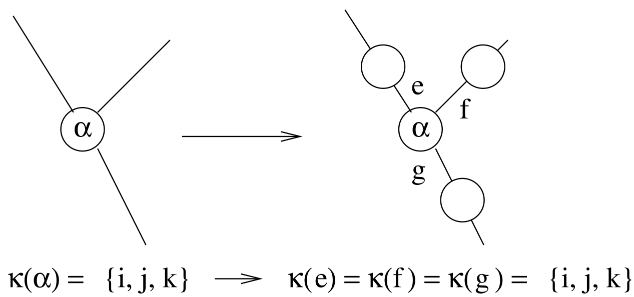


Fig. 3. Modeling control of vertices

Remark 5 (Veto players). A more general control of edges by veto players renders our results invalid. Consider a player set $\{1, 2, 3\}$ and suppose that there is common tree available to everyone, which consists of just one zero-cost edge connecting player 1 to a public vertex. The edge can be sanctioned by player 1 (the veto player), in conjunction with any player in $\{2, 3\}$. The only benefit B is obtained by player 1 when he gets connected to the public vertex. In this game $w(1) = 0$ and $w(1, 2) = w(1, 3) = w(1, 2, 3) = B$. Hence $w(1, 2, 3) + w(1) = B < 2B = w(1, 2) + w(1, 3)$, showing that w is not supermodular.

Remark 6 (Dropping distributivity). In the special case where $\mathcal{L}$ is the cross product of the lattices $\mathcal{L}(\alpha)$ over $\alpha \in V$, our results hold without postulating that $\wedge$ distributes over $\vee$. But in general distributivity is indispensable.

Remark 7 (Enhancement of information). So far we have taken information to be fixed a priori. But it could well happen that the information of an agent gets enhanced by virtue of the information he receives from others. He can turn around and send his enhanced information back to them, enhancing theirs', and so on. Even in this setting, under suitable hypotheses, the induced cooperative game is well-defined (i.e., the enhancement sequence converges) and is supermodular, as we shall show in a sequel paper.

References

1. Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *Symposium on Foundations of Computer Science (FOCS 2004)*, October 2004.
2. Elliot Anshelevich, Anirban Dasgupta, Éva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *ACM Symposium on Theory of Computing (STOC 2003)*, pages 511–520, June 2003.
3. Garrett Birkhoff. Lattice theory. *Encyclopaedia Brittanica, Macropaedia, Volume 1, Encyclopaedia Brittanica Inc., Helen Hemingway Benton, London*, pages 519–524, 1977.
4. Pradeep Dubey and Rahul Garg. Games of connectivity. Technical Report RI04006, IBM Research Report, July 2004.
5. Joan Feigenbaum, Christos Papadimitriou, and Scott Shenker. Sharing the cost of multicast transmissions. In *ACM Symposium on Theory of Computing (STOC'00)*, Portland, Oregon, USA, May 2004.
6. Lloyd Shapley. A value for n-person games. *Contributions to the Theory of Games, edited by H. W. Kuhn and A. W. Tucker. Annals of Mathematics Studies*, 2(28), 1953.
7. Lloyd Shapley. Cores of convex games. *International Journal of Game Theory*, 1:11–26, 1971.
8. Y. Wang and Q. Zhu. Error control and concealment for video communications: A review. *Proceedings of IEEE, special issue on Multimedia Signal Processing*, pages 974–997, May 1998.

Assignment Problems in Rental Markets*

David Abraham[1], Ning Chen[2], Vijay Kumar[3], and Vahab S. Mirrokni[4]

[1] Department of Computer Science, Carnegie Mellon University
`dabraham@cs.cmu.edu`
[2] Department of Computer Science & Engineering, University of Washington
`ning@cs.washington.edu`
[3] Strategic Planning and Optimization Team, Amazon.com
`vijayk@amazon.com`
[4] Microsoft Research, Redmond, USA
`mirrokni@theory.csail.mit.edu`

Abstract. Motivated by the dynamics of the ever-popular online movie rental business, we study a range of assignment problems in rental markets. The assignment problems associated with rental markets possess a rich mathematical structure and are closely related to many well-studied one-sided matching problems. We formalize and characterize the assignment problems in rental markets in terms of one-sided matching problems, and consider several solution concepts for these problems. In order to evaluate and compare these solution concepts (and the corresponding algorithms), we define some "value" functions to capture our objectives, which include fairness, efficiency and social welfare. Then, we bound the value of the output of these algorithms in terms of the chosen value functions.

We also consider models of rental markets corresponding to static, online, and dynamic customer valuations. We provide several constant-factor approximation algorithms for the assignment problem, as well as hardness of approximation results for the different models. Finally, we describe some experiments with a discrete event simulator compare the various algorithms in a practical setting, and present some interesting experimental results.

1 Introduction

Online movie rental services such as Blockbuster.com, Netflix.com and Amazon.co.uk are perhaps the most familiar instances of *rental markets* in the Internet. The primary function of centralized rental markets such as these is to repeatedly allocate a rental inventory in accordance with customer demand at successive time instances. Customers return assigned items after some time steps and a central authority reassigns the items to other customers. The basic model behind these markets involves (partial) customer preferences over items, and the rental service aims to satisfy these preferences within the constraints of available

* This work was done when the authors were visiting Amazon.com.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 198–213, 2006.
© Springer-Verlag Berlin Heidelberg 2006

inventory. Other objectives include trying to maximize overall resource (inventory) utilization and limiting (perceived) unfairness in the allocation process.

Given the collection of competing objectives, resource constraints and challenging business characteristics (popularity of movies tends to be highly non-uniform and extremely short-lived; there is a deep catalog with very sparse demand in the tail), it is natural that the allocation process involves complex decisions. As we shall see, specific considerations involved in these tradeoffs are to some extent captured by familiar matching problems from mathematical literature. Thus several natural questions of the following form arise: how well does maximization of one objective (such as inventory utilization) serve another (such as fairness)? How can one objective be generalized to include another? And when one objective (such as fairness or popularity) does not have a unique maximum, how do the different maxima compare under another objective? In this paper, we consider a range of such issues, identify several interesting questions, and (partly) answer many of them.

Formally speaking, a rental service repeatedly computes a matching between the two sides of the market (i.e., customers and items), given the preference lists of one (and only one) side of the market. This type of matching markets are called *one-sided matching markets* as only one side of the market has preference over the other side. This is contrast to *two-sided matching markets* in which both sides of the market have preferences over the other side.

The preferences of customers are often *ordinal* in that they only explain the relative ranking of the items for individual customers. As noted above, optimality in allocation is not clearly defined as two matchings only based on the ordinal preferences of customers may not be comparable. This nuance underlies several notions of one-sided matching objectives studied in recent literature, such as *pareto-optimal matchings* [1], *fair matchings* [15], *rank-maximal matchings* [11], and so forth. In this context, we examine different measures that may be used to choose between non-comparable matchings and analyze different one-sided matching algorithms in terms of these measures.

Two leading criteria to measure the allocation performance of the rental service are that of *social welfare* and the *fairness* of allocation. In this following, we consider different algorithms for a single one-sided matching seeking a reasonable social welfare and fairness and compare the *value* of the output of these algorithms. For this purpose we require the measures of the *value* of the allocations, which must be defined in respect to the preference lists of customers to capture the social welfare and the fairness of the output. Under different measurese, we analyze the value of the output of different one-sided matchings for a single assignment; and then extend the results to repeated matchings for the rental market problem. Our metric to measure these matching algorithms is similar to that of the competitive analysis. That is, we study the ratio of the value of the matchings resulting from the one-sided matching algorithms over the value of the optimal matching.

1.1 One-Sided Matching Markets

Consider the following classical *one-sided matching problem*: we are given a set A of m customers and a set B of n items with one copy[1] for each $j \in B$. Each customer $i \in A$ has a preference list $\mathcal{L}_i = (b_i^1, \ldots, b_i^{\ell_i})$ over different items, where $\ell_i = |\mathcal{L}_i|$ and $b_i^j \in B$ for all $1 \le j \le \ell_i$. In a *matching*, items are assigned to customers so that each customer i gets at most one item and each item j is assigned to at most one customer. Since the vertices of one (and only one) side of the corresponding bipartite graph has a preference list, we call the matching in this setting *one-sided matching*.

Roughly speaking, our goal is to assign the customers to items which are among the top of their lists. More formally, let us consider associating a value $v(i,j)$ for assigning item j to customer i, and let our general goal be to find one-sided matchings to maximize the total value of the assignments in terms of the given valuations. We denote such valuations on the items to customers by $\boldsymbol{v}$.

This valuation function, however, should have some desired properties. The first natural property is that the function should be non-decreasing, i.e., $v(i, b_i^1) \ge v(i, b_i^2) \ge \cdots \ge v(i, b_i^{\ell_i}) > 0$ and $v(i,j) = 0$ for all other items j that are not on the list $\mathcal{L}_i$, where equality only reflects ties among items. Secondly, the customers tend to have stronger preference over the top choices in their preference list. We can model this fact by considering the concave valuation functions, i.e., $v(i, b_i^j) - v(i, b_i^{j+1}) \ge v(i, b_i^{j+1}) - v(i, b_i^{j+2})$, for $1 \le j < \ell_i - 1$. Moreover, in the valuation function, we would not want to "favor" any customer too much. For simplicity, let us say we would like to give the same value to the first choices of all customers and the same value to the second choices of all customers and so on. We call such functions satisfying the above conditions by the *universal ranking valuation functions*.

In particular, we are interested in the following special universal ranking valuation functions: for each customer i, the value of her jth-ranked item is $(n-j+1)^k$ (i.e., $v(i, b_i^j) = (n-j+1)^k$), for all $1 \le i \le n$, $1 \le j \le \ell_i$, and some fixed constant $k \ge 0$. We denote this valuation vector by $\boldsymbol{x^k}$. Note that when $k = 0$, this valuation function models the cardinality of the matching (i.e., $v(i, b_i^j) = 1$).

In this paper, we consider and analyze several one-sided matching frameworks that are listed below:

Maximum Weighted Matching. As described above, in the maximum weighted matching, we associate a valuation vector $\boldsymbol{v}$ to items and customers and maximize the total value of the one-sided matching. The maximum weighted matching associated with valuation vector $\boldsymbol{v}$ is denoted by $\mathsf{MaxWeightMatch}(\boldsymbol{v})$.

Rank-Maximal Matching. The *profile* of a one-sided matching M is a vector, where the jth element of the profile is the number of customers allocated to their jth-ranked item by M. M is *rank-maximal* if it has the lexicographically maximum profile. The rank-maximal matching M is denoted

[1] Note that all our results in the paper apply to the multiple copies case.

by RankMaxMatch. This solution concept for one-sided matching has been suggested by Irving [12] and later explored by Irving et al. [11].

Weighted Rank-Maximal Matching. Given a valuation vector $\boldsymbol{v}$, the *weighted profile* of a one-sided matching M for $\boldsymbol{v}$ is a vector, where the jth element of the profile is the value of the jth largest value among the values of the pairs of M. M is *weighted rank-maximal* for vector $\boldsymbol{v}$ if it has the lexicographically maximum weighted profile for the value vector $\boldsymbol{v}$. A weighted rank-maximal matching is denoted by WeightRankMaxMatch.

Fair Matching. A fair matching has the fewest number of unmatched customers (i.e., it has maximum-cardinality), and subject to this, matches the fewest number of customers to their nth-ranked item, and subject to this, matches the fewest number of customers to their $(n-1)$th-ranked item, and so on. (This definition can be formalized in terms of lexicographically-minimum reverse profiles, where we pad each customer's preference list with dummy items). The fair matching is denoted by FairMatch. Mehlhorn and Michail [15] considered this solution concept for the one-sided matching problems.

Order-Based Matching. Consider an arbitrary ordering $\pi : A \rightarrow \{1, \ldots, m\}$ of customers, the order-based matching algorithm for the ordering π goes over the list of customers according to π and for each customer i, it assigns the first available item on i's preference list to i. This algorithm is very simple and scalable to implement. Moreover, in order to achieve different goals in the assignment, we could change the ordering of customers. For example, in order to favor the new customers or the more profitable customers, we can put them at the beginning of the ordering. A matching resulted from the order-based matching algorithm for the ordering π is denoted by $\mathsf{OrderMatch}(\pi)$.

Note that the ordering of customers may differ at different time steps and may depend on the allocations of the previous time steps. For example, in order to achieve some fairness properties, we can favor the customers who did not get their first choices recently in the ordering and put them at the beginning of the ordering.

Stable Matching. Stable matchings are the well-known solution concepts for two-sided matching problems. In a two-sided matching problem, both sides have preference lists over the elements of the other side. In order to extend our setting from a one-sided matching to a two-sided matching problem, we need to define a preference list over customers for each item. We define the preference list for an item j by first listing the customers who have item j as their first choice in an arbitrary order, then listing all customers who have item j as their second choice, and so on. By defining these preference lists for items, we can apply the stable matching algorithm on the two-sided matching instance and output the resulting assignment. This matching is denoted by StableMatch.

The above solutions are the main algorithms that we study in this paper.

1.2 The Rental Market Problem

Rental markets seek to compute one-sided matching, of course, but they also have a time dimension. Roughly speaking, rental markets are frameworks for repeated one-sided matching. More formally, let us say that we are given a set B of n items and a set A of m customers with preference lists $\mathcal{L}_i$ over items. In the *rental market problem*, we need to assign a matching of items to customers at each discrete time step $t = 1, 2, \ldots, T$, where T is the common deadline. We assume that customers use the items for one time step and items can be reused after that. Besides the requirements that at each time, one customer can be assigned at most one item and one item can be assigned to at most one customer, the other requirement in the rental market problem is that one item can be assigned to one customer at most once.

We associate a value $v_t(i, j)$ for assigning item j to customer i at time step t. Our goal in the rental market problem is to find a set of matchings for all time steps to maximize the total value. We consider three different types of valuations: the static, online and the dynamic valuations. Roughly speaking, in the *static valuation model*, the value $v_t(i, j)$ is determined at the beginning $t = 1$, whereas in the *dynamic valuation model*, $v_t(i, j)$ directly depends on the position of j on i's preference list at time step t (i.e., it may change according to the assignments of previous steps). On the other hand, in the *online valuation model*, customers can update their preference lists (add new items or remove available items). We will elaborate the details of these models in Section 3.

1.3 Our Contribution

In this paper, we formalize the rental market problem as a repeated one-sided matching problem. We propose some value functions to measure the performance of the assignments in the rental market problem to capture the fairness and the social value of the output of different algorithms. We analyze several one-sided matching algorithms and give (almost) tight bounds on the performance in terms of those value functions for a single one-sided matching problem. These bounds are summarized in Table 1.

Then, we formalize the rental market in three models: The static valuation setting, online valuation setting and the dynamic valuation setting. In the static valuation setting, we show that there exists a 2-approximation algorithm by a reduction from the problem to the weighted 3-dimensional matching problem. As a hardness result, we prove the APX-hardness of the rental market problem. For the online valuation model, we derive a 2-competitive online algorithm to maximize the total value of the assignments. For the dynamic valuation model, we observe that the problems are similar to general variants of the job shop scheduling problems. As a result, we get a constant-factor approximation for the problem of minimizing the number of time steps to satisfy all the demand where at each step, we are allowed to assign one of the few top choices of each customer to her. We also give a hardness result of approximation for this model.

Finally, we give a description of our discrete event simulator for measuring the performance of most of these algorithms on a sample data (and will report some practical evaluations of our algorithms). We conclude the paper with some directions and open problems in the last section.

1.4 Related Work

As mentioned earlier, stable matchings are extensively studied as a solution concept for two-sided markets in which both sides of the market have preferences over the other side [5,7]. For one-sided matchings, Irving [12] introduced the concept of the rank maximal matchings and observed that they can be found by computing the maximum weighted matching in an edge-weighted bipartite graph where the edge weights are exponentially decreasing with respect to the preferences. Irving et al. [11] derived an algorithm with the running time $O(nm2\sqrt{m+n})$ for this problem. Mehlhorn and Michail [15] studied fair matchings and gave some efficient algorithms to find them. To the best of our knowledge, none of the above work analyzed the value of these one-sided matching algorithms for their worst-case performance. The only related paper in this regard is by Abraham et al. [1] in which the authors studied the structure of pareto-optimal solutions and pareto-optimality of some of the one-sided matching algorithms.

1.5 Notations

Recall that for a given valuation vector $\boldsymbol{v}$, $\mathsf{MaxWeightMatch}(\boldsymbol{v})$ denotes the one-sided matching with the maximum value. If the value of all items in the preference list is one, i.e., $v(i, b_i^j) = 1$ for all $1 \le i \le n$ and $1 \le j \le \ell_i$, the maximum-value one-sided matching is indeed the maximum cardinality matching and denoted by $\mathsf{MaxCardMatch} = \mathsf{MaxWeightMatch}(\mathbf{1})$. In addition, the value of a one-sided matching M for the valuation vector $\boldsymbol{v}$ is denoted by $\mathsf{Val}(M, \boldsymbol{v})$, and the cardinality of a one-sided matching M is denoted by $\mathsf{Card}(M)$.

2 Single Matching Algorithms

To understand the performance of different one-sided matching algorithms in the rental market problem, we need to define some universal objective functions to evaluate these matching algorithms. In particular, we evaluate the performance of a one-sided matching in terms of the value function over the pairs of customers and items. As discussed in the Introduction, we are interested in the special universal ranking valuation functions $\boldsymbol{x^k}$, for some fixed constant $k \ge 0$, where for each customer i, the value of its jth-ranked item is $(n-j+1)^k$ (i.e., $v(i, b_i^j) = (n-j+1)^k$), for all $1 \le i \le n$ and $1 \le j \le \ell_i$.

In this section, we prove several bounds on the ratio of the value of our proposed algorithms by the worst-case analysis. We summarize the results of this section in Table 1.

Table 1. The performance of one-sided matching algorithms

	Approximation factor				Pareto-optimal	Running time
	Card	Val($\boldsymbol{x}$)	Val($\boldsymbol{x^4}$)	Val($\boldsymbol{x^k}$)		
MaxCardMatch	1^*	ϵ^*	ϵ^*	ϵ^*	Exist	$O(e\sqrt{v})$ [9]
MaxWeightMatch($\boldsymbol{x}$)	$0.5, \frac{2}{3}$	1^*	$0.2, 1$	-	Yes	$O(e\sqrt{v}\log v)$ [4]
MaxWeightMatch($\boldsymbol{x^4}$)	$0.5, \frac{4}{7}$	$\frac{n+1}{4n}, 1$	1^*	-	Yes	$O(e\sqrt{v}\log v)$ [4]
MaxWeightMatch($\boldsymbol{x^{k'}}$)	0.5^*	$\frac{n+1}{4n}, 1$	$0.1, 1$	-	Yes	$O(ek'\sqrt{v}\log v)$ [15]
RankMaxMatch	0.5^*	0.5^*	0.5^*	0.5^*	Yes	$O(ev)$ [11]
WeightRankMaxMatch	0.5^*	0.5^*	0.5^*	0.5^*	Yes	$O(ev)$ [11]
FairMatch	1^*	$\frac{n+1}{2n}^*$	-	ϵ^*	Yes	$O(e\sqrt{v}\log v)$ [15]
OrderMatch	0.5^*	0.5^*	0.5^*	0.5^*	Exist	$O(e+v)$
StableMatch	0.5^*	0.5^*	0.5^*	0.5^*	Yes	$O(e+v)$ [5]

Notes: (i) The star symbol ($*$) implies that the ratio is (almost) tight.
(ii) Two numbers a, b implies the best known lower and upper bound.
(iii) "ϵ" means that the ratio can be arbitrarily close to zero.
(iv) In the running time, $v = \max\{m, n\}$ and $e = \sum_{i=1}^{m} \ell_i$.

2.1 Approximation Factor: Lower and Upper Bounds

Due to space limit, we leave the discussions of the tight bounds for cardinality of the maximum weighted matching, order-based matching, stable matching, and (weighted) rank-maximal matching to the full version. In the following discussions, we consider the bounds for the fair and maximum weighted matching. To prove our bounds, We first establish the following two lemmas.

Lemma 1. *Let M be either a* FairMatch *or a* MaxWeightMatch *w.r.t valuation function $\boldsymbol{x}$, and $|M| = \ell$. For any w, where $n - \ell + 1 \leq w \leq n$, we have*

$$|\{(a_i, b_i) \in M \mid v(a_i, b_i) \leq w\}| \leq w - n + \ell.$$

Basically, the lemma says that in the FairMatch or MaxWeightMatch M w.r.t valuation function $\boldsymbol{x}$, there is at most one edge with value smaller than or equal to $n - \ell + 1$, at most two edges with value smaller than or equal to $n - \ell + 2$, and so on.

For any constant $k \geq 1$, we can show similarly the following result.

Lemma 2. *For any constant $k \geq 1$, let M be a* MaxWeightMatch($\boldsymbol{x^k}$) *and $|M| = \ell$. For any w, where $n - \ell + 1 \leq w \leq n$, we have*

$$\left|\{(a_i, b_i) \in M \mid v(a_i, b_i) \leq w^k\}\right| \leq w - n + \ell.$$

Note that any FairMatch first try to minimize the number of unmatched customers, thus it's essentially a MaxCardMatch. To compare the FairMatch with MaxWeightMatch($\boldsymbol{x^k}$), we will first give an example to show the upper bound for any $k \geq 1$, and then study the lower bound for the case of $k = 1$.

Assume there are n customers $(a_1, \ldots, a_n)$ and n items $(b_1, \ldots, b_n)$. Each customer a_i prefers items $b_i, b_{i+1}, \ldots, b_n$ on her list. For $i = 1, \ldots, n-1$, customer a_i puts b_i the last choice and b_{i+1} the first choice on her list, respectively. All other items on the list can be ranked arbitrarily. Thus, $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ is a $\mathsf{FairMatch}$ with total value $1^k + 2^k + \cdots + n^k$. The $\mathsf{MaxWeightMatch}$ is $\{(a_1, b_2), \ldots, (a_{n-1}, b_n)\}$ with total value $(n-1)n^k$. Note that when $k = 1$, the ratio is $\frac{n+1}{2n-2}$; when $k = 4$, the ratio approaches to $1/5$ when n goes to infinity; and when n and k are sufficiently large, the ratio can be arbitrarily close to zero.

Now let's consider the $\mathsf{FairMatch}$ and $\mathsf{MaxWeightMatch}(\boldsymbol{x})$. Assume M is a $\mathsf{MaxWeightMatch}(\boldsymbol{x})$ and $|M| = \ell$. Note that $\mathsf{Val}(M) \leq \ell n$. Let M^* be a $\mathsf{FairMatch}$. Since the $\mathsf{FairMatch}$ is also a $\mathsf{MaxCardMatch}$, we know $|M^*| \geq \ell$. Due to Lemma 1, we know $\mathsf{Val}(M^*) \geq \sum_{w=n-\ell+1}^{n} w$. Thus,

$$\frac{\mathsf{Val}(M^*)}{\mathsf{Val}(M)} \geq \frac{(n-\ell+1)+\cdots+n}{\ell \cdot n} = \frac{2n-\ell+1}{2n} \geq \frac{n+1}{2n}$$

We conclude the above analysis as the following proposition.

Proposition 1. *For the universal ranking valuation function $\boldsymbol{x}$, we have*

$$\mathsf{Val}(\mathsf{FairMatch}, \boldsymbol{x}) \geq \frac{n+1}{2n} \cdot \mathsf{Val}(\mathsf{MaxWeightMatch}(\boldsymbol{x}))$$

Finally, we give some bounds for the value of the maximum weighted matching algorithms with valuation functions $\boldsymbol{x}$ and $\boldsymbol{x^4}$. We first consider the valuation function $\boldsymbol{x}$. Let M be a $\mathsf{MaxWeightMatch}(\boldsymbol{x})$ where $|M| = \ell$. Due to Lemma 1, we know that $\mathsf{Val}\left(M, \boldsymbol{x^4}\right) \geq \sum_{w=n-\ell+1}^{n} w^4$. On the other hand, consider the $\mathsf{MaxWeightMatch}(\boldsymbol{x^4})$ M^*, note that $\mathsf{Val}(M^*, \boldsymbol{x}) \leq \mathsf{Val}(M, \boldsymbol{x}) \leq \ell n$, which implies that $\mathsf{Val}\left(M^*, \boldsymbol{x^4}\right) \leq \ell n^4$. Thus,

$$\begin{aligned}\frac{\mathsf{Val}\left(M, \boldsymbol{x^4}\right)}{\mathsf{Val}\left(M^*, \boldsymbol{x^4}\right)} &\geq \frac{\sum_{w=n-\ell+1}^{n} w^4}{\ell \cdot n^4} \\ &\geq \frac{\sum_{w=1}^{n} w^4}{n \cdot n^4} \\ &= \frac{1/30 \cdot n(n+1)(2n+1)(3n^2+3n-1)}{n \cdot n^4} \\ &\geq 1/5\end{aligned}$$

Proposition 2. $\mathsf{Val}(\mathsf{MaxWeightMatch}(\boldsymbol{x}), \boldsymbol{x^4}) \geq 1/5 \cdot \mathsf{Val}(\mathsf{MaxWeightMatch}$ $(\boldsymbol{x^4}), \boldsymbol{x^4})$.

On the other hand, Let M be a $\mathsf{MaxWeightMatch}(\boldsymbol{x^k})$ where $|M| = \ell$, for any constant $k > 1$. Due to Lemma 2, we know that $\mathsf{Val}\left(M, \boldsymbol{x^4}\right) \geq \sum_{w=n-\ell+1}^{n} w^4$. Consider the $\mathsf{MaxWeightMatch}(\boldsymbol{x^4})$ M^*, it is easy to see that $|M^*| \leq 2|M| = 2\ell$, which implies that $\mathsf{Val}\left(M^*, \boldsymbol{x^4}\right) \leq 2\ell n^4$. Thus,

$$\frac{\mathsf{Val}\left(M, \boldsymbol{x^4}\right)}{\mathsf{Val}\left(M^*, \boldsymbol{x^4}\right)} \geq \frac{\sum_{w=n-\ell+1}^{n} w^4}{2\ell \cdot n^4} \geq 1/10$$

Proposition 3. $\mathsf{Val}\left(\mathsf{MaxWeightMatch}(\boldsymbol{x^k}), \boldsymbol{x^4}\right) \geq 0.1 \cdot \mathsf{Val}\left(\mathsf{MaxWeightMatch}(\boldsymbol{x^4}), \boldsymbol{x^4}\right)$, *for any* $k > 1$.

Consider another case: For any constant $k \geq 1$, let M^* be a $\mathsf{MaxWeightMatch}(\boldsymbol{x^k})$ where $|M^*| = \ell$. Due to Lemma 2, we know $\mathsf{Val}(M^*, \boldsymbol{x}) \geq \sum_{w=n-\ell+1}^{n} w$. Let M be a $\mathsf{MaxWeightMatch}(\boldsymbol{x})$. Again, note that $|M| \leq 2|M^*| = 2\ell$, thus, $\mathsf{Val}(M, \boldsymbol{x}) \leq 2\ell n$. Therefore,

$$\frac{\mathsf{Val}\left(M^*, \boldsymbol{x}\right)}{\mathsf{Val}\left(M, \boldsymbol{x}\right)} \geq \frac{\sum_{w=n-\ell+1}^{n} w}{2\ell \cdot n} = \frac{\ell(2n - \ell + 1)}{4\ell \cdot n} \geq \frac{n+1}{4n}$$

Proposition 4. *For any* $k \geq 1$,

$$\mathsf{Val}\left(\mathsf{MaxWeightMatch}(\boldsymbol{x^k}), \boldsymbol{x}\right) \geq \frac{n+1}{4n} \cdot \mathsf{Val}\left(\mathsf{MaxWeightMatch}(\boldsymbol{x}), \boldsymbol{x}\right)$$

2.2 Pareto-optimality

We say an allocation of items to customers is *Pareto-optimal* if there is no other allocation with some customers better and no one worse.

Proposition 5. *There is a* MaxCardMatch *that is Pareto-optimal.*

Proposition 6. *For any universal ranking valuation function* $\boldsymbol{v}$*,* $\mathsf{MaxWeightMatch}(\boldsymbol{v})$*,* RankMaxMatch*,* WeightRankMaxMatch*,* FairMatch*,* OrderMatch*, and* StableMatch *are Pareto-optimal.*

Note that for the OrderMatch with ties, we can show similar to Proposition 5 that there exists a Pareto-optimal solution. But in general, the OrderMatch is not Pareto-optimal. However, if it is not allowed to have ties, the OrderMatch guarantees Pareto-optimal.

3 The Rental Market Problem

In this section, we study the rental market problem with a focus on static, online, and dynamic valuations, respectively.

3.1 Static Valuation Model

In the following discussion, we study a static valuation model to evaluate the performance of the rental market problem. In the *static valuations model*, at each time t, $t = 1, \ldots, T$, where T is the deadline that customers can get the items, there is a valuation $v_t(i, j)$, which is determined at the beginning, associated with the pair (i, j), for any $(i, j) \in A \times B$, representing the valuation of the customer $i \in A$ for item $j \in B$ at time step t. Our goal is to select one-sided matchings M_t for each time t that maximizes $\sum_{t=1}^{T} \sum_{e \in M_t} v_t(e)$, given the condition that every pair is selected at most once, that is, each item can be assigned to each customer at most once. We denote the rental market problem in the static valuation model by StaticRentMark.

We reduce StaticRentMark problem with arbitrary valuation functions to the weighted 3-dimensional matching problem (W3DM). This implies a local search 2-approximation algorithm for this problem. In an instance of W3DM, given a subset D of triples in set $X \times Y \times Z$ where X, Y, and Z are disjoint sets, and a weight w_e for each triple of D, we need to find a set of triples $C \subseteq D$ with the maximum weight such that no two elements of C agree in any coordinate. W3DM is APX-complete [13] and a local search two-approximation algorithm is known for it [2].

Theorem 1. *For any static valuation function $\boldsymbol{v}$, there exists a 2-approximation algorithm for the* StaticRentMark *problem.*

Proof. Given an instance $\mathcal{S}(A, B; T, \boldsymbol{v})$ of the StaticRentMark problem, where T is the deadline time and $\boldsymbol{v}$ is the valuation function, we construct an instance $\mathcal{G}(\mathcal{S})$ of W3DM as follows: Let $[T] = \{1, \ldots, T\}$. Define $X = (A \times B)$, $Y = (A \times [T])$ and $Z = (B \times [T])$. For any triple $e = ((i, j), (i', t), (j', t')) \in X \times Y \times Z$, define the weight

$$w_e = \begin{cases} v_t(i, j) & \text{if } i = i', j = j', t = t' \\ 0 & \text{otherwise} \end{cases}$$

Now, it is easy to check that there exists a set of T one-sided matchings as the solution to the instance $\mathcal{S}$ with the total value w, if and only if, there exists a weighted 3-dimensional matching of weight w in the instance $\mathcal{G}(\mathcal{S})$. As a result, we can use the local search two-approximation algorithm of Arkin and Hassin [2] to achieve a two-approximation algorithm for StaticRentMark with any valuation function. □

Next, we complement this result by showing that StaticRentMark with arbitrary valuation functions is APX-Hard.

Theorem 2. *The* StaticRentMark *problem is APX-hard.*

3.2 Online Valuation Model

The 2-approximation algorithm for StaticRentMark above is based on a local search algorithm and does not provide an online algorithm. In this section, we consider a *online valuation model* in which customers can arbitrarily update their preference lists (add new items or remove available items[2]) for every time step, and at time t, we only know the preference lists and the corresponding valuations till this time. We denote this problem by OnlineRentMark.

Similar to the above models, our goal in OnlineRentMark is to assign items to customers at each time step to maximize the total value of assignments. In order

[2] For a more general setting in which users can change their preferences arbitrarily, we cannot hope to get any bounded competitive ratio.

to evaluate the performance of the online the algorithm, we follow the approach of the competitive analysis that compares the efficiency of the solution with a global optimum (assuming the knowledge of the future in advance).

Before studying the online algorithm for OnlineRentMark, we need to specify the valuation functions. In general, the valuations of customers over items decrease as time passes by. Therefore, it is reasonable to assume *non-increasing valuation functions*, i.e., $v_{t'}(i,j) \geq v_t(i,j)$, for any $1 \leq t' < t \leq T$, when item j is on customer i's list at both time t' and t.

A natural greedy strategy is that at each time t, we compute the current maximum weighted matching in terms of the available pairs and valuations at time t, and allocate the items to customers according to that matching. As the following theorem shows, this greedy algorithm has a good competitive ratio.

Theorem 3. *For any non-increasing valuation function $\boldsymbol{v}$, the above greedy algorithm gives a 2-competitive algorithm to the* OnlineRentMark *problem.*

Proof. Let OPT_t be the set of pairs selected by the optimal solution at time step t, and ALG_t be the set of pairs selected by the greedy online algorithm at time t. Let

$$OPT^* = \sum_{t=1}^{T} \sum_{e \in OPT_t} v_t(e)$$

be the total value of the optimal offline solution, and

$$ALG^* = \sum_{t=1}^{T} \sum_{e \in ALG_t} v_t(e)$$

be the total value of the greedy online algorithm. Let

$$X_t = OPT_t \cap \left(\bigcup_{i=1}^{t} ALG_i \right).$$

That is, X_t is the set of selected pairs in the optimal solution at time t that appear in the greedy online algorithm no later than time step t. Let $Y_t = OPT_t - X_t$.

For any $e = (i,j) \in X_t$, assume $e \in ALG_{t'}$, where $t' \in \{1, \ldots, t\}$. Note that item j appears on customer i's list at both time t' and t. Due to the non-increasing property, we have $v_t(e) \leq v_{t'}(e)$. Therefore,

$$\sum_{t=1}^{T} \sum_{e \in X_t} v_t(e) \leq \sum_{t=1}^{T} \sum_{e \in ALG_t} v_t(e) = ALG^*.$$

For the set Y_t, note that all pairs in Y_t are available in the greedy online algorithm at time t. Thus, Y_t is a feasible candidate set of the greedy algorithm. Due to the maximum weighted matching strategy, we have

$$\sum_{e \in Y_t} v_t(e) \leq \sum_{e \in ALG_t} v_t(e),$$

for any t. Thus, we have

$$\sum_{t=1}^{T}\sum_{e\in Y_t} v_t(e) \le \sum_{t=1}^{T}\sum_{e\in ALG_t} v_t(e) = ALG^*.$$

Therefore,

$$OPT^* = \sum_{t=1}^{T}\sum_{e\in X_t} v_t(e) + \sum_{t=1}^{T}\sum_{e\in Y_t} v_t(e) \le 2ALG^*,$$

which completes the proof of the theorem. □

Another advantage of the above greedy online algorithm is that it can be easily modified by changing the one-sided matching algorithm at each time step to get another competitive algorithm which may satisfy some extra desirable properties. We can combine the proofs of Section 2 and the above proof to bound the efficiency of the assignments resulting from using one of the aforementioned one-sided matching algorithms at each time step. For example, we could run a stable matching algorithm to find an assignment at each time step. The resulting algorithm is thus a $\frac{1}{4}$-competitive online algorithm for any non-increasing universal ranking valuation function.

3.3 Dynamic Valuation Model

A drawback of StaticRentMark and OnlineRentMark is that it ignores the effect of allocations in the previous time steps on the valuations of later time steps (OnlineRentMark essentially reflects the perspectives and changes of customers, but not allocations). We illustrate this by the following example. Let the preference list of customer i be (b_1, b_2, b_3). If we assign items b_1, b_2, and b_3 to customer i at the first three time steps, respectively, we have assigned the first choice of i to her every time step. In other words, the value of assigning b_2 to i at time step 2 for customer i is larger if item b_1 is assigned to i at time step 1. To capture this aspect, we formalize the rental market problem with dynamic valuations, denoted by DynamicRentMark, as follows: Let $r_t(i,j)$ be the jth-ranked item on customer i's preference list at time t. For every time step t, $t = 1, \ldots, T$, the value of assigning $r_t(i,j)$ to customer i at time t is $g(i,j)$.

The main difference between DynamicRentMark and the other two models is that the value of assigning an item in DynamicRentMark only depends on the position of the item on the preference list of the customer at the time of the assignment (that is, $r_t(i,j)$ is a dynamic function in terms of the previous allocation), but in StaticRentMark and OnlineRentMark, the value depends on the time step of the assignment and not directly on the position of the item at the time of the assignment[3].

[3] A more general model is that of the combination of OnlineRentMark and DynamicRentMark, where the value of assigning at item only depends on its current position on the list and customers can update their preference lists. We do not study this model in this paper.

First, we observe that a special case of the DynamicRentMark problem is the job-shop scheduling problem with unit-length jobs on parallel machines (JobShopSch) [14,17,6,18]. In the JobShopSch problem, we have a set of m jobs and n machines. Each machine can run at most one job at a time. Each job i consists of n_i operations o_j^i. Each operation o_i^j has a type t_i^j and can only be scheduled on machine with t_i^j. We need to schedule the operations of each job in the order $(o_i^1, o_i^2, \ldots, o_i^{n_i})$. There are two variations of this problem: In the minimization variant (MinJobShopSch [14]), we need to schedule all the operations of all jobs in the minimum number of time steps, that is we need to minimize the makespan of the schedule. In the maximization variant (MaxJobShopSch), we want to maximize the number (or the total value) of the operations that are scheduled before a deadline T. Constant-factor approximation algorithms are known for MinJobShopSch [14], but no constant-factor approximation algorithm is known for MaxJobShopSch.

JobShopSch is a special case of DynamicRentMark in which $g(i,1) = 1$ and $g(i,j) = -M$ for any $j > 1$ and sufficiently large value M. Each machine corresponds to an item in DynamicRentMark. Jobs in JobShopSch correspond to customers in DynamicRentMark and their operations correspond to the preference list of customers. As a result, for this value function, the known results for MinJobShopSch give a good approximation algorithm for the minimization version of DynamicRentMark. Moreover, designing a constant-factor approximation for the maximization version of DynamicRentMark will solve the open problem of approximating MaxJobShopSch.

Here, we formalize a more general value function and prove similar results for the DynamicRentMark problem. Consider the following dynamic value function: Given any constant $k \geq 1$, let $g(i,j) = 1$ for any $j \leq k$ and $g(i,j) = -M$ for any $j > k$ and a sufficiently large value M. In other words, at each step, we can assign only one of the first k choices of any customer to her. Our goal is to minimize the number of time steps for assigning all the items to customers (with the restriction of assigning only the first k choices). We call this problem MinDynamicRentMark(k). The MinJobShopSch problem corresponds to MinDynamicRentMark(1). We observe that the constant-factor approximation for MinJobShopSch can be used to give a constant-factor approximation for MinDynamicRentMark(k) for any $k \geq 1$.

Corollary 1. *For any constant $k \geq 1$, there exists a polynomial-time constant-factor approximation algorithm for* MinDynamicRentMark(k) *.*

In the following, we give a hardness result for MinDynamicRentMark(2).

Theorem 4. *It is NP-hard to approximate the* MinDynamicRentMark(2) *problem within a factor better than* 1.2.

4 Practical Evaluation

In this section, we describe our discrete event simulator and report the performance of different algorithms.

4.1 Discrete Event Simulator

DVD rental businesses are more complicated than the theoretical models we have analyzed here. For example, customers typically have a choice of *subscription plans*, which determine, say, how many DVDs they can borrow at once and in a given month. These plans have a big influence on the DVD *return times* - i.e. the time between when a DVD is borrowed and returned. This complicates matching decisions: should we allocate a DVD to a customer today, or wait until tomorrow when we may be able to give them a better DVD?

In our theoretical models, there is a fixed collection of DVDs available for rental. However, rental businesses have control over their inventory levels: if there aren't enough DVDs of a particular title, more can be purchased. Of course, with this control comes the problem of determining optimum inventory levels, which to some degree involves trading-off between customer satisfaction and financial sustainability. Inventory levels must also take into account return times. Customers with slower return times may keep a high-demand DVD for several days beyond watching it. Knowing this, additional copies of the DVD must be bought, even though only a fraction of the DVDs are actively being watched on a given day.

These are just some of the complications dealt with by a real-world DVD Rental business. We cannot hope to fully capture the underlying model and analyze it theoretically. Instead, we have built a discrete event simulator to see the effects of various subscription plans, matching algorithms, inventory planning strategies and so on. The simulator can be seeded by real-world data, including actual customer preference lists, distributions of return times and forecasts of demand. This simulator is used by the DVD Rental business unit at Amazon.com.

4.2 Performance of Different Algorithms

The following table contains some sample results from our simulator. We constructed a small instance from real-world data containing 2000 customers (with existing rental histories and preference lists), 150 DVD titles (5000 DVDs in total) and two types of subscription plans (one with a maximum of 4 DVDs per month, and at most 2 borrowed at any one time; the other with unlimited DVDs per month, and at most 3 borrowed at any one time). Using forecast demand data, we then ran the simulator for a (virtual) three-month period to test the different matching algorithms.

The three objective functions are $\mathsf{Val}(\boldsymbol{x})$, $\mathsf{Val}(\boldsymbol{x}^{\mathbf{4}})$ and the total number of *skips*. A skip occurs for each higher-ranked DVD a customer misses out on when we perform a matching. If an eligible customer receives no DVD, a skip is recorded for each DVD on his/her preference list. We report this value for different matching algorithms as an alternative measure to compare the results. Note that in Table 1, we report the worst case analysis for a single matching, but in Table 2, we report the total value of matchings for several time steps. As expected from the theoretical results in Table 1, the total value for different algorithms are close to each other.

Table 2. Simulation results

	Total skips	$\mathsf{Val}(\boldsymbol{x})$	$\mathsf{Val}(\boldsymbol{x^4})$
$\mathsf{MaxWeightMatch}(\boldsymbol{x})$	12,522	1,137,456	3.7921e+012
$\mathsf{MaxWeightMatch}(\boldsymbol{x^4})$	12,962	1,137,555	3.7963e+012
RankMaxMatch	15,802	1,139,654	3.8078e+012
FairMatch	12,316	1,135,668	3.7861e+012
OrderMatch	17,059	1,141,082	3.8148e+012
StableMatch	25,788	1,139,025	3.8126e+012

Although these objective functions capture the social welfare, they do not reveal the utility variability amongst the customers. Figure 1 shows the number of skips experienced by the 50 customers with the most number of skips. It is of interest to note that fair matching is substantially better for these customers. This is achieved with very little loss in utility w.r.t. $\mathsf{Val}(\boldsymbol{x})$ and $\mathsf{Val}(\boldsymbol{x^4})$.

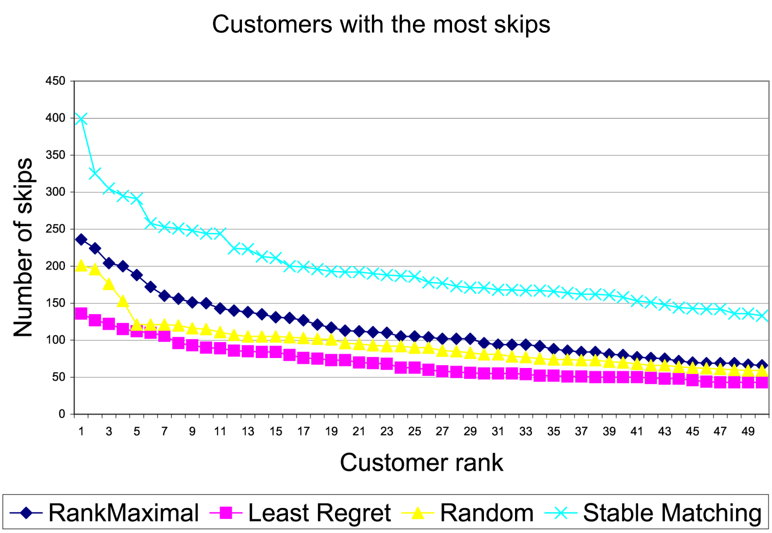


Fig. 1. The worst customer experience in each scenario

5 Conclusions

In this paper, we studied different algorithms for the rental market problem, defined universal measures to compare these algorithms, and analyzed them theoretically and practically. An open problem of this paper is to design a constant-factor approximation algorithm for the maximization version of

DynamicRentMark. Such a constant-factor approximation algorithm also gives a constant-factor approximation for MaxJobShopSch.

Designing algorithms with extra fairness properties is an interesting subject of study. For example, we would like to minimize the maximum number of skips that any customer observes. Dealing with strategic agents is another interesting topic. This can be done by proving that for random preference lists, the probability that a customer has incentive to lie tends to zero as the number of customers approaches to ∞.

References

1. D. Abraham, K. Cechlarov, D. F. Manlove and K. Mehlhorn. *Pareto Optimality in House-Allocation Problems*. ISAAC 2004, 3-15.
2. E. M. Arkin and R. Hassin, *On Local Search for Weighted Packing Problems*. Math. Oper. Res., V.10(3), 640-648, 1998.
3. D. Avis. *A Survey of Heuristics for the Weighted Matching Problem*. Networks, V.13, 475C493, 1983.
4. H. N. Gabow and R. E. Tarjan. *Faster Scaling Algorithms for Network Problems*. SIAM Journal on Computing, 18(5), 1013-1036, 1989
5. D. Gale and L. S. Shapley. *College Admissions and the Stability of Marriage*. American Mathematical Monthly, V.69, 9-15, 1962.
6. L. Goldberg, M. Paterson, A. Srinivasan and E. Sweedyk. *Better Approximation Guarantees for Job-Shop Scheduling*. SODA 1997, 599-608.
7. D. Gusfield and R. W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.
8. Holyer. *The NP-Completeness of Some Edge Partitioning Problems*. SIAM journal of Computing, V.10(3), 713-717, 1981.
9. J. E. Hopcroft and R. M. Karp. *An $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs*. SIAM J. Computing, V.4, 225-231, 1973.
10. N. Immorlica, M. Mahdian, and V. S. Mirrokni. *Cycle Cover with Short Cycles*. STACS 2005, 641-653.
11. R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail and K. Paluch. *Rank-Maximal Matchings*. SODA 2004, 68-75.
12. R. W. Irving. *Greedy Matchings*. Technical Report TR-2003-136, University of Glasgow, 2003.
13. V. Kann. *Maximum Bounded 3-Dimensional Matching is MAX SNP-Complete*. Inform. Process. Lett., V.37, 27-35, 1991.
14. F. T. Leighton, B. M. Maggs and S. B. Rao. *Packet Routing and Job-Shop Scheduling in O(Congestion +Dilation) Steps*. Combinatorica, V.14(2), 167-180, 1994.
15. K. Mehlhorn and D. Michail. *Network Problems with Non-Polynomial Weights and Applications*. Manuscript, 2005.
16. R. Preis. *Linear Time 1/2-Approximation Algorithm for Maximum Weighted Matching in General Graphs*. STACS 1999, 259-269.
17. D. B. Shmoys, C. Stein and J. Wein. *Improved Approximation Algorithms for Shop Scheduling Problems*. SIAM Journal of Computing, V.23, 617-632, 1994.
18. D. Williamson, L. Hall, J. Hoogeveen, C. Hurkens, J. Lenstra, S. Sevastianov and D. Shmoys. *Short Shop Schedules*. Operations Research, V.45(2), 288-294, 1997.

On Portfolio's Default-Risk-Adjusted Duration and Value: Model and Algorithm Based on Copulas*

Ping Li[1], Hou-Sheng Chen[1], Guang-Dong Huang[2], and Xiao-Jun Shi[1]

[1] School of Economics and Management, Beihang University,
Beijing 100083, P.R. China
liping124@buaa.edu.cn

[2] School of Information Sciences, China University of Geosciences (Beijing),
Beijing 100083, P.R. China

Abstract. In this paper, we propose a new approach, copulas, to calculating the default-risk-adjusted duration and present value for a portfolio of bonds vulnerable to default risk. A copula function is used to determine the default dependence structure and simulate correlated default time from individual obligor's default distribution. This approach is verified to be effective and applicable by a numerical example, in which we demonstrate how to calculate the default-risk-adjusted duration and present value for a given portfolio. In the process we take into account of the settlement time when default happens, the choice of copula function and the correlation between obligors, and make a sensitive analysis of the influence of Kendall's tau and copula functions on the default-risk-adjusted duration and present value. Results show that the duration and present value simulated from Gaussian copula fluctuates larger than that from Clayton and Gumbel copulas when Kendall's tau varies from zero to one.

1 Introduction

Technically, duration is the weighted-average time to maturity taking the relative present value of the cash flows as the weights. As such, a duration is thought of as a measure of sensitivity of an asset's value with respect to interest rate. Duration is a fundamental tool for banks and other financial institutions to manage their assets and liabilities.

Recently, there is a rapid growth in the market of defaultable corporate bonds, especially low-grade junk bonds, which prompts the study of the influence of default risk on the duration of default-prone bonds. Among the literatures on this topic, Jonkhart (1979) developed a formulation for the term structure of interest rates considering the influence of default risk on the duration of bonds that are not default-free. Bierwag and Kaufman (1988) obtained an expression for default-adjusted duration for various patterns of expected defaults assuming a flat term

* This work is supported by the National Natural Science Foundation of China, grant 70501003.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 214–224, 2006.
© Springer-Verlag Berlin Heidelberg 2006

structure. They showed that the timing of defaults, the size and pattern of the cash flow recovered from default are critical in modelling default-risk-adjusted duration. Also, Chance (1990) used a contingent claim approach to derive the duration for zero-coupon bonds subject to default risk. His work assumed that default triggers an immediate lump sum settlement. Fooladi et. al(1997) introduced risk aversion into Jonkhart's (1979) model using a structure of certainty equivalent (CE) factors, and derived a general default-adjusted duration model. Jacoby (2003) removed three limitations of the model developed by Fooladi et. al(1997), and obtained a model for the valuation of coupon-bearing corporate bonds. In his model, he assumed that the defaultable bond is equivalent to a portfolio of a riskless coupon bond and a short position in a series of European options on the firm's value.

But all of the above models only considered a single bond. For a bank or financial institution, they usually hold a portfolio of bonds or debts. To manage the portfolio using duration immunization strategy, a simple hypothesis is that all of the bonds or debts are independent. This assumption simplifies the calculation of portfolio's duration, but it is not consistent with the practice that the default risk among obligors are intensively related to others, especially in a down market. This dependence may be due to both of macroeconomic (the overall economy) and microeconomic (sectorial and even firm-specific) aspects. Thus the default dependence structure among obligors in a portfolio plays a crucial role in the quantification of the portfolio's duration, and the problem we care about is how to calibrate the default dependence structure?

Many existing literatures use liner correlation coefficient to define the default correlation. Lucas (1995) applied discrete event approach of liner correlation coefficient to measure the correlation between defaults. But because of the instinctive disadvantages of liner correlation coefficient, Lucas' method hasn't been extensively applied in finance.

Recently, a statistical tool, copula function, which has the distinctive charm of measuring dependence, has been gradually applied in finance. Some researchers attempted to characterize the default dependence structure by copulas. Li (2000) proposed to use Gaussian copula to capture the default dependence in the collateral portfolio, and developed a method for the pricing of multi-name credit derivatives. It may be considered as an extended version of the Credit Metrics framework. Schönbucher and Schubert (2001) presented a method to incorporate dynamic default dependence in intensity-based default risk models. They used a copula function to simulate default time combining with single obligor's intensity-based default model. Meneguzzo and Vecchiato (2004) adopted Li's method to price collateralized debt obligations and basket default swaps and made a sensitive analysis.

In this paper we provide a method based on copulas to calculate the duration and present value for a portfolio of bonds that are not default-free. Since the default process is different from each other, we follow each obligor's specification to determine their default probability distributions, then combine them by a copula function to obtain the joint default probability distribution. The copulas will also be used to simulate the default time of each obligor. In existing related literatures many researchers employ Gaussian copulas for its simplicity in sampling. Here we will also use Clayton and Gumbel copulas because of their properties of asymmetry and heavy tail which are more accordant with practice.

The rest of the paper is organized as follows. In next section, we offer a simple review of copulas including three common-used copulas, and then in Section 3 we introduce the model for marginal default distribution developed by Li (2000), based on which we propose the model for portfolio's joint-default-risk-adjusted duration and present value in Section 4. In the process we will consider the practice of discrete credit spread. Section 5 offers a numerical example to demonstrate the calculation of portfolio's joint-default-risk-adjusted duration and present value using three copulas and make sensitive analysis for different Kendall's taus. Section 6 concludes the paper and gives some suggestion for the application.

2 Copulas

A copula is a function that combines marginal distributions into a joint distribution. In what follows, we give the definition and some properties of copula functions, and three common-used copulas. Readers can refer to Nelsen (1999) for details.

2.1 Definition and Property

Definition 1. *An n-dimensional copula is a function $C : [0,1]^n \to [0,1]$, which satisfies the following three properties:*

(1) (Grounded) $C(u_1, \cdots, u_n) = 0$ if there is one coordinate $u_i = 0$ for $i = 1, 2, \cdots, n$;

2) $C(1, \cdots, 1, u_i, 1, \cdots, 1) = u_i, \quad \forall i = 1, \cdots, n$;

3) (Increasing) The C-volume of any box with vertices in $[0,1]$ is nonnegative.

The following Sklar theorem shows that a copula function is in fact the joint distribution of n $[0,1] - uniform$ random variables.

Theorem 1. *Let F be an n-dimensional distribution function with continuous margins $F_1, ..., F_n$, then there exists an n-dimensional copula function C satisfying*

$$F(x_1, ..., x_n) = C(F_1(x_1), ..., F_2(x_n)). \tag{1}$$

From Sklar theorem we see that, the joint distribution of multiple random variables can be represented by their marginal distributions and a copula function. Or from another point of view, a copula function represents the joint distribution function of n standard uniform random variables $U_1, U_2, ..., U_n$:

$$C(u_1, u_2, ..., u_n) = Pr(U_1 \leq u_1, U_2 \leq u_2, ..., U_n \leq u_n).$$

This provides an easy approach to obtaining the multi-dimensional joint distribution and dependence structure when marginal distributions and a copula function are given.

2.2 Measure of Dependence

Copulas are intimately related to standard measures of dependence between a pair of random variables X_1 and X_2 whose copula is C. Note that the traditional Pearson correlation coefficient only captures the linear dependence and is not invariant under non-linear strictly increasing transformations (see Embrechts et al 1999), but the dependence characterized by copulas can overcome these pitfalls. Here we only introduce Kendall's tau since it is one of the most important measures and will be used in this paper.

Definition 2. *Kendall's tau is the probability of concordance minus the probability of discordance, and can be expressed by copulas as follows*

$$\tau = 4 \int\int_{[0,1]^2} C(u_1, u_2) dC(u_1, u_2) - 1.$$

More properties of Kendall's tau and other measures of dependence see Nelsen (1999).

2.3 Three Common-Used Copulas

Here we only introduce three most common-used copulas which will be used in our numerical example.

(1) Gaussian copula

The n-variate Gaussian (or normal) copula is defined as follows:

$$C_{Ga}(u_1, ..., u_n) = \Phi_R(\phi^{-1}(u_1), ..., \phi^{-1}(u_n)),$$

where, Φ_R is the standardized $n-$variate normal distribution with correlation matrix R, ϕ^{-1} is the inverse of standard univariate normal distribution ϕ.

(2) Clayton copula

With $\psi(u) = u^{-\alpha} - 1,\ \alpha > 0$, being the generator, the n-variate Clayton copula is given by:

$$C_{Cl}(u_1, ..., u_n) = [\sum_{i=1}^{n} u_i^{-\alpha} - n + 1]^{-1/\alpha}$$

(3) Gumbel copula

With $\psi(t) = (-\ln t)^{\alpha},\ \alpha \geq 1$, being the generator, the n-variate Gumbel copula is given by:

$$C_{Gu}(u_1, ..., u_n) = \exp\{-[(-\ln u_1)^{\alpha} + (-\ln u_2)^{\alpha} +, ..., +(-\ln u_n)^{\alpha}]\}$$

Clayton copula and Gumbel copula are Archimedean copulas, and have a lot of good properties such as asymmetric density distributions, and hence will be used in our numerical example.

3 Modelling the Marginal Default Distribution

In the event of defaults, both of the timing and the amount of cash flow promised by obligor will be changed significantly, thus the duration and immunization strategy should be adjusted by default risk. In the process the default distribution of the specific obligor (or bond) plays an important role.

According to option pricing theory of Merton (1974), the value of a defaultable bond can be regarded as a portfolio of risk-free asset and a put option. Thus the default probability can be derived by Black-Scholes model. But this method ignores the non-tradeable value of a firm. Fooladi et al (1997) and the extended version of Jacoby (2003) indicated to calculate the expected cash flow at specific default time from a default probability, and determine the certainty equivalent by multiplying the expected cash flow with equivalent factor. Then the price of a dafaultable bond and the default-risk-adjusted duration can be derived. But both of the two articles did not interpret how to get the default probability.

On the other hand, in order to calculate the joint-default-risk-adjusted duration for a portfolio of defaultable bonds, the default dependence structure is of utmost importance. Neither of the above two models can be directly applied to multi-obligor case, while copulas can be employed.

Before we obtain the joint default probability distribution we first need to model the single obligor default distribution. Just as Li (2000) we use a random variable T to denote the time to default, or equivalently, the survival time of a security. $F(t)$ denotes the distribution of T, that is,

$$F(t) = Pr(T \leq t), t \geq 0.$$

Set

$$S(t) \widehat{=} 1 - F(t) = Pr(T > t),\ \ t \geq 0.$$

The function $S(t)$ is called the survival function, which denotes the probability that a security will survive t year(s). The probability density of $F(t)$ is given by:

$$f(t) = F'(t) = -S'(t)$$

From

$$Pr(x < T \leq x + \triangle x | T > x) = \frac{F(x + \triangle x) - F(x)}{1 - F(x)} \approx \frac{f(x)\triangle x}{1 - F(x)}$$

we can see that the function

$$h(x) \widehat{=} \frac{f(x)}{1 - F(x)} = -\frac{S'(x)}{S(x)} \tag{2}$$

represents the instantaneous default probability for a security that has attained age x. In this sense $h(x)$ is called the hazard rate function.

From (2) we can express the survival function $S(t)$ by hazard rate function $h(t)$ as follows:

$$S(t) = \exp\left\{-\int_0^t h(x)dx\right\}. \tag{3}$$

Thus the modelling of survival function is transformed to the modelling of hazard rate function. There are some other similar models for the default process. Lando (1998) modelled the default process by a Cox process, in which the hazard rate function $h(x)$ is a stochastic process; while Duffie and Garleanu (2001) applied a jump diffusion process to model the default risk, and $h(x)$ is dynamic.

There are three approaches to the modelling of hazard rate function $h(x)$: 1) Obtaining the historical default information directly from rating organizations; 2) Using Merton's option pricing approach (see Geske (1977), Delianedis and Geske (1998)); 3) Using the implicit approach implied from market observable information, such as asset swap spreads and credit spread (see Li, 1998). The former two approaches are seldom applied in practice because of their unsatisfactory effect and complexity. Alternatively, we will use the third approach which is common used to price credit derivatives or manage default risk. Specifically we assume that the hazard rate function is given by:

$$h(t) = \frac{S_t}{1 - R_t} \tag{4}$$

where S_t and R_t denote the credit spread and recovery rate at time t, respectively.

From (4) we can see that it is easy to obtain the hazard rate function $h(t)$ if the credit spread S_t and recovery rate R_t are given. S_t can be acquired from the bond price or credit default swap (CDS), while the recovery rate is determined by the specific obligor in the industry. For simplicity, we always assume that the recovery rate is constant in a period.

In general, we can not have observations of the hazard rate for all periods of time but only for some time points. So in practice, for each obligor's hazard rate we always consider a stepwise constant function using the observed values of $h_i(t_j)$, where subscript i and j denote obligor and time point, respectively. Then from (3) we can rewrite the continuous form of the default probability distribution $F_{T_i}(t)$ for the time to default T_i of obligor i as follows:

$$F_{T_i}(t) = 1 - \exp\left\{ -\sum_{j=1}^{k_i} h_i(t_j) \triangle t_j \right\}, \quad i = 1, \cdots, n, \tag{5}$$

where, $\triangle t_j = t_j - t_{j-1}$ and $k_i = 1$, if $T_i \leq t_1$; $k_i = 2$, if $t_1 \leq T_i \leq t_2$,..., $k_i = N$, if $T_i \geq t_{N-1}$.

After obtaining the marginal distribution, we can get the joint default distribution from a given copula function C. Then we can simulate the correlated time to default T_i from the joint distribution $C(T_1, T_2, ..., T_n)$ and further to calculate the joint-default-risk-adjusted duration and present value for a portfolio, which is the task of our next section.

4 Model and Algorithm for Joint-Default-Risk-Adjusted Duration and Portfolio Value

In last section we have got the default distribution $F_{T_i}(t)$ for each obligor $i = 1, \cdots, n$ by equation (5), now we will obtain the joint default distribution and the correlated time to default.

From Sklar Theorem, if C is a copula function, then

$$F(T_1, T_2, ..., T_n) = C(F_1(T_1), F_2(T_2), ..., F_n(T_n)) \tag{6}$$

is the joint default distribution of $T_1, T_2, ..., T_n$.

To simplify the simulation of T_i, we put

$$Y_1 = F_1(T_1),\ Y_2 = F_2(T_2)\ , ...,\ Y_n = F_n(T_n),$$

then (6) becomes

$$F(F_1^{-1}(Y_1), F_2^{-1}(Y_2), ..., F_n^{-1}(Y_n)) = C(Y_1, Y_2, ..., Y_n) \tag{7}$$

Thus we can get the correlated time to default $T_1, T_2, ..., T_n$ from the following simulation:

1) Simulate $Y_1, Y_2, ..., Y_n$ from a given copula function C and (7);
2) Obtain $T_1, T_2, ..., T_n$ from $T_1 = F_1^{-1}(Y_1), T_2 = F_2^{-1}(Y_2), ..., T_n = F_n^{-1}(Y_n)$.

From the simulated time to default, we can calculate the joint-default-risk-adjusted duration and present value for a portfolio as follows.

Suppose that a firm i has survived to present without default. The face value of this firm's debt is A_i, and the coupon rate is r_i. The payment is made at the end of each year, and the recovery rate is R_i. Once the debt defaults at time T_i before maturity, there will be a payment $R_i \times A_i$ at some time $T_i + S$, where S represents the number of years it takes to make final settlement for default.

From the fair pricing theory, the present value of the bond is

$$P_i = \sum_{j=1}^{k_i} \frac{A_i \times r_i}{(1 + I_j)^j} + \frac{A_i \times R_i}{(1 + I_{T_i+S})^{T_i+S}}, \quad i = 1, \cdots, n, \tag{8}$$

where, $k_i = [T_i]$ is the integer part of survival time T_i, I_j is the annual risk-free forward rate, measuring from time 0 to time j.

Employing the formula for duration calculation, we can obtain the duration for bond i which defaults at time T_i

$$D_i = \sum_{j=1}^{k_i} j \times \frac{A_i \times r_i}{P_i(1 + I_j)^j} + (T_i + S) \times \frac{A_i \times R_i}{P_i(1 + I_{T_i+S})^{T_i+S}}, \quad i = 1, \cdots, n. \tag{9}$$

Hence, the present value P and duration D for a portfolio can be easily obtained:

$$D = \sum_{i=1}^{n} w_i D_i, \quad P = \sum_{i=1}^{n} w_i P_i \tag{10}$$

where, w_i is the weight of bond i in the portfolio.

Since the time to default $T_1, T_2, ..., T_n$ is simulated from the joint distribution, which characterize the default dependence structure, the portfolio duration obtained from (10) is a joint-default-risk-adjusted duration. On the other hand, to implement the immunization strategy accurately, portfolio's present value also needs to be adjusted as above, because default risk will affect the value of the ingredient bonds.

5 Numerical Example

In this section, we give a numerical example to demonstrate the calculation of a portfolio's joint-default-risk-adjusted duration and present value by simulation. We only consider a simple case of two bonds (or debts).

Suppose that each bond has a maturity of ten years and face value of 1. The coupon rate and risk-free interest rate are both 10%. For simplicity, we assume that both of the two obligors have a recovery rate of 0.3 and have the same credit spread structure as presented in table 1.

Table 1. 3-, 5- 10-year credit spread for two obligors

	obligor 1	obligor 2
3-year credit spread	450	450
5-year credit spread	500	500
10-year credit spread	600	600

Assume further that the two obligors haven't experienced default up to now, and all payments are delivered at the end of each year. Consider a portfolio composed of the two bonds with an equal weight of 0.50. In practice, if an obligor defaults, the settlement cannot be done immediately. Therefore, we consider three circumstances of $S = 1, 2, 3$, which mean the recovery are made at the end of the first, the second and the third year after the year of default. Thus we do a little modification to the model for default-risk-adjusted duration D_i and present value P_i given in section 4:

$$P_i = \sum_{j=1}^{k_i} \frac{A_i \times r_i}{(1 + I_j)^j} + \frac{A_i \times R_i}{(1 + I_{[T_i]+S})^{[T_i]+S}}, \quad i = 1, \cdots, n, \tag{11}$$

$$D_i = \sum_{j=1}^{k_i} j \times \frac{A_i \times r_i}{P_i(1 + I_j)^j} + ([T_i] + S) \times \frac{A_i \times R_i}{P_i(1 + I_{[T_i]+S})^{[T_i]+S}}. \tag{12}$$

From the given credit spread, recovery rate, equation (4) and (5), we can obtain the default probability distribution $F_{T_i}(t)$ for each obligor i, and then simulate the time to default T_1 and T_2 from a copula function. Here we choose

Gaussian copula, Clayton copula and Gumbel copula to meet our demand, and select eleven values of Kendall's tau from 0 to 1 with increments of 0.1 for the correlation between the two obligors. We use Monte Carlo approach to simulate 10,000 times under different copula functions and different Kendall's taus to generate the time to default for two obligors, then calculate the joint-default-risk-adjusted duration and present value from formulas (10), (11) and (12) under different settlement time S. Results are reported in Table 2 and Table 3.

Table 2. Joint-default-risk-adjusted duration under different circumstances

	Clayton copula			Gumbel copula			Gaussian copula		
Kendall tau	$S = 0$	$S = 1$	$S = 2$	$S = 0$	$S = 1$	$S = 2$	$S = 0$	$S = 1$	$S = 2$
0	4.8714	5.0741	5.2605	4.8714	5.0741	5.2605	4.8426	5.047	5.2346
0.1	4.8719	5.0745	5.2608	4.8696	5.0724	5.2589	4.8439	5.0482	5.2357
0.2	4.8713	5.0742	5.2610	4.8735	5.0761	5.2626	4.8472	5.051	5.2381
0.3	4.8763	5.0781	5.2636	4.8727	5.0749	5.2608	4.8513	5.0541	5.2401
0.4	4.8759	5.0775	5.263	4.8726	5.0745	5.2601	4.8635	5.0641	5.2480
0.5	4.8722	5.0741	5.2597	4.8734	5.0752	5.2607	4.8884	5.0839	5.2629
0.6	4.8733	5.0748	5.2599	4.8745	5.0759	5.2609	4.9217	5.1109	5.2838
0.7	4.8697	5.0717	5.2573	4.8722	5.0738	5.2591	4.9675	5.1495	5.3156
0.8	4.8689	5.0710	5.2568	4.8731	5.0747	5.2600	5.0149	5.1903	5.3505
0.9	4.8680	5.0702	5.2561	4.8713	5.0729	5.2582	5.0319	5.2035	5.3601
1.0	4.8699	5.0717	5.2570	4.8661	5.0685	5.2546	4.9557	5.1309	5.2892

From Table 2 we can see that the adjusted duration increases moderately when the settlement is delayed, which is accordant with the intuition. The unadjusted Macaulay duration in our example is 6.6291 which is larger than that adjusted by default risk. When using Clayton and Gumbel copula, there is no evident trend for adjusted duration as Kendall's tau increases; but when using Gaussian copula the adjusted duration increases as Kendall's tau increases except that the two obligors are perfectly correlated. Adjusted durations calculated from Clayton and Gumbel copulas fluctuate little in each column when Kendall's tau changes, while durations calculated from Gaussian copula have a larger volatility as tau varies. For example, the differences between the largest and smallest adjusted durations in column 2 and column 5 are only 0.0083 and 0.0085, respectively, while the difference between the largest and smallest durations in column 8 is 0.1892.

Table 3 indicates that the default-risk-adjusted portfolio value decrease when the settlement is delayed, which is also consistent with our intuition. There are some similar characteristics between Table 2 and Table 3. The values obtained from Clayton and Gumbel copulas change irregularly and have little volatility

Table 3. Joint-default-risk-adjusted portfolio value under different circumstances

	Clayton copula			Gumbel copula			Gaussian copula		
Kendall tau	$S=0$	$S=1$	$S=2$	$S=0$	$S=1$	$S=2$	$S=0$	$S=1$	$S=2$
0	0.7164	0.7074	0.6992	0.7164	0.7074	0.7074	0.7118	0.7026	0.6942
0.1	0.7165	0.7074	0.6992	0.7159	0.7069	0.7069	0.7121	0.7029	0.6945
0.2	0.7164	0.7073	0.6991	0.7167	0.7077	0.7077	0.7126	0.7034	0.6950
0.3	0.7169	0.7079	0.6997	0.7163	0.7072	0.7072	0.7127	0.7035	0.6952
0.4	0.7169	0.7078	0.6996	0.7162	0.7071	0.7071	0.7138	0.7047	0.6964
0.5	0.7163	0.7073	0.6990	0.7164	0.7074	0.7074	0.7160	0.7069	0.6986
0.6	0.7164	0.7074	0.6992	0.7166	0.7076	0.7076	0.7188	0.7098	0.7016
0.7	0.7159	0.7068	0.6986	0.7162	0.7072	0.7072	0.7225	0.7136	0.7056
0.8	0.7157	0.7067	0.6984	0.7163	0.7073	0.7073	0.7249	0.7161	0.7081
0.9	0.7156	0.7065	0.6983	0.7160	0.7070	0.7070	0.7196	0.7105	0.7023
1.0	0.7157	0.7067	0.6984	0.7153	0.7062	0.7062	0.6903	0.6800	0.6707

when Kendall's tau changes, while values simulated from Gaussian copula in each column increase when Kendall's tau increases except for 0.9 and 1.

In summary, the settlement time, the recovery rate in event of default, the correlation among obligors and the copula function are four main factors need to be considered when calculating the duration and present value for a portfolio of bonds vulnerable to default risk. In our another paper (Li and Chen, 2005) we consider the effects of recovery rate on the price of Collateralized Debt Obligation (CDO). The idea and method in that paper can also be applied here, but in this paper we focus on the other three factors.

6 Conclusion

Ignoring default correlation among obligors when calculating the duration for a portfolio of bonds vulnerable to default risk may lead to an inefficient immunization strategy because there exists dependence among economic entities. But it is more difficult to model the joint default distribution than for single default case. In this paper, we employ copula functions to measure the default dependence structure among obligors, and use Monte Carlo approach to determine the time to default for each obligor in the portfolio, from which the joint-default-risk-adjusted durations and present value of the portfolio are obtained.

A numerical example indicates the difference between the joint-default-risk-adjusted duration and unadjusted duration. We take into consideration of the effects of the delay of settlement in event of defaults on portfolio's adjusted duration and present value. The choice of copula function is also crucial in simulation. In our example, we use Gaussian, Clayton and Gumbel copulas to characterize the dependence structure between obligors. We also make a sensitive analysis under different Kendall's taus. Results show that durations and values simulated from Gaussian copula fluctuate larger than that from Clayton and Gumbel copulas.

Our approach is applicable in monitoring default risk related to bank loans, debts and other credit assets, since the data needed is easy to obtain. Results can also be extended to gap management for financial institution's assets and liabilities. This method provides an attractive framework for risk managers to use active strategies to manage portfolios vulnerable to default risk.

References

1. Bierwag, G. O., G. G. Kaufman: Durations of non-default-free securities. Financial Analysts Journal. **44** (1988) 39-46.
2. Chance, D. M.: Default risk and the duration of zero coupon bonds. Journal of Finance. **45** (1990) 265-274.
3. Delianedis, G., Geske, R.: Credit risk and risk neutral default probabilities: Information about rating migrations and defaults, UCLA: The Anderson School. (1998).
4. Duffie, D., Garleanu, N.: Risk and valuation of collateralized debt obligations. Financial Analyst's Journal. **57** (2001) 41-59.
5. Embrechets, P., McNeil, A., D. straumann: Correlation: Pitfalls and Alternatives, Department Mathematik, ETH Zentrum, CH-8092 Zürich. (1999) 1-8.
6. Fooladi, I. J., G. S. Roberts, R. Skinner: Duration for bonds with default risk. Journal of Banking and Finance **21** (1997) 1-16.
7. Geske, R.: The valuation of corporate liabilities as compound options, Journal of Financial and Quantitative Analysis. **19** (1977) 541-552.
8. Jacoby, G.: A duration model for defaultable bonds, The journal of financial research. **26** (2003) 129-146.
9. Jonkhart, M. J. L.: On the term structure of interest rates and the risk of default. Journal of Banking and Finance. **3** (1979) 253-62.
10. Lando, D.: On Cox processes and credit risky securities, Review of Derivatives Research. **2** (1998) 99-120.
11. Li, D. X.: Constructing a credit curve, Risk, Special report on Credit Risk.November (1998) 40-44.
12. Li, D. X.: On default correlation: A copula function approach. Journal of Fixed income, 2000, 9: 43-54.
13. Li, P., H.S. Chen: A Copula Approach to the Pricing of Collateralized Debt Obligation. Lecture Notes in Decision Science, (2005).
14. Lucas, D.: Default Correlation and Credit Analysis. Journal of Fixed Income.**11** (1995) 76-87.
15. Meneguzzo, D., Vecchiato, W.: Copula sensitivity in collateralized debt obligation and basket default swaps. The journal of futures Markets.**24** (2004) 37-70.
16. Merton, R. C.: On the pricing of corporate debt: The risk structure of interest rates, Journal of Finance, **29** (1974) 449-470.
17. Nelsen, R. B.: An introduction to copulas, volume 139 of Lecture Notes in Statistics. Springer, New York. (1999)
18. Schönnbucher, P. J., Schubert, D.: Copula-dependent default risk in intensity models, Bonn University. Ecomonics Faculty, Department of Statistics, Working paper (2001).

Price Roll-Backs and Path Auctions: An Approximation Scheme for Computing the Market Equilibrium

Rahul Garg[1] and Sanjiv Kapoor[2]

[1] IBM India Research Lab., New Delhi
[2] Illinois Institute of Technology, Chicago, USA

Abstract. In this paper we investigate the structure of prices in approximate solutions to the market equilibrium problem. The bounds achieved allow a scaling approach for computing market equilibrium in the Fisher model. Our algorithm computes an exact solution and improves the complexity of previously known combinatorial algorithms for the problem. It consists of a price roll-back step combined with the auction steps of [11]. Our approach also leads to an efficient polynomial time approximation scheme. We also show a reduction from a flow problem to the market equlibrium problem, illustrating its inherent complexity.

1 Introduction

We consider the *market equilibrium problem* with non-negative and linear utilities. The combinatorial structure of the problem has become of interest recently in the design of algorithms, though interest in the structure of this problem dates back further. Arrow and Debreu [1] showed the existence of equilibrium prices under weak assumptions, using a non-constructive argument. Arrow et al. [2], showed that the set of equilibrium prices is convex if the utility functions satisfy the *weak gross substitutability (WGS)* property.

Active interest in designing efficient algorithms to solve the market equilibirum problem dates back to the works of Fisher [3] who designed a hydraulic apparatus to compute market equilibrium. Eaves [8] formulated the market equilibrium problem with linear utilities as a linear complementarity problem and used Lemke's algorithm to solve it. Recently, polynomial-time algorithms have been designed for a number of special cases using a variety of algorithmic techniques.

The techniques may be broadly classified into combinatorial techniques and techniques using convex programming methods. The characterization of equilibirum prices by Arrow et al. [2] leads to an ellipsoid method for computation of market equilibrium in polynomial time (for WGS utilities). This approach has been surveyed in a recent note by Codenotti et al. [4]. Algorithms utilizing circumscribed and inscribed ellipsoids have been discussed in [16,17].

Another convex programming approach is to transform the market equilibirum problem into a convex feasibility (or optimization) problem. This includes the

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 225–238, 2006.
© Springer-Verlag Berlin Heidelberg 2006

works of of Eisenberg and Gale[10,9] who reduced the market equilibirum problem in the Fisher model with linear utilities into a convex optimization problem. Similarly, the market equilibrium problem for the general Walrasian model with linear utilities can be transformed into a convex feasibility problem [15,14,21].

The combinatorial techniques solve the problem for restricted cases and often give approximate solution to the problem. These may be classified into (a) primal-dual techniques algorithms based on maximum flows [5,20,7] and (b) the auction based approaches [11,12]. Devanur et al. [5] solve the market equilibirum problem for the Fisher model with linear utilities using a primal-dual technique based on maximum flows. Using this approach, an algorithm for approximate market equilibirum for the Walrasian model with linear utilities was presented in [13]. This algorithm was further improved in [6].

An efficient auction algorithm was designed to find approximate market equilibrium in the Walrasian model with linear utilities [11]. The auction approach is also applicable to separable gross substitute utilities in the Fisher model [12].

In this paper we study the structure of prices achieved via a typical approximation method. Realizing reasonable bounds on the difference between the approximate and optimal prices allows us to design an approximation scheme for efficient computation of market equilibrium in the Fisher model with linear utilities. We give a definition of approximate market equilibrium and establish a bound on the ratio of prices in approximate and exact equilibirum. We show that $\frac{1}{(1+\epsilon)^n}p_j^* \leq p_j \leq (1+\epsilon)^{n-1}p_j^*$ where p_j, p_j^* are the prices of item j achieved by the approximate and optimal schemes, respectively. Achieving these bounds appears challenging enough itself and achieving similar bounds for the general exchange model is more complicated due to a feedback effect of the error on the prices on the endowment.

Using the price bounds, we devise a price-roll back mechanism and design an algorithm, using the distributed auction mechanism [11,12], to give an approximation scheme for efficiently computing the market equilibrium. This algorithm uses the auction approach to find an approximate solution. It then successively improves the approximation by a price roll-back and uses the auction approach again with a smaller bid increment (ϵ). This algorithm finds an $(1+\epsilon)$ approximate market equilibrium in $O((n^3m+n^2m\log M)\log(1/\epsilon))$ steps, where n and m are the number of buyers and items respectively and $M(\leq ev_{max}a_{max}/e_{min}v_{min}a_{min})$ is a bound on the ratio of initial to final prices. Note that the time complexity of our algorithm depends logarithmically on the approximation factor ϵ. Since the equilibrium prices are rational numbers (assuming that the input is rational), ϵ may be chosen sufficient small to give an exact market equilibrium. This leads to an exact algorithm of complexity $O((n^3m+n^2m\log M)L)$ (L is the bit complexity of the input data, the data being assumed to be rational), which improves the current best complexity, $O(n^4)$ max-flow computation, of the best known combinatorial algorithm ([5]) and matches the currently best known complexity of $O(n^4L)$ [21], which uses an interior point approach.

We further improve on the auction mechanism by designing *path-auction* mechanisms. In this mechanism, the amount of good j that a buyer bids for, is dependent on the re-allocation possible along a sequence of buyer-good pairs $(b_1, g_1, b_2, g_2 \ldots b_k, g_k)$ where every even pair, (g_i, b_{i+1}) represents the return of good i to buyer $i+1$ and the pair (b_i, g_i) represents the bid for good i by buyer i. These path auctions are efficiently implemented using a variant of the dynamic tree data structure of Sleater and Tarjan [18]. The data structure is modified to handle multiplier flows. The path auctions are then carried out using multiplier flows on dynamic trees. This leads to a time complexity of $O((n^3 + n^2 \log M)L)$. In comparison to the best known algorithm in [21], which is based on interior point methods and presumably requires careful handling of numerical precision, our algorithm is combinatorial in nature, has a natural economic interpretation, improves the approximation in successive iterations and is $O(n)$ times faster.

Finally, we show a reduction from the maximum flow problem to the market equilibrium problem (in the Fisher model with linear utilities). This throws some light on the hardness of the market equilibrium problem.

The rest of the paper is organized as follows: In Section 2 we formally define the market equilibrium problem considered by us. This is followed by a discussion of approximate market equilibrium in Section 3. In this section, we establish the bound on prices of approximate market equilibrium. In Section 4 we describe our price-rollback scheme, outline the proof of its correctness and time complexity. In section 5 we describe the algorithm based on the path-auction mechanism. We show how to use multiplier flows and dynamic trees to achieve the improved time bound. In Section 6 we present the reduction from the maximum flows to the market equilibrium problem.

2 Market Model and Preliminaries

We now review the market model of Fisher [3] with linear utilities. Consider a market consisting of a set B of n buyers and a set S of m divisible goods. Buyer i has an amount of money equal to e_i. The amount of good j available in the market is a_j. Assume that the utility of buyers on these goods is linear. Buyer i has a per-unit utility of v_{ij} on good j. Assume that the buyers have no utility for money. The buyers use their money to purchase the goods that maximize their utility.

Given prices $p_1, p_2, \ldots, p_m$ of these m goods, the buyers use their money to purchase goods that maximize their individual utilities. Thus a buyer i will select a good j that maximizes v_{ij}/p_j. Let x_{ij} represent the amount of good j purchased by buyer i. We say that the pair $(\underline{X}, \underline{P})$ forms a market equilibrium if (a) the buyers have spent all their money; (b) there is neither a surplus or a deficiency of any good; (c) all the buyers get items that maximize their utility per unit money spent. The prices $\underline{P}$ are called market clearing prices and the allocation $\underline{X}$ is called the equilibrium allocation.

The condition for market equilibrium can be mathematically represented as:

$$\forall j : \sum_{i=1}^{n} x_{ij} = a_j \tag{1}$$

$$\forall i : \sum_{j=1}^{m} x_{ij} p_j = e_i \tag{2}$$

$$x_{ij} > 0 \Rightarrow v_{ij}/p_j \geq v_{ij}/p_k \forall k \tag{3}$$

where $x_{ij} \geq 0, p_j \geq 0$. The equations (1) and (2) imply that all the goods are sold and all the buyers have exhausted their budget. Equation (3) implies that every buyer gets only those goods that maximize its total utility. It was shown in [10] that the market equilibrium price for the above model is unique.

The market equilibrium problem can be formulated as a solution to a specific primal-dual program (derived from a family of Linear Programs)[11,12]. The equations corresponding to the "restricted" dual program are given by:

$$\forall i, j : \alpha_i p_j \geq v_{ij} \tag{4}$$

and the corresponding complementary slackness conditions are given by:

$$\forall i, j : x_{ij} > 0 \Rightarrow \alpha_i p_j = v_{ij} \tag{5}$$

where $\alpha_i \geq 0$ are the dual variables.

Theorem 1. *Any solution* $(\underline{X}, \underline{P}, \underline{\alpha})$ *with* $\underline{X} \geq 0, \underline{P} \geq 0$ *and* $\underline{\alpha} \geq 0$*, satisfying the conditions (1), (2), (4) and (5), constitutes a market equilibrium.*

For the proof the reader is referred to [11,12]

3 Approximate Market Equilibrium

Define a solution $(\underline{X}, \underline{P}, \underline{\alpha})$ to be $(1+\epsilon)$-approximate market equilibrium if it satisfies (1), (4) and the following "ϵ-relaxation" of the conditions (2) and (5):

$$\forall i : \frac{e_i}{1+\epsilon} \leq \sum_{j=1}^{m} x_{ij} p_j \leq e_i \tag{6}$$

$$\forall i, j : x_{ij} > 0 \Rightarrow v_{ij} \leq \alpha_i p_j \leq (1+\epsilon) v_{ij} \tag{7}$$

Theorem 2. *Let* $\underline{P}^*$ *be the unique market equilibrium price. Let* $\underline{X}^*$ *be corresponding equilibrium allocation. Let* $(\underline{P}, \underline{X}, \underline{\alpha})$ *be a* $(1+\epsilon)$*-approximate market equilibrium (satisfying (1), (4), (6) and (7)). Then, for all* j *the following must be true:*

$$\frac{p_j^*}{(1+\epsilon)^n} \leq p_j \leq (1+\epsilon)^{n-1} p_j^*.$$

For the proof of the above theorem, we first construct a directed weighted bipartite graph (called the assignment graph G_a) with the set of buyers B and the set of goods S as vertices on the two side. We use the market equilibrium $(\underline{X}^*, \underline{P}^*)$ and the approximate market equilibrium $(\underline{X}, \underline{P})$ to define the weight on the edges of G_a. We then construct a reduced (weighted) directed acyclic subgraph G_r by successively removing cycles in G_a. We prove some properties of the graph G_r and use these properties to prove Theorem 2.

Define:

$$\gamma_i = \frac{e_i}{\sum_{j=1}^{m} x_{ij} p_j}$$

Since the solution $(\underline{X}, \underline{P})$ satisfies (6), $1 \le \gamma_i \le 1+\epsilon$ for all i. The weight function $w_a : ((B \times S) \cup (S \times B) \rightarrow R_+$ of the graph G_a is defined as follows:

$$w_a(i,j) = x_{ij} p_j \gamma_i \quad \forall (i,j) \in B \times S$$

$$w_a(j,i) = x^*_{ij} p^*_j \quad \forall (j,i) \in S \times B$$

The edge e is present in the graph G_a iff $w(e) > 0$. Note that, with this construction following is true:

$$\sum_{j \in S} w_a(i,j) = \sum_{j \in S} w_a(j,i) = e_i$$

Define the average price of an item j ($\hat{p}_j$) in the approximate market equilibrium as follows:

$$\hat{p}_j = \frac{\sum_{i \in B} w_a(i,j)}{a_j} = \sum_{i \in B} \frac{\gamma_i x_{ij} p_j}{a_j}$$

Since $1 \le \gamma_i \le 1+\epsilon$, and $\sum_{i=1}^{n} x_{ij} = a_j$, we have:

$$p_j \le \hat{p}_j \le (1+\epsilon) p_j \tag{8}$$

The reduced acyclic sub-graph G_r is constructed from G_a using a sequence of "cycle removal" steps leading to graphs $G_a \equiv G_0, G_1, G_2, \ldots G_k, G_{k+1} \equiv G_r$. At step l, a cycle in the graph G_{l-1} is located. Let e_l be the minimum weight edge in this cycle. The graph G_l is obtained from G_{l-1} by subtracting $w(e_l)$ from all edges in the cycle and removing the zero weight edges (including the edge e_l). Every step removes at least one edge (e_l) of the graph G_{l-1}. Therefore, this procedure is guaranteed to terminate giving an acyclic graph G_r. We now prove some properties of G_r.

Lemma 1. *No maximal path in G_r can start or end at a buyer vertex $i \in B$.*

Lemma 2. *If a maximal path in G_r starts at an item $k \in S$ and ends at an item $k' \in S$ then $p^*_k > \hat{p}_k$ and $p^*_{k'} < \hat{p}_{k'}$.*

Lemma 3. *If an item $k \in S$ is disconnected in G_r then $p^*_k = \hat{p}_k$.*

Lemma 4. *If there is a path of length $2l$ from item k to item k' in the graph G_a then*

$$\frac{p_{k'}}{p_k} \le (1+\epsilon)^l \frac{p^*_{k'}}{p^*_k}$$

Proof of Theorem 2. Consider any item $j \in S$. If j is disconnected in G_r, then Lemma 3 gives $p_j^* = \hat{p}_j$. Using (8) we get:

$$\frac{p_j^*}{1+\epsilon} \leq p_j \leq p_j^*$$

If j is connected in G_r, then consider any maximal path in G_r containing j. According to Lemma 1, this path must start at an item (say k) and end at an item (say k'). Using Lemma 2 and (8) we get:

$$p_k \leq \hat{p}_k < p_k^* \tag{9}$$

$$p_{k'}^* < \hat{p}_{k'} \leq (1+\epsilon)p_{k'} \tag{10}$$

The length of the path from k to j is bounded by $2(n-1)$. Therefore, Lemma 4 can be used to get:

$$\frac{p_j}{p_k} \leq (1+\epsilon)^{n-1}\frac{p_j^*}{p_k^*}$$

$$\Rightarrow \frac{p_j}{p_j^*} \leq (1+\epsilon)^{n-1}\frac{p_k}{p_k^*}$$

The above inequality alongwith with (9) gives:

$$p_j \leq (1+\epsilon)^{n-1}p_j^*$$

Similarly, there is a path from item j to item k' of length at most $2(n-1)$. Again, using Lemma 4 we have:

$$\frac{p_{k'}}{p_{k'}^*} \leq (1+\epsilon)^{n-1}\frac{p_j}{p_j^*}$$

Using (10) the above reduces to:

$$p_j \geq \frac{1}{(1+\epsilon)^n}p_j^*$$

This proves that for all items $j \in S$,

$$\frac{1}{(1+\epsilon)^n}p_j^* \leq p_j \leq (1+\epsilon)^{n-1}p_j^*$$

4 Successive Approximations by Price Rollback

We now describe the algorithm `roll-back` (shown in Figure 1), which provides a polynomial method for solving the market equilibrium problem exactly (or to within any specified accuracy). This algorithm relies on algorithm listed as algorithm `auction` in Figure 2. This algorithm is a simplification of `main` in [12].

algorithm roll-back
  initialize
  $\epsilon = \epsilon_0 = 1$
  call algorithm auction(ϵ)
  while ($\epsilon > \delta$) **do**
    $\forall j : p_j \leftarrow p_j/(1+\epsilon)^{2n}$
    $\forall i, j : y_{ij} = y_{ij} + h_{ij}; h_{ij} = 0;$
    $\epsilon \leftarrow \epsilon/2$
    $\forall i : r_i = e_i - \sum_{j=1}^{m} y_{ij} p_j/(1+\epsilon)$
    call algorithm auction(ϵ)
  end while
end algorithm roll-back

Fig. 1. The price rollback algorithm

The algorithm begins with $\epsilon = 1$ and the initial solution as in [12]. It then calls algorithm `auction` to get a 2-approximate market equilibrium. It then scales down the prices by a factor $(1+\epsilon)^{2n}$ and reduces ϵ to half of its current value. It then calls `auction` to with the new value of ϵ to give a $(1+\epsilon)$-approximate market equilibrium. This process continues until ϵ is sufficiently small.

Lemma 5. *For any* $0 \leq \epsilon_1 \leq \epsilon_0$*, if the algorithm* `auction` *is called with* $\epsilon = \epsilon_1$ *and an initial solution satisfying (1), (4), (7), (11) and (12)*

$$\forall i : \sum_{j=1}^{m} x_{ij} p_j \leq e_i \tag{11}$$

$$\forall j : p_j \leq \frac{p_j^*}{(1+\epsilon)^{n+1}} \tag{12}$$

with $\epsilon = \epsilon_0$*, then it terminates with* $(1+\epsilon_1)$*-approximate market equilibrium.*

This leads to the following result.

Theorem 3. *Iteration* k *of procedure* `roll-back` *finds a* $(1+\epsilon_k)$*-approximate market equilibrium, where* $\epsilon_k = \epsilon_0/2^k$*.*

To prove the complexity of the algorithm, we need to bound the time taken by a call to algorithm `auction`. For this we recall the relevant portions of the analysis in [11]. An item is sold at two prices and h_{ij}, y_{ij} respectively represent the amounts of item j sold to buyer i at prices p_j and $p_j/(1+\epsilon)$. For a buyer i its demand set is defined as $D_i = \arg\max_j v_{ij}/p_j$. Define D to be the set of demand edges, X the set of assignment edges, Y a subset of the set of assignment edges and Z a subset of Y as follows.

$$\begin{aligned}
(i,j) \in D &\text{ iff } j \in D_i \\
(j,i) \in X &\text{ iff } x_{ij} > 0 \\
(j,i) \in Y &\text{ iff } y_{ij} > 0 \\
(j,i) \in Z &\text{ iff } y_{ij} > 0 \text{ and } j \notin D_i
\end{aligned}$$

```
procedure initialize
    ∀i, ∀j : y_ij = 0; ∀i ≠ 1, ∀j : h_ij = 0
    ∀j : h_1j = a_j
    ∀j : α_1j = (Σ_j a_j v_1j)/e_1
    ∀j : p_j = v_1j/α_1 ; ∀i : α_i = max_j v_ij/p_j
    ∀i ≠ 1 : r_i = e_i ; r_1 = 0
end procedure initialize

algorithm auction(ε);
    repeat
        forall buyers i do
            D_i = arg max_j v_ij/p_j
            α_i = max_j v_ij/p_j
            while r_i > 0 do
                pick j ∈ D_i
                if ∃k : y_kj > 0
                    outbid(i, j, k)
                else
                    raise_price(j)
                    ∀k : D_k = arg max_j v_kj/p_j; α_k = max_j v_kj/p_j
                endif
            end while
        end for
    until Σ_i r_i > ε(Σ_i e_i)
end algorithm auction

  procedure outbid(i, j, k)
    if j ∉ D_k and i ≠ k then
        t = min(y_kj, r_i/p_j)
        h_ij = h_ij + t ; y_kj = y_kj − t
        r_i = r_i − t p_j
        r_k = r_k + t p_j/(1 + ε)
    else
        t = min((ε/(1+ε)) y_kj, r_i/p_j)
        h_kj = h_kj + t/ε
        y_kj = y_kj − t(1 + ε)/ε ; h_ij = h_ij + t
        r_i = r_i − t p_j
    endif
end procedure
procedure raise_price(j)
    ∀i : y_ij = h_ij; h_ij = 0;
    p_j = (1 + ε) p_j
end procedure raise_price
```

Fig. 2. The basic auction mechanism

Define a directed bipartite graph $G = (B, S, D \cup Z)$ where B is the set of buyers, S the set of goods.

Lemma 6. *The graph G is acyclic.*

The proof can be carried on same lines as [11].

Note that procedure `outbid` transfers the surplus (unspent money) from one buyer to another only along paths in G. Moreover the surplus reduces by a factor $(1 + \epsilon)$ every time it travels form one node to another in G. When `outbid(`i`,`

j, k) is called either r_i goes to zero or an edge from Y is removed (either y_{ij} goes to zero or y_{kj} goes to zero). Define a call to `outbid` as complete when an edge in Y is removed and incomplete if r_i goes to zero.

The steps performed by the algorithm `auction` are classified into three types (a) price rise of an item; (b) complete call to `outbid` (y_{kj} limits the bid); (c) incomplete call (surplus r_i goes to zero) to `outbid` and (d) computation of α_i and D_i for all i. The time complexity of the algorithm can be bounded as follows: steps of type (a) take $O(n)$ time; there can be atmost n steps of type (b) for every step of type (a). Using the directed acyclic graph argument of [11] it can be shown that there can be atmost n^2 steps of type (c) for every step of type (a). Steps of type (d) can be naively implemented in $O(nm)$ time and there can be atmost one step of type (d) for every step of type (a).

This gives a time complexity of $O((n^2 + nm)W)$ where W is the number of steps of type (a) (i.e., the number of price rises).

We now bound W. Let $e_{min} = \min_i e_i$, $e = \sum_i e_i$, $v_{min} = \min_{i,j:v_{ij}>0} v_{ij}$, $v_{max} = \max_{i,j} v_{ij}$, $a_{max} = \max_j a_j$ and $a_{min} = \min_j a_j$. The maximum and the minimum price of any item is bounded, respectively, by $\frac{e}{a_{min}}$ and $\frac{(e_{min}v_{min})}{(a_{max}v_{max})}$. Therefore, the number of price raise for any item in the first iteration is bounded by $O(\log((ev_{max}a_{max})/(e_{min}v_{min}a_{min})))$. Using Theorem 2 and the fact that prices are rolled back by factor $(1+\epsilon_k)^{2n}$ after iteration k and $\epsilon_{k+1} = \epsilon_k/2$, the number of price raises for any item in one call to algorithm `auction` at iterations subsequent to the first, can be bounded by $7n$. This gives a bound of $O(nm + m\log((ev_{max}a_{max})/(e_{min}v_{min}a_{min}))\log(1/\delta))$ for w. Therefore, we have the following time complexity of algorithm `roll-back`.

Theorem 4. *Algorithm* `roll-back` *terminates in* $O(nm(n+m)(n + \log((ev_{max}a_{max})/(e_{min}v_{min}a_{min})))\log(1/\delta))$ *steps.*

A bound on $1/\delta$ can be provided when the input is rational with numbers bounded by M. By an analysis similar to that of Lemma 8 in [5], it can be shown that price of an item j, p_j is related to p_k, price of k, in the connected component of $G_o = (B, S, E)$ where E comprises edges (i, j) s.t. $x^*_{ij} > 0$ in the equilibrium allocation. In fact $p_j = \frac{a}{b}p_k$ where a and b are product of utility values of length l when the items j and k are connected by a path of length $2l$. Thus $1/\delta$ is bounded by nV^n where $V = \max_{ij}\{v_{ij}\}$. We can thus obtain an exact algorithm for the market clearing problem.

5 A Faster Algorithm Using Path-Auctions

The market equilibrium problem (for Fisher's model with linear utilities) is strikingly similar to the maximum flow problem and the auction algorithm described in the previous section resembles the preflow push algorithms for the maximum flow problem. The procedure `outbid` may be viewed as a step where the surplus (excess flow) is transferred from one buyer (vertex) to another. However, there are two key differences between the maximum flow algorithms and the `auction` algorithm. Unlike the traditional flows, the surplus is not conserved; it decreases by a factor $(1+\epsilon)$ when transferred from one vertex to another. Secondly, the graph G on which the algorithm works changes dynamically, due to the changes in demand sets and assignments.

Consider the graph $G = (B, S, D \cup Z)$. With each edge is associated a capacity which represents the maximum amount of money that can be pushed along the edge. For an edge in D there is no bound whereas for an edge (u, v) in Z the capacity is bounded by $y_{vu} * p_u$. Consider a path $(u_1, v_1, u_2, v_2 \ldots, u_k, v_k)$ in G, beginning at a buyer u_1 with positive surplus and ending at an item v_k that has no outgoing edges. Let the prices of these items be $p_1, p_2, \ldots, p_k$ respectively. Consider a bidding sequence where u_1 outbids u_2 on item v_1, who inturn outbids u_3 on item v_2, and so on, till u_{k-1} outbids u_k on item v_{k-1}. To acquire x units of item v_1, u_1 needs xp_1 amount of money. This releases $p_1x/(1+\epsilon)$ surplus for u_2, which if spent fully on acquiring v_2, will generate a surplus of $p_1x/(1+\epsilon)^2$ at u_3. If such a bidding is carried till the end of the path, the amount of surplus generated at u_k will be $p_1x/(1+\epsilon)^{k-1}$. Since v_k has no outgoing edge, all the buyers of v_k have v_k in their demand sets. If these buyers are outbid by u_k then they can reduce their allocations of item v_k by a factor $(1+\epsilon)$ and switch from the lower price $(p_k/(1+\epsilon))$ to the higher price (p_k). Therefore the maximum amount u_k can bid on v_k is given by $\frac{\epsilon}{1+\epsilon}\sum_{w\in B} y_{wk}p_k$. To model this, we add a special vertex t (called the sink) to G. For every vertex v which does not have an outgoing edge, we add an edge (v, t) of capacity $\frac{\epsilon}{1+\epsilon}\sum_{w\in B} y_{wv}p_v$.

Path auctions are defined by such sequences of bidding along paths in G. An auction where bidding is done along paths in G such the surplus of all the nodes except the first node remains unchanged, is called a *path auction*.

We define the capacity of an auction path as the maximum amount of money that the first buyer can bid along the path, without changing the surplus of other buyers or the price of any item on the path. For the path $(u_1, v_1, \ldots, u_k, v_k)$ the capacity is equal to $\min(\min_{1\le j\le k} y_{(j+1)j}(1+\epsilon)^{j-1}p_j, \epsilon(1+\epsilon)^{k-2}\sum_{w\in B} y_{wk}p_k)$.

In order to compute the path capacities and carry out bidding on paths efficiently, we use *multiplier* flows which are used to model the fact that the surplus is not conserved as it traverses a path in the graph G. We then define operations on dynamic trees [18] with multiplier flows. We use these operations to efficiently implement the algorithm *path auctions.* We finally outline how the dynamic trees may be modified to support multiplier flows efficiently.

Multiplier Flows

Given a directed graph $G = (V, E)$ with edge capacities $c : E \rightarrow R_+$ and multipliers $\eta : E \rightarrow R_+$. The multiplier of a path $P = (e_1, e_2, \ldots, e_k)$ is defined as $\eta(P) = \Pi_{i=1}^{k}\eta(e_i)$. A multiplier flow of value f through a directed path P carries a flow of value $\Pi_{j=i}^{k}\eta(e_j)f$ through the edge e_i. Note that if the multiplier $\eta(e_j) = 1+\epsilon$, for all j then a multiplier flow of value f on the path P translates into a flow of value $(1+\epsilon)^{(k-j)}f$ through the edge e_j for all $j \le k$. We define the *multiplier capacity* of the path P as $\min_{1\le i\le k} c(e_i)/\eta(e_i, \ldots e_k)$. Note that the capacity of an auction path is equal to the product of the multiplier of the path and its multiplier capacity.

Dynamic Trees, Multiplier Flows and Efficient Path-Auctions
An efficient data structure to maintain dynamic trees was proposed in [18]. This data structure maintains a collection of vertex disjoint trees to efficiently (in $O(\log n)$ amortized time) carry out operations of combining/splitting the trees and updating capacity on paths from leaves to root of a tree. We adapt this data structure for multiplier flows by defining operations on trees.

We need the following definitions:

`link`(u, v, c, η) : Join the tree rooted at u to the tree node v by adding the edge (u, v) of capacity c and multiplier η.
`cut`(u, v) : Split the tree containing the (u, v) edge into two trees by removing the edge.
`parent`(u) : Returns the parent of u in the tree (nil if u is a root node).
`root`(u) : Returns the root of tree containing the vertex u.
`children`(u) : Returns the set of children of node u.
`capacity`(u, v) : Returns the capacity of the edge (u, v).
`multiplier`(u) : Returns the multiplier of the path from u to `root`(u).
`find-min`(u) : Let V be the set of edges in the path from u to `root`(u). This function returns $\arg\min_{v \in V}$ `capacity`$(v,$ `parent`$(v))$ / `multiplier`(v).
`update`(u, x) : Let V be the set of edges in the path from u to `root`(u). This function updates the capacities for all the edges in the path as $\forall v \in V$: `capacity`$(v, \text{parent}(v)) += x$ `multiplier`(v)

The algorithm discovers a path from a vertex u with positive surplus ($r_u > 0$) to the sink t. It then finds out the maximum amount that may be bid along the path using its multiplier capacity. After the bidding there are three possibilities (as in algorithm `auction`) (a) price of last item needs to be raised; (b) y_{wv} goes to zero for some edge in the path, or (c) the surplus r_u of vertex u goes to zero. In case (c) the algorithm moves to another buyer with positive surplus. In case (b) the corresponding edge is removed from the graph G and another path from v to t is found. In case (a) the price of the item is raised and the required data-structures and variables updated suitably. We now present the algorithm in greater detail.

The algorithm first creates the graph G using the initial solution. The set D is maintained in the graph by keeping a heap of items for all $u \in B$, sorted by v_{uv}/p_v. The sets Z and Y are maintained by keeping two lists for each item $v \in S$. We add a special vertex t called as the sink in the graph G. All the vertices $v \in S$ which do not have an outgoing edge (in Z) are implicitly assumed to be connected to t. Initially all the trees are singleton vertices.

The algorithm picks a buyer vertex u with positive surplus and creates a path from u to the sink t by linking the vertices using the edges in G. For every buyer vertex in G, the demand set is well defined and hence there is always an outgoing edge from the vertex. For an item vertex v in G either there is an outgoing edge in Z or it is connected to t. Since the graph G is acyclic, a path to t can always be discovered.

For the edges in D that get linked in the process, the capacity is set to M and multiplier is set to 1. For the edges (v, u) in Z, the capacity is set to $\hat{y}_{uv} p_v$

and the multiplier is set to $(1+\epsilon)$. For edges of the form (v,t) the capacity is set to $\frac{\epsilon}{1+\epsilon}\sum_{u\in B}\hat{y}_{uv}p_v$ and the multiplier is set to 1. Note that the capacity of auction path $(u,\ldots,t)$ is equal to $\texttt{multiplier}(u)\ \texttt{find\_min}(u)$.

For an item v such that $(v,t) \in G$, define $\gamma_v = (1+\epsilon)\texttt{capacity}(v,t)/(\epsilon\sum_{u\in B}\hat{y}_{uv}p_v)$. For all other items v, define $\gamma_v = 0$. Now define the assignments h_{uv} and y_{uv} using the variables $\hat{h}_{uv}$ and $\hat{y}_{uv}$ and the capacities of edges in trees as follows: if $(v,u) \in Z$ is in a tree, then $y_{uv} = \texttt{capacity}(v,u)$ else $y_{uv} = \hat{y}_{uv}(1-\gamma_v)$. If $(u,v) \in D$ is in a tree then $h_{uv} = (M - \texttt{capacity}(u,v))/p_v + \gamma_v\hat{y}_{uv}/(1+\epsilon)$ else $h_{uv} = \gamma_v\hat{y}_{uv}/(1+\epsilon)$.

With these definitions, it can be verified that the update step in the algorithm is indeed equivalent to a path auction from buyer u on the path to t. After the update step, the algorithm either raises the price of the last item in the path, or cuts an edge from the tree or moves to another buyer with positive surplus. In each of these cases, the variables and data structures are updated suitably.

Efficient Implementation of Dynamic Trees

We next show how to modify the dynamic tree data structure of Sleater and Tarjan [18] to implement the requirements of pushing multiplier flows.

The collection of auction paths form a collection of vertex-disjoint trees and change over time. The dynamic tree data structure[18] applies in this context. Each path is represented by binary trees. Each node, v, of the binary tree data structure represents a sub-path $P(v)$, i.e. a sequence of edges corresponding to the edges stored at the leaves of the subtree rooted at v. At each node v we maintain a variable corresponding to the minimum capacity of the flow path, $Min(v)$, a variable representing a composite multiplier for the path P termed $Mult(v)$, and updates which are applicable to each edge of the path $P(v)$, $UP(v)$. The effective minimum capacity of the path corresponding to a node x in the tree is obtained as $EMin(x) = Min(x) - \sum_{y\in Y} UP(y) * Mult(y)/Mult(x)$, where Y is the set of nodes on the path from node (x) to the root. The second term denote the effective update variable at node x. The minimum value at a node v is effectively $Min(v) = \min(Min(u) - Up(v), (Min(w) - Up(w)) * Mult(u))$, where u and w are the left and right children of the node v, respectively. Further, $Mult(v) = Mult(u) * Mult(w)$.

If the tree data structure used is represented by splay trees [19], nodes along the path from the root to a particular node, say v, in the tree are affected. Let the path be $P(v)$. To implement the changes along the path, say at a node $y \in P(v)$, the effective update variable is computed at each child for node y, say w and z and $UP(w), UP(z)$ computed at these nodes. $UP(y)$ is set to zero. This makes the value of the update variable zero along the path and path changes can be now be made locally. Following [19], this results in an implementation of a sequence of k tree operation involving multiplier flows requiring $O(k\log n)$ operations where n are the maximum number of nodes in the splay trees.

Complexity of Path-Auctions

To bound the complexity of the algorithm we carry out amortized analysis of different operations. If W is the number of times prices are raised, then the number of dynamic tree operations are bounded by $O(nW)$. Each of the dynamic tree operation can be implemented in $O(\log n)$ amortized time. The time required to update the data structures is $O(n)$. This gives a overall time complexity of $O(nW \log n)$.

If we use the roll-back mechanism, W is bounded by $O(nm + m\log((ev_{max}a_{max})/(e_{min}v_{min}a_{min}))\log(1/\delta))$, where $1/\delta$ is bounded by nV^n and $V = \max_{ij}\{v_{ij}\}$ Thus we have the following result.

Theorem 5. *Using path-auctions the market equilibrium problem can be solved exactly in* $O(n(nm + m\log((ev_{max}a_{max})/(e_{min}v_{min}a_{min}))\log(1/\delta))\log n)$.

6 Reduction from Max-Flows

In order to show the inherent complexity of the market equilibrium problem, we reduce the problem of maximum flows with vertex capacities to the market equilibrium problem. Given a graph $G = (V, E)$ with vertex capacities $c : V \rightarrow R_+$, two special vertices s, t (called the source and the sink) and a flow value f the problem is to decide if it is possible to route f units of flow from s to t without sending a flow of value more than $c(v)$ from any vertex v. This problem can be reduced to the following market equilibrium.

For each vertex in $V - \{t\}$ create a buyer in B and for each vertex in $V - \{s\}$, create an item in S. For every vertex $j \in V - \{s, t\}$, let $e_j = a_j = c(j)$. Let $v_{ij} = 1$ if $(i, j) \in E$ and $v_{ij} = 0$ otherwise. Let $v_{ii} = 1$ for all $i \in V - \{s, t, \}$. Let $e_s = f$ and $a_t = f$. Note that the equilibrium prices are unique in the Fisher model. Therefore, it is easy to verify that a flow f can be routed from s to t in G iff all the equilibrium prices in the corresponding market equilibrium problem are unity.

References

1. K. Arrow and G. Debreu. Existence of an Equilibrium for a Competitive Economy. *Econometrica*, 22:265–290, 1954.
2. K.J. Arrow, H.D. Black, and L. Hurwicz. On the Stability of the Competetive Equilibrium, II. *Econometrica*, 27:82–109, 1959.
3. W. C. Brainard and H. E. Scarf. How to Compute Equilibrium Prices in 1891. Cowles Foundation Discussion Paper (1272), 2000.
4. Bruno Codenotti, Sriram Pemmaraju, and Kasturi Varadarajan. A note on the Computation of Equilibria in Exchange Markets with Gross Substitutibility. Preprint, 2004.
5. N. Devanur, C. Papadimitriou, A. Saberi, and V. Vazirani. Market Equilibrium via a Primal-Dual-Type Algorithm. In *43rd Symposium on Foundations of Computer Science (FOCS 2002)*, pages 389–395, November 2002.

6. N. R. Devanur and V. Vazirani. An Improved Approximation Scheme for Computing the Arrow-Debreu Prices for the Linear Case. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2003)*, 2003.
7. N. R. Devanur and V.V. Vazirani. The Spending Constraint Model for Market Equilibrium: Algorithmic, Existence and Uniqueness Results. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, 2004.
8. B. Eaves. A Finite Algorithm for the Linear Exchange Model. *Journal of Mathematical Economics*, 3:197–203, 1976.
9. E. Eisenberg. Aggregation of utility functions. *Management Sciences*, 7(4):337–350, 1961.
10. E. Eisenberg and D. Gale. Consensus of Subjective Probabilities: The Pari-Mutuel Method. *Annals of Mathematical Statistics*, 30:165–168, 1959.
11. Rahul Garg and Sanjiv Kapoor. Auction Algorithms for Market Equilibrium. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, 2004.
12. Rahul Garg, Sanjiv Kapoor, and Vijay Vazirani. An Auction-Based Market Equilibrium Algorithm for the Separable Gross Substitutibility Case. In *Proceedings of the 7th. International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'04)*, 2004.
13. K. Jain, M. Mahdian, and A. Saberi. Approximating Market Equilibrium. In *Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2003)*, 2003.
14. Kamal Jain. A Polynomial Time Algorithm for Computing the Arrow-Debreau Market equilibrium for Linear Utilities. Preprint.
15. E.I. Nenakov and M.E. Primak. One algorithm for finding solutions of the Arrow-Debreu model. *Kibernetica*, 3:127–128, 1983.
16. D.J. Newman and M.E. Primak. Complexity of Circumscribed and Inscribed Ellipsod Methods for Solving Equilibrium Economical Models. *Applied Mathematics and Computations*, 52:223–231, 1992.
17. M.E. Primak. A Converging Algorithm for a Linear Exchange Model. *Applied Mathematics and Computations*, 52:223–231, 1992.
18. Daniel D. Sleator and Robert Endre Tarjan. A data structure for dynamic trees. *J. Comput. Syst. Sci.*, 26(3):362–391, 1983.
19. Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *J. ACM*, 32(3):652–686, 1985.
20. V. V. Vazirani. Market Equilibrium When Buyers Have Spending Constraints . Submitted, 2003.
21. Yinyu Ye. A Path to the Arrow-Debreu Competetive Market Equilibrium. Preprint, 2004.

New Results on Rationality and Strongly Polynomial Time Solvability in Eisenberg-Gale Markets*

Deepamab Chakrabarty, Nikhil Devanur, and Vijay V. Vazirani

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332–0280
{deepc, nikhil, vazirani}@cc.gatech.edu

Abstract. We study the structure of EG[2], the class of Eisenberg-Gale markets with two agents. We prove that all markets in this class are rational and they admit strongly polynomial algorithms whenever the polytope containing the set of feasible utilities of the two agents can be described via a combinatorial LP. This helps resolve positively the status of two markets left as open problems by [JV]: the capacity allocation market in a directed graph with two source-sink pairs and the network coding market in a directed network with two sources.

Our algorithms for solving the corresponding nonlinear convex programs are fundamentally different from those obtained by [JV]; whereas they use the primal-dual schema, we use a carefully constructed binary search.

1 Introduction

The need for developing an algorithmic theory of market equilibria is by now well established and much work has been done along these lines within algorithmic game theory over the last five years. Since one of the main motivations for this work is new markets defined on the Internet which typically have massive computational power available for running these markets in a centralized or distributed manner, this work should not only address traditional market models studied within mathematical economics but also define new models. The latter is a difficult task – the new models need to not only capture some of the idiosyncracies of these markets in a simple manner but also be amenable to efficient computation. Until we are in a position to produce such results, it is important to define mathematical clean market models in order to grow our toolkit of algorithms for computing equilibria efficiently.

An important step in this direction was recently taken by Jain and Vazirani [JV] who defined the notion of Eisenberg-Gale markets. Eisenberg and Gale [EG59] had given a remarkable convex program whose optimal solution gives equilibrium allocations for the linear utilities case of Fisher's market model; the latter is considered one of the most fundamental market models in mathematical economics.

* Work supported by NSF Grants 0311541, 0220343 and 0515186.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 239–250, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Equilibrium for an Eisenberg-Gale market is captured via a convex program that has the same form as the Eisenberg-Gale program, i.e., it maximizes the money weighted geometric mean of buyers' utilities subject to linear packing constraints with the additional conditions of free disposal and utility homogeneity (see Section 2 for definitions). [JV] studied the class of Eisenberg-Gale markets from the five viewpoints of solvability via strongly polynomial algorithms, rationality, efficiency, fairness and competition monotonicity, and they found a surprisingly rich structure. They also stated a host of open problems whose resolution should lead to a deeper understanding not only of these markets but also of the issue of solvability of nonlinear convex programs via strongly polynomial algorithms.

In this paper we investigate Eisenberg-Gale markets further and settle two open problems of [JV]. A remarkable property of the Eisenberg-Gale program is that, despite its being nonlinear, it always has a rational solution if all the input parameters are rational. Therefore, the linear case of Fisher's market model also has this property. We will say that a market or a nonlinear program is *rational* if it has this property and *irrational* otherwise.

Interestingly enough, rationality is not unique to the Eisenberg-Gale program. [JV] showed that several natural markets in Kelly's [Kel97] resource allocation framework, which are also Eisenberg-Gale markets, are rational. Several other such markets are irrational [GJTV05, JV] (see Section 1.1 for a detailed description). Two markets whose status was not known, and were left as open problems by [JV], were: the capacity allocation market in a directed graph with two source-sink pairs and the network coding market in a directed network with two sources (see Section 2 for definitions). Among the markets characterized so far, an important distinction between the rational and irrational markets was that combinatorial problems underlying the former satisfied max-min theorems, which were used critically to establish rationality, and those for the latter didn't.

The two markets left open do not support max-min theorems. Surprisingly enough, despite this, both of them turn out to be rational. More generally, in this paper we show that all markets in EG[2], the class of Eisenberg-Gale markets with two agents, are rational. We also show that whenever the polytope containing the set of feasible utilities of the two agents can be described via a combinatorial LP, the market admits a strongly polynomial algorithm; both markets described above admit such LP's. In Section 1.1 we give an overview of the structural and algorithmic ideas we use to prove these results.

1.1 A Comparison of Structural and Algorithmic Ideas

There is a fundamental difference between markets for which [JV] gave strongly polynomial algorithms and the ones for which we do – combinatorial problems underlying the former do support max-min thoerems whereas the latter don't. This difference manifests itself in the algorithmic ideas needed in the two cases – whereas [JV] use the primal-dual schema and their algorithms can be viewed as ascending price auctions, we use a carefully constructed binary search. The algorithms of [JV] are combinatorial whereas ours are not, ours require a subroutine for solving combinatorial LP's. The latter can be accomplished in strongly polynomial time using Tardos' algorithm [Tar86].

We start by giving a brief overview of the natural markets studied in [JV]. These markets belong to the resource allocation framework given by Kelly [Kel97] for modeling and understanding TCP congestion control. Resources in these markets are edge capacities and agents want to build combinatorial objects such as source-sink flow paths or spanning trees or branchings, e.g., for establishing TCP connections or broadcasting messages to all nodes in the network.

The following market, called the *capacity allocation market*, is of special significance within Kelly's framework: Given a network (directed or undirected) with edge capacities specified and a set of source-sink pairs, each with initial endowment of money specified, find equilibrium flow and edge-prices. The equilibrium must satisfy:

- Only saturated edges can have positive prices.
- All flows are sent along a minimum cost path from source to sink.
- The money of each source-sink pair is fully spent.

[JV] generalized the above market to the broadcasting setting where the agents are nodes of the network with money and they want to buy (possibly fractional) spanning trees (in the undirected case) or branchings (in the directed case). Once again, the equilibrium prices must satisfy conditions similar to the ones given above, that is, only saturated edges have prices, agents buy only the cheapest trees or arborescence and money of each agent is fully spent.

Table 1 summarizes the results about equilibrium in the two capacity allocation markets. We also record the complexity of the finding the minimum cut separating the source-sink pairs in the setting.

Table 1. Table of Results about Rationality of Equilibrium in Capacity Allocation markets. The table also notes the complexity of finding the min-cut separating the source sink pairs from each other.

	One Source Multiple Sink	Two Source Two Sink	Multiple Source Multiple Sink
Directed Networks	– Rational [JV] – Polynomial Time [FF56]	– **Rational** (This paper) – **NP**-hard [GVY94]	– Irrational [GJTV05] – **NP**-hard [DJP+94]
Undirected Networks	– Rational [JV] – Polynomial Time [EFS56]	– Rational [JV] – Polynomial Time [Hu63]	– Irrational [GJTV05] – **NP**-hard [DJP+94]

Note that previous to this paper, each case of rational equilibrium also has a polynomial time algorithm to find the minimum cut and the cases which are **NP**-hard have irrational equilibrium. Also each polynomial time algorithm is associated with a max-flow-min-cut result. Indeed [JV] use this duality crucially in their proofs of rationality of equilibrium.

For the broadcasting capacity allocation markets, [JV] proved rationality for the case of undirected networks with arbitrary agents and for directed networks with two or less agents. Once again, the main tool used were duality results about packing trees and arborescence by [NW61, Tut61] and [JV] respectively.

In fact, in the two source-sink case of the capacity allocation market, where there is a gap between the max-flow and the min-cut, it was expected that the equilibrium would be irrational.

In this paper, we prove that the two source-sink capacity allocation markets, and in general any EG market with two agents including the *network coding market* which generalizes the broadcasting capacity allocation market (see Section 2), are rational. Our algorithm circumvents the lack of underlying max-min theorems by using the more general LP-Duality Theorem itself; on the flip side, our methods work only for the case of two agents.

Let $\mathcal{M}$ be a market in EG[2] and $\mathcal{P}$ be the polytope, in 2-dimensions, capturing the set of feasible utilities of the two agents. For the rest of the section, consider $\mathcal{M}$ to be the capacity allocation market described above with two source-sink pairs. We consider a simple description of the set of facets of $\mathcal{P}$ using only one parameter, α. Think of α as the slope of the line segment representing the facet. As it will turn out, the issue of pricing edges of the network is equivalent to pricing the facets of $\mathcal{P}$ which satisfy conditions similar to the ones for the capacity allocation market (Details in Section 3 and the full paper [CDV]). We show that for any instance of $\mathcal{M}$, at most two (adjacent) facets of $\mathcal{P}$ need to be assigned positive prices. Since $\mathcal{P}$ is a projection of the flow-polytope onto two dimensions, it can be expected that the number of facets would not be too large. For instance, in the case of undirected networks with two source-sink pairs, it follows from a theorem of Hu [Hu63] that the number of facets is bounded by 3 (Note that this gives another algorithm for undirected two source-sink pairs). However we show that in the case of directed networks, $\mathcal{P}$ can have exponentially many facets ([CDV]). Thus simply enumerating over all the facets won't give a polynomial time algorithm. We get around this via binary search as follows.

Number the facets by increasing α values. In order to use binary search to find the right facets, we establish a monotonicity between the number of the required facets and the ratio of the money of the two agents. The next difficulty is that we don't know of any efficient procedures for counting the number of facets or for finding the kth facet. Instead, we do a binary search on α rather than on the order of the facets, to find the right facets. For this purpose, we give an efficient procedure that given an α, gives the two (or one) facet in the immediate neighborhood of α. We also show that the size of α, which is a real number in general, is polynomially bounded in the capacity allocation market case, and in general is polynomially bounded when the markets can be described via combinatorial LP's (We refer the reader to Section 4 for more details). Thus the binary search algorithm(which runs in time polynomial the size of α) runs in polynomial time.

Finally, we translate prices on facets to prices on edges as follows. Corresponding to each facet, we give an LP whose dual assigns weights to the edges of the network. The price of each facet, multiplied by these dual variables, yields prices of the edges. (Refer to the full paper [CDV] for details).

2 Definitions and Results

Jain and Vazirani [JV] define a class of abstract markets, called the Eisenberg-Gale or EG Markets.

Definition 1. EG Markets. *An EG Market $\mathcal{M}$ with the set of buyers (agents) $[n]$ is such that the set of feasible utilities of the buyers $\boldsymbol{u} \in \mathbf{R}_+^n$ for $\mathcal{M}$ is captured by a polytope $\mathcal{P}$ defined by linear equations of the form*

$$\forall\ j \in J, \sum_{i \in [n]} a_{ij} u_i + \sum_{k \in K} a_{kj} t_k \le b_j,$$
$$\forall i \in [n], k \in K,\ u_i, t_k \ge 0,$$

such that it satisfies the following two conditions:

- Free disposal: *if $\boldsymbol{u}$ is feasible, then so is any other $\boldsymbol{u}'$ dominated by $\boldsymbol{u}$.*
- Utility Homogeneity: *for all $j \in J$, if for some $i \in [n]$, $a_{ij} > 0$ then $b_j = 0$.*

The auxiliary variables t_k might be used for instance, to give a more efficient representation of the feasible region, or as a means to provide semantics for the market. For example, in the Fisher model of a market where there are buyers and divisible goods, the auxiliary variables denote the amount of each good every buyer gets.

An instance of $\mathcal{M}$ is given by the moneys $\boldsymbol{m}$ of the buyers. The equilibrium utility allocation of an EG market is captured by the following convex program similar to the one considered by Eisenberg and Gale [EG59] for the Fisher market with linear utilities.

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{n} m_i \log u_i \\ \text{subject to} & \forall j \in J,\ \displaystyle\sum_{i \in [n]} a_{ij} u_i + \sum_{k \in K} a_{kj} t_k \le b_j, \\ & \forall i \in [n], k \in K,\ u_i, t_k \ge 0. \end{array}$$

Remark: Since the equilibrium of an EG market is captured by a convex program, the equilibrium always exists (even if the constraints are not finite). Given proper separation oracles, the equilibrium could also be approximated to arbitrary small additive error via the ellipsoid method. Moreover, since the objective function above is strictly concave, the equilibrium is unique.

Agent preferences are traditionally represented by a utility function for goods. However in some cases, like in the capacity allocation market described in Section 1.1, it might be more convenient to simply represent the set of all feasible utilities of agents, thus abstracting away the semantics of the market. In EG Markets, the notion of a good being bought or sold has been subsumed by the various constraints on the utilities of agents. As a result, these markets can be manipulated and reasoned with in an abstract setting.The usefulness of such a setting is demonstrated by the general nature of the result we obtain.

Since there are no goods in EG markets, each agent instead pays for the constraints influencing his utility. Thus, each constraint has a *price*. Interpreting the prices as Lagrangian variables and applying the Karasch-Kuhn-Tucker (KKT) conditions (KKT conditions provide a characterization of the optimum solution in convex programs. See, for example, [BV04]) we get the following equivalent definition of an equilibrium allocation in EG markets.

Definition 2. *A feasible utility* $\boldsymbol{u}$ *is an* equilibrium allocation *if there exist witness* $\boldsymbol{t} \in \mathbf{R}_+^K$ *and prices* $\boldsymbol{p} \in \mathbf{R}_+^J$ *such that*

- $\forall\ i \in [n], m_i = rate(i)u_i$, *where* $rate(i) = \sum_j a_{ij}p_j$.
- $\forall\ j \in J, p_j > 0 \implies \sum_{i\in[n]} a_{ij}u_i + \sum_{k\in K} a_{kj}t_k = b_j$.
- $\forall\ t \in K, t_k > 0 \implies \sum_{k\in K} a_{kj}p_j = 0$, *and* $\sum_{k\in K} a_{kj}p_j \geq 0$ *otherwise.*

In an equilibrium allocation, all money of each agent must be exhausted. This is captured by the first requirement above. Moreover, if a constraint is priced, then it must not be "under utilized" and the second requirement above implies this.

The third condition above is a technicality which arises due to the auxiliary variables. Think of these variable as corresponding to dummy agents with no money. The third condition above states that if an allocation gives any "utility" to a dummy agent, the total price paid by him must be zero. In concrete instances of markets, this condition normally translates to the premise that in equilibrium an agent chooses the *best* bundle of goods. For example, in the Fisher market, the third condition would imply that each buyer buys goods of maximum "bang-per-buck"; in the capacity allocation market of Section 1.1, the condition corresponds to the fact that each agent chooses the cheapest source-sink path.

We now give more examples of EG markets. We have already seen the Fisher market with linear utilities and the capacity allocation market are examples.

Examples of EG Markets

1. Fisher markets with linear, Leontief and linear substitution utilities. The first two are special cases of the last. A linear substitution utility is of the form
$$u(\boldsymbol{x}) = \min_k \sum_j u_j^k x_j,$$
where, x_j is the amount of good j allocated to the buyer. A linear utility is one of the form $\sum_j u_j x_j$. A Leontief utility is of the form $u(\boldsymbol{x}) = \min_j x_j/a_j$.
2. The Network Coding Market. We are given a directed graph $G = (V, E)$; E is the set of resources, with capacities $c : E \to \mathbf{R}_+$. The set V is partitioned

into two sets, terminals and Steiner nodes, denoted T and R, respectively. A set $S \subseteq T$ is the set of sources with money m_v, $v \in S$ specified. Source v broadcasts messages to all terminals at rate r by picking a *generalized branching rooted at* v: a fractional subgraph of G specified via a function $b : E \to \mathbf{R}_+$ such that $b(e) \leq c(e)$ for all edges e and a flow of r units is possible in the subgraph from v to every terminal u. Generalized branchings rooted at vertices of S, $b_1, \ldots, b_k$ are said to form a *feasible packing* for G if

$$\forall e \in E, b_1(e) + \ldots + b_k(e) \leq c(e).$$

Edge e is said to be *saturated* if this inequality holds with equality. Given prices p_e for $e \in E$, the price of generalized branching b is defined to be $\sum_{e \in E} b(e)p_e$.

The network coding market asks for a feasible packing of generalized branchings and prices on edges such that

- The generalized branchings rooted at each source are cheapest possible.
- Only saturated edges have positive prices.
- The money of each source is fully used up.

We denote the class of EG markets with k buyers as EG$[k]$. Recall markets having rational equilibrium allocations are called rational markets.We prove the following theorem in Section 3.

Theorem 1. *EG[2] markets are rational.*

Note that the polytope of feasible utilities can be described by a linear program. If this linear program is combinatorial[1], then we call the EG market corresponding to it a *combinatorial* market. In Section 4, we give a strongly polynomial time algorithm to find the equilibrium prices for combinatorial EG[2] markets.

Theorem 2. *If an EG[2] market is combinatorial, then the equilibrium prices can be found in strongly polynomial time.*

3 Rationality of EG[2] Markets

The main results of this section are that EG markets with 2 agents are rational. Let the polytope of feasible utilities be

$$\mathcal{P} = \{\boldsymbol{x} : A\boldsymbol{x} \leq \mathbf{b}, \boldsymbol{x} \geq 0\},$$

with $u_1 = x_1$ and $u_2 = x_2$ being the utilities of agents 1 and 2 respectively and the rest being auxiliary variables. Let $\mathbf{c}$ be a vector such that $c_1 = 1, c_2 = \alpha$, and $c_i = 0$ otherwise. This is defined so that $\mathbf{c} \cdot \boldsymbol{x} = u_1 + \alpha u_2$. Let $\mathcal{L}(\alpha) = \max\{\mathbf{c} \cdot \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\} = \min\{\mathbf{b} \cdot \mathbf{y} : \mathbf{y} \in \mathcal{D}\}$, where $\mathcal{D}$ is the dual polytope $\{\mathbf{y} : A^T\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq 0\}$. In particular, $\mathcal{L}(0) = \max\{u_1 : \boldsymbol{x} \in \mathcal{P}\}$ and $\mathcal{L}(\infty) = \max\{u_2 : \boldsymbol{x} \in \mathcal{P}\}$.

[1] An LP of the form $\max\{cx : Ax \leq b, x \geq 0\}$ is combinatorial if the entries in A have binary encoding length polynomial in the dimension of A.

Let the projection of $\mathcal{P}$ on (u_1, u_2) be

$$\mathcal{P}_u = \{(u_1, u_2) : u_2 \leq \beta_0, u_1 + \alpha_l u_2 \leq \beta_l, 1 \leq l \leq m, u_1 \leq \beta_{m+1}\}.$$

Observe that $\beta_l = \mathcal{L}(\alpha_l)$ for all $0 \leq l \leq m+1$ if we define $\alpha_0 = \infty$ and $\alpha_{m+1} = 0$. We may assume that we only consider facet inducing inequalities: for all $1 \leq l \leq m$, $u_1 + \alpha_l u_2 = \beta_l$ is a facet of $\mathcal{P}_u$. Call it facet l. Without loss of generality, assume that the α_l's and β_l's are strictly decreasing.

We assert that $\mathcal{P}_u$ defines the same market as $\mathcal{P}$. Moreover, as mentioned in Section 1.1, when we price the constraints (facets) in $\mathcal{P}_u$, these prices can be used to get the prices for constraints of $\mathcal{P}$. Moreover if the prices of the facets are rational, then so are the prices of constraints in $\mathcal{P}$. For more details, refer the full paper [CDV]. Thus in the remaining of the paper, we discuss methods of pricing the facets.

In the remaining of the section we show that no matter what the moneys of the two agents are, at most two facets need to be priced. Indeed these prices appear as variables in simultaneous linear equations and thus are rational. Although the number of facets in the projection $\mathcal{P}_u$ maybe exponential (in the full version [CDV] we do give the construction of such an example), in Section 4, we show how to find these prices in polynomial time.

Definition 3. *Let* $(\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_m, \alpha_{m+1})$ *with* $\alpha_l > \alpha_{l+1}$ *be the* profile *of* $\mathcal{P}_u$ *which completely describes it.*

Let the facets l and $l+1$ intersect at the point (u_1^l, u_2^l). Thus the endpoints of facet l are (u_1^{l-1}, u_2^{l-1}) and (u_1^l, u_2^l). Associate subintervals of $[0,1]$ to the facets as follows.

Definition 4

$$\forall 1 \leq l \leq m, I_l := \left[\frac{u_1^{l-1}}{\beta_l}, \frac{u_1^l}{\beta_l}\right], I_{l,l+1} := \left[\frac{u_1^l}{\beta_l}, \frac{u_1^l}{\beta_{l+1}}\right].$$

$$I_{0,1} := \left[0, 1 - \frac{\alpha_1 \beta_0}{\beta_1}\right].$$

The main idea is that if m_1, m_2 are the moneys of the two agents, then $\frac{m_1}{m_1+m_2}$ falls in exactly one of the intervals I_l or $I_{l,l+1}$. In the first case, we price only the facet l, while in the second we price only the facets l and $l+1$. We now make this precise in a series of lemmas.

Lemma 1. *If* $\frac{m_1}{m_1+m_2} \in I_l, 1 \leq l \leq m$, *then* $p_l = \frac{m_1+m_2}{\beta_l}$ *(and 0 otherwise) is an equilibrium price.*

Proof. Its not too hard to check that the utilities $u_1^* := m_1/p_l$ and $u_2^* := m_2/(\alpha_l p_l)$ are equilibrium utilities and lie on facet l.

Lemma 2. *If* $\frac{m_1}{m_1+m_2} \in I_{l,l+1}, 1 \leq l \leq m$, *then there exists an equilibrium price with only* p_{l+1} *and* p_l *having non-zero prices.*

Proof. The equilibrium utility allocation is (u_1^l, u_2^l). To show this, we want p_l and p_{l+1} that satisfy the following two equations. $m_1 = u_1^l(p_l + p_{l+1})$, and $m_2 = u_2^l(\alpha_l p_l + \alpha_{l+1} p_{l+1})$. Note that this system of two equations in two unknowns has a unique solution since they are linearly independent and are positive exactly when $\frac{m_1}{m_2} \in \left[\frac{u_1^l}{\alpha_l u_2^l}, \frac{u_1^l}{\alpha_{l+1} u_2^l}\right]$, which happens when $\frac{m_1}{m_1+m_2}$ is in the interval $I_{l,l+1}$.

$I \leq I'$ means interval I ends where I' begins. $I < I'$ means interval I ends before I' begins. $I \leq x$ means interval I ends before or at x. $x \leq I$ means interval I starts after or at x. We note the following for future reference.

Observation 1

$$I_l \leq I_{l,l+1} \leq I_{l+1}.$$

Proof of Theorem 1. Proof follows from noting that the intervals I_l, for $1 \leq l \leq m$, and $I_{l,l+1}$, for $0 \leq l \leq m$, cover the entire unit interval (Observation 1). Thus for any instance of moneys, the equilibrium prices are rational. ∎

4 Algorithms for Combinatorial EG[2] Markets

4.1 Binary Search Algorithm

In this section we give a binary search algorithm for finding equilibrium prices. We also give a strongly polynomial time algorithm for finding the equilibrium prices in EG[2] markets that are combinatorial. The algorithm takes as input, the moneys of the buyers, m_1 and m_2, a description of the polytope $\mathcal{P}$, and two parameters, M and ϵ such that we are guaranteed that $M \geq \alpha_1$, and $\alpha_l - \alpha_{l+1} \geq 2\epsilon$ for all l.

We now describe the algorithm at a high level. The algorithm does a binary search on α. First, it finds the facets adjacent to α, say l and $l+1$ such that $\alpha \in [\alpha_l, \alpha_{l+1}]$, and their endpoints. Now, it checks if the equilibrium can be attained by pricing these two facets, using Lemmas 1 and 2. If yes, the algorithm outputs those prices and halts. Otherwise, the monotonicity of the intervals (Observation 1) allows us to restrict our attention to a smaller range. The rest of the section describes how to implement Lines 2 and 3 in Algorithm 1. Let the entries of the matrix A be $A_{(i,j)} = a_{ij}$. Recall that $\mathcal{P} = \{\boldsymbol{x} : A\boldsymbol{x} \leq \mathbf{b}, \boldsymbol{x} \geq 0\}$ and $\mathcal{D} = \{\mathbf{y} : A^T\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq 0\}$. Also recall $\mathcal{L}(\alpha) = \max\{\mathbf{c} \cdot \boldsymbol{x} : \boldsymbol{x} \in \mathcal{P}\} = \min\{\mathbf{b} \cdot \mathbf{y} : \mathbf{y} \in \mathcal{D}\}$ Given any $\boldsymbol{x} \in \mathcal{P}$, define the polytope $\mathcal{Q}(\boldsymbol{x})$ as the set of all vectors $(\mathbf{y}, \alpha)$ that satisfy

$$\forall i,\ \sum_j a_{ij} y_j \leq c_i, \qquad \forall j, y_j \geq 0.$$

$$\forall i:\ x_i > 0, \sum_j a_{ij} y_j = c_i,$$

$$\forall j:\ \sum_i a_{ij} x_i < b_j, y_j = 0.$$

Note that the first two constraints imply that $\mathbf{y} \in \mathcal{D}$. The last two constraints imply that $\boldsymbol{x}$ and $\mathbf{y}$ satisfy the complementary slackness conditions. However, in $\mathcal{Q}(\boldsymbol{x})$, α is treated as a variable. The algorithm to find the facets adjacent to any given α makes use of the following Lemmas 3 and 4. The proofs are not too hard and are omitted in the extended abstract for sake of brevity.

input $m_1, m_2, \mathcal{P}, M, \epsilon$.
⋮
$U \leftarrow M$;
$L \leftarrow 0$;
$\rho \leftarrow \frac{m_1}{m_1+m_2}$;
repeat
  $\alpha \leftarrow (U + L)/2$;
  Find l such that $\alpha \in [\alpha_l, \alpha_{l+1}]$;
  Find the endpoints of the facets l and $l + 1$;
  if $\rho \in I_l \cup I_{l,l+1} \cup I_{l+1}$ **then**
    Assign prices to the facets l and $l + 1$ as in Lemmas 1 and 2, and halt;
  end
  else if $\rho < I_l$ **then**
    $L \leftarrow \alpha_l$;
  end
  else
    $U \leftarrow \alpha_{l+1}$;
  end
until $U - L < \epsilon$;

Algorithm 1. The Binary Search Algorithm

Lemma 3. *Let x be any feasible extension of (u_1^l, u_2^l), that is $x \in \mathcal{P}$, $x_1 = u_1^l$ and $x_2 = u_2^l$. Then $\alpha_l = \min\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$, and $\alpha_{l+1} = \max\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$.*

Lemma 4. *$\mathcal{L}(\alpha) = u_1^l + \alpha u_2^l$ if and only if $\alpha \in [\alpha_l, \alpha_{l+1}]$.*

Now given α, one can find the facets adjacent to it, that is, l such that $\alpha \in [\alpha_l, \alpha_{l+1}]$. First find x that maximizes $cx = u_1 + \alpha u_2$ such that $x \in \mathcal{P}$. Then find $\alpha_l = \min\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$, and $\alpha_{l+1} = \max\{\alpha : (y, \alpha) \in \mathcal{Q}(x)\}$. We now give a lemma that enables us to find the endpoints of a facet.

Lemma 5. *$\mathcal{L}(\alpha_l + \epsilon) = u_1^{l-1} + (\alpha_l + \epsilon)u_2^{l-1}$ and $\mathcal{L}(\alpha_l - \epsilon) = u_1^l + (\alpha_l - \epsilon)u_2^l$.*

Let T be the time required to optimize any linear objective function over the polytopes $\mathcal{P}$ and $\mathcal{Q}(x)$. The following theorem characterizes the running time of the algorithm.

Theorem 3. *The running time of the algorithm is $O\left(T \log\left(\frac{M}{\epsilon}\right)\right)$.*

4.2 Combinatorial Markets

In this section, we show that for combinatorial EG[2] markets, the equilibrium price can be found in strongly polynomial time. Let $\nu(\cdot)$ denote the binary encoding length.

Lemma 6. *$\forall\ l, \nu(\alpha_l) = \nu(A)^{O(1)}$. That is, the size of the α_l's is polynomially bounded in the size of the matrix entries.*

Proof. Note that $\mathcal{Q}(x)$ is described by the a_{ij}'s. Theorem follows from Lemma 3 and standard application of Cramer's rule.

Lemma 7. *One can find M and ϵ such that* $\log\left(\frac{M}{\epsilon}\right) = \nu(A)^{O(1)}$.

Proof. Let c be the constant in the $O(1)$ in Lemma 6. M can be chosen to be the largest integer with a binary encoding length $\nu(A)^c$. Clearly $\alpha_1 \leq M$. ϵ can then be chosen to be $1/(2M)$. α_l's have their denominators at most M and hence $\alpha_l - \alpha_{l+1} \geq 1/M = 2\epsilon$.

Theorem 2 follows from this lemma and Theorem 3. As a corollary, we get that there is a strongly polynomial time algorithm for the capacity allocation market in directed graphs with two source-sink pairs and the network coding market in a directed network with two sources.

5 Discussion

In this paper, we extend the study of Eisenberg-Gale markets defined by [JV] and prove that EG[2] markets have rational equilibrium. Moreover, when the polytope describing the feasible utilities is combinatorial, we provide a strongly polynomial time algorithm to obtain the equilibrium allocation.

Our resolution of the open problems from [JV] raises some interesting questions. The restriction of Fisher's linear utilities market to two agents is an EG[2] market which does not admit a combinatorial LP; however it does admit a strongly polynomial algorithm, since Deng, Papadimitriou and Safra [DPS02] have shown that Fisher's linear utilities market on a bounded number of agents always has a strongly polynomial algorithm. Do all markets in EG[2] admit strongly polynomial algorithms? Alternatively, can some evidence be given to establish the contrary?

For EG[k] with $k > 2$, [GJTV05, JV] showed that the equilibrium can be irrational, which implies that an exact strongly polynomial time algorithm for finding equilibria is not expected. By definition, one can approximate the equilibria up to an arbitrary additive error via the ellipsoid method. Is there a strongly polynomial time algorithm which approximates the equilibria? Our method of iterating over facets does not seem to generalize to higher k. A positive answer to this question would probably involve designing new tools which could be useful for solving convex programs as well.

References

[BV04] Stephen Boyd and Lieven Vandenberghe. **Convex Optimization** Cambridge University Press, 2004.

[CDV] D. Chakrabarty, N. Devanur, and V. V. Vazirani. Eisenberg-gale markets: Rationality, strongly polynomial time solvability and competition monotonicity. Manuscript available from `http://www.cc.gatech.edu/~nikhil/`.

[DJP+94] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[DPS02] Xiaotie Deng, Christos Papadimitriou, and Shmuel Safra. On the complexity of equilibria. In *Proceedings of ACM Symposium on Theory of Computing*, 2002.

[EFS56] P. Elias, A. Feinstein, and C. E. Shannon. Note on maximum flow through a network. *IRE Transactions on Information Theory IT-2*, pages 117–199, 1956.

[EG59] E. Eisenberg and D. Gale. Consensus of subjective probabilities: the Pari-Mutuel method. *The Annals of Mathematical Statistics*, 30:165–168, 1959.

[FF56] L. R. Jr. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, pages 99–404, 1956.

[GJTV05] D. Garg, K. Jain, K. Talwar, and V. Vazirani. A primal-dual algorithm for computing Fisher equilibrium in absence of gross substitutability property. In *Proceedings of wine*, 2005.

[GVY94] N. Garg, V.V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *Proc. 21st International Colloquium on Automata, Languages, and Programming*, volume 820 of *Lecture Notes in Computer Science*, pages 487–498. Springer-Verlag, Berlin, 1994.

[Hu63] T.C. Hu. Multicommodity network flows. *Operations Research*, 14:344–360, 1963.

[JV] K. Jain and V. V. Vazirani. Combinatorial solvability in resource allocation markets, with applications to efficiency and fairness. Manuscript, 2005 available from `http://www.cc.gatech.edu/∼vazirani/`.

[Kel97] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[NW61] C. S. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. London Math Soc*, 36:445–450, 1961.

[Tar86] E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34(2):250–256, 1986.

[Tut61] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *J. London Math Soc*, 36:221–230, 1961.

Making Economic Theory Operational

Xiaotie Deng

Department of Computer Science
City University of Hong Kong, Hong Kong
deng@cs.cityu.edu.hk

Abstract. We focus on the opportunity the Internet has provided for a fully operational Economic Theory. We should review a few of the current development of the algorithmic approach in its study and postulate on the important issues that lie ahead of us.

The Internet has now secured its place as a major global platform for commerce, with the help of great information and communication technology advances. Bill Gates of Microsoft describes such a market as a friction-free Capitalism [1], citing the availability of the huge communication bandwidth provided by today's network that has significantly reduced the information barrier. In addition, electronic trading processes have made economic operations accessible automatically at micro-level operations, and with significantly reduced transaction costs. Program trading allows optimization techniques be applied in precision that closely matches that of perfect rational decision makers. We are closer to the Economists' dream of a perfect market than ever before. The perfect market would allow for a state of equilibrium where prices and allocations of commercial products are determined in a way, dependent on the market model, such that all goods are cleared and no individual would be better off changing to another feasible state.

How to reach such an ideal state is another matter that has been a major source of disagreement among economists. A tatonnement process was originally proposed by Walras to allow a fictitious auctioneer to propose a sequence of virtual prices until the price converges to an equilibrium. Technically, the idea can actually be implemented by a continuously declared price during the pre-opening period of a computerized trading system, such as Oskar Lange had advocated since the dawn of the computer age [2]. Alternatively, Herbert Scarf developed methodologies for the use of numeric algorithms to compute the equilibrium price vector and commodity allocations in general equilibrium systems [3].

But there is a huge gap between an existential theorem for which a solution can be found in a finite number of convergence steps and computationally feasible solutions. In theoretical computer science, the latter is suggested to be characterized by polynomial time algorithms to derive output from input data which may be regarded as a rational computational resource bound in decision making [4,5]. However, the pervasiveness of the negative results in the possibility of a

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 251–261, 2006.
© Springer-Verlag Berlin Heidelberg 2006

polynomial time algorithm for general economic problems has brought in a sense of frustration and may cast doubt on the methodology. Many of the solution concepts in cooperative game theory are known to be hard to evaluate in general [6]. The existence of the core [7], a central solution concept in cooperative game theory, as well as that of Walrasian equilibrium [8], depends on whether a linear system has an integer solution, a condition that is in general NP-hard to decide. General equilibrium price determination is difficult for many cases [9,10,11], even though solvable in polynomial time for some limited utility functions [12,13,14]. Even arbitrage is proven, under various realistic constraints, to be hard to determine [15]. Recently, computation of Nash equilibrium for a constant number of players (all the way down to two players) is proven to be PPAD-complete (equivalent to the computation of the fixed point problem) [16,17], where the two player case was previously expected to be polynomially solvable. Though immediate interests have been drawn to such astonishing results, gradually serious doubt may edge in to question the usefulness of this new paradigm, borrowed from computer science to study fundamental topics in Economics, previously regarded as well solved in consensus of the community.

Under such circumstance, a major challenge that lies in front of our paradigm is whether we can make useful contributions to Economics, both in theory and in practice. In some sense, the complexity approach faces a situation similar to the impossibility theorem of Kenneth Arrow which investigates the consistency of axioms that govern the relationship of the collective decision of a group with the free wills of its constituent individuals. Its impossibility in the unrestricted domain without a dictator has since aroused intensive interests in the subject. At the first sight, every effort trying to turn the impossible around seems quite naive and stubborn minded. The simply stated and clearly proven theorem seems to make it pale all arguments to mend the brutal truth by replacing domain conditions or restricting decision making processes. To many, it spells the end to a potential social choice theory. Surprisingly, further studies under the framework set by Arrow have developed into the commonly referred to as Arrovian Social Choice. A particularly prominent role is played by Amartya Sen, who carried the approach of interpersonal comparisons of utility on to escape from the impossibility and to study collective choices based on real life necessities [18]. In another direction, James Buchanan and Gordon Tullock studied the political decision making process, in particular that of the US, as a market institution involved with individuals' valuation of public goods and their shares of the associated costs [19]. In both cases, the advances in our knowledge of the relationship of the public institutions and the individuals of the society have been guided in clear comparison with the Arrow's celebrated theorem.

In contrast, we are in a much better situation in regard to the use of the algorithmic complexity in the classical model of Economics. The methodology does not only result in negative conclusions, but it also brings in a rich class of methodologies such as randomized and approximate algorithms that have successfully tackled computationally difficult problems. The positive results in algorithms for problems in Economics are abundant. They encourage and foster positive interactions with Economics in complement to the role of negative complexity results which point to the necessity of active effort to ensure the efficiency and stability of economic systems, such is the case of E-commerce enabled through the Internet. Setting it in a different aspect, the difficulty in deriving equilibrium may delay the convergence toward equilibrium but at the same time it creates more potential profitable opportunities for sophisticated profit seekers. Exactly because of the computational difficulties, we would expect to see, or putting it more proactively, we would advocate heavy involvement of computing resources and efforts in the electronic economic systems. Finally, in new Economic activities seen in adword auctions, as well as in virtual Economics as of online games, we encounter completely new economic environments, where computer science has been making great contributions in creating new theories and in providing new efficient protocols.

In this talk, I focus on a selection of topics in this exciting interaction that I am most familiar with or have participated in. I shall also limit myself to issues that are well formulated and fundamental. Current success and open problems will be discussed but without detailed introduction of the mathematical and computational complexity concepts. The readers not familiar with the jargons would be referred to the relevant literatures. We shall start with Nash equilibrium for which the recent development has provided us a clear picture of the computational complexity. Nash equilibrium is an example where we have seen the most creative interaction of complexity theory and game theory, which can well be described as " Game World is Flat" [20]. Next, we discuss the price equilibrium of market where there has been continuous progress in the past few years in characterizing its computational complexity under different constraints. Finally, we consider operational issues related to market processes, such as incentive compatibility and arbitrage.

1 The Power of PPAD on Nash Equilibrium

The computational complexity of non-cooperative game in bimatrix form has been a long standing open problem known since Lemke-Howson's algorithm was first designed, which was shown to have a class of examples to require an exponential number of steps even for the best starting position by Rahul Savani and Bernhard von Stengel [21]. In addition, Nash equilibrium with various restrictions, most noticeably the uniqueness of Nash equilibrium, are known to be NP-hard previously due the wonderful work of Gilboa and Zemel [22]. The problem that stood open for a long time is the computational complexity of finding a Nash equilibrium that is guaranteed by Nash's original theorem.

The root of the solution was initiated in 1991 by Papadimitriou in his introduction of the complexity class PPAD (polynomial parity argument, directed), which turns out to be exactly the class for Nash equilibrium in bimatrix form as revealed by a series of recent works, starting with the big breakthrough of the four player case by Daskalakis, Goldberg, and Papadimitriou [16], and end with the two player game PPAD-hard proof by Chen and Deng [17]. The result places

PPAD at the center of the algorithmic study of the Nash equilibrium problem. In addition, for games with a large number of players and a succinct representation, the same class PPAD fully characterizes their computational complexity, as summarized in "Game World is Flat" by Daskalakis, Fabrikant and Papadimitriou [20].

The concept and techniques developed here turned out powerful enough to move the frontier of another very important field, smoothed analysis, significantly. Chen, Deng and Teng [23], with several other important new innovative ideas, proved that Nash equlibrium has no polynomial smoothed algorithms unless PPAD is in RP, as a next significant result to complement the polynomial time smoothed complexity of linear program (or zero sum games) by the pioneer work of Spielman and Teng [24] in this field. The work is also extendable to win-loss game with new ideas [25] built on a connection by Tim Abbott, Daniel Kane, Paul Valiant [26]. The result can also be applied to derive similar results for General equilibrium by Huang and Teng [27].

In comparison with other problems we are going to discuss here, PPAD provides Nash equilibrium with a clean complexity characterization. The clear next frontier is whether there is a polynomial time approximation as we know no full polynomial time approximation exists unless PPAD is in P [23]. In some sense, the PPAD complete proof of 2NASH may not necessarily be a negative result as games with many players are equivalent in complexity to games with two players. This result itself invites further algorithmic studies in term of two player games. Two results [28,29] in this promising direction have already appeared in the same conference WINE 2006 with the first constant approximation-ratio algorithms for two player Nash Equilibrium. They are only the beginning of the future explorations into the question whether a polynomial time approximation scheme can be found for two player Nash equilibrium, though it is known not possible to have a fully polynomial time approximation scheme. In addition, questions also open up whether absolute constant approximation algorithms for 2NASH would lead to approximation algorithms with more players for which the exact solution is equivalent to, in a way similar to the MAX-SNP concept of equally approximable algorithms? Finally, it also poses a challenge to the complexity community to exactly characterize PPAD against other complexity classes.

There have been several important innovations in this series of work leading to the proof of the PPAD characterization of two player Nash equilibrium [16,17,23]. The most non-intuitive one is the reliance on a new concept of the approximate Nash equilibrium introduced in [16]. An interesting question is whether we can derive a direct reduction for the exact Nash equilibrium? In addition, a tricky methodology in the proofs is the definition and the use of discrete fixed points in the proof. A discrete fixed point is defined on a set of vertices on a unit cube [16] (or $k+1$ point on a k-dimension unit cube [23]) satisfying a property that all the function values are present on those vertices. Note that the range of functions considered are $\{e_1, e_2, \cdots, e_k, -\sum_{i=1}^{k} e_i\}$, where e_i is the unit vector in the positive direction of the i-th coordinate. In the reduction proof, however, a restricted version was used. In [16,17], such a set of vertices on a unit cube allow for an averaging cube that sums up to zero of the functions values on its points. In [23], the $k+1$ vertices of the cube allow for an equiangular line segment that averages to zero of the functions values on its points. Such a strange approach simplifies the proofs significantly, and seems to be necessary for the proof of the no FPTAS results in [23] though alternative approaches are possible for the exact NASH results via a concept [30] of badness of cube-function pairs similar to that of degree theory of Brouwer. It would be mathematically interesting to know the structures of the set of fixed points that averaging to zero on a small cube or an equiangular line segment. Finally, the encoding of Boolean variables using Nash equilibrium probability values in paired strategies is the main idea that made it possible to carry out the proof of PPAD-completeness of the two player Nash equilibrium. Such encoding is still efficient with respect to polynomial factor approximation [23] but seems not possible to extend further. It would be interesting to develop encoding schemes for more players which parameterize with respect to subpolynomial approximation factor schemes.

2 Is There a Central Complexity Theorem for General Equilibrium?

Deng, Papadimitriou and Safra started the current effort, and explored several aspects, of complexity issues of the general equilibrium pricing problem [9]. The algorithmic issue of the standard continuous variable competitive general equilibrium has since attracted a lot of efforts and works starting with the linear utility case (for the Fisher model by Devanur, Papadimitriou, Saberi and Vazirani [13], and for the Arrow Debreu model by Jain [14]). An excellent survey conducted by Codenotti, Pemmaraju and Varadarajan made an exploration of connections of the recent effort with previously mathematical works (e.g., that of Nenakov and Primak [12]), cast in the context of algorithmic complexity [31], and discussed some of the excellent works done upto then, as well as proposed several interesting open problems. Since then, a brilliant discovery by Ye of a link of a linear complementary relation to the solutions to an exchange market equilibria with Leontief utilities, has allowed computational complexity results known for various versions of Nash equilibrium extended to the computational complexity results for general equilibrium pricing of continuous variables [10,11]. It also naturally extends the PPAD-hardness results for Nash equilibrium to the exchange economy's general equilibrium. The non-approximability results and the negative smoothed complexity results can also be carried out through this connection by a significant effort of Huang and Teng [27].

As the mathematical methodologies, and algorithms, for these two problems are closedly related, it is reasonable to see related computational complexity results. Unlike Nash equilibrium, however, there is a lack of a clear frontier in the computational complexity of the general equilibrium pricing problem. The positive results are much related to convex program formulations of the linear utility function case [12,14], and its extensions (often related to Fisher utility functions). On the other hand, the negative results are based on the connection with Nash equilibrium [10,11]. Between those two special forms of utility functions, there are still many possibilities.

An obvious open problem in the computational complexity issue of the general equilibrium problem is the possibility of a sharp result that separates polynomial time solvability and otherwise (either by NP-completeness or by PPAD-completeness). The most convincing positive result would be expected to match up to the work of Arrow and Debreu in the existential theorems [32], and complemented by a negative complexity result, most probably of PPAD-hardness, for slightly more general utility functions. One immediate technical open question is whether we would be able to explore the Uzawa's solution of the fixed point problem using Walrasian equilibrium [33] to a tighter PPAD-hardness result than those already known.

Finally, other frontiers studied in [9] would be worthy of further explorations, such as the complexity issues involved with integer variables and communication complexity in the framework of the general equilibrium paradigm. In addition, with the communication pattern well spelt out on the Internet, it would be feasible to study, in greater depth than ever before, how would communication protocols affect the equilibrium price process.

3 Market Forms and Efficiencies

Human commercial activities have shown a rich collection of different institutions. Most of them operate, in theory, on three fundamental principles: non-arbitrageness, competitive equilibrium, and incentive compatibility. The algorithmic complexity paradigm is especially relevant at this operational level of the market mechanisms. While equilibrium is discussed above, we should discuss the remaining two related concepts, arbitrage (the possibility of making a riskless risk) and incentive compatibility (the truth telling properties of agents), in the subsequent subsections.

3.1 Arbitrage and Time Factor in Market

Here we are interested to see how computational complexity varies in different models of exchange markets, and how in reality the exchange markets have evolved. In the simplest model of frictionless markets, an arbitrage opportunity is a three way triangle that can be identified quickly. In a slightly more general model where some currencies can only be changed in one way to another, arbitrage is equivalent to a negative cycle in a directed graph, which can be found, e.g., by dynamic programming techniques, in polynomial time. In addition, such approach remains polynomial even if some forms of friction are allowed.

In reality, however, friction does exist to the extent that the above dynamic programming approach does not work any more. In fact, three particularly interesting constraints are in existence of many exchange markets. We shall adopt the foreign exchange market as an example for the sake of definiteness in terminologies. Integrality is one of the most common constraints in that trading volume is often based on lots of a fixed number of a currency or its round sum multiples as a unit. The bound constraint requires a maximum number of units to be bought or sold in a certain price. One may have to pay more or sell for less in the next levels of trading. Finally, the ask-bid spread constraint characterizes the fact that the buying rate and the selling rate are usually different. Under such a general model of frictional exchange market, it is computationally difficult, more specifically, NP-hard, to find arbitrage opportunity [15]. Even though this may be counter-intuitive to what most would expect, one may relate the result to a barter market where merchants profited significantly so that commerce could take off. Complexity in commercial activities may well be an important factor in the evolution of commodity exchange markets.

Of course, an NP-hardness result may not rule out the possibility that difficult-to-find arbitrage opportunities are insignificant. The hardness result would be deemed useless in presence of such situation. One may argue against such a result that there may exist insignificant arbitrage opportunities but it is not worthwhile to exploit it by spending an enormous amount of computational resource to find it. In this regard, *complexity on approximability offers an answer to the proposal of bounded rationality in computation.* We would then be interested to determine whether an approximation (within in a factor of $(1+\epsilon)$ for some fixed constant ϵ) to the optimal solution is not possible. Therefore, in order that a quest into the arbitrage condition via computational complexity approach conform to the principle of bounded rationality, the issue would be whether we are not able to find a significant large arbitrage opportunities (inapproximability), or more computational time would result in a better approximation (PTAS). This is again NP-hard under the above general assumptions [15].

Still, the hard non-approximability result on arbitrage of in general exchange markets may not necessarily imply that realistic frictional exchange markets are full of arbitrage opportunities. Two models closely related to the current foreign exchange markets do admit polynomial time algorithms for finding arbitrage opportunities: One is the star-shaped exchange market models, corresponding to a major currency and others valuated against the major currency (a model of a dominating currency such as in the case of dollar region or euro region.) Another is a market with a constant number of currencies (but the number of ask/bid rates on the market could be a non-constant). Though the special cases are quite simple, they concur with the reality to some extent. Obviously, when the number of currencies is bounded by a fixed constant, the computation of arbitrage becomes easier than the general case of an unbounded number of currencies. As a matter of fact, we have a moderate number of currencies in the world. Therefore, the complexity of arbitrage may not be as pessimistic as the hardness results show. In addition, the number of currencies tends to be reduced gradually, as in the creation of Euro. In corresponding to the star-shaped digraph, one may also relate the central vertex to the role of money in a commodity market.

The lesson learnt through this study is that, different foreign exchange systems exhibit quite different computational complexities in locating an arbitrage opportunity. They would definitely affect the time to restore to a non-arbitrage state. Such operations issues would be interesting future works. The problem becomes much more involved when futures are introduced. The simple star graph model becomes NP-hard in such case. We would be interested to identify systems that would be easy to located arbitrage in presence of a futures market at exchange systems. Those results may shed new light on how monetary system models were adopted and how they evolved in reality, and are of potential applications in a merging world market.

3.2 Trading Systems and Auction Protocols

The most convincing study of the market process, both in theory and in reality, is that of a single seller, with the second price auction, often referred to as the Vickery auction, based on the concept of incentive compatibility: The highest bidder wins, and pays the seller the price of the second highest bidder. Every buyer would submit its own value of the goods as its bidding price with no regret, as there is no possibility to gain in utility by changing the bid. This concept of incentive compatibility first exemplified its usefulness in Vickery auction has become the theoretical foundation of the mechanism design discipline in Economics and Management science.

The market with many buyers and many sellers, often referred to as a double auction market, has not been characterized as such. The most successful understanding of the double auction model, especially in a dynamic setting, according to Friedman and Rust [34], is by experimental research scientists such as Vernon Smith and Charles Plott, in that the price of goods and their allocations converge to the competitive equilibrium almost universally in continuous auction experimental settings. We have also seen a rich range of theoretical research activities in understanding double auctions but they have not been able to draw as clear a picture of the double auction market as has done by experimental studies. In some sense, there is a long overdue theory on continuous auction markets that would explain reality without bending principles. The emergence of the E-Commerce with the Internet technology over the last decade has created much large double auction markets in a more diversified environment than ever before, bringing in both opportunities for and extra complexities in their challenges to theoretical studies. Such opportunities have already attracted much attention of theoreticians in applying new ideas to the double auction market.

Competitive auction [35], is a particularly interesting approach that aims at maximizing the returns of the seller while maintaining incentive compatibility of the participants. The idea is to achieve a total revenue that is within a constant factor of the optimum for the seller who knows all the private values of the buyers on the goods. There is a technical requirement that the optimum is the single price revenue under the condition that at least two buyers are allocated with the item. The most attractive property is that the revenue is guaranteed to be within a constant factor of that optimum no matter what the private values of the buyers are. The solution is achieved with introduction of a randomized protocol, with a distortion of the market condition: Buyers can be excluded from allocation of the goods even if they bid higher than the trading price. Even though such protocols are quite normal in other resource allocation problems studied in computer science, it is quite unusual in Economics. If it indeed generates more revenue for the seller in reality, there is no reason why the seller would be deterred from applying such protocols, especially in an Internet environment where few laws are available that are globally applicable. It therefore poses a more general challenge to the trading process of the market: To what extent the new approaches can be accepted in the market place? Clearly, if a secondary market is available, a resale of the goods by some of the buyers to those barred from obtaining the goods because of the procedural process would affect the primary market behavior of the participants and would most probably destroy the incentive compatibility property. On the other hand, especially in markets of digital goods, there is a potential that resales could be blocked via technological means. Therefore, there is a possibility that arbitrage arguments for ordinary goods may not be applicable in such cases.

We may not end up with the randomized protocols proposed in the competitive auction approach as in [35]. It needs verification in realistic environment. It may not work well at all. Nevertheless, its challenge to the traditional auction market assumptions and protocols may be much more important than its own practicality. Efficiency and profitability have been the motivations in such newly proposed pricing schemes and protocols, largely due to the monopolistic features of many such markets. Indeed, for information goods and services, many different pricing strategies have already appeared in the market, such as discriminative pricing, fixed fees versus unit pricing, bundling pricing, etc. Huber [36] most succinctly summarizes those as "Information just doesn't obey the ordinary laws of economics...". As it appears that none of the requirements in non-arbitrage, incentive compatibility, competitive price equilibrium cannot be broken in information goods pricing, what would we reliably hold as the fundamental principles and the ultimate building blocks of those new markets?

4 Conclusions

The Internet provided a perfect opportunity for Economic Theories to be tested and applied at a precision that has never been seen before. Algorithmic complexity that developed in the context of Computer Science could play an very important role in the operational issues we see at the micro-economic levels. The negative results in complexity of the economic solutions may have their constructive values, in addition to enrichment our knowledge, in encouraging deeper effort in improving computational methods and computing resources. In addition, a lot of computer science methodologies in dealing with hard problems can also be applied in such situations.

Acknowledgements

Research is supported by a research grant (Project No.7001838) of CityU of Hong Kong. I would like to thank Tian-ming Bu, Xi Chen, Wei Sun for their help and all the discussions related to this work.

References

1. Gates, B.: The Road Ahead. Penguin (1996)
2. Lange, O.: The Computer and the Market. In: Socialism, Capitalism and Economic Growth: Essays Presented to Maurice Dobb. New York: Cambridge University Press (1961) 158–161
3. Scarf, H.: On the Computation of Equilibrium Prices. In: Ten Economic Studies in the Tradition of Irving Fisher. New York: Wiley (1967) 207–230
4. Simon, H.: Models of Man. John Wiley (1957)
5. Papadimitriou, C.H., Yannakakis, M.: On complexity as bounded rationality (extended abstract). In: Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing, Montréal, Québec, Canada (1994) 726–733
6. Deng, X., Papadimitriou, C.H.: On the complexity of cooperative solution concepts. Math. Oper. Res. **19** (1994) 257–266
7. Deng, X., Ibaraki, T., Nagamochi, H.: Combinatorial optimization games. In: Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, New York, Philadelphia, ACM-SIAM (1997) 720–729
8. Bikhchandani, S., Mamer, J.W.: Competitive equilibrium in an exchange economy with indivisibilities. Journal of Economic Theory **74**(2) (1997) 385–413
9. Deng, X., Papadimitriou, C.H., Safra, S.: On the complexity of price equilibria. Journal of Computer System and Science **67**(2) (2003) 311–324
10. Ye, Y.: Exchange market equilibria with leontief's utility: Freedom of pricing leads to rationality. In Deng, X., Ye, Y., eds.: Proceedings of First International Workshop of Internet and Network Economics (WINE-2005). Volume 3828 of Lecture Notes in Computer Science., Hong Kong, China, Springer (2005) 14–23
11. Codenotti, B., Saberi, A., Varadarajan, K.R., Ye, Y.: Leontief economies encode nonzero sum two-player games. In: Proceedings of the seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-06), Miami, FL, ACM Press (2006) 659–667
12. Nenakov, E.I., Primak, M.E.: One algorithm for finding solutions of the arrow-debreu model. Kibernetica **3** (1983) 127–128
13. Devanur, N.R., Papadimitriou, C.H., Saberi, A., Vazirani, V.V.: Market equilibrium via a primal-dual-type algorithm. In: Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS-02), Los Alamitos, IEEE COMPUTER SOCIETY (2002) 389–395
14. Jain, K.: A polynomial time algorithm for computing an Arrow–Debreu market equilibrium for linear utilities. In: Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS-04), Rome, Italy, IEEE Computer Society Press (2004) 286–294
15. Cai, M., Deng, X.: Arbitrage in frictional foreign exchange market. Electronic Notes in Theoretical Computer Science **78** (2003)
16. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. In: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, Seattle, W.A, USA, ACM Press (2006) 71–78
17. Chen, X., Deng, X.: Settling the complexity of 2-player nash-equilibrium. To appear in the 46th Symposium on Foundations of Computer Science (FOCS-06) (2006)
18. Sen, A.K.: Collective choice and social welfare. Holden-Day (1970)
19. Buchanan, J.M., Tullock, G.: The Calculus of Consent: Logical Foundations of Constitutional Democracy. University of Michigan Press (1962)

20. Daskalakis, C., Fabrikant, A., Papadimitriou, C.H.: The game world is flat: The complexity of nash equilibria in succinct games. In Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., eds.: Proceedings of the 33rd International Colloquium of Automata, Languages and Programming (ICALP-2006), Part I. Volume 4051 of Lecture Notes in Computer Science., Venice, Italy, Springer (2006) 513–524
21. Savani, R., Stengel, B.: Hard-to-solve bimatrix games. Econometrica **74**(2) (2006) 397–429
22. Gilboa, I., Zemel, E.: Nash and correlated equilibria: Some complexity considerations. Games and Economic Behavior **1** (1989) 80–93
23. Chen, X., Deng, X., Teng, S.H.: Computing nash equilibria: Approximation and smoothed complexity. To appear in the 46th Symposium on Foundations of Computer Science (FOCS-06) (2006)
24. Spielman, D.A., Teng, S.H.: Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. Journal of the ACM **51**(3) (2004) 385–463
25. Chen, X., Teng, S.H., Valiant, P.: The approximation complexity of win-lose games. To appear in the the eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-07) (2007)
26. Abbott, T., Kane, D., Valiant, P.: On the complexity of two-player win-lose games. In: Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS-05). (2005) 113–122
27. Huang, L.S., Teng, S.H.: On the approximation and smoothed complexity of leontief market equilibria. ECCC (2006)
28. Kontogiannis, S., Panagopoulou, P., Spirakis, P.: Polynomial algorithms for approximating nash equilibria of bimatrix games. In: Proceedings of the 2nd Workshop on Internet and Network Economics (WINE2006), Patra, Greece (2006)
29. Constantinos Daskalakis, A.M., Papadimitriou, C.: A note on approximate nash equilibria. In: Proceedings of the 2nd Workshop on Internet and Network Economics (WINE2006), Patra, Greece (2006)
30. Chen, X., Deng, X.: On algorithms for discrete and approximate brouwer fixed points. In Gabow, H.N., Fagin, R., eds.: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, M.D., USA, ACM (2005) 323–330
31. Codenotti, B., Pemmaraju, S., Varadarajan, K.: The computation of market equilibria. SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory) **35**(4) (2004)
32. Arrow, K., Debreu, G.: Existence of an equilibrium for a competitive economy. Econometrica **22**(3) (1954) 265–290
33. Uzawa, H.: Walras's existence theorem and brouwer's fixed point theorem. Economic Studies Quarterly **13** (1962) 59–62
34. Friedman, D., Rust, J.: The Double Auction Market: Institutions, Theories, and Evidence. Addison Wesley Publishing Company (1993)
35. Goldberg, A.V., Hartline, J.D., Wright, A.: Competitive auctions and digital goods. In: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-01), New York, ACM Press (2001) 735–744
36. Huber, P.: Two cheers for price discrimination. Forbes (September 27, 1993) 142

Sparse Games Are Hard

Xi Chen[1], Xiaotie Deng[2], and Shang-Hua Teng[3]

[1] Department of Computer Science, Tsinghua University, Beijing
xichen00@mails.tsinghua.edu.cn
[2] Department of Computer Science, City University of Hong Kong, Hong Kong
deng@cs.cityu.edu.hk
[3] Department of Computer Science, Boston University, Boston
steng@cs.bu.edu

Abstract. A two-player game is sparse if most of its payoff entries are zeros. We show that the problem of computing a Nash equilibrium remains **PPAD**-hard to approximate in fully polynomial time for sparse games. On the algorithmic side, we give a simple and polynomial-time algorithm for finding exact Nash equilibria in a class of sparse win-lose games.

1 Introduction

Motivated by the growing possibilities in both Internet applications and network computations, Game Theory has attracted a great deal of attention from Theoretical Computer Science community. Central to such game theoretical applications is the problem of computing a Nash equilibrium in a non-cooperative game.

A series of significant progress on the complexity of this problem was initiated by a recent work of Daskalakis, Goldberg, and Papadimitriou [1,2] who introduced a reduction technique and showed that a Nash equilibrium in a four-player game is hard to find, unless **PPAD** [3] is in **P**. Shortly afterward, this hardness result was extended to three-player games [4,5]. Chen and Deng [6] finally settled a long-term open problem, and proved that computing a Nash equilibrium in a two-player game is **PPAD**-complete.

These breakthrough work left the problem of computing approximate Nash equilibria with less than exponential accuracy as a central remaining open question in the area of computing Nash equilibria. In a recent paper [7], we solved this problem by showing that two-player games do not have a fully polynomial-time approximation scheme unless **PPAD** is in **P**. Hence, it is unlikely that the $n^{O(\log n/\epsilon^2)}$-time algorithm of Lipton, Markakis, and Mehta [8], the fastest algorithm known today for approximating Nash equilibria, can be further improved to poly$(n, 1/\epsilon)$. This result also implies that, unlike the simplex algorithm for zero-sum two-player games [9], the smoothed complexity of the classical Lemke-Howson algorithm for non-zero-sum two-player games is not polynomial, unless **PPAD** $\subseteq$ **RP**. Thus the average-case polynomial-time result of Barany, Vempala, and Vetta [10] is not likely extendible to the smoothed model. Recently,

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 262–273, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Chen, Teng, and Valiant [11] extended this result and proved that win-lose two-player games, in which the payoff entries are either 0 or 1, are **PPAD**-hard to approximate in fully polynomial time.

A two-player game is specified by two $m \times n$ matrices $\mathbf{A} = (a_{i,j})$ and $\mathbf{B} = (b_{i,j})$. They state the payoffs when the first player makes a choice of a row and the second player makes a choice of a column. In general, each player can pick a distribution over its choices in advance, and during the playing time, selects a choice according to this distribution, simultaneously. The concept of a Nash equilibrium captures the notion of rational play in such non-cooperative games. It is rather a strong notion of rationality, stating the condition that neither player can gain by changing its own distribution, when the opponent's distribution is revealed. Each two-player game has at least one Nash equilibrium [12].

In this paper, we consider sparse games in which most of the payoff entries are zeros. Particularly, we focus on sparse two-player games in which each row and column of the two payoff matrices has at most a constant number of non-zero entries. We prove that a Nash equilibrium in such sparse games is equally hard to compute and essentially equally hard to approximate as in general two-player games. Our result shows that sparsity alone does not make game easier to solve and that sparse two-player games do not have a fully polynomial-time approximation scheme unless $\mathbf{PPAD} \subseteq \mathbf{P}$.

To establish our complexity result, we construct a set of new arithmetic and logic gadgets, for the reduction from a discrete Brouwer's fixed point problem to an equilibrium computation problem. These new gadgets enable us to reduce the degree of influence in the simulation of arithmetic and logic computations in two-player games, resulting in hard sparse instances.

On the positive side, we give a polynomial-time algorithm for computing an exact Nash equilibrium for a subclass of sparse win-lose games. Our algorithm takes advantage of the 0-1 payoff structure and effectively reduces the computation of a Nash equilibrium of a two-player win-lose game to the computation of an equilibrium in a smaller game. We were informed by the conference committee that Codenotti, Leoncini, and Resta [13] very recently and independently obtained the same result for finding Nash equilibria in this subclass of sparse win-lose games. As our algorithm appears to be simpler than theirs, we decide to keep our algorithm and its analysis in this conference version.

2 Sparse Two-Player Games and Our Main Result

Definition 1 (Sparse Normalized Games). *A bimatrix game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is normalized if every entry of matrices $\mathbf{A}$ and $\mathbf{B}$ is between -1 and 1. A matrix $\mathbf{A}$ is row (column) sparse if there are at most 10 nonzero entries in every row (column). $\mathbf{A}$ is sparse if it is both row sparse and column sparse. A two-player game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is sparse if both $\mathbf{A}$ and $\mathbf{B}$ are sparse.*

We use $\mathbb{P}^n$ to denote the set of all *probability vectors* in $\mathbb{R}^n$, i.e., non-negative vectors whose entries sum to 1. Recall that an *ϵ-approximate Nash equilibrium* of game $(\mathbf{A}, \mathbf{B})$ is a pair $(\mathbf{x}^* \in \mathbb{P}^m, \mathbf{y}^* \in \mathbb{P}^n)$ such that, for all probability vectors $\mathbf{x} \in \mathbb{P}^m, \mathbf{y} \in \mathbb{P}^n$,

$$(\mathbf{x}^*)^T \mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T \mathbf{A}\mathbf{y}^* - \epsilon \quad \text{and} \quad (\mathbf{x}^*)^T \mathbf{B}\mathbf{y}^* \geq (\mathbf{x}^*)^T \mathbf{B}\mathbf{y} - \epsilon.$$

Following [7], an *ϵ-well-supported Nash equilibrium* of game $(\mathbf{A}, \mathbf{B})$ is a pair $(\mathbf{x}^*, \mathbf{y}^*)$, such that for all i, j, $\langle \mathbf{b}_i | \mathbf{x}^* \rangle > \langle \mathbf{b}_j | \mathbf{x}^* \rangle + \epsilon \Rightarrow y_j^* = 0$, and $\langle \mathbf{a}_i | \mathbf{y}^* \rangle > \langle \mathbf{a}_j | \mathbf{y}^* \rangle + \epsilon \Rightarrow x_j^* = 0$, where $\mathbf{a}_i$ and $\mathbf{b}_i$ denote the i^{th} row of $\mathbf{A}$ and the i^{th} column of $\mathbf{B}$, respectively. Motivated by the next lemma proved in [7]. we define the following search problem called SPARSE BIMATRIX.

Lemma 1 ([7]). *In a normalized game $(\mathbf{A}, \mathbf{B})$, for every $0 \leq \epsilon \leq 1$, (1) every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ is also an ϵ-approximate Nash equilibrium; (2) from every $\epsilon^2/(8n)$-approximate Nash equilibrium $(\mathbf{u}, \mathbf{v})$, one can find in polynomial time an ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$.*

Definition 2 (SPARSE BIMATRIX). *The input instance is a bimatrix game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ which is both normalized and sparse. $\mathbf{A}$ and $\mathbf{B}$ are $n \times n$ matrices.*
The output is an n^{-6}-well-supported Nash equilibrium of game $\mathcal{G}$.

Our main result is the following theorem.

Theorem 1 (Main). *Problem* SPARSE BIMATRIX *is* **PPAD***-complete.*

Clearly, SPARSE BIMATRIX belongs to **PPAD** [6]. To prove its completeness, we will reduce the **PPAD**-complete problem BROUWER^f [7] to it, where $f(n) = 3$. We also notice that, in contrast, a $(10/n)$-approximate Nash equilibrium of a sparse normalized game can be found in polynomial time.

3 Review of the Reduction in [7]

In this section, we review the reduction in [7], from BROUWER^f to the problem of finding an n^{-6}-well-supported Nash equilibrium in a normalized game.

Let $U = (C, 0^{3n})$ be an input instance of BROUWER^f, where C is a boolean circuit. Let m be the smallest integer such that $2^m > \text{Size}[C] > n$. Here we let $\text{Size}[C]$ denote the number of gates plus the number of input and output variables in C. In the reduction, we construct a game $\mathcal{G}^U = (\mathbf{A}^U, \mathbf{B}^U)$ in polynomial time, where $\mathbf{A}^U$ and $\mathbf{B}^U$ are $N \times N = 2^{6m+1} = 2K$ matrices, satisfying

Property $\mathbf{P}_1$: $|a_{i,j}^U|, |b_{i,j}^U| \leq N^3$ for all i, j: $1 \leq i, j \leq N$;

Property $\mathbf{P}_2$: From every ϵ-well-supported Nash equilibrium of $\mathcal{G}^U$, where $\epsilon = 2^{-18m} = 1/K^3$, one can find a panchromatic simplex P of circuit C in polynomial time.

Then we normalize $\mathcal{G}^U$ to obtain $\overline{\mathcal{G}^U} = (\overline{\mathbf{A}^U}, \overline{\mathbf{B}^U})$ by setting $\overline{\mathbf{A}^U} = \mathbf{A}^U/N^3$ and $\overline{\mathbf{B}^U} = \mathbf{B}^U/N^3$. **Property $\mathbf{P}_2$** implies that, from any $1/N^6$-well-supported Nash equilibrium of $\overline{\mathcal{G}^U}$, one can find a panchromatic simplex of circuit C efficiently.

As a result, the problem of finding an n^{-6}-well-supported Nash equilibrium in a normalized bimatrix game is **PPAD**-hard.

The construction of $\mathcal{G}^U$ starts with a zero-sum game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$ called Matching Pennies with payoff parameter $M = 2^{18m+1} = 2K^3$. $\mathbf{A}^*$ is a $K \times K$ block diagonal matrix, where each block is a 2×2 matrix of all M's, and $\mathbf{B}^* = -\mathbf{A}^*$. Ultimately, we obtain $\mathcal{G}^U$ by perturbing the payoff entries of $\mathcal{G}^*$.

At a high level, we partition the rows of $\mathcal{G}^*$ and hence of $\mathcal{G}^U$ into K groups: the i^{th} group consists of rows $2i-1, 2i$. Every row group $(2i-1, 2i)$ is referred as an *arithmetic node* v. Let V_A denote the set of all such nodes ($|V_A| = K$), and $\mathcal{C}_A$ denote the one-to-one correspondence from V_A to $\{1, 2...K\}$ such that v corresponds to the $\mathcal{C}_A(v)^{th}$ row group, for all $v \in V_A$. We also partition the columns of $\mathcal{G}^*$ into K groups: the j^{th} group consists of columns $2j-1, 2j$, and every group is referred as an *internal node* w. Let V_I denote the set of internal nodes and $\mathcal{C}_I$ denote the one-to-one correspondence from V_I to $\{1, 2...K\}$.

Let $(\mathbf{x} \in \mathbb{P}^N, \mathbf{y} \in \mathbb{P}^N)$ be a profile of mixed strategies. For each $v \in V_A$, we let $\mathbf{x}[v] = x_{2k-1}$ and $\mathbf{x}_C[v] = x_{2k-1} + x_{2k}$ denote the *value* and *capacity* of v in $(\mathbf{x}, \mathbf{y})$, respectively, where $k = \mathcal{C}_A(v)$. For each $w \in V_I$, we let $\mathbf{y}[w] = y_{2t-1}$ and $\mathbf{y}_C[w] = y_{2t-1} + y_{2t}$ denote the *value* and *capacity* of w in $(\mathbf{x}, \mathbf{y})$, respectively, where $t = \mathcal{C}_I(w)$. For $x, y \in \mathbb{R}$ and $c \in \mathbb{R}^+$, by $x = y \pm c$, we mean that $y - c \leq x \leq y + c$. All our perturbations of $\mathcal{G}^*$ have the following nice property.

Lemma 2 ([7]). *Let* $(\mathbf{A}, \mathbf{B})$ *be a game with* $0 \leq \mathbf{A} - \mathbf{A}^*, \mathbf{B} - \mathbf{B}^* \leq 1$. *For any* $t \leq 1$, *let* $(\mathbf{x}, \mathbf{y})$ *be a t-well-supported Nash equilibrium of* $(\mathbf{A}, \mathbf{B})$, *then it must satisfy constraint* $\mathcal{P} = [\,\mathbf{x}_C[v] = 1/K \pm \epsilon,\ \mathbf{y}_C[w] = 1/K \pm \epsilon,\ \forall\ v \in V_A, w \in V_I\,]$.

To construct $\mathcal{G}^U$, we transform the prototype game $\mathcal{G}^*$ by adding "gadget" games: we first build a collection of gadgets $\mathcal{S}^U = \{T_1..., T_l\}$ for some $l < K$. Each $T \in \mathcal{S}^U$ defines [7] an $N \times N$ "gadget" game $(\mathbf{L}[T], \mathbf{R}[T])$. We then build game $\mathcal{G}^U$ by invoking function BuildGame on $\mathcal{S}^U$. BuildGame takes a collection $\mathcal{S}$ of gadgets and returns a bimatrix game $(\mathbf{A}, \mathbf{B})$ as

$$\mathbf{A} = \mathbf{A}^* + \textstyle\sum_{T \in \mathcal{S}} \mathbf{L}[T] \quad \text{and} \quad \mathbf{B} = \mathbf{B}^* + \textstyle\sum_{T \in \mathcal{S}} \mathbf{R}[T].$$

A gadget T is a 6-tuple (G, v_1, v_2, v, c, w). Here G is the type of the gadget where $G \in \{G_\zeta, G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\wedge, G_\vee, G_\neg\}$. $v_1 \in V_A \cup \{nil\}$ and $v_2 \in V_A \cup \{nil\}$ are the first and second input nodes of T, respectively. $v \in V_A$ is the output node, and $w \in V_I$ is the internal node. Parameter $c \in \mathbb{R} \cup \{nil\}$ is only used in G_ζ and $G_{\times\zeta}$ gadgets: when $G = G_\zeta$, $0 \leq c \leq 1/K - \epsilon$; when $G = G_{\times\zeta}$, $0 \leq c \leq 1$; otherwise, $c = nil$.

Every gadget $T = (G, v_1, v_2, v, c, w)$ implements an arithmetic or logic constraint $\mathcal{P}[T]$, which requires the values of nodes v, v_1 and v_2 to satisfy certain functional relationship. All the nine types of constraints are listed in Figure 1. Among the nine types of gadgets, $G_\wedge, G_\vee$ and $G_\neg$ are logic gadgets. They are used to simulate the logic gates in C. Associated with probability vectors $(\mathbf{x}, \mathbf{y})$, the value of $v \in V_A$ represents boolean 1 ($\mathbf{x}[v] =_B 1$) if $\mathbf{x}[v] = \mathbf{x}_C[v]$; it represents boolean 0 ($\mathbf{x}[v] =_B 0$) if $\mathbf{x}[v] = 0$.

G_+: $\mathcal{P}[T] = [\ \mathbf{x}[v] = \min(\mathbf{x}[v_1] + \mathbf{x}[v_2], \mathbf{x}_C[v]) \pm \epsilon\ ]$

G_ζ: $\mathcal{P}[T] = [\ \mathbf{x}[v] = c \pm \epsilon\ ]$

$G_{\times\zeta}$: $\mathcal{P}[T] = [\ \mathbf{x}[v] = \min(c\mathbf{x}[v_1], \mathbf{x}_C[v]) \pm \epsilon\ ]$

$G_=$: $\mathcal{P}[T] = [\ \mathbf{x}[v] = \min(\mathbf{x}[v_1], \mathbf{x}_C[v]) \pm \epsilon\ ]$

$G_<$: $\mathcal{P}[T] = [\ \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] < \mathbf{x}[v_2] - \epsilon;\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] > \mathbf{x}[v_2] + \epsilon\ ]$

G_-: $\mathcal{P}[T] = [\ \min(\mathbf{x}[v_1] - \mathbf{x}[v_2], \mathbf{x}_C[v]) - \epsilon \le \mathbf{x}[v] \le \max(\mathbf{x}[v_1] - \mathbf{x}[v_2], 0) + \epsilon\ ]$

$G_\neg$: $\mathcal{P}[T] = [\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 1;\ \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 0\ ]$

$$G_\vee\text{:}\ \ \mathcal{P}[T] = \begin{bmatrix} \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1 \text{ or } \mathbf{x}[v_2] =_B 1; \\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0 \text{ and } \mathbf{x}[v_2] =_B 0 \end{bmatrix}$$

$$G_\wedge\text{:}\ \ \mathcal{P}[T] = \begin{bmatrix} \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0 \text{ or } \mathbf{x}[v_2] =_B 0; \\ \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1 \text{ and } \mathbf{x}[v_2] =_B 1 \end{bmatrix}$$

Fig. 1. Constraint $\mathcal{P}[T]$, where $T = (G, v_1, v_2, v, c, w)$

The collection $\mathcal{S}^U$ we construct is *valid*, that is, for each pair $T = (G, v_1, v_2, v, w, c)$ and $T' = (G', v_1', v_2', v', c', w')$ in $\mathcal{S}^U$, $v \neq v'$ and $w \neq w'$. In [7], we prove the following two lemmas for valid collections of gadgets.

Lemma 3 ([7]). *Let $\mathcal{S}$ be a valid collection and $\mathcal{G} = (\mathbf{A}, \mathbf{B}) =$ BuildGame$(\mathcal{S})$, then we have $0 \le \mathbf{A} - \mathbf{A}^*, \mathbf{B} - \mathbf{B}^* \le 1$. So, by Lemma 2, each ϵ-well-supported Nash equilibrium of $\mathcal{G}$ satisfies constraint $\mathcal{P}$.*

Lemma 4 ([7]). *Let $\mathcal{S}$ be a valid collection of gadgets, and $(\mathbf{x}, \mathbf{y})$ be any ϵ-well-supported Nash equilibrium of* BuildGame$(\mathcal{S})$*, then for each $T \in \mathcal{S}$, constraint $\mathcal{P}[T]$ as defined in Figure 1 is satisfied by $(\mathbf{x}, \mathbf{y})$.*

Property $\mathbf{P}_1$ follows directly from Lemma 3. From Lemma 3 and 4, every ϵ-well-supported Nash equilibrium of $\mathcal{G}^U$ satisfies a set of $|\mathcal{S}^U| + 1$ constraints: $\{\mathcal{P}, \mathcal{P}[T_1], ..., \mathcal{P}[T_l]\}$, which can be used to prove **Property $\mathbf{P}_2$**.

4 The New Reduction

Although the prototype game $\mathcal{G}^*$ is sparse (for each row and column, there are exactly two nonzero entries), $\mathcal{G}^U$ constructed in [7] is not always sparse:

1. There are three types of "bad" gadgets used in the construction of $\mathcal{G}^U$: G_ζ, $G_\wedge$ and $G_\vee$. For every $T = (G, v_1, v_2, v, c, w) \in \mathcal{S}^U$ with $G \in \{G_\zeta, G_\wedge, G_\vee\}$, every entry in the $(2\mathcal{C}_I(w))^{th}$ column of matrix $\mathbf{R}[T]$ is non-zero [7]. As a result, $\mathbf{B}^U$ is not column sparse.
2. There exist some arithmetic nodes $v \in V_A$ which are used by more than 5 gadgets in $\mathcal{S}^U$ as one of their input nodes. Suppose they are $T_1 ... T_k \in \mathcal{S}^U$, then in both the $(2\mathcal{C}_A(v) - 1)^{st}$ and $(2\mathcal{C}_A(v))^{th}$ rows of $\sum_{1 \le i \le k} \mathbf{R}[T_i]$, we have $2k > 10$ non-zero entries [7]. As a result, $\mathbf{B}^U$ is not row sparse.

In this section, we will reduce problem Brouwer^f to Sparse Bimatrix. The reduction is very similar to the one in [7]. We will develop new "gadget" games to overcome the first obstacle above. Then we will perturb the prototype game $\mathcal{G}^*$ to build a sparse game $\mathcal{H}^U$ which satisfies both **Property** $\mathbf{P}_1$ and $\mathbf{P}_2$. One can normalize the sparse game $\mathcal{H}^U$ to prove Theorem 1.

4.1 New Gadgets and Constraints

To build game $\mathcal{H}^U$, we transform the prototype game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$ by adding "gadget" games. We first build a collection $\mathcal{T}^U = \{T_1, ..., T_l\}$ of gadgets. For every gadget T, we construct a "gadget" game $(\mathbf{M}[T], \mathbf{N}[T])$ according to Figure 2. Given any collection of gadgets $\mathcal{T}$, one can construct a two-player game $(\mathbf{A}, \mathbf{B}) = \text{BuildGame}(\mathcal{T})$ by setting

$$\mathbf{A} = \mathbf{A}^* + \textstyle\sum_{T \in \mathcal{T}} \mathbf{M}[T] \quad \text{and} \quad \mathbf{B} = \mathbf{B}^* + \textstyle\sum_{T \in \mathcal{T}} \mathbf{N}[T].$$

From $\mathcal{T}^U$, we obtain game $\mathcal{H}^U = \text{BuildGame}(\mathcal{T}^U)$.

Here a gadget is a 7-tuple $T = (G, v_1, v_2, v_3, v, c, w)$. $v_3 \in V_A \cup \{nil\}$ is the auxiliary input node of T, while the meanings of all the other components are the same as those in the previous reduction. In the new reduction, we have totally eleven types of gadgets: $G \in \{G_+, G_-, G_=, G_<, G_{\times\zeta}, G_\neg, G^*_\zeta, G^*_\wedge, G^*_\vee, G_H, G_{B=}\}$. Similarly, every gadget T implements an arithmetic or logic constraint $\mathcal{R}[T]$, which requires the values of v_1, v_2, v_3, v to satisfy certain functional relationship. Before describing constraints $\mathcal{R}[T]$ for each type of gadgets, we claim the following two lemmas, whose proofs are very similar to those of Lemma 3 and Lemma 4 in [7]. Here a collection $\mathcal{T}$ is *valid* if for every pair $T = (G, v_1, v_2, v_3, v, c, w)$ and $T' = (G', v'_1, v'_2, v'_3, v', c', w')$ in $\mathcal{T}$, $v \neq v'$ and $w \neq w'$.

Lemma 5. *Let $\mathcal{T}$ be a valid collection and $(\mathbf{A}, \mathbf{B}) = \text{BuildGame}(\mathcal{T})$, then we have $0 \leq \mathbf{A} - \mathbf{A}^*, \mathbf{B} - \mathbf{B}^* \leq 1$. So from Lemma 2, every ϵ-well-supported Nash equilibrium of $(\mathbf{A}, \mathbf{B})$ satisfies constraint $\mathcal{P}$.*

Lemma 6 (Gadget Constraints). *Let $\mathcal{T}$ be a valid collection of gadgets, and $(\mathbf{x}, \mathbf{y})$ be an ϵ-well-supported Nash equilibrium of* BuildGame$(\mathcal{T})$*, then for each $T \in \mathcal{T}$, constraint $\mathcal{R}[T]$ is satisfied by $(\mathbf{x}, \mathbf{y})$.*

By Lemma 5 and 6, every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game BuildGame$(\mathcal{T})$ satisfies $|\mathcal{T}|+1$ constraints: $\{\mathcal{P}, \mathcal{R}[T], T \in \mathcal{T}\}$. Let $T = (G, v_1, v_2, v_3, v, c, w)$ be a gadget in $\mathcal{T}$, then $\mathcal{R}[T]$ on $(\mathbf{x}, \mathbf{y})$ is described as follows:

- If type $G \in \{G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\neg\}$, then $v_3 = nil$. We have $\mathbf{M}[T] = \mathbf{L}[T']$ and $\mathbf{N}[T] = \mathbf{R}[T']$, where $T' = (G, v_1, v_2, v, c, w)$, and naturally, constraint $\mathcal{R}[T]$ is the same as $\mathcal{P}[T']$.
- If $G = G_{B=}$, then $v_1 \in V_A$ and $v_2 = v_3 = c = nil$. $(\mathbf{x}, \mathbf{y})$ satisfies constraint $\mathcal{R}[T] = [\,\mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1;\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0\,]$.
- If $G = G^*_\zeta$, then the auxiliary input node $v_3 \in V_A$, $v_1 = v_2 = nil$, and $0 \leq c \leq 1/K - \epsilon$. $(\mathbf{x}, \mathbf{y})$ satisfies $\mathcal{R}[T] = [\,\mathbf{x}[v] = c \pm 4\epsilon \text{ if } \mathbf{x}[v_3] = 1/(2K) \pm \epsilon\,]$. So, $\mathcal{R}[T]$ is very close to constraint $\mathcal{P}[T']$, where $T' = (G_\zeta, nil, nil, v, c, w)$, when the value of the auxiliary input node v_3 in $(\mathbf{x}, \mathbf{y})$ is close to $1/(2K)$.

Construction of $\mathbf{M}[T]$ and $\mathbf{N}[T]$, where $T = (G, v_1, v_2, v_3, v, c, w)$

Set $\mathbf{M}[T] = (M_{i,j}) = \mathbf{N}[T] = (N_{i,j}) = 0$

$k = \mathcal{C}_A(v)$, $k_1 = \mathcal{C}_A(v_1)$, $k_2 = \mathcal{C}_A(v_2)$, $k_3 = \mathcal{C}_A(v_3)$ and $t = \mathcal{C}_I(w)$

$G_+ \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k_1-1,2t-1} = N_{2k_2-1,2t-1} = N_{2k-1,2t} = 1$

$G_- \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k_1-1,2t-1} = N_{2k_2-1,2t} = N_{2k-1,2t} = 1$

$G_= \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k_1-1,2t-1} = N_{2k-1,2t} = 1$

$G_< \;:\; M_{2k-1,2t} = M_{2k,2t-1} = N_{2k_1-1,2t-1} = N_{2k_2-1,2t} = 1$

$G_{\times\zeta} \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k-1,2t} = 1,\; N_{2k_1-1,2t-1} = c$

$G_\neg \;:\; M_{2k-1,2t} = M_{2k,2t-1} = N_{2k_1-1,2t-1} = N_{2k_1,2t} = 1$

$G^*_\zeta \;:\; M_{2k-1,2t} = M_{2k,2t-1} = 1,\; N_{2k-1,2t-1} = 1/2,\; N_{2k_1-1,2t} = Kc$

$G^*_\wedge \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k_3-1,2t} = 1,\; N_{2k_1-1,2t-1} = N_{2k_2-1,2t-1} = 1/3$

$G^*_\vee \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k_1-1,2t-1} = N_{2k_2-1,2t-1} = N_{2k_3-1,2t} = 1$

$G_{B=} \;:\; M_{2k-1,2t-1} = M_{2k,2t} = N_{2k_1-1,2t-1} = N_{2k_1,2t} = 1$

$G_H \;:\; M_{2k-1,2t} = M_{2k,2t-1} = N_{2k-1,2t-1} = N_{2k,2t} = 1$

Fig. 2. Construction of "Gadget" Game $(\mathbf{M}[T], \mathbf{N}[T])$

– If $G = G^*_\vee$, then $v_1, v_2, v_3 \in V_A$ and $c = nil$. $(\mathbf{x}, \mathbf{y})$ satisfies constraint $\mathcal{R}[T]$

$$\left[\; \mathbf{x}[v_3] = \frac{1}{2K} \pm \epsilon \implies \left\{ \begin{array}{c} \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1 \text{ or } \mathbf{x}[v_2] =_B 1 \\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0 \text{ and } \mathbf{x}[v_2] =_B 0 \end{array} \right\} \right].$$

Similarly, if $G = G^*_\wedge$, then $(\mathbf{x}, \mathbf{y})$ must satisfy constraint $\mathcal{R}[T]$

$$\left[\; \mathbf{x}[v_3] = \frac{1}{2K} \pm \epsilon \implies \left\{ \begin{array}{c} \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1 \text{ and } \mathbf{x}[v_2] =_B 1 \\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0 \text{ or } \mathbf{x}[v_2] =_B 0 \end{array} \right\} \right].$$

Clearly, constraint $\mathcal{R}[T]$ is the same as $\mathcal{P}[T']$ where $T' = (G_\vee$ or $G_\wedge, v_1, v_2, v, c, w)$, when the value of v_3 in $(\mathbf{x}, \mathbf{y})$ is close to $1/(2K)$.

– If type $G = G_H$, then $v_1 = v_2 = v_3 = nil$. $(\mathbf{x}, \mathbf{y})$ satisfies $\mathcal{R}[T] = [\, \mathbf{x}[v] = 1/2K \pm \epsilon\,]$. We will use G_H gadgets to "generate" auxiliary nodes for G^*_ζ, $G^*_\wedge$ and $G^*_\vee$ gadgets to simulate the old G_ζ, $G_\wedge$ and $G_\vee$ gadgets used in [7].

We next show that, if a valid collection $\mathcal{T}$ also satisfies the following property, then bimatrix game BuildGame($\mathcal{T}$) must be sparse.

Definition 3. *Collection $\mathcal{T}$ is said to be* sparse *if for every v^* in V_A, there exist at most two gadgets $T = (G, v_1, v_2, v_3, v, c, w) \in \mathcal{T}$ such that $v^* \in \{v_1, v_2, v_3\}$.*

Lemma 7. *If $\mathcal{T}$ is both valid and sparse, then* BuildGame($\mathcal{T}$) *is sparse.*

$\text{COPY}_A(\mathcal{T}; v; v_1, v_2, ..., v_k)$
1: pick unused nodes $v'_1, v'_2, ..., v'_{k-1} \in V_A$ and $w', w'', w_1 ..., w_{k-1}, w'_1 ..., w'_{k-2} \in V_I$
2: $\text{INSERT}(\mathcal{T}, (G_=, v, nil, nil, v_1, nil, w'))$ and $\text{INSERT}(\mathcal{T}, (G_=, v, nil, nil, v'_1, nil, w''))$
3: **for** i from 1 to $k-1$, $\text{INSERT}(\mathcal{T}, (G_=, v'_i, nil, nil, v_{i+1}, nil, w_i))$
4: **for** i from 1 to $k-2$, $\text{INSERT}(\mathcal{T}, (G_=, v'_i, nil, nil, v'_{i+1}, nil, w'_i))$

Fig. 3. Function COPY_A

Proof. For each $T = (G, v_1, v_2, v_3, v, c, w) \in \mathcal{T}$, $(\mathbf{M}[T], \mathbf{N}[T])$ satisfies:

Property 1. *Let $k = \mathcal{C}_A(v)$, $t = \mathcal{C}_I(w)$ and $k_i = \mathcal{C}_A(v_i)$ for every $1 \le i \le 3$. In matrices $\mathbf{M}[T] = (M_{i,j})$ and $\mathbf{N}[T] = (N_{i,j})$, only the following entries are possibly nonzero: $\{M_{2k-1,2t-1}, M_{2k-1,2t}, M_{2k,2t-1}, M_{2k,2t}\}$ and $\{N_{2l-1,2t-1}, N_{2l-1,2t}, N_{2l,2t-1}, N_{2l,2t}$, where $l \in \{k_1, k_2, k_3, k\}\}$, and all these entries are in $[0, 1]$.*

Property 1 follows directly from the construction of $\mathbf{M}[T]$ and $\mathbf{N}[T]$ in Figure 2. Let $(\mathbf{A} = (a_{i,j}), \mathbf{B} = (b_{i,j})) = \text{BUILDGAME}(\mathcal{T})$.

Let v be an arithmetic node in V_A and $k = \mathcal{C}_A(v)$. According to Property 1, for any $1 \le j \le 2K$, $a_{2k,j} \ne a^*_{2k,j}$ implies that there exists a gadget $T \in \mathcal{T}$ whose output node is v and internal node w satisfies $j \in \{2\mathcal{C}_I(w), 2\mathcal{C}_I(w) - 1\}$. Since $\mathcal{T}$ is valid, there can be at most one gadget whose output node is v and thus, there are at most two integers $1 \le j \le 2K$ such that $a_{2k,j} \ne a^*_{2k,j}$. On the other hand, the $(2k)^{th}$ row of $\mathbf{A}^*$ has exactly two nonzero entries. As a result, the number of nonzero entries in the $(2k)^{th}$ row of $\mathbf{A}$ is at most four. The case for the $(2k-1)^{st}$ row can be proved similarly, and thus, $\mathbf{A}$ is row sparse. One can prove similarly that both $\mathbf{A}$ and $\mathbf{B}$ are column sparse.

Let v be an arithmetic node in V_A and $k = \mathcal{C}_A(v)$. According to Property 1, $b_{2k,j} \ne b^*_{2k,j}$ implies there exists a gadget $T = (G, v_1, v_2, v_3, v, c, w) \in \mathcal{T}$ such that $v \in \{v_1, v_2, v_3, v\}$ and $j \in \{2\mathcal{C}_I(w), 2\mathcal{C}_I(w) - 1\}$. Since $\mathcal{T}$ is both valid and sparse, there can be at most three gadgets $T = (G, v_1, v_2, v_3, v, c, w) \in \mathcal{T}$ such that $v \in \{v_1, v_2, v_3, v\}$, and at most six integers j such that $b_{2k,j} \ne b^*_{2k,j}$. So the number of nonzero entries in the $(2k)^{th}$ row of $\mathbf{B}$ is at most eight. The case for the $(2k-1)^{st}$ row can be proved similarly, and thus, $\mathbf{B}$ is row sparse.

4.2 The Copy Network

In this subsection, we build a network of gadgets which will be referred to as a *copy network*. Let us define some notations that will be useful.

Let $\mathcal{T}$ be a valid collection of gadgets. An arithmetic node $v \in V_A$ (or an internal node $w \in V_I$) is *unused* in $\mathcal{T}$ if none of the gadgets in $\mathcal{T}$ uses v (or w) as its output node (or internal node). We use $\text{UNUSED}[\mathcal{T}]$ to denote the number of unused nodes $v \in V_A$ in $\mathcal{T}$. Suppose $T \notin \mathcal{T}$ is a gadget such that $\mathcal{T} \cup \{T\}$ is still valid. We use $\text{INSERT}(\mathcal{T}, T)$ to denote the insertion of gadget T into $\mathcal{T}$.

set $\mathcal{T} = \emptyset$
for every gadget $T = (G, v_1, v_2, v, c, w) \in \mathcal{S}^U$ constructed in [7] **do**
  if $G \in \{G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\neg\}$ **then**
    $\textsc{Insert}(\mathcal{T}, (G, v_1, v_2, nil, v, c, w))$
  else [if $G = G_\zeta$ (or $G_\wedge, G_\vee$), we use G^* to denote G^*_ζ (or $G^*_\wedge, G^*_\vee$)]
    pick nodes $v' \in V_A$ and $w' \in V_I$, which are unused in both $\mathcal{S}^U$ and $\mathcal{T}$
    $\textsc{Insert}(\mathcal{T}, (G_H, nil, nil, nil, v', nil, w'))$, $\textsc{Insert}(\mathcal{T}, (G^*, v_1, v_2, v', v, c, w))$

Fig. 4. Step 1: from $\mathcal{S}^U$ to $\mathcal{T}$

set $\mathcal{T}^U = \mathcal{T}$
for every $v \in V_A$ which is used by $k > 2$ gadgets in $\mathcal{T}^U$ as their input nodes **do**
  suppose these gadgets are $T_1, T_2, ..., T_k \in \mathcal{T}^U$
  pick k nodes $v_1, v_2, ..., v_k \in V_A$ which are unused in $\mathcal{T}^U$
  for every $1 \le i \le k$, replace the v in $T_i \in \mathcal{T}^U$ by v_i
  if we intend to store a boolean value in v (which should be clear from [7])
    $\textsc{Copy}_B(\mathcal{T}^U; v; v_1, v_2, ..., v_k)$
  else
    $\textsc{Copy}_A(\mathcal{T}^U; v; v_1, v_2, ..., v_k)$

Fig. 5. Step 2: from $\mathcal{T}$ to $\mathcal{T}^U$

Let $\mathcal{T}$ be a valid collection with $\textsc{Unused}[\mathcal{T}] \ge 2k-1$, and $k \ge 3$. Let $v \in V_A$, and $v_1, v_2, ..., v_k \in V_A$ be k unused nodes in $\mathcal{T}$. We insert $2k-1$ gadgets into $\mathcal{T}$ by invoking the function $\textsc{Copy}_A(\mathcal{T}; v; v_1, v_2, ..., v_k)$ in Figure 3. We let $\mathcal{T}'$ denote the collection $\mathcal{T}$ after executing $\textsc{Copy}_A(\mathcal{T}; v; v_1, v_2, ..., v_k)$, then

Lemma 8. *In every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of bimatrix game $\textsc{BuildGame}(\mathcal{T}')$, $\mathbf{x}[v_i] = \mathbf{x}[v] \pm 3t\epsilon$ for all $1 \le t \le k$.*

Furthermore, by replacing every $G_=$ gadget in $\textsc{Copy}_A$ with a $G_{B=}$ gadget, we immediately get a function $\textsc{Copy}_B(\mathcal{T}; v; v_1, v_2 ... v_k)$ for inserting a boolean copy network into $\mathcal{T}$, such that

Lemma 9. *In every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of bimatrix game $\textsc{BuildGame}(\mathcal{T}')$, if $\mathbf{x}[v] =_B b$ where $b \in \{0, 1\}$, then $\mathbf{x}[v_t] =_B b$, $\forall\ 1 \le t \le k$.*

4.3 Construction of $\mathcal{T}^U$ and $\mathcal{H}^U$

Let $U = (C, 0^{3n})$ be an input instance of search problem $\textsc{Brouwer}^f$, and $\mathcal{S}^U$ be the collection of gadgets constructed in [7]. We now convert it into a new collection $\mathcal{T}^U$ that is both valid and sparse, such that $\mathcal{H}^U = \textsc{BuildGame}(\mathcal{T}^U)$

satisfies both **Property** $\mathbf{P}_1$ and $\mathbf{P}_2$. Notice that, since $\mathcal{T}^U$ is valid and sparse, game $\mathcal{H}^U$ is sparse by Lemma 7. We build $\mathcal{T}^U$ with a two-step construction:

Step 1 [Figure 4]. We build a collection $\mathcal{T}$ by replacing each $G_\zeta, G_\wedge$ and $G_\vee$ gadget in $\mathcal{S}^U$ with two gadgets: one G_H gadget and one $G^*_\zeta, G^*_\wedge$ or $G^*_\vee$ gadget. Note that, every $G^*_\zeta, G^*_\wedge$ or $G^*_\vee$ gadget in $\mathcal{T}$ has a "private" G_H gadget.
Step 2 [Figure 5]. For every $v \in V_A$ which is used by $k > 2$ gadgets in $\mathcal{T}$ as one of their input nodes, we pick k unused nodes $v_1, ..., v_k$ in V_A and insert a copy network to connect v with $v_1, ..., v_k$. Then, each of the k gadgets gets one node in $\{v_1, v_2, ..., v_k\}$ as its input node.

4.4 Correctness of the Reduction

The main item we need to check carefully is the number of nodes used in $\mathcal{T}^U$. The number of nodes used in $\mathcal{S}^U$ is $O(\text{Size}\,[C]^4)$ [7]. Thus the number of nodes used in $\mathcal{T}$ is also $O(\text{Size}\,[C]^4)$. On the other hand, every $T \in \mathcal{T}$ can appear in line 3 of Figure 5 for at most three times, since T only has three input nodes. The number of nodes used in $\mathcal{T}^U$ is still $O(\text{Size}\,[C]^4) \ll K$. So we always have $\textsc{Unused}[\mathcal{T}] > 0$ and $\textsc{Unused}[\mathcal{T}^U] > 0$, during the construction of $\mathcal{T}$ and $\mathcal{T}^U$.

Because $\mathcal{S}^U$ is valid, one can check that $\mathcal{T}^U$ is both valid and sparse. As a corollary of Lemma 5, $\mathcal{H}^U$ satisfies **Property** $\mathbf{P}_1$. Following the line of proof in [7], we can show that, with the same procedure used in [7], one can recover a panchromatic simplex of C from every ϵ-well-supported Nash equilibrium of two-player game $\mathcal{H}^U$. Therefore, $\mathcal{H}^U$ also satisfies **Property** $\mathbf{P}_2$.

By Lemma 7, we know that game $\mathcal{H}^U$ is sparse. Finally, we get a reduction from problem $\textsc{Brouwer}^f$ to $\textsc{Sparse Bimatrix}$, and Theorem 1 is proven.

5 An Algorithm for Very Sparse Win-Lose Games

In this section, we describe an algorithm for finding exact Nash equilibria in a class of very sparse win-lose games.

A bimatrix game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is a *win-lose game* if every entry of $\mathbf{A}$ and $\mathbf{B}$ is either 0 or 1. A win-lose game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is *very sparse* if in each row of $\mathbf{A}$ and each column of $\mathbf{B}$, there are at most *two* non-zero entries. We use $\mathcal{P}$ to denote the set of very sparse win-lose games. First we define a subclass $\mathcal{Q}$ of $\mathcal{P}$. Every game in $\mathcal{Q}$ has an exact Nash equilibrium that can be computed easily.

Definition 4. *Let* $\mathbf{A}$ *be a* $\{0,1\}$*-matrix. The row* i *of* $\mathbf{A}$ *is said to be dominated if one of the following conditions is true:* **1).** *all the entries in it are zero;* **2).** *only one entry* $a_{i,j} = 1$ *is non-zero, and there exists another* $i' \neq i$ *such that* $a_{i',j} = 1$*. Similarly, the column* j *of matrix* $\mathbf{B}$ *is dominated if the row* j *of* $\mathbf{B}^T$ *is dominated. A bimatrix game* $\mathcal{G} = (\mathbf{A}, \mathbf{B}) \in \mathcal{P}$ *belongs to* $\mathcal{Q}$ *if none of the rows of* $\mathbf{A}$ *is dominated, and none of the columns of* $\mathbf{B}$ *is dominated.*

For every game $\mathcal{G} = (\mathbf{A}, \mathbf{B}) \in \mathcal{Q}$ where $\mathbf{A}$ and $\mathbf{B}$ are $n \times m$ matrices, we build a pair of vectors $(\mathbf{x}^* \in \mathbb{R}^n, \mathbf{y}^* \in \mathbb{R}^m)$ as follows: **1)** For each $1 \leq j \leq m$, if there

SparseWinLose ($\mathcal{G} = (\mathbf{A}, \mathbf{B}) \in \mathcal{P}$)

if $n = 1$ or $m = 1$ **then**
  output a Nash equilibrium of $\mathcal{G}$
else if $\mathcal{G} \in \mathcal{Q}$ **then**
  output a Nash equilibrium of $\mathcal{G}$ using Lemma 10
else if the row i of $\mathbf{A}$ is dominated **then**
  $(\mathbf{x}', \mathbf{y}') =$ **SparseWinLose** ($\mathcal{G}'$), $\mathcal{G}'$ is obtained by deleting row i from $\mathcal{G}$
  output a Nash equilibrium of $\mathcal{G}$ using Lemma 11
else [assume the column j of $\mathbf{B}$ is dominated]
  $(\mathbf{x}', \mathbf{y}') =$ **SparseWinLose** ($\mathcal{G}'$), $\mathcal{G}'$ is obtained by deleting column j from $\mathcal{G}$
  output a Nash equilibrium of $\mathcal{G}$ using Lemma 12

Fig. 6. An Algorithm for Very Sparse Win-Lose Games

exists an $1 \leq i \leq n$ such that the row i of $\mathbf{A}$ has exactly one non-zero entry: $a_{i,j} = 1$, then $y_j^* = 2$, otherwise $y_j^* = 1$; **2)** For each $1 \leq i \leq n$, if there is an $1 \leq j \leq m$ such that the column j of matrix $\mathbf{B}$ has exactly one nonzero entry: $b_{i,j} = 1$, then $x_i^* = 2$, otherwise $x_i^* = 1$.

Lemma 10. *For every $\mathcal{G} = (\mathbf{A}, \mathbf{B}) \in \mathcal{Q}$, let $(\mathbf{x}^* \in \mathbb{R}^n, \mathbf{y}^* \in \mathbb{R}^m)$ be the pair of vectors constructed above, then $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium of $\mathcal{G}$ where*

$$x_i = x_i^* \Big/ \sum_{1 \leq k \leq n} x_k^* \quad \textit{and} \quad y_j = y_j^* \Big/ \sum_{1 \leq k \leq m} y_k^*, \quad \forall\ 1 \leq i \leq n,\ 1 \leq j \leq m.$$

Proof. From the definition of $\mathcal{Q}$, one can check that, for every $1 \leq i \leq n$ and $1 \leq j \leq m$, $\langle \mathbf{A}_i | \mathbf{y} \rangle = 2 / \sum_{1 \leq k \leq m} y_k^*$ and $\langle \mathbf{x} | \mathbf{B}_j \rangle = 2 / \sum_{1 \leq k \leq n} x_k^*$, where $\mathbf{A}_i$ denotes the i^{th} row vector of matrix $\mathbf{A}$, and $\mathbf{B}_j$ denotes the j^{th} column vector of $\mathbf{B}$. This implies that $(\mathbf{x}, \mathbf{y})$ is an exact Nash equilibrium of game $\mathcal{G}$.

Our algorithm is recursive. If the input game is small ($n = 1$ or $m = 1$) or belongs to $\mathcal{Q}$, then a Nash equilibrium can be found easily. Otherwise, we delete one row or column from $\mathcal{G}$, and obtain a smaller game $\mathcal{G}'$. From every Nash equilibrium $(\mathbf{x}', \mathbf{y}')$ of the new game $\mathcal{G}'$, one can "recover" a solution to the old one quickly. The algorithm is supported by the following two lemmas. Here we use $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ to denote a game in $\mathcal{P}$, where $\mathbf{A}$ and $\mathbf{B}$ are $n \times m$ matrices.

Lemma 11. *If the row k of $\mathbf{A}$ is dominated, letting $(\mathbf{x}' \in \mathbb{P}^{n-1}, \mathbf{y}' \in \mathbb{P}^m)$ be an exact Nash equilibrium of $\mathcal{G}'$, where $\mathcal{G}'$ is obtained by deleting row k from game $\mathcal{G}$, then $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium of $\mathcal{G}$, where $\mathbf{y} = \mathbf{y}'$, $x_k = 0$, $x_i = x_i'$ for all $1 \leq i < k$, and $x_i = x_{i-1}'$ for all $k < i \leq n$.*

Lemma 12. *If the column k of $\mathbf{B}$ is dominated, letting $(\mathbf{x}' \in \mathbb{P}^n, \mathbf{y}' \in \mathbb{P}^{m-1})$ be a Nash equilibrium of $\mathcal{G}'$, where $\mathcal{G}'$ is obtained by deleting column k from game*

$\mathcal{G}$, then $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium of $\mathcal{G}$, where $\mathbf{x} = \mathbf{x}'$, $y_k = 0$, $y_i = y_i'$ for all $1 \leq i < k$, and $y_i = y_{i-1}'$ for all $k < i \leq m$.

Acknowledgements

We thank Li-Sha Huang for a discussion on market equilibria which led us to study this problem. Part of Teng's work was done while visiting Microsoft Beijing Research Lab. His work is supported by the NSF grants CCR-0311430 and ITR CCR-0325630. Part of Deng's work was done while visiting Tsinghua University.

References

1. Goldberg, P., Papadimitriou, C.: Reducibility among equilibrium problems. (In: STOC 2006)
2. Daskalakis, C., Goldberg, P., Papadimitriou, C.: The complexity of computing a nash equilibrium. (In: STOC 2006)
3. Papadimitriou, C.: On the complexity of the parity argument and other inefficient proofs of existence. Journal of Computer and System Sciences (1994) 498–532
4. Chen, X., Deng, X.: 3-nash is ppad-complete. ECCC, TR05-134 (2005)
5. Daskalakis, C., Papadimitriou, C.: Three-player games are hard. ECCC, TR05-139 (2005)
6. Chen, X., Deng, X.: Settling the complexity of two-player nash-equilibrium. (In: FOCS 2006)
7. Chen, X., Deng, X., Teng, S.H.: Computing nash equilibria: Approximation and smoothed complexity. (In: FOCS 2006)
8. Lipton, R.J., Markakis, E., Mehta, A.: Playing large games using simple strategies. (In: ACM EC 2003) 36–41
9. Spielman, D.A., Teng, S.H.: Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. J. ACM **51** (2004) 385–463
10. Barany, I., Vempala, S., Vetta, A.: Nash equilibria in random games. (In: FOCS 2005)
11. Chen, X., Teng, S.H., Valiant, P.A.: The approximation complexity of win-lose games. Manuscript: Tsinghua-BU-MIT (2006)
12. Nash, J.: Equilibrium point in n-person games. Porceedings of the National Academy of the USA **36** (1950) 48–49
13. Codenotti, B., Leoncini, M., Resta, G.: Efficient computation of nash equilibria for very sparse win-lose bimatrix games. (In: ESA 2006)

Market Equilibria with Hybrid Linear-Leontief Utilities

Xi Chen[1], Li-Sha Huang[1,⋆], and Shang-Hua Teng[2,⋆⋆]

[1] State Key Laboratory of Intelligent Technology and Systems,
Dept. of Computer Science and Technology, Tsinghua Univ., Beijing, 10084, China
[2] Computer Science Department, Boston University, Boston, Massachusetts, USA

Abstract. We introduce a new family of utility functions for exchange markets. This family provides a natural and "continuous" hybridization of the traditional linear and Leontief utilities and might be useful in understanding the complexity of computing and approximating market equilibria. Because this family of utility functions contains Leontief utility functions as special cases, finding approximate Arrow-Debreu equilibria with hybrid linear-Leontief utilities is **PPAD**-hard in general. In contrast, we show that, when the Leontief components are grouped, finite and well-conditioned, we can efficiently compute an approximate Arrow-Debreu equilibrium.

1 Introduction

In recent years, the problem of computing market equilibria has attracted many computer scientists. In an exchange market, there is a set of traders and each trader comes with an initial endowment of commodities. They interact through some exchange process in order to maximize their own utility functions. In the state of an equilibrium, the traders can simply sell their initial endowments at a determined market price and buy the commodities to maximize their utilities. Then, the market will clear — the price is so wisely set that the supplies exactly satisfy the demands. This price is called the *equilibrium price*.

Arrow-Debreu [1] proved the existence of equilibrium prices under a mild condition. Since then, efficient algorithms have been developed for various settings. Naturally, the complexity for finding an equilibrium price is determined not just by the initial endowments, but also by traders' utility functions.

1.1 From Linear to Leontief Utilities

Two popular families of utility functions are the linear and Leontief utilities. Both utilities can be specified by an $m \times n$ demand matrix $\mathbf{D} = (d_{i,j})$, for m

⋆ Supported by Natural Science Foundation of China (No.60135010,60321002) and the Chinese National Key Foundation Research and Development Plan (2004CB318108).
⋆⋆ Also affiliated with Akamai Technologies Inc. Cambridge, Massachusetts, USA. Partially supported by NSF grants CCR-0311430 and ITR CCR-0325630. Part of this work done while visiting Tsinghua University and Microsoft Research Asia Lab.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 274–285, 2006.
© Springer-Verlag Berlin Heidelberg 2006

goods and n traders. If trader $1 \leq j \leq n$ receives a bundle of goods $\mathbf{x}_j$, then its linear utility is $u_j(\mathbf{x}_j) = \sum_i x_{i,j}/d_{i,j}$, while its Leontief utility is $u_j(\mathbf{x}_j) = \min_i(x_{i,j}/d_{i,j})$. Both linear and Leontief utility functions are members a large family of utilities functions, referred to as CES utilities. The CES utility function with parameter $\rho \in (-\infty, 1] - \{0\}$ is:

$$u_j^\rho(\mathbf{x}_j) = \left(\sum_i d_{i,j} x_{i,j}^\rho\right)^{1/\rho}.$$

As $\rho \to -\infty$, CES utilities become the Leontief utilities. When $\rho = 1$, the utility functions are linear functions.

Although the Leontief utility functions and linear utility functions look similar, the complexity for finding their market equilibria might be very different. A market equilibrium with linear utilities can be approximated and computed in polynomial time, thanks to a collection of great algorithmic works by Nenakhov and Primak [16], Devanur *et al.* [10], Jain, Mahdian and Saberi [14], Garg and Kapoor [11], Jain [13], and Ye [17].

However, approximating market equilibria with Leontief utilities has proven to be hard, under some reasonable complexity assumptions. In particular, by analyzing a reduction of Codenotti, Saberi, Varadarajan and Ye [5] from Nash equilibria to market equilibria, Huang and Teng [12] showed that approximating Leontief market equilibria is as hard as approximating Nash equilibria of general two-player games. Thus, by a recent result of Chen, Deng, and Teng [3], it is **PPAD**-hard to approximate a Leontief market equilibrium in fully polynomial time. In fact, the smoothed complexity of finding a market equilibrium in Leontief economies cannot be polynomial unless **PPAD** $\subset$ **RP**.

1.2 Hybrid Linear-Leontief Utilities and Our Results

In this paper, we introduce a new family of utility functions and study the computation and approximation of equilibria in exchange markets with these utilities. Our work is partially motivated by the complexity discrepancy of linear and Leontief utilities. In our market model, each trader's utility function is a linear combination of a collection of Leontief utility functions. We parameterize such a utility function by the maximum number of terms in its Leontief components. If the number of terms in any of its Leontief components is at most k, we refer to it as a *k-wide linear-Leontief function.* We further focus on *grouped* hybridizations in which the commodities are divided into groups. Each trader's utility is the summation over the Leontief utilities of all groups. If each group has at most k commodities, we refer to the hybrid functions as *grouped k-wide linear-Leontief functions.*

Intuitively, the new utility function combines an "easy" linear function with several "hard" Leontief utility functions. Clearly, a 1-wide linear-Leontief function is a linear function, and hence a market equilibrium with 1-wide linear-Leontief functions can be found in polynomial time. On the other hand, market equilibria with general hybrid linear-Leontief utilities are **PPAD**-hard to find.

A market with grouped linear-Leontief utility functions can be viewed as a linear combination of several Leontief markets, one for each group of commodities. In an equilibrium, the supplies exactly satisfy the demands for each group of commodities. However, the trader can invest the surplus it earned from one Leontief market to other Leontief markets.

We present two algorithmic results on the computation and approximation of equilibria in markets with hybrid linear-Leontief utilities.

- We show that a Fisher equilibrium of an exchange market with n traders, M commodities and hybrid linear-Leontief utility functions can be found in $O(\sqrt{Mn}(M+n)^3 L)$ time.
- We also show that, in the grouped hybridizations when the Leontief component is well-conditioned, we can compute an approximate Arrow-Debreu equilibrium in polynomial time either in M or n. (An interesting observation is that a recent result of Chen, Deng, and Teng [4] on sparse two-player games implies that it is **PPAD**-hard to approximate Arrow-Debreu equilibria in an exchange market with 10-wide linear-Leontief utilities in fully polynomial-time.)

In this paper, we only give formal definition for grouped linear-Leontief utility functions. It is easy to extend the definition and the first algorithmic result to hybrid ones.

1.3 Notations

We will use bold lower-case Roman letters such as $\mathbf{x}$, $\mathbf{a}$, $\mathbf{b}_j$ to denote vectors. Whenever a vector, say $\mathbf{a} \in \mathbb{R}^n$ is present, its components will be denoted by lower-case Roman letters with subscripts, such as $a_1, \ldots, a_n$. Matrices are denoted by bold upper-case Roman letters such as $\mathbf{A}$ and scalars are usually denoted by lower-case Roman letters.

We now enumerate some other notations that are used in this paper.

- $\mathbb{R}^m_+$: the set of m-dimensional vectors with non-negative real entries;
- $\mathbb{P}^n$: the set of vectors $\mathbf{x} \in \mathbb{R}^n_+$ with $\sum_{i=1}^n x_i = 1$;
- $\langle \mathbf{a} | \mathbf{b} \rangle$: the dot-product of two vectors in the same dimension;
- $\|\mathbf{x}\|_p$: the p-norm of vector $\mathbf{x}$, that is, $\left(\sum |x_i^p|\right)^{1/p}$ and $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

2 Grouped Linear-Leontief Markets

Assume there are n traders in the market, denoted by $\mathbf{T} = \{1, 2, \ldots, n\}$. The market contains m groups of commodities, denoted by $\mathbf{G} = \{G_1, \ldots, G_m\}$. Each group G_j contains k_j kinds of commodities.

The trader i's initial endowment of goods is a collection of m vectors: $\{\mathbf{e}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$, where $\mathbf{e}_{j,k}^i$ is the amount of good k in group j held by trader i. For each group j, let the matrix $\mathbf{E}_j = (\mathbf{e}_j^1, \ldots, \mathbf{e}_j^n)$ denote the traders' initial endowments in the groups. We assume that the amount of each commodity is normalized to 1, i.e., $\langle \mathbf{E}_j | \mathbf{1} \rangle = \mathbf{1}$, or equivalently,

$$\sum_{i=1}^{n} \mathbf{e}_{j,k}^{i} = 1, \quad \forall 1 \le j \le m, 1 \le k \le k_j.$$

Similar to the initial endowments, the allocation to trader i is a collection of m vectors, denoted by $\mathbf{x}^i = \{\mathbf{x}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$. The trader i's utility function is characterized by a tuple $\{\mathbf{a}^i \in \mathbb{R}_+^m, \{\mathbf{d}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}\}$. Given an allocation $\mathbf{x}^i = \{\mathbf{x}_j^i \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$, trader i's utility is defined as follows:

$$u_i(\mathbf{x}^i) = \sum_{j=1}^{m} a_j^i v_j^i, \text{ where } v_j^i = \min \left\{ \frac{x_{j,k}^i}{d_{j,k}^i} \,\middle|\, k = 1, 2, \ldots, k_j \right\}$$

In other words, trader i's utility function is a linear combination of m Leontief utility functions.

Locally, each group j is a Leontief economy. That is, every trader i demands the goods in group j in proportion to the vector $\mathbf{d}_j^i$. Therefore, we can introduce the matrix $\mathbf{D}_j = (\mathbf{d}_j^1, \ldots, \mathbf{d}_j^n)$ to characterize the traders' demands in group j. Let $\mathbf{v}_j = (v_j^1, v_j^2, \ldots, v_j^n)^\top$ be an n-dimensional column vector, which can be viewed as an allocation of goods in group j. Then a feasible allocation $\mathbf{v}_j$ of goods in group j should satisfy $\mathbf{D}_j \mathbf{v}_j \le \mathbf{1}$. The allocation of the whole market is denoted by $\mathbf{v} = \{\mathbf{v}_j \in \mathbb{R}_+^n \mid 1 \le j \le m\}$.

Let $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m)$, $\mathbf{E} = (\mathbf{E}_1, \ldots, \mathbf{E}_m)$ and $\mathbf{A} = (a_j^i)$, then the market can be denoted by a tuple $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$. Now we define the exchange equilibrium and approximate equilibrium in this market model.

Definition 1 (Exchange Equilibrium). *An equilibrium is a pair* $(\mathbf{p}, \mathbf{v})$*, where* $\mathbf{p} = \{\mathbf{p}_j \in \mathbb{R}_+^{k_j} \mid 1 \le j \le m\}$ *is a collection of* m *price vectors and* $\mathbf{v} = \{\mathbf{v}_j \in \mathbb{R}_+^n \mid 1 \le j \le m\}$ *is the allocation of the whole market, satisfying that:*

$$\begin{cases} u_i = \sum\limits_{j=1}^{m} a_j^i v_j^i, \forall i = 1, \ldots, n \\ u_i = \max\{\sum\limits_{j=1}^{m} a_j^i z_j^i \mid \sum\limits_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle z_j^i \le \sum\limits_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle\}, \forall i = 1, \ldots, n \\ \mathbf{D}_j \mathbf{v}_j \le \mathbf{1}, \forall j = 1, \ldots, m \end{cases}$$

Definition 2 (ε-approximate Equilibrium). *An* ε*-equilibrium is a pair* $(\mathbf{p}, \mathbf{v})$*, satisfying that:*

$$\begin{cases} u_i = \sum\limits_{j=1}^{m} a_j^i v_j^i, \forall i = 1, \ldots, n \\ u_i \ge (1-\varepsilon)\max\{\sum\limits_{j=1}^{m} a_j^i z_j^i \mid \sum\limits_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle z_j^i \le \sum\limits_{j=1}^{m} \langle \mathbf{p}_j | \mathbf{e}_j^i \rangle\}, \forall i = 1, \ldots, n \\ \mathbf{D}_j \mathbf{v}_j \le (1+\varepsilon)\mathbf{1}, \forall j = 1, \ldots, m \end{cases}$$

3 An Equivalent Equilibrium Condition

Next, we prove a necessary and sufficient condition of an equilibrium. This condition will be useful in our equilibrium computation algorithms.

Theorem 1. *A pair* $(\mathbf{p}, \mathbf{v})$ *is an equilibrium if and only if it satisfies that*

$$\begin{cases} \mathbf{D}_j \mathbf{v}_j \leq \mathbf{1}, & \forall j \\ u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i = \sum_{j=1}^{m} \left\langle \mathbf{p}_j | \mathbf{e}_j^i \right\rangle, & \forall i \\ w_i a_j^i \leq u_i \left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle, & \forall i, j \end{cases} \tag{1}$$

Proof. For each trader i, the pair $(\mathbf{p}, \mathbf{v})$ maximizes his utility if and only if

$$\sum_{j=1}^{m} v_j^i \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle \leq \sum_{j=1}^{m} \left\langle \mathbf{e}_j^i | \mathbf{p}_j \right\rangle \tag{2}$$

$$v_j^i > 0 \Rightarrow a_j^i / \left\langle \mathbf{d}_j^i | \mathbf{p}_j \right\rangle \geq a_k^i / \left\langle \mathbf{d}_k^i | \mathbf{p}_k \right\rangle (\forall k) \tag{3}$$

The first equation is the trader's budget constraint, and the second equation implies that the trader buys only those groups that maximizes his utility gained per unit money spent on the groups.

Note that the equation (2) and (3) can be replaced equivalently by

$$\begin{cases} u_i = \sum_{j=1}^{m} a_j^i v_j^i, \quad w_i = \sum_{j=1}^{m} \left\langle \mathbf{p}_j | \mathbf{e}_j^i \right\rangle \\ \dfrac{a_j^i}{\left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle} \leq \dfrac{u_i}{w_i}, \ \forall j \\ \dfrac{a_j^i}{\left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle} v_j^i \leq \dfrac{u_i}{w_i} v_j^i, \ \forall j \end{cases}$$

Therefore, $(\mathbf{p}, \mathbf{v})$ is an equilibrium if and only if

$$\begin{cases} \mathbf{D}_j \mathbf{v}_j \leq \mathbf{1}, & \forall j \\ u_i = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i = \sum_{j=1}^{m} \left\langle \mathbf{p}_j | \mathbf{e}_j^i \right\rangle, & \forall i \\ w_i a_j^i \leq u_i \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle, & \forall i, j \\ w_i a_j^i v_j^i \leq u_i \langle \mathbf{p}_j | \mathbf{d}_j^i \rangle v_j^i, & \forall i, j \end{cases}$$

Now, it suffices to prove that the last equation can be derived from the other four equations. By $w_i a_j^i \leq u_i \left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle$, we have

$$\begin{array}{llll} & w_i a^i_j v^i_j & \leq u_i \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j, & \forall i, j \\ \Rightarrow & w_i \sum_{j=1}^{m} a^i_j v^i_j & \leq u_i \sum_{j=1}^{m} \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j, & \forall i \\ \Rightarrow & w_i = \sum_{j=1}^{m} \left\langle \mathbf{e}^i_j | \mathbf{p}_j \right\rangle & \leq \sum_{j=1}^{m} \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j, & \forall i \\ \Rightarrow & \sum_{i=1}^{n} \sum_{j=1}^{m} \left\langle \mathbf{e}^i_j | \mathbf{p}_j \right\rangle & \leq \sum_{i=1}^{n} \sum_{j=1}^{m} \left\langle \mathbf{d}^i_j v^i_j | \mathbf{p}_j \right\rangle & \\ \Rightarrow & \sum_{j=1}^{m} \left\langle \mathbf{1} | \mathbf{p}_j \right\rangle & \leq \sum_{j=1}^{m} \left\langle \mathbf{D}_j \mathbf{v}_j | \mathbf{p}_j \right\rangle & \end{array}$$

Since $\mathbf{D}_j \mathbf{v}_j \leq \mathbf{1}$ for all j, we have

$$\begin{cases} \left\langle \mathbf{D}_j \mathbf{v}_j | \mathbf{p}_j \right\rangle = \left\langle \mathbf{1} | \mathbf{p}_j \right\rangle \\ w_i = \sum_{j=1}^{m} \left\langle \mathbf{e}^i_j | \mathbf{p}_j \right\rangle = \sum_{j=1}^{m} \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j, \forall i \end{cases}$$

Again, by $w_i a^i_j \leq u_i \left\langle \mathbf{p}_j | \mathbf{d}^i_j \right\rangle$, we have

$$\begin{array}{lll} w_i a^i_j v^i_j & \leq u_i \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j, & \forall i, j \\ \Rightarrow w_i u_i = w_i \sum_{j=1}^{m} a^i_j v^i_j & \leq u_i \sum_{j=1}^{m} \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j = u_i w_i, & \forall i \end{array}$$

This forced that $w_i a^i_j v^i_j = u_i \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle v^i_j$ for all i, j. ∎

3.1 Solving the Fisher's Model

The Fisher's model is a special case of the Arrow-Debreu's exchange market model. In the Fisher's model, the commodities are held by a seller initially. The traders come to the market with the initial endowments of money, instead of the endowments of commodities in the general setting. The traders buy goods from the seller to maximize each's utility, under the budget constraints. The market is in an equilibrium if the supplies satisfy the demands. Usually, the computation of equilibria in the Fisher's setting is much easier than that in the general case.

Assume in the Fisher's model, trader i has w_i dollars initially. As shown in [15], the equilibrium can be approximated by solving the following convex programming problem:

$$\begin{aligned} &\max \sum_{i=1}^{n} w_i \log(u_i) \\ &\text{s.t.} \begin{cases} u_i = \sum_{j=1}^{m} a^i_j v^i_j & , \forall i = 1, \ldots, n \\ \mathbf{D}_j \mathbf{v}_j \leq \mathbf{1} & , \forall j = 1, \ldots, m \\ \mathbf{v}_j \geq 0 & , \forall j = 1, \ldots, m \end{cases} \end{aligned} \tag{4}$$

With the same argument as in Ye [17], we can prove that

Theorem 2 (Fisher's Equilibrium). *The Fisher's model can be solved by the interior-point algorithm in time $O(\sqrt{Mn}(M+n)^3 L)$, where $M = \sum_{j=1}^{m} k_j$ is the total number of commodities, n is the number of traders and L is the bit-length of the input data.* ∎

4 An Approximation Algorithm

Since Leontief economy is a special case of the hybrid linear-Leontief economy, the hardness results [5,8,12] in Leontief economy can be imported to our case. For example, it is NP-hard to determine the existence of equilibria [5], and there is no algorithm to compute the equilibrium in smoothed polynomial time, unless **PPAD** $\subset$ **RP** [12]. In this section, we propose an approximation algorithm for the grouped linear-Leontief economy, running in

$$\min\left\{O\left((\tau\varepsilon)^{-M+m}\text{poly}(M,n)\right), O\left(\left(\frac{\log(1/\tau)}{\varepsilon}\right)^{2mn}\text{poly}(M,n)\right)\right\} \text{ time,}$$

where $M = \sum_{j=1}^{m} k_j$ is the total number of commodities and $\tau = \min_{i,j,k}\{\mathbf{e}^i_{j,k}, \mathbf{d}^i_{j,k}\}$.

4.1 Intuition

Since the market can be viewed as a linear combination of several Leontief markets, we may expect that it can be reduced to a linear market when the equilibrium information of the sub-markets are given. We first discuss this intuition in this subsection.

Assume the market is $\mathbf{M} = (\mathbf{T}, \mathbf{G}, \mathbf{D}, \mathbf{E}, \mathbf{A})$ and $(\mathbf{p}, \mathbf{v})$ is one of its equilibria. In the following discussion, it is more convenient to replace $\mathbf{p}_j$ by $q_j\mathbf{p}_j$, where $q_j \in \mathbb{R}_+$ and $\|\mathbf{p}_j\|_1 = 1$ is a normalized vector in $\mathbb{R}^{k_j}_+$. Thus the equilibrium $(\mathbf{p}, \mathbf{v})$ is replaced by $(\mathbf{q}, \mathbf{p}, \mathbf{v})$, where $\mathbf{q} \in \mathbb{R}^m_+$.

We define a market $\hat{\mathbf{M}}$ with linear utilities as follows. The set of traders are same as $\mathbf{M}$. For each group $G_j \in \mathbf{G}$, we introduce a commodity j to $\hat{\mathbf{M}}$. The trader i's initial endowment of commodity j is defined by $\hat{e}^i_j = \left\langle \mathbf{e}^i_j | \mathbf{p}_j \right\rangle$ and his preference to commodity j is $\hat{a}^i_j = a^i_j / \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle$. The following lemma is obvious.

Lemma 1 (Market Reduction). *Let* $\hat{x}^i_j = v^i_j \left\langle \mathbf{d}^i_j | \mathbf{p}_j \right\rangle$ *for all* $1 \le i \le n$ *and* $1 \le j \le m$, *then* $(\mathbf{q}, \hat{\mathbf{x}})$ *is an equilibrium for the linear market* $\hat{\mathbf{M}}$. ■

The above lemma shows that if we are so lucky that we know the *internal* price $\mathbf{p}_j$ of every group G_j, the hybrid market can be transformed to a linear market, where every group G_j in the market $\mathbf{M}$ is replaced by a special commodity j in $\hat{\mathbf{M}}$, which plays the role of currency of this group. The traders' endowments and preferences to this group are changed to the endowments and preferences to this currency. $\hat{\mathbf{M}}$ can be viewed as the foreign currency exchange market. The equilibrium price $\mathbf{q}$ of $\hat{\mathbf{M}}$ is the exchange rate between groups, which can be computed in polynomial time, since the market $\hat{\mathbf{M}}$ is linear.

This fact leads to the following approximation heuristic. We exhaustively *enumerate* the internal prices of every group j in the simplex $\mathbb{P}^{k_j}$. With the collection of sampled internal prices $\mathbf{p} = \{\mathbf{p}_j \mid 1 \leq j \leq m\}$, we transform the market $\mathbf{M}$ to the linear market $\hat{\mathbf{M}}$. Then we compute the equilibrium price $\mathbf{q}$ and allocation $\hat{\mathbf{x}}$ in the market $\hat{\mathbf{M}}$. Let $v_j^i = \hat{x}_j^i / \langle \mathbf{d}_j | \mathbf{p}_j \rangle$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$. For every group j, let $\mathbf{v}_j = (v_j^1, \ldots, v_j^n)^\top$. Finally, we check that if $\mathbf{D}_j \mathbf{v}_j \leq \mathbf{1}$ are approximately satisfied. If true, we have found an approximate equilibrium $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ for the original market $\mathbf{M}$.

Before we explicitly present the algorithm, we prepare two important tools in the following two subsections.

4.2 Efficient Sampling Problem

Problem: Given $\varepsilon > 0$ and n vectors $\{\mathbf{x}_i \in \mathbb{P}^k \mid i = 1, \ldots, n\}$, called anchor points, find a sampling set $S \subseteq \mathbb{P}^k$ such that for any $\mathbf{p} \in \mathbb{P}^k$, there exists a sample point $\hat{\mathbf{p}} \in S$ satisfying $1 - \varepsilon \leq \langle \mathbf{p} | \mathbf{x}_i \rangle / \langle \hat{\mathbf{p}} | \mathbf{x}_i \rangle \leq 1 + \varepsilon$ for any anchor point $\mathbf{x}_i$, $1 \leq i \leq n$. The set S is called the *efficient sampling set* of $\{\mathbf{x}_i\}$ and ε. Our goal is to minimize the size of S.

We give two constructions for set S.

Lemma 2. *If $\tau = \min_{i,j}\{\mathbf{x}_{i,j}\} > 0$, then we can construct an efficient sampling set S of size $O(\frac{\log(1/\tau)}{\varepsilon})^n$.*

Proof. For $\mathbf{x}_1$ and any $\mathbf{p} \in \mathbb{P}^k$, we have $\tau \leq \langle \mathbf{p} | \mathbf{x}_1 \rangle \leq 1$.

Define $\log(1/\tau)/\log(1+\varepsilon) \approx \log(1/\tau)/\varepsilon$ planes:

$$\begin{cases} a_0 = \tau, & \text{plane}_0 = \{\mathbf{y} \mid \langle \mathbf{y} | \mathbf{x}_1 \rangle = a_0\}; \\ a_i = (1+\varepsilon)a_{i-1}, & \text{plane}_i = \{\mathbf{y} \mid \langle \mathbf{y} | \mathbf{x}_1 \rangle = a_i\}. \end{cases}$$

These planes cut $\mathbb{P}^k$ into $O(\log(1/\tau)/\varepsilon)$ polytopes, denoted by $P_0, P_1, \ldots$.

For $\mathbf{x}_2$, we similarly define $O(\log(1/\tau)/\varepsilon)$ planes which cut each P_i into at most $O(\log(1/\tau)/\varepsilon)$ polytopes, denoted by $P_{i,0}, P_{i,1}, \ldots$.

Repeat this process for n rounds, we divide simplex $\mathbb{P}^k$ to $O(\log(1/\tau)/\varepsilon)^n$ polytopes. The sampling set S is constructed by picking an inner point from each polytope. ■

Lemma 3. *If $\tau = \min_{i,j}\{\mathbf{x}_{i,j}\} > 0$, then we can construct an efficient sampling set S of size $O((\tau\varepsilon)^{1-k})$.*

Proof. The sampling set S is constructed by meshing the simplex $\mathbb{P}^k$, such that for any $\mathbf{p} \in \mathbb{P}^k$, there exists a $\hat{\mathbf{p}} \in S$ satisfying $\|\mathbf{p} - \hat{\mathbf{p}}\|_\infty \leq \varepsilon\tau$. Obviously, S is an efficient sampling set and is of size $O((\tau\varepsilon)^{1-k})$. ■

In our algorithm, we are going to construct the efficient sampling set S_j for $\{\mathbf{e}_j^i \mid i = 1, \ldots, n\} \cup \{\mathbf{d}_j^i \mid i = 1, \ldots, n\}$ and ε. Let S be $S_1 \times \cdots \times S_m$. The time complexity of the algorithm is dominated by the size of S.

4.3 Convex Optimization Problem

Consider the equilibrium condition in Theorem 1. As in the above discussion, an equilibrium $(\mathbf{p}, \mathbf{v})$ is replaced by a 3-tuple $(\mathbf{q}, \mathbf{p}, \mathbf{v})$. We now introduce the following optimization problem:

$$\begin{array}{rll} min & \theta & \\ s.t.\ \mathbf{D}_j \mathbf{v}_j & \leq (1+\theta)\mathbf{1}, & \forall j \\ u_i & = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i & = \sum_{j=1}^{m} q_j \left\langle \mathbf{p}_j | \mathbf{e}_j^i \right\rangle, & \forall i \\ w_i a_j^i & \leq u_i q_j \left\langle \mathbf{p}_j | \mathbf{d}_j^i \right\rangle, & \forall i, j \\ \langle \mathbf{p}_j | \mathbf{1} \rangle & = 1,\ \mathbf{q} > 0, & \forall j \end{array} \tag{5}$$

The quantity θ can be viewed as the surplus of the demands. We can prove that θ is always nonnegative for any feasible solution of problem (5). The proof is omitted here since it is similar to the one in Ye [17].

Lemma 4. *For any feasible solution* $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ *of (5),* $\theta \geq 0$. *Moreover,* $(\mathbf{q}, \mathbf{p}, \mathbf{v})$ *is an equilibrium if and only if* $\theta = 0$.

Assume we have *guessed* a set of internal prices $\hat{\mathbf{p}} = \{\hat{\mathbf{p}}_j \mid 1 \leq j \leq m\}$, then problem (5) is reduced to the following convex optimization problem, denoted by $\mathrm{Opt}(\hat{\mathbf{p}})$:

$$\begin{array}{rll} min & \theta & \\ s.t.\ \mathbf{D}_j \mathbf{v}_j & \leq (1+\theta)\mathbf{1}, & \forall j \\ u_i & = \sum_{j=1}^{m} a_j^i v_j^i, & \forall i \\ w_i & = \sum_{j=1}^{m} q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{e}_j^i \right\rangle, & \forall i \\ w_i a_j^i & \leq u_i q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{d}_j^i \right\rangle, & \forall i, j \\ \mathbf{q} & > 0, & \forall j \end{array} \tag{6}$$

$\mathrm{Opt}(\hat{\mathbf{p}})$ can be solved in polynomial time [17]. Note that since $\hat{\mathbf{p}}$ may not be an equilibrium internal prices, the optimum of $\mathrm{Opt}(\hat{\mathbf{p}})$ may not be zero.

4.4 The Algorithm

Finally, our algorithm is described in Figure 1. Its correctness is guaranteed by the following lemma. The lemma shows that there exists an internal price $\hat{\mathbf{p}} \in S$ such that the solution of $\mathrm{Opt}(\hat{\mathbf{p}})$ is an ε-approximate equilibrium, according to Definition 2.

Lemma 5. *Assume* $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$ *is an equilibrium. If* $\hat{\mathbf{p}}$ *satisfies that*

$$1 - \varepsilon \leq \frac{\left\langle \mathbf{p}_j^* | \mathbf{d}_j^i \right\rangle}{\left\langle \hat{\mathbf{p}}_j | \mathbf{d}_j^i \right\rangle} \leq 1 + \varepsilon \quad and \quad 1 - \varepsilon \leq \frac{\left\langle \mathbf{p}_j^* | \mathbf{e}_j^i \right\rangle}{\left\langle \hat{\mathbf{p}}_j | \mathbf{e}_j^i \right\rangle} \leq 1 + \varepsilon$$

for all i *and* j, *then the optimum of the problem* $\mathit{Opt}(\hat{\mathbf{p}})$ *satisfies* $\hat{\theta} \leq 3\varepsilon$.

for *each group* G_j **do**
    Construct the efficient sampling set S_j for
    $\{\mathbf{e}^i_j \mid i = 1, \ldots, n\} \cup \{\mathbf{d}^i_j \mid i = 1, \ldots, n\}$ and $\varepsilon/3$.
end
Let $S = S_1 \times \cdots \times S_m$.
for *each* $\hat{\mathbf{p}} \in S$ **do**
    Solve the convex optimization problem $\mathrm{Opt}(\hat{\mathbf{p}})$;
    If the optimum $\hat{\theta} < \varepsilon$, break the loop and output.
end

Fig. 1. An Approximation Algorithm

Proof. Since $(\mathbf{q}^*, \mathbf{p}^*, \mathbf{v}^*)$ is an equilibrium, it should satisfy that

$$\begin{cases} \mathbf{D}_j \mathbf{v}^*_j \leq \mathbf{1}, & \forall j \\ u^*_i \;\; = \sum_{j=1}^{m} a^i_j v^{i*}_j, & \forall i \\ w^*_i \;\; = \sum_{j=1}^{m} q^*_j \left\langle \mathbf{p}^*_j | \mathbf{e}^i_j \right\rangle, & \forall i \\ \lambda^*_i \;\; = w_i / u_i, & \forall i \\ \lambda^*_i \;\; = \min\{q^*_j \left\langle \mathbf{p}^*_j | \mathbf{d}^i_j \right\rangle / a^i_j \mid 1 \leq j \leq m\}, & \forall i \end{cases}$$

We explicitly construct a feasible solution $(\mathbf{q}, \mathbf{v})$ of the problem (6) as follows:

$$\begin{cases} q_j = q^*_j, \quad \forall j \\ \lambda_i = \min\{q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{d}^i_j \right\rangle / a^i_j \mid 1 \leq j \leq m\}, \quad \forall i \\ w_i = \sum_{j=1}^{m} q_j \left\langle \hat{\mathbf{p}}_j | \mathbf{e}^i_j \right\rangle, \quad \forall i \\ u_i = w_i / \lambda_i, \quad \forall i \\ v^i_j = v^{i*}_j \dfrac{u_i}{u^*_i}, \quad \forall i, j \end{cases}$$

Since $\lambda_i / \lambda^*_i \geq 1 - \varepsilon$ and $w_i / w^*_i \geq 1 + \varepsilon$, we have

$$\frac{u_i}{u^*_i} = \frac{w_i}{w^*_i} \frac{\lambda^*_i}{\lambda_i} \leq \frac{1 + \varepsilon}{1 - \varepsilon} \leq 1 + 3\varepsilon$$

and thus, $\mathbf{D}_j \mathbf{v}_j \leq \mathbf{D}_j \mathbf{v}^*_j (1 + 3\varepsilon) \leq 1 + 3\varepsilon$. Therefore, the optimum of (6) must be less or equal to 3ε. ■

The time complexity of our algorithm is $|S| \mathrm{poly}(M, n)$, where $\mathrm{poly}(M, n)$ is spent on solving each optimization problem $\mathrm{Opt}(\hat{\mathbf{p}})$ and $M = \sum_{j=1}^{m} k_j$ is the total number of commodities. According to Lemma 2 and Lemma 3, the size of S is

$$\min \left\{ O\left((\tau\varepsilon)^{-M+m} \right), O\left((\log(1/\tau)/\varepsilon)^{2mn} \right) \right\},$$

where $\tau = \min_{i,j,k} \{ \mathbf{e}^i_{j,k}, \mathbf{d}^i_{j,k} \}$.

5 Discussion

In this paper, we introduce a new family of utility functions — hybrid linear-Leontief functions. We study the computation and approximation of exchange equilibria in markets with grouped linear-Leontief utilities, which are special cases of the hybrid ones. We show that equilibria in the Fisher's model can be found in polynomial time. We also develop an approximation algorithm for approximating equilibria in the Arrow-Debreu's exchange market model. The time complexity of this approximation algorithm depends on the answer to the efficient sampling problem, which is described in Section 4.2. At this moment, it is exponential to either the number of commodities or the number of traders. Any improvement to the sampling problem will improve the performance of our approximation algorithm.

As a grouped hybrid market is a linear combination of Leontief economies, given the fact that linear markets are easy to solve [16,13,17], we conjecture that there exists an approximation algorithm that runs in polynomial time to the number of groups and the number of traders, with an access to an oracle that can compute equilibria in Leontief economies.

More generally, we can extend the concept of hybrid linear-Leontief utility functions to *hierarchical* linear-Leontief utility functions. Such a function can be specified by a tree whose internal vertices are either plus or max operators. Each of its leaves is associated with one commodity. Given an allocation vector, one can evaluate the utility function from bottom up. Clearly, we can use the family of hierarchical utility functions to characterize more complicated market behaviors. With the same technique used in Section 3.1, an equilibrium in the Fisher's setting can be computed efficiently. We hope that, the study to these utilities will lead us to a better understanding of the complexity of computing market equilibria.

References

1. Arrow, J., Debreu, G.: Existence of an equilibrium for a competitive economy. *Econometrica*, 22(3):265–290, 1954.
2. Chen, X., Deng, X.: Settling the Complexity of 2-Player Nash-Equilibrium. To appear in FOCS 2006.
3. Chen, X., Deng, X., Teng, S.-H.: Computing Nash Equilibria: Approximation and smoothed complexity. To appear in FOCS 2006.
4. Chen, X., Deng, X., Teng, S.-H.: Sparse games are hard. To appear in WINE 2006.
5. Codenotti, B., Saberi, A., Varadarajan, K., Ye, Y.: Leontief economies encode nonzero sum two-player games. In: *Proceedings of SODA'06*, pages 659–667, 2006.
6. Codenotti, B., Varadarajan, K.: Efficient computation of equilibrium prices for markets with Leontief utilities. In: *Proceedings of ICALP'04*, pages 371–382, 2004.
7. Deng, X., Papadimitriou, C., Safra, S.: On the complexity of price equilibria. *Journal of Computer and System Sciences*, 67(2):311–324, 2003.
8. Deng, X., Huang, L.-S.: On Complexity of Market Equilibria with Maximum Social Welfare. *Information Processing Letters*, 97 : 4–11, 2006.

9. Deng, X., Huang, L.-S.: Approximate Economic Equilibrium Algorithms. In T. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*. 2005.
10. Devanur, N. R., Papadimitriou, C., Saberi, A., Vazirani, V. V.: Market equilibria via a primal-dual-type algorithm. In *Proceedings of FOCS'02*, pages 389–395, 2002.
11. Garg, R., Kapoor, S.: Auction Algorithms for Market Equilibrium. In: *Proceeding of STOC'04*, pages 511–518, 2004.
12. Huang, L.-S., Teng, S.-H.: On the Approximation and Smoothed Complexity of Leontief Market Equilibria. ECCC TR 06-031, 2006. Also available online http://arXiv.org/abs/cs/0602090, 2006
13. Jain, K.: A polynomial time algorithm for computing the Arrow-Debreu market equilibrium for linear utilities. In: *Proceeding of FOCS'04*, pages 286–294, 2004.
14. Jain, K., Mahdian, M., Saberi, A.: Approximating market equilibria. In: *Proceedings of APPROX'03*, pages 98-108, 2003.
15. Jain, K., Vazirani, V. V., Ye, Y.: Market equilibria for homothetic, quasi-concave utilities and economies of scale in production. In: *Proceedings of SODA'05*, pages 63–71, 2005.
16. Nenakhov, E., Primak, M.: About one algorithm for finding the solution of the Arrow-Debreu model. *Kibernetica*, (3):127–128, 1983.
17. Ye, Y.: A path to the Arrow-Debreu competitive market equilibrium. *Math. Programming*, 2004.
18. Ye, Y.: On exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. In: *Proceedings of WINE'05*, pages 14–23, 2005.

Polynomial Algorithms for Approximating Nash Equilibria of Bimatrix Games*

Spyros C. Kontogiannis[1,3], Panagiota N. Panagopoulou[2,3], and Paul G. Spirakis[2,3]

[1] Computer Science Department, University of Ioannina, 45110 Ioannina, Greece
kontog@cs.uoi.gr
[2] Department of Computer Engineering and Informatics, Patras University, Greece
[3] Research Academic Computer Technology Institute
N. Kazantzaki Str., Patras University, Rion, Patras GR 26500, Greece
panagopp@cti.gr, spirakis@cti.gr

Abstract. We focus on the problem of computing an ϵ-Nash equilibrium of a bimatrix game, when ϵ is an absolute constant. We present a simple algorithm for computing a $\frac{3}{4}$-Nash equilibrium for any bimatrix game in strongly polynomial time and we next show how to extend this algorithm so as to obtain a (potentially stronger) parameterized approximation. Namely, we present an algorithm that computes a $\frac{2+\lambda}{4}$-Nash equilibrium, where λ is the minimum, among all Nash equilibria, expected payoff of either player. The suggested algorithm runs in time polynomial in the number of strategies available to the players.

1 Introduction

Motivation, Framework and Overview. Non-cooperative game theory has been extensively used in understanding the phenomena observed when decision makers interact. A game consists of a set of players, and, for each player, a set of strategies available to her as well as a payoff function mapping each strategy profile (i.e. each combination of strategies, one for each player) to a real number that captures the preferences of the player over the possible outcomes of the game. The most important solution concept in non-cooperative game theory is the notion of *Nash equilibrium* [11]: it is a strategy profile such that no player would have an incentive to unilaterally deviate from her strategy, i.e. no player could increase her payoff by choosing another strategy while the rest of the players persevered their strategies.

Despite the certain existence of such equilibria [11], the problem of finding any Nash equilibrium even for games involving only two players has been recently

* Partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (ARRIVAL) and 015964 "Algorithmic Principles for Building Efficient Overlay Computers" (AEOLUS), and by the General Secretariat for Research and Technology of the Greek Ministry of Development within the programme PENED 2003.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 286–296, 2006.
© Springer-Verlag Berlin Heidelberg 2006

proved to be complete in the PPAD (polynomial parity argument, directed version) class, introduced by Papadimitriou [12]. This fact emerged the computation of *approximate* Nash equilibria, also referred to as ϵ-Nash equilibria. An ϵ-Nash equilibrium is a strategy profile such that no deviating player could achieve a payoff higher than the one that the specific profile gives her, plus ϵ.

In this work, we focus on the problem of approximating Nash equilibria of 2-player games. We propose simple and efficient algorithms for computing ϵ-Nash equilibria of such games, for sufficiently small absolute constants ϵ.

Previous Work. Nash [11] introduced the concept of Nash equilibria in non-cooperative games and proved that any game possesses at least one such equilibrium; however, the computational complexity of finding a Nash equilibrium used to be a wide open problem for several years. A well-known algorithm for computing a Nash equilibrium of a game with 2 players is the Lemke-Howson algorithm [9], however it has exponential worst-case running time in the number of available pure strategies. A simple Las Vegas algorithm for finding a Nash equilibrium in 2-player random games was presented in [2]; this algorithm always finds an equilibrium, and it runs in polynomial time with high probability.

Recently, Daskalakis, Goldberg and Papadimitriou [5] showed that the problem of computing a Nash equilibrium in a game with 4 or more players is PPAD-complete; this result was later extended to games with 3 players [7]. Eventually, Chen and Deng [3] proved that the problem is PPAD-complete for bimatrix games in which each player has n available pure strategies.

In [10], following similar techniques as in [1], it was shown that, for any bimatrix game and for any *constant* $\epsilon > 0$, there exists an ϵ-Nash equilibrium with only logarithmic support (in the number n of available pure strategies). This result directly yields a quasi-polynomial ($n^{O(\ln n)}$) algorithm for computing such an approximate equilibrium.

In [4] it was shown that the problem of computing a $\frac{1}{n^{\Theta(1)}}$-Nash equilibrium is PPAD-complete, and that bimatrix games are unlikely to have a fully polynomial time approximation scheme (unless PPAD $\subseteq$ P). However, it was conjectured that it is unlikely that finding an ϵ-Nash equilibrium is PPAD-complete when ϵ is an absolute constant.

Daskalakis, Mehta and Papadimitriou [6], independently to our work, show how to compute an 1/2-Nash equilibrium of a bimatrix game.

Our Results. In this work, we deal with the problem of computing an ϵ-Nash equilibrium of a bimatrix game, for some constant ϵ. We first present a simple algorithm for computing a $\frac{3}{4}$-Nash equilibrium for any bimatrix game in strongly polynomial time (Lemma 1).

Next we show how to extend this result so as to obtain a parameterized and potentially stronger approximation. More specifically, we present an algorithm that computes a $\frac{2+\lambda}{4}$-Nash equilibrium, where λ is the minimum, among all Nash equilibria, expected payoff of either player (Theorem 3). The suggested algorithm runs in time polynomial in the number of strategies available to the players.

Organization. In Section 2 we present the notation used throughout this paper, together with the definitions of bimatrix games, Nash equilibria and approximate Nash equilibria, and we formally state and discuss the results of [4] and [10] on the problem of approximating Nash equilibria.

Our first algorithm for computing a $\frac{3}{4}$-Nash equilibrium is described in Section 3, while in Section 4 we present an extension of this algorithm that can give a stronger approximation. We conclude, in Section 5, with a discussion of our results and suggestions for further research.

2 Background

2.1 Notation

For an integer n, let $[n] = \{1, 2, \ldots, n\}$. For a $n \times 1$ vector $\mathbf{x}$ we denote by $x_1, x_2, \ldots x_n$ the components of $\mathbf{x}$ and by $\mathbf{x}^T$ the transpose of $\mathbf{x}$. For an $n \times m$ matrix A, we denote $a_{i,j}$ the element in the i-th row and j-th column of A. For an $n \times m$ matrix A and a constant $c \in \mathbb{R}$, we denote cA the $n \times m$ matrix resulting after multiplying each element of A by c. Let $\mathbb{P}^n$ be the set of all probability vectors in n dimensions, i.e.

$$\mathbb{P}^n \equiv \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \sum_{i=1}^{n} x_i = 1 \ \text{ and } x_i \geq 0 \ \text{ for all } \ i \in [n] \right\} \ .$$

Denote $\mathbb{R}^{n \times m}_{[0:1]}$ the set of all $n \times m$ matrices with real entries between 0 and 1, i.e.

$$\mathbb{R}^{n \times m}_{[0:1]} \equiv \left\{ A \in \mathbb{R}^{n \times m} \ : \ 0 \leq a_{i,j} \leq 1 \ \text{ for all } \ i \in [n], j \in [m] \right\} \ .$$

2.2 Bimatrix Games

A *noncooperative game* $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ consists of (i) a finite set of *players* N, (ii) a nonempty finite set of *pure strategies* S_i for each player $i \in N$ and (iii) a *payoff function* $u_i : \times_{i \in N} S_i \to \mathbb{R}$ for each player $i \in N$.

Bimatrix games [8,9] are a special case of 2-player games (i.e. $|N| = 2$) such that the payoff functions can be described by two real $n \times m$ matrices A and B, where $n = |S_1|$ and $m = |S_2|$. More specifically, the n rows of A, B represent the pure strategies of the first player (the *row player*) and the m columns represent the pure strategies of the second player (the *column player*). Then, when the row player chooses strategy i and the column player chooses strategy j, the former gets payoff $a_{i,j}$ while the latter gets payoff $b_{i,j}$. Based on this observation, bimatrix games are denoted by $\Gamma = \langle A, B \rangle$.

A *mixed strategy* for player $i \in N$ is a probability distribution on the set of her pure strategies S_i. In a bimatrix game $\Gamma = \langle A, B \rangle$, a mixed strategy for the row player can be expressed as a probability vector $\mathbf{x} \in \mathbb{P}^n$ while a mixed strategy for the column player can be expressed as a probability vector $\mathbf{y} \in \mathbb{P}^m$. When the row player chooses mixed strategy $\mathbf{x}$ and the column player chooses

$\mathbf{y}$, then the players get *expected payoffs* $\mathbf{x}^T A\mathbf{y}$ (row player) and $\mathbf{x}^T B\mathbf{y}$ (column player). The *support* of a mixed strategy is the set of pure strategies that are assigned non-zero probability.

2.3 Nash Equilibria and ϵ-Nash Equilibria

A Nash equilibrium [11] for a game Γ is a combination of (pure or mixed) strategies, one for each player, such that no player could increase her payoff by unilaterally changing her strategy. We formally give the definition of a Nash equilibrium and an ϵ-Nash equilibrium for a bimatrix game.

Definition 1 (Nash equilibrium). *A pair of strategies* $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ *is a* Nash equilibrium *for the bimatrix game* $\Gamma = \langle A, B\rangle$ *if*

(i) For every (mixed) strategy $\mathbf{x}$ *of the row player,* $\mathbf{x}^T A\tilde{\mathbf{y}} \leq \tilde{\mathbf{x}}^T A\tilde{\mathbf{y}}$ *and*
(ii) For every (mixed) strategy $\mathbf{y}$ *of the column player,* $\tilde{\mathbf{x}}^T B\mathbf{y} \leq \tilde{\mathbf{x}}^T B\tilde{\mathbf{y}}$.

Definition 2 (ϵ-Nash equilibrium). *For any* $\epsilon > 0$ *a pair of strategies* $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ *is an* ϵ-Nash equilibrium *for the bimatrix game* $\Gamma = \langle A, B\rangle$ *if*

(i) For every (mixed) strategy $\mathbf{x}$ *of the row player,* $\mathbf{x}^T A\hat{\mathbf{y}} \leq \hat{\mathbf{x}}^T A\hat{\mathbf{y}} + \epsilon$, *and*
(ii) For every (mixed) strategy $\mathbf{y}$ *of the column player,* $\hat{\mathbf{x}}^T B\mathbf{y} \leq \hat{\mathbf{x}}^T B\hat{\mathbf{y}} + \epsilon$.

Remark 1. From now on we denote by $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ an arbitrary Nash equilibrium of $\langle A, B\rangle$ and by $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ an arbitrary ϵ-Nash equilibrium of $\langle A, B\rangle$, for some constant $\epsilon > 0$ that will be clear from the context.

Positively Normalized Bimatrix Games. As pointed out in [4], since the notion of ϵ-Nash equilibria is defined in the additive fashion, it is important to consider bimatrix games with normalized matrices so as to study their complexity. That is, the absolute value of each entry in the matrices is bounded, for example by 1, and there exists an entry in each matrix equal to 1. [10] also used a similar normalization, which we adopt in this paper and describe it below.

Consider the $n \times m$ bimatrix game $\Gamma = \langle A, B\rangle$ and let c, d be two arbitrary positive real constants. Suppose that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is a Nash equilibrium for Γ and $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is an ϵ-Nash equilibrium for Γ. Let $\mathbf{x}$ and $\mathbf{y}$ be any strategy of the row and column player respectively. Now consider the game $\Gamma' = \langle cA, dB\rangle$. Then it holds that

$$\mathbf{x}^T(cA)\tilde{\mathbf{y}} = c\mathbf{x}^T A\tilde{\mathbf{y}} \leq c\tilde{\mathbf{x}}A\tilde{\mathbf{y}} = \tilde{\mathbf{x}}^T(cA)\tilde{\mathbf{y}}$$

and, similarly,

$$\tilde{\mathbf{x}}^T(dB)\mathbf{y} \leq \tilde{\mathbf{x}}^T(dB)\tilde{\mathbf{y}}\ .$$

Moreover,

$$\mathbf{x}^T(cA)\hat{\mathbf{y}} \leq \hat{\mathbf{x}}^T(cA)\hat{\mathbf{y}} + c\epsilon$$

and

$$\hat{\mathbf{x}}^T(dB)\mathbf{y} \leq \hat{\mathbf{x}}^T(dB)\hat{\mathbf{y}} + d\epsilon\ .$$

Hence Γ and Γ' have precisely the same set of Nash equilibria; furthermore, any ϵ-Nash equilibrium for Γ is a $\ell\epsilon$-Nash equilibrium for Γ' (where $\ell = \max\{c, d\}$) and vice versa.

Now let C be an $n \times m$ matrix such that, for all (columns) $j \in [m]$, $c_{i,j} = c_j \in \mathbb{R}$ for all $i \in [n]$. Similarly, let D be an $n \times m$ matrix such that, for all (rows) $i \in [m]$, $d_{i,j} = d_i \in \mathbb{R}$ for all $j \in [m]$. Note that, for every pair $\mathbf{x} \in \mathbb{P}^n$ and $\mathbf{y} \in \mathbb{P}^m$,

$$\mathbf{x}^T C\mathbf{y} = \sum_{j=1}^{m}\sum_{i=1}^{n} c_{i,j}x_i y_j = \sum_{j=1}^{m} y_j \sum_{i=1}^{n} c_j x_i = \sum_{j=1}^{m} c_j y_j$$

and

$$\mathbf{x}^T D\mathbf{y} = \sum_{i=1}^{n}\sum_{j=1}^{m} d_{i,j}x_i y_j = \sum_{i=1}^{n} x_i \sum_{j=1}^{m} d_i y_j = \sum_{i=1}^{n} d_i x_i \ .$$

Consider now the game $\Gamma'' = \langle C + A, D + B\rangle$. Then, for all $\mathbf{x} \in \mathbb{P}^n$,

$$\mathbf{x}^T(C + A)\tilde{\mathbf{y}} = \mathbf{x}^T C\tilde{\mathbf{y}} + \mathbf{x}^T A\tilde{\mathbf{y}} \leq \sum_{j=1}^{m} c_j\tilde{y}_j + \tilde{\mathbf{x}}^T A\tilde{\mathbf{y}} = \tilde{\mathbf{x}}^T(C + A)\tilde{\mathbf{y}}$$

and similarly, for all $\mathbf{y} \in \mathbb{P}^m$,

$$\tilde{\mathbf{x}}^T(D + B)\mathbf{y} \leq \tilde{\mathbf{x}}^T(D + B)\tilde{\mathbf{y}} \ .$$

Also, for all $\mathbf{x} \in \mathbb{P}^n$ it holds that

$$\mathbf{x}^T(C + A)\hat{\mathbf{y}} = \mathbf{x}^T C\hat{\mathbf{y}} + \mathbf{x}^T A\hat{\mathbf{y}} \leq \sum_{j=1}^{m} c_j\hat{y}_j + \hat{\mathbf{x}}^T A\hat{\mathbf{y}} + \epsilon = \hat{\mathbf{x}}^T(C + A)\hat{\mathbf{y}} + \epsilon$$

and similarly, for all $\mathbf{y} \in \mathbb{P}^m$,

$$\hat{\mathbf{x}}^T(D + B)\mathbf{y} \leq \hat{\mathbf{x}}^T(D + B)\hat{\mathbf{y}} + \epsilon \ .$$

Thus Γ and Γ'' are equivalent as regards their sets of Nash equilibria, as well as their sets of ϵ-Nash equilibria.

This equivalence allows us to focus only on bimatrix games where the payoffs are between 0 and 1, i.e. on games $\langle A, B\rangle$ where $A, B \in \mathbb{R}_{[0:1]}^{m\times n}$. Such games are referred to as *positively normalized* [4].

2.4 Tractability of ϵ-Nash Equilibria

Consider a bimatrix game $\Gamma = \langle A, B\rangle$ and let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ be a Nash equilibrium for Γ. Fix a positive integer k and assume that we form a multiset S_1 by sampling k times from the set of pure strategies of the row player, independently at random according to the distribution $\tilde{\mathbf{x}}$. Similarly, assume we form a multiset S_2 by sampling k times from set of pure strategies of the column player, independently at random according to the distribution $\tilde{\mathbf{y}}$. Let $\hat{\mathbf{x}}$ be the mixed strategy for the row player that assigns probability $1/k$ to each member of S_1 and 0 to all other pure strategies, and let $\hat{\mathbf{y}}$ be the mixed strategy for the column player that assigns probability $1/k$ to each member of S_2 and 0 to all other pure strategies. Clearly, if a pure strategy occurs α times in the multiset, then it is assigned probability α/k. Then $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are called *k-uniform* [10] and the following holds:

Theorem 1 ([10]). *For any Nash equilibrium $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ of a positively normalized $n \times n$ bimatrix game and for every $\epsilon > 0$, there exists, for every $k \geq \frac{12 \ln n}{\epsilon^2}$, a pair of k-uniform strategies $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ such that $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is an ϵ-Nash equilibrium.*

However,

Theorem 2 ([4]). *The problem of computing a $\frac{1}{n^{\Theta(1)}}$-Nash equilibrium of a positively normalized $n \times n$ bimatrix game is* PPAD*-complete.*

Theorem 2 asserts that, unless $\mathsf{PPAD} \subseteq \mathsf{P}$, there exists no fully polynomial time approximation scheme for computing equilibria in bimatrix games. However, this does not rule out the existence of a polynomial approximation scheme for computing an ϵ-Nash equilibrium when ϵ is an absolute constant, or even when $\epsilon = \Theta\left(\frac{1}{poly(\ln n)}\right)$. Furthermore, as observed in [4], if the problem of finding an ϵ-Nash equilibrium were PPAD-complete when ϵ is an absolute constant, then, due to Theorem 1, all PPAD problems would be solved in quasi-polynomial time, which is unlikely to be the case.

3 A $\frac{3}{4}$-Nash Equilibrium

In this section we present a straightforward method for computing a $\frac{3}{4}$-Nash equilibrium for any positively normalized bimatrix game.

Lemma 1. *Consider any positively normalized $n \times m$ bimatrix game $\Gamma = \langle A, B\rangle$ and let $a_{i_1,j_1} = \max_{i,j} a_{i,j}$ and $b_{i_2,j_2} = \max_{i,j} b_{i,j}$. Then the pair of strategies $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ where $\hat{x}_{i_1} = \hat{x}_{i_2} = \hat{y}_{j_1} = \hat{y}_{j_2} = \frac{1}{2}$ is a $\frac{3}{4}$-Nash equilibrium for Γ.*

Proof. First observe that

$$\begin{aligned}
\hat{\mathbf{x}}^T A \hat{\mathbf{y}} &= \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{x}_i \hat{y}_j a_{i,j} \\
&= \hat{x}_{i_1} \hat{y}_{j_1} a_{i_1,j_1} + \hat{x}_{i_1} \hat{y}_{j_2} a_{i_1,j_2} + \hat{x}_{i_2} \hat{y}_{j_1} a_{i_2,j_1} + \hat{x}_{j_1} \hat{y}_{j_1} a_{i_2,j_2} \\
&= \frac{1}{4} \left(a_{i_1,j_1} + a_{i_1,j_2} + a_{i_2,j_1} + a_{i_2,j_2}\right) \\
&\geq \frac{1}{4} a_{i_1,j_1} \quad .
\end{aligned}$$

Similarly,

$$\hat{\mathbf{x}}^T B \hat{\mathbf{y}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{x}_i \hat{y}_j b_{i,j}$$

$$= \hat{x}_{i_1}\hat{y}_{j_1}b_{i_1,j_1} + \hat{x}_{i_1}\hat{y}_{j_2}b_{i_1,j_2} + \hat{x}_{i_2}\hat{y}_{j_1}b_{i_2,j_1} + \hat{x}_{j_1}\hat{y}_{j_1}b_{i_2,j_2}$$
$$= \frac{1}{4}\left(b_{i_1,j_1} + b_{i_1,j_2} + b_{i_2,j_1} + b_{i_2,j_2}\right)$$
$$\geq \frac{1}{4}b_{i_2,j_2} \ .$$

Now observe that, for any (mixed) strategies $\mathbf{x}$ and $\mathbf{y}$ of the row and column player respectively,

$$\mathbf{x}^T A\hat{\mathbf{y}} \leq a_{i_1,j_1} \quad \text{and} \quad \hat{\mathbf{x}}^T B\mathbf{y} \leq b_{i_2,j_2}$$

and recall that $a_{i,j}, b_{i,j} \in [0,1]$ for all $i \in N$, $j \in M$. Hence

$$\mathbf{x}^T A\hat{\mathbf{y}} \leq a_{i_1,j_1} = \frac{1}{4}a_{i_1,j_1} + \frac{3}{4}a_{i_1,j_1} \leq \hat{\mathbf{x}}^T A\hat{\mathbf{y}} + \frac{3}{4}$$

and

$$\hat{\mathbf{x}}^T B\mathbf{y} \leq b_{i_2,j_2} = \frac{1}{4}b_{i_2,j_2} + \frac{3}{4}a_{i_2,j_2} \leq \hat{\mathbf{x}}^T B\hat{\mathbf{y}} + \frac{3}{4} \ .$$

Thus $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a $\frac{3}{4}$-Nash equilibrium for Γ. □

4 A Parameterized Approximation

We now proceed in extending the technique used in the proof of Lemma 1 so as to obtain a parameterized, stronger approximation.

Theorem 3. *Consider a positively normalized $n\times m$ bimatrix game $\Gamma = \langle A, B\rangle$. Let λ_1^* (λ_2^*) be the minimum, among all Nash equilibria of Γ, expected payoff for the row (column) player and let $\lambda = \max\{\lambda_1^*, \lambda_2^*\}$. Then, there exists a $\frac{2+\lambda}{4}$-Nash equilibrium that can be computed in time polynomial in n and m.*

Proof. Observe that, for any pair of strategies $\mathbf{x}, \mathbf{y}$ of the row and column player respectively, it holds that $\mathbf{x}^T A\mathbf{y} \in [0,1]$ and $\mathbf{x}^T B\mathbf{y} \in [0,1]$. Consider the following linear programs LP1 and LP2:

LP1	LP2
minimize t subject to $\sum_{j=1}^m a_{i,j}y_j \leq t \quad \forall i \in [n]$ $\sum_{j=1}^m y_j = 1$ $y_j \geq 0 \quad \forall j \in [m]$	minimize s subject to $\sum_{j=1}^n b_{i,j}x_i \leq s \quad \forall j \in [m]$ $\sum_{i=1}^n x_i = 1$ $x_i \geq 0 \quad \forall i \in [n]$

Let $t^*, \mathbf{y}^*$ and $s^*, \mathbf{x}^*$ correspond to the optimal solutions of LP1 and LP2 respectively. Then, there exists at least one row $r \in [n]$ such that $\sum_j a_{r,j}y_j^* = t^*$. Similarly, there exists at least one column $c \in [m]$ such that $\sum_i b_{i,c}x_i^* = s^*$.

Let λ_1^* be the minimum, among all Nash equilibria of Γ, expected payoff for the row player and let $(\tilde{\mathbf{x}}', \tilde{\mathbf{y}}')$ be the corresponding Nash equilibrium. Then $\lambda_1^*, \tilde{\mathbf{y}}'$ is a feasible solution for LP1, thus $t^* \leq \lambda_1^*$. Similarly, let λ_2^* be the minimum, among all Nash equilibria of Γ, expected payoff for the row player and let $(\tilde{\mathbf{x}}'', \tilde{\mathbf{y}}'')$ be the corresponding Nash equilibrium. Then $\lambda_2^*, \tilde{\mathbf{y}}''$ is a feasible solution for LP2, thus $s^* \leq \lambda_2^*$. Now let $\lambda = \max\{\lambda_1^*, \lambda_2^*\}$. Thus $t^* \leq \lambda$ and $s^* \leq \lambda$.

Now consider the pair of strategies $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ for the row and column player respectively defined as follows:

$$\begin{aligned}
\hat{x}_i &= \frac{x_i^*}{2} \quad \forall i \in [n] - \{r\} \\
\hat{x}_r &= \frac{x_r^*}{2} + \frac{1}{2} \\
\hat{y}_j &= \frac{y_j^*}{2} \quad \forall j \in [m] - \{c\} \\
\hat{y}_c &= \frac{y_c^*}{2} + \frac{1}{2} \ .
\end{aligned}$$

Observe that

$$\begin{aligned}
\hat{\mathbf{x}}^T A \hat{\mathbf{y}} &= \sum_{i=1}^{n} \hat{x}_i \sum_{j=1}^{m} \hat{y}_j a_{i,j} \\
&= \sum_{i \neq r} \frac{x_i^*}{2} \sum_{j \neq c} \frac{y_j^*}{2} a_{i,j} + \sum_{i \neq r} \frac{x_i^*}{2} \left(\frac{y_c^*}{2} + \frac{1}{2} \right) a_{i,c} \\
&\quad + \left(\frac{x_r^*}{2} + \frac{1}{2} \right) \sum_{j \neq c} \frac{y_j^*}{2} a_{r,j} + \left(\frac{x_r^*}{2} + \frac{1}{2} \right) \left(\frac{y_c^*}{2} + \frac{1}{2} \right) a_{r,c} \\
&\geq \frac{1}{4} \sum_{j=1}^{m} a_{r,j} y_j^* \\
&= \frac{t^*}{4} \ .
\end{aligned}$$

Furthermore, for each row $i \in [n]$,

$$\begin{aligned}
\sum_{j=1}^{m} \hat{y}_j a_{i,j} &= \sum_{j=1}^{m} \frac{y_j^*}{2} a_{i,j} + \frac{1}{2} a_{i,c} \\
&\leq \frac{t^*}{2} + \frac{1}{2} \\
&\leq \hat{\mathbf{x}}^T A \hat{\mathbf{y}} + \frac{2 + t^*}{4} \\
&\leq \hat{\mathbf{x}}^T A \hat{\mathbf{y}} + \frac{2 + \lambda}{4} \ .
\end{aligned}$$

Similarly,

$$\begin{aligned}
\hat{\mathbf{x}}^T B \hat{\mathbf{y}} &= \sum_{j=1}^{m} \hat{y}_j \sum_{i=1}^{n} \hat{x}_i b_{i,j} \\
&= \sum_{j \neq c} \frac{y_j^*}{2} \sum_{i \neq r} \frac{x_i^*}{2} b_{i,j} + \sum_{j \neq c} \frac{y_j^*}{2} \left(\frac{x_r^*}{2} + \frac{1}{2} \right) b_{r,j}
\end{aligned}$$

$$
\begin{aligned}
&+\left(\frac{y_c^*}{2}+\frac{1}{2}\right)\sum_{i\neq r}\frac{x_i^*}{2}b_{i,c}+\left(\frac{y_c^*}{2}+\frac{1}{2}\right)\left(\frac{x_r^*}{2}+\frac{1}{2}\right)b_{r,c}\\
&\geq \frac{1}{4}\sum_{i=1}^{n} b_{i,c}x_i^*\\
&= \frac{s^*}{4}
\end{aligned}
$$

and, for each column $j \in [m]$,

$$
\begin{aligned}
\sum_{i=1}^{n}\hat{x}_i b_{i,j} &= \sum_{i=1}^{n}\frac{x_i^*}{2}b_{i,j}+\frac{1}{2}b_{r,j}\\
&\leq \frac{s^*}{2}+\frac{1}{2}\\
&\leq \hat{\mathbf{x}}^T B\hat{\mathbf{y}}+\frac{2+s^*}{4}\\
&\leq \hat{\mathbf{x}}^T B\hat{\mathbf{y}}+\frac{2+\lambda}{4}\ .
\end{aligned}
$$

Thus, $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a $\frac{2+\lambda}{4}$-Nash equilibrium for $\mathit{\Gamma}$ that can be computed in polynomial time. $\square$

Note that, for any bimatrix game $\mathit{\Gamma} = \langle A, B\rangle$, we can check in polynomial time whether there exists a Nash equilibrium in which each player chooses with probability 1 one of her pure strategies (i.e. a *pure Nash equilibrium*). If there exists such an equilibrium, then we can find it in polynomial time and there is no point in searching for ϵ-Nash equilibria. On the other hand, if all Nash equilibria are not pure, then the payoff of either player is strictly less than 1, hence $\lambda = \max\{\lambda_1^*, \lambda_2^*\} < 1$. Thus $\frac{2+\lambda}{4} < \frac{3}{4}$, assuring that the the algorithm described in the above proof yields a stronger approximation than the one presented in Section 3.

An Application. The approximation factor achieved by the algorithm we just described depends on λ_1^* and λ_2^*. We believe that, in most situations, there exists a Nash equilibrium such that the payoff of the row player is small, and that there exists a (possibly different) Nash equilibrium such that the payoff of the column player is small, and thus the approximation achieved is close to $\frac{1}{2}$.

As an example, consider the $n \times n$ *generalized matching pennies* game $\mathit{\Gamma} = \langle A, B\rangle$ where A and B are described as follows:

$$
\begin{aligned}
a_{i,j} &= \begin{cases} 1 \text{ if } i = j \\ 0 \text{ else} \end{cases}\\
b_{i,j} &= \begin{cases} 1 \text{ if } j = i(\mathrm{mod} n) + 1 \\ 0 \text{ else} \end{cases}\ .
\end{aligned}
$$

Observe that the pair of strategies $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ where $\tilde{x}_i = \tilde{y}_i = \frac{1}{n}$ for all $i \in [n]$ is a Nash equilibrium of the generalized matching pennies game. Indeed, for any $\mathbf{x}, \mathbf{y} \in \mathbb{P}^n$,

$$\mathbf{x}^T A\tilde{\mathbf{y}} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j} = \frac{1}{n^2}n = \frac{1}{n} = \tilde{\mathbf{x}}^T A\tilde{\mathbf{y}}$$

$$\tilde{\mathbf{x}}^T B\mathbf{y} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{i,j} = \frac{1}{n^2}n = \frac{1}{n} = \tilde{\mathbf{x}}^T B\tilde{\mathbf{y}} \ .$$

Thus $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is a Nash equilibrium[1] that gives each player a payoff equal to $\frac{1}{n}$. By applying Theorem 3, we can obtain in polynomial time a $\frac{1+1/n}{2}$-Nash equilibrium. Thus we can guarantee an approximation factor that tends to $\frac{1}{2}$ as $n \to \infty$.

5 Conclusions

In this paper we tried to approximate, within a constant additive factor, the problem of computing a Nash equilibrium in an arbitrary $n \times m$ bimatrix game.

The (additive) approximation parameter achieved by the algorithm described in the above proof of Theorem 3 depends on λ_1^* and λ_2^*, i.e. the minimum payoff, over all Nash equilibria, for the row and column player respectively. Observe that, so long as not *all* Nash equilibria of the game give payoffs very close to 1 for either player, the algorithm gives an approximation very close to $\frac{1}{2}$. In other words, it suffices that there exists a Nash equilibrium that gives row player a payoff close to 0 and a Nash equilibrium (not necessarily the same!) that gives column player a payoff close to 0 so that the approximation achieved can be assured to be close to $\frac{1}{2}$. Furthermore, this is just a *sufficient* and not a necessary condition: recall that we only used λ_1^* and λ_2^* so as to prove the existence of feasible solutions to some linear constraints.

Furthermore, for both Lemma 1 and Theorem 3, we used a factor of $\frac{1}{2}$ to deal with the underlying Linear Complementarity Problem. More specifically, we tried to compute independently for each player a strategy that guarantees her a sufficiently large payoff, and then we "merged" in an equivalent way the strategies found with the ones needed by the other player so as to approximate a Nash equilibrium. We observed that, for the specific algorithms presented in these results, this factor of $\frac{1}{2}$ is optimal.

Albeit simple, we believe that the techniques described here are a first step towards establishing whether there exists any approximation scheme for computing an ϵ-Nash equilibrium and that our methods can be extended in order to achieve stronger approximations to the problem of finding Nash equilibria of bimatrix games.

Acknowledgments. We thank Dimitris Fotakis for contributing to our work an improvement to Theorem 3.

[1] In fact it can be proved that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is the *unique* Nash equilibrium of the generalized matching pennies game.

References

1. I. Althöfer. “On Sparse Approximations to Randomized Strategies and Convex Combinations”. *Linear Algebra and Applications*, 199:339355, 1994.
2. Imre Barany, Santosh Vempala and Adrian Vetta: “Nash Equilibria in Random Games”. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 123–131, 2005.
3. Xi Chen and Xiaotie Deng: “Settling the Complexity of 2-Player Nash-Equilibrium”. *Electronic Colloquium on Computational Complexity (ECCC)*, TR05-140, 2005.
4. Xi Chen, Xiaotie Deng and Shang-Hua Teng: “Computing Nash Equilibria: “Approximation and Smoothed Complexity”. *Electronic Colloquium on Computational Complexity (ECCC)*, TR06-023, 2006.
5. Constantinos Daskalakis, P. W. Goldberg and Christos Papadimitriou: “ The Complexity of Computing a Nash Equilibrium”. *Electronic Colloquium on Computational Complexity (ECCC)*, TR05-115, 2005.
6. Constantinos Daskalakis, Aranyak Mehta and Christos Papadimitriou: “A Note on Approximate Nash Equilibria”. *2nd Workshop on Internet and Network Economics (WINE'06)*, 2006.
7. Constantinos Daskalakis and Christos Papadimitriou: “Three-player Games are Hard”. *Electronic Colloquium on Computational Complexity (ECCC)*, TR05-139, 2005.
8. C. E. Lemke: “Bimatrix Equilibrium Points and Mathematical Programming”. *Management Science*, Vol. 11, pp. 681–689, 1965.
9. C. E. Lemke and J. T. Howson: “Equilibrium Points of Bimatrix Games”. *J. Soc. Indust. Appl. Math.*, Vol. 12, pp. 413–423, 1964.
10. Richard J. Lipton, Evangelos Markakis and Aranyak Mehta: “Playing Large Games using Simple Startegies”. In *EC '03: Proceedings of the 4th ACM Conference on Electronic Commerce*, pp. 36–41, 2003.
11. J. Nash: “Noncooperative Games”. *Annals of Mathematics*, Vol. 54, pp. 289–295, 1951.
12. C. H. Papadimitriou: “On Inefficient Proofs of Existence and Complexity Classes”. In *Proceedings of the 4th Czechoslovakian Symposium on Combinatorics*, 1991.

A Note on Approximate Nash Equilibria

Constantinos Daskalakis[1], Aranyak Mehta[2], and Christos Papadimitriou[1]

[1] University of California, Berkeley, USA*
[2] IBM Almaden Research Center, San Jose, USA

Abstract. In view of the intractability of finding a Nash equilibrium, it is important to understand the limits of approximation in this context. A subexponential approximation scheme is known [LMM03], and no approximation better than $\frac{1}{4}$ is possible by any algorithm that examines equilibria involving fewer than $\log n$ strategies [Alt94]. We give a simple, linear-time algorithm examining just two strategies per player and resulting in a $\frac{1}{2}$-approximate Nash equilibrium in any 2-player game. For the more demanding notion of *well-supported approximate equilibrium* due to [DGP06] no nontrivial bound is known; we show that the problem can be reduced to the case of win-lose games (games with all utilities $0-1$), and that an approximation of $\frac{5}{6}$ is possible contingent upon a graph-theoretic conjecture.

1 Introduction

Since it was shown that finding a Nash equilibrium is PPAD-complete [DGP06], even for 2-player games [CD05], the question of approximate Nash equilibrium emerged as the central remaining open problem in the area of equilibrium computation. Assume that all utilities have been normalized to be between 0 and 1 (this is a common assumption, since scaling the utilities of a player by any positive factor, and applying any additive constant, results in an equivalent game). A set of mixed strategies is called an *ϵ-approximate Nash equilibrium*, where $\epsilon > 0$, if for each player all strategies have expected payoff that is at most ϵ more than the expected payoff of the given strategy. Clearly, any mixed strategy combination is a 1-approximate Nash equilibrium, and it is quite straightforward to find a $\frac{3}{4}$-approximate Nash equilibrium by examining all supports of size two. In fact, [KPS06] provides a scheme that yields, for every $\epsilon > 0$, in time polynomial in the size of the game and $\frac{1}{\epsilon}$, a $\frac{2+\epsilon+\lambda}{4}$-approximate Nash equilibrium, where λ is the minimum, among all Nash equilibria, expected payoff of either player. In [LMM03] it was shown that, for every $\epsilon > 0$, an ϵ-approximate Nash equilibrium can be found in time $O(n^{\frac{\log n}{\epsilon^2}})$ by examining all supports of size $\frac{\log n}{\epsilon^2}$. It was pointed out in [Alt94] that, even for zero-sum games, no algorithm that examines supports smaller than about $\log n$ can achieve an approximation better than $\frac{1}{4}$. Can this gap between $\frac{1}{4}$ and $\frac{3}{4}$ be bridged by looking at small supports? And how can the barrier of $\frac{1}{4}$ be broken in polynomial time?

* Supported by NSF grant CCF-0515259.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 297–306, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this note we concentrate on 2-player games. We point out that a straightforward algorithm, looking at just three strategies in total, achieves a $\frac{1}{2}$-approximate Nash equilibrium. The algorithm is very intuitive: For any strategy i of the row player let j be the best response of the column player , and let k be the best response of the row player to j. Then the row player plays an equal mixture of i and k, while the column player plays j. The proof of $\frac{1}{2}$-approximation is rather immediate.

We also examine a more sophisticated approximation concept due to [GP06, DGP06], which we call here the *well-supported ϵ-approximate Nash equilibrium*, which does not allow in the support strategies that are suboptimal by at least ϵ. For this concept no approximation constant better than 1 is known. We show that the problem is reduced — albeit with a loss in the approximation ratio — to the case in which all utilities are either zero or one (this is often called the "win-lose case"). We also prove that, assuming a well-studied and plausible graph-theoretic conjecture, in win-lose games there is a well-supported $\frac{2}{3}$-approximate Nash equilibrium with supports of size at most three (and of course it can be found in polynomial time). This yields a well-supported $\frac{5}{6}$-approximate Nash equilibrium for any game.

2 Definitions

We consider normal form games between two players, the *row player* and the *column player,* each with n strategies at his disposal. The game is defined by two $n \times n$ payoff matrices, R for the row player, and C for the column player. The pure strategies of the row player correspond to the n rows and the pure strategies of the column player correspond to the n columns. If the row player plays row i and the column player plays column j, then the row player receives a payoff of R_{ij} and the column player gets C_{ij}. Payoffs are extended linearly to pairs of mixed strategies — if the row player plays a probability distribution x over the rows and column player plays a distribution y over the columns, then the row player gets a payoff of $x^T Ry$ and the column player gets a payoff of $x^T Cy$.

A *Nash equilibrium* in this setting is a pair of mixed strategies, x^* for the row player and y^* for the column player, such that neither player has an incentive to unilaterally defect. Note that, by linearity, the best defection is to a pure strategy. Let e_i denote the vector with a 1 at the ith coordinate and 0 elsewhere. A pair of mixed strategies (x^*, y^*) is a Nash equilibrium if

$$\forall\ i = 1..n, \quad e_i^T Ry^* \ \leq \ {x^*}^T Ry^*$$

$$\forall\ i = 1..n, \quad {x^*}^T Ce_i \ \leq \ {x^*}^T Cy^*$$

It can be easily shown that every pair of equilibrium strategies of a game does not change upon multiplying all the entries of a payoff matrix by a constant, and upon adding the same constant to each entry. We shall therefore assume that the entries of both payoff matrices R and C are between 0 and 1.

For $\epsilon > 0$, we define an *ϵ-approximate Nash equilibrium* to be a pair of mixed strategies x^* for the row player and y^* for the column player, so that the incentive to unilaterally deviate is at most ϵ:

$$\forall\ i = 1..n,\quad e_i^T R y^* \ \le\ {x^*}^T R y^* + \epsilon$$

$$\forall\ i = 1..n,\quad {x^*}^T C e_i \ \le\ {x^*}^T C y^* + \epsilon$$

A stronger notion of approximately equilibrium strategies was introduced in [GP06, DGP06]: For $\epsilon > 0$, a *well-supported ϵ-approximate Nash equilibrium*, or an *ϵ-well-supported Nash equilibrium*, is a pair of mixed strategies, x^* for the row player and y^* for the column player, so that a player plays only approximately best-response pure strategies with non-zero probability:

$$\forall\ i:\ x_i^* > 0 \Rightarrow\quad e_i^T R y^* \ge e_j^T R y^* - \epsilon,\quad \forall\ j$$

$$\forall\ i:\ y_i^* > 0 \Rightarrow\quad {x^*}^T C e_i \ge {x^*}^T C e_j - \epsilon,\quad \forall\ j$$

If only the first set of inequalities holds, we say that every pure strategy in the support of x^* is *ϵ-well supported against y^**, and similarly for the second. This is indeed a stronger definition, in the sense that every ϵ-well supported Nash equilibrium is also an ϵ-approximate Nash equilibrium, but the converse need not be true. However, the following lemma from [CDT06] shows that there does exist a polynomial relationship between the two:

Lemma 1. *[CDT06] For every 2 player normal form game, for every $\epsilon > 0$, given an $\frac{\epsilon}{8n}$-approximate equilibrium we can compute in polynomial time an ϵ-well-supported equilibrium.*

Since in this paper we are interested in constant ϵ, this lemma is of no help to us; indeed, our results for approximate equilibria are stronger and simpler than those for well-supported equilibria.

3 A Simple Algorithm

We provide here a simple way of computing a $\frac{1}{2}$-approximate Nash equilibrium: Pick an arbitrary row for the row player, say row i. Let $j = \arg\max_{j'} C_{ij'}$. Let $k = \arg\max_{k'} R_{k'j}$. Thus, j is a best-response column for the column player to the row i, and k is a best-response row for the row player to the column j.

The equilibrium is $x^* = \frac{1}{2}e_i + \frac{1}{2}e_k$ and $y^* = e_j$, i.e., the row player plays row i or row k with probability $\frac{1}{2}$ each, while the column player plays column j with probability 1.

Theorem 1. *The strategy pair (x^*, y^*) is a $\frac{1}{2}$-approximate Nash equilibrium.*

Proof. The row player's payoff under (x^*, y^*) is ${x^*}^T R y^* = \frac{1}{2}R_{ij} + \frac{1}{2}R_{kj}$. By construction, one of his best responses to y^* is to play the pure strategy on row k, which gives a payoff of R_{kj}. Hence his incentive to defect is equal to the difference:

$$R_{kj} - (\tfrac{1}{2}R_{ij} + \tfrac{1}{2}R_{kj}) = \tfrac{1}{2}R_{kj} - \tfrac{1}{2}R_{ij} \quad \leq \quad \tfrac{1}{2}R_{kj} \quad \leq \quad \tfrac{1}{2}$$

The column player's payoff under (x^*, y^*) is ${x^*}^T C y^* = \frac{1}{2}C_{ij} + \frac{1}{2}C_{kj}$. Let j' be a best pure strategy response of the column player to x^*: this strategy gives the column player a value of $\frac{1}{2}C_{ij'} + \frac{1}{2}C_{kj'}$, hence his incentive to defect is equal to the difference:

$$\begin{aligned}(\tfrac{1}{2}C_{ij'} + \tfrac{1}{2}C_{kj'}) - (\tfrac{1}{2}C_{ij} + \tfrac{1}{2}C_{kj}) &= \tfrac{1}{2}(C_{ij'} - C_{ij}) + \tfrac{1}{2}(C_{kj'} - C_{kj}) \\ &\leq 0 + \tfrac{1}{2}(C_{kj'} - C_{kj}) \\ &\leq \tfrac{1}{2} \qquad (1)\end{aligned}$$

Here the first inequality follows from the fact that column j was a best response to row i, by the first step of the construction.

4 Well Supported Nash Equilibria

The algorithm of the previous section yields equilibria that are, in the worst case, as bad as 1-well supported. In this section we address the harder problem of finding ϵ-well supported equilibria for some $\epsilon < 1$.

Our construction has two components. In the first we transform the given 2-player game into a new game by rearranging and potentially discarding or duplicating some of the columns of the original game. The transformation will be such that well supported equilibria in the new game can be mapped back to well supported equilibria of the original game; moreover, the mapping will result in some sort of *decorrelation of the players*, in the sense that computation of well supported equilibria in the decorrelated game can be carried out by looking at the row player. The second part of the construction relies in mapping the original game into a win-lose game (game with $0-1$ payoffs) and computing equilibria on the latter. The mapping will guarantee that well supportedness of equilibria is preserved, albeit with some larger ϵ.

4.1 Player Decorrelation

Let (R, C) be a 2-player game, where the set of strategies of both players is $[n]$.

Definition 1. *A mapping* $f : [n] \to [n]$ *is a* best response mapping for the column player *iff, for every* $i \in [n]$,

$$C_{if(i)} = \max_j C_{ij}.$$

Definition 2 (Decorrelation Transformation). *The* decorrelated game (R^f, C^f) *corresponding to the best response mapping* f *is defined as follows*

$$\forall i, j \in [n]: \qquad \begin{aligned} R^f_{ij} &= R_{if(j)} \\ C^f_{ij} &= C_{if(j)} \end{aligned}$$

Note that the decorrelation transformation need not be a permutation of the columns of the original game. Some columns of the original game may very well be dropped and others duplicated. So, it is not true in general that (exact) Nash equilibria of the decorrelated game can be mapped into Nash equilibria of the original game. However, some specially structured well supported equilibria of the decorrelated game can be mapped into well supported equilibria of the original game as we explore in the following lemmas.

In the following discussion, we assume that we have fixed a best response mapping f for the column player and that the corresponding decorrelated game is (R^f, C^f). Also, if $S \subseteq [n]$, we denote by $\Delta(S)$ the set of probability distributions over the set S. Moreover, if $x \in \Delta(S)$, we denote by $supp(x) \triangleq \{i \in [n] | x(i) > 0\}$, the support of x.

Lemma 2. *In the game (R^f, C^f), for all sets $S \subseteq [n]$, the strategies of the column player in S are $\frac{|S|-1}{|S|}$-well supported against the strategy x^* of the row player, where x^* is defined in terms of the set $S' = \left\{ i \in S | C^f_{ii} = 0 \right\}$ as follows*

- *if $S' \neq \emptyset$, then x^* is uniform over the set S'*
- *if $S' = \emptyset$, then*

$$x^*(i) = \begin{cases} \frac{1}{Z} \frac{1}{C^f_{ii}}, & \text{if } i \in S \\ \\ 0, & \text{otherwise} \end{cases}$$

where $Z = \sum_{i \in S} \frac{1}{C^f_{ii}}$ is a normalizing constant.

Proof. Suppose that $S' \neq \emptyset$. By the definition of S', it follows that $\forall i \in S', j \in [n], C^f_{ij} = 0$. Therefore, for all $j \in [n]$,

$$x^{*T} C^f e_j = 0,$$

which proves the claim.

The proof of the $S' = \emptyset$ case is based on the following observations.

- $\forall j \in S$:
$$x^{*T} C^f e_j \geq x^*(j) C^f_{jj} = \frac{1}{Z} \frac{1}{C^f_{jj}} C^f_{jj} = \frac{1}{Z}$$
- $\forall j \in [n]$:
$$x^{*T} C^f e_j = \frac{1}{Z} \sum_{i \in S} \frac{1}{C^f_{ii}} C^f_{ij} \leq \frac{1}{Z} \sum_{i \in S} \frac{1}{C^f_{ii}} C^f_{ii} = \frac{|S|}{Z}$$
- $Z = \sum_{i \in S} \frac{1}{C^f_{ii}} \geq |S|$ because every entry of C^f is at most 1.

Therefore, $\forall j_1 \in S, j_2 \in [n]$,

$$x^{*T} C^f e_{j_2} - x^{*T} C^f e_{j_1} \leq \frac{|S| - 1}{Z} \leq \frac{|S| - 1}{|S|},$$

which completes the claim.

The following lemma is an immediate corollary of Lemma 2.

Lemma 3 (Player Decorrelation). *In the game (R^f, C^f), if there exists a set $S \subseteq [n]$ and a mixed strategy $y \in \Delta(S)$ for the column player such that the strategies in S are $\frac{|S|-1}{|S|}$-well supported for the row player against the distribution y, then there exists a strategy $x \in \Delta(S)$ for the row player so that the pair (x, y) is an $\frac{|S|-1}{|S|}$-well supported Nash equilibrium.*

The next lemma describes how well supported equilibria in the games (R, C) and (R^f, C^f) are related.

Lemma 4. *$\forall S \subseteq [n]$, if the pair (x^*, y^*), where x^* is defined as in the statement of Lemma 2 and y^* is the uniform distribution over S, constitutes an $\frac{|S|-1}{|S|}$-well supported Nash equilibrium for the game (R^f, C^f), then the pair of distributions (x^*, y') is an $\frac{|S|-1}{|S|}$ well supported Nash equilibrium for the game (R, C), where y' is the distribution defined as follows*

$$y'(i) = \sum_{j \in S} y^*(j) \mathcal{X}_{f(j)=i}, \forall i \in [n],$$

where $\mathcal{X}_{f(j)=i}$ is the indicator function of the condition "$f(j) = i$".

Proof. We have to verify that the pair of distributions (x^*, y') satisfies the conditions of well supportedness for the row and column player in the game (R, C).

Row Player: We show that the strategies y^* and y' of the column player give to every pure strategy of the row player the same payoff in the two games. And, since the support of the row player stays the same set S in the two games, the fact that the strategy of the row player is well supported in the game (R^f, C^f) guarantees that the strategy of the row player will be well supported in the game (R, C) as well.

$$\begin{aligned}
\forall i \in [n] : \qquad e_i^T R y' &= \sum_{k=1}^{n} R_{ik} \cdot y'(k) \\
&= \sum_{k=1}^{n} R_{ik} \cdot \sum_{j \in S} y^*(j) \mathcal{X}_{f(j)=k} \\
&= \sum_{j \in S} y^*(j) \sum_{k=1}^{n} R_{ik} \cdot \mathcal{X}_{f(j)=k} \\
&= \sum_{j \in S} y^*(j) R_{if(j)} \\
&= \sum_{j \in S} y^*(j) R^f_{ij} = e_i^T R^f y^*
\end{aligned}$$

Column Player: As in the proof of Lemma 2, the analysis proceeds by distinguishing the cases $S' \neq \emptyset$ and $S' = \emptyset$. The case $S' \neq \emptyset$ is easy, because, in this regime, it must hold that $\forall i \in S', j \in [n], C^f_{ij} = 0$ which implies that, also, $C_{ij} = 0$, $\forall i \in S', j \in [n]$. And, since, the row player plays the same distribution as in the proof of Lemma 2, we can use the arguments applied there.

So it is enough to deal with the $S' = \emptyset$ case. The support of y' is clearly the set $S'' = \{j | \exists i \in S \text{ such that } f(i) = j\}$. Moreover, observe the following:

$$\forall j \in S'' : \qquad x^{*T} C e_j = \sum_{i \in S} x^*(i) C_{ij} \geq \sum_{\substack{i \in S \text{ s.t.} \\ f(i)=j}} x^*(i) C_{if(i)} = \sum_{\substack{i \in S \text{ s.t.} \\ f(i)=j}} x^*(i) C^f_{ii} \geq \frac{1}{Z}$$

The final inequality holds because there is at least one summand, since $j \in S''$. On the other hand,

$$\forall j \notin S'' : \qquad x^{*T} C e_j = \sum_{i \in S} x^*(i) C_{ij} \leq \sum_{i \in S} x^*(i) C_{if(i)} = \sum_{i \in S} x^*(i) C^f_{ii} = \sum_{i \in S} \frac{1}{Z} \frac{1}{C^f_{ii}} C^f_{ii} = \frac{|S|}{Z}$$

Moreover, as we argued in the proof of Lemma 2, $\frac{|S|}{Z} - \frac{1}{Z} \leq \frac{|S|-1}{|S|}$. This completes the proof, since the strategy of the column player is, thus, also well supported.

4.2 Reduction to Win-Lose Games

We now describe a mapping from a general 2-player game to a win-lose game so that well supported equilibria of the win-lose game can be mapped to well supported equilibria of the original game. A bit of notation first. If A is an $n \times n$ matrix with entries in $[0, 1]$, we denote by *round*(A) the 0-1 matrix defined as follows, for all $i, j \in [n]$,

$$\mathit{round}(A)_{ij} = \begin{cases} 1, & \text{if } A_{ij} \geq \frac{1}{2} \\ 0, & \text{if } A_{ij} < \frac{1}{2} \end{cases}$$

The following lemma establishes a useful connection between well supported equilibria of the 0-1 game and those of the original game.

Lemma 5. *If (x, y) is an ϵ-well supported Nash equilibrium of the game $(round(R), round(C))$, then (x, y) is a $\frac{1+\epsilon}{2}$-well supported Nash equilibrium of the game (R, C).*

Proof. We will show that the strategy of the row player in game (R, C) is well supported; similar arguments apply to the second player. Denote $R' = round(R)$ and $C' = round(C)$.

The following claim follows easily from the rounding procedure.

Claim. $\forall i, j \in [n] : \frac{R'_{ij}}{2} \leq R_{ij} \leq \frac{1}{2} + \frac{R'_{ij}}{2}$

Therefore, it follows that, $\forall i \in [n]$,

$$\frac{1}{2} e_i^T R' y \leq e_i^T R y \leq \frac{1}{2} + \frac{1}{2} e_i^T R' y \tag{2}$$

We will use (2) to argue that the row player is well supported. Indeed, $\forall j \in supp(x)$, and $\forall\ i \in [n]$

$$e_i^T Ry - e_j^T Ry \leq \frac{1}{2} + \frac{1}{2} e_i^T R' y - \frac{1}{2} e_j^T R' y \leq \frac{1}{2} + \frac{1}{2} \cdot (e_i^T R' y - e_j^T R' y) \leq \frac{1}{2} + \frac{1}{2} \cdot \epsilon$$

where the last implication follows from the fact that (x, y) is an ϵ-well supported Nash equilibrium.

4.3 Finding Well Supported Equilibria

Lemmas 2 through 5 suggest the following algorithm, *ALG-WS*, to compute well supported Nash equilibria for a given two player game (R, C):

1. Map game (R, C) to the win-lose game $(round(R), round(C))$.
2. Map game $(round(R), round(C))$ to the game $(round(R)^f, round(C)^f)$, where f is any best response mapping for the column player.
3. Find a subset $S \subseteq [n]$ and a strategy $y \in \Delta(S)$ for the column player such that all the strategies in S are $\frac{|S|-1}{|S|}$ well supported for the row player in $(round(R)^f, round(C)^f)$ against the strategy y for the column player.
4. By a successive application of lemmas 3, 4 and 5, get an $\frac{1}{2} + \frac{1}{2}\frac{|S|-1}{|S|} = 1 - \frac{1}{2|S|}$ well supported Nash equilibrium of the original game.

The only non-trivial step of the algorithm is step 3. Let us paraphrase what this task entails:

"Given a 0-1 matrix $round(R)^f$, find a subset of the columns $S \subset [n]$ and a distribution $y \in \Delta(S)$, so that all rows in S are $\frac{|S|-1}{|S|}$ well supported against the distribution y over the columns."

It is useful to consider the $0-1$ matrix $round(R)^f$ as the adjacency matrix of a directed graph G on n vertices. We shall argue next that the task above is easy in two cases: When G has a small sycle, and when G has a small *undominated* set of vertices, that is, a set of vertices such that no other vertex has edges to all of them.

1. Suppose first that G has a cycle of length k, and let S be the vertices on the cycle. Then it is easy to see that all the k strategies in S are $\frac{k-1}{k}$-well supported for the row player against y, where y is the uniform strategy for the column player over the set S. The reason is that each strategy in S has expected payoff $\frac{1}{k}$ against y, and thus no other strategy can dominate it by more than $\frac{k-1}{k}$. This, via the above algorithm, implies a $(1-\frac{1}{2k})$-well-supported Nash equilibrium.
2. Second, suppose that there is a set S of ℓ undominated vertices. Then every strategy in S is $(1-\frac{1}{\ell})$-well supported for the row player against the uniform strategy y of the column player on S, simply because there is no row that has payoff better than $1-\frac{1}{\ell}$ against y. Again, via the algorithm, this implies that we can find a $(1-\frac{1}{2\ell})$-well-supported Nash equilibrium.

This leads us to the following graph theoretic conjecture:

Conjecture 1. There are integers k and ℓ such that every digraph either has a cycle of length at most k or an undominated set of ℓ vertices.

Now, the next result follows immediately from the preceding discussion:

Theorem 2. *If Conjecture 1 is true for some values of k and ℓ, then Algorithm* ALG-WS *returns in polynomial time (e.g. by exhaustive search) a* $\max\{1-\frac{1}{2k}, 1-\frac{1}{2\ell}\}$*-well-supported Nash equilibrium which has support of size* $\max\{k,\ell\}$*.*

The statement of the conjecture is false for $k=\ell=2$, as can be seen by a small construction (7 nodes). The statement for $k=3, \ell=2$ is already non-trivial. In fact, it was stated as a conjecture by Myers [Mye03] in relation to solving a special case of the Caccetta-Häggkvist Conjecture [CH78]. Moreover, it has recently been proved incorrect in [Cha05] via an involved construction. The case of a constant bound on k for $\ell=2$ has been left open.

While stronger forms of Conjecture 1 seem to be related to well-known and difficult graph theoretic conjectures, we believe that the conjecture itself is true, and even that it holds for some small values of k and ℓ, such as $k=\ell=3$.

What we can prove is the case of $\ell=\log n$ by showing that every digraph has a set of $\log n$ undominated vertices. This gives a $(1-\frac{1}{2\log n})$-well-supported equilibrium, which does not seem to be easily obtained via other arguments. We can also prove that the statement is true for $k=3, \ell=1$ in the special case of tournament graphs; this easily follows from the fact that every tournament is either transitive or contains a directed triangle.

5 Open Problems

Several open problems remain: Can we achieve a better than 1/2-approximate Nash equilibrium using constant sized supports, and if so, what is the limit, in view of the lower bound of 1/4-approximate equilibria [Alt94]? If constant supports do not suffice, then can we extend our techniques for larger supports? One attempt would be a natural extension of our simple algorithm from Section 3: Continue the iterations for a larger number of steps - in every step, if there is a good defection for either player, then give that pure strategy a non-zero probability and include it in the support. A second idea is to run the Lemke-Howson algorithm for some polynomial number of steps and return the best pair of strategies (note that our algorithm may be interpreted as running three steps of the Lemke Howson algorithm with an extension or truncation of the last step). Towards this, we have the following result: Recall that an imitation game is a two-player game in which the column player's matrix is the identity matrix. We can show that we can find a $\frac{1}{4}$-approximate Nash equilibrium in an imitation game by running the Lemke Howson method for 6 steps. As a final question, can we find in polynomial time a constant well-supported equilibrium, either by proving our conjecture, or independent of it?

References

[Alt94] I. Althofer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199, 1994.

[CH78] L. Caccetta and R. Haggkvist. On minimal digraphs with given girth. *Congressus Numerantium*, XXI, 1978.

[Cha05] P. Charbit. Circuits in graphs and digraphs via embeddings. *Doctoral dissertation*, University of Lyon I, 2005.

[CD05] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. *FOCS*, 2006.

[CDT06] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Computing nash equilibria:approximation and smoothed complexity. *ECCC*, 2006.

[DGP06] Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. The complexity of computing a nash equilibrium. *STOC*, 2006.

[GP06] Paul Goldberg and Christos Papadimitriou. Reducibility among equilibrium problems. *STOC*, 2006.

[LMM03] Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. *ACM Electronic Commerce*, 2003.

[Mye03] J.S. Myers. Extremal theory of graph minors and directed graphs. *Doctoral dissertation*, 2003.

[KPS06] Spyros C. Kontogiannis, Panagiota N. Panagopoulou, and Paul G. Spirakis. *Workshop on Internet and Network Economics, WINE*, 2006.

Ranking Sports Teams and the Inverse Equal Paths Problem*

Dorit S. Hochbaum

Department of Industrial Engineering and Operations Research and
Walter A. Haas School of Business, University of California, Berkeley
hochbaum@ieor.berkeley.edu

Abstract. The problem of rank aggregation has been studied in contexts varying from sports, to multi-criteria decision making, to machine learning, to academic citations, to ranking web pages, and to descriptive decision theory. Rank aggregation is the mapping of inputs that rank subsets of a set of objects into a consistent ranking that represents in some meaningful way the various inputs. In the ranking of sports competitors, or academic citations or ranking of web pages the inputs are in the form of pairwise comparisons. We present here a new paradigm using an optimization framework that addresses major shortcomings in current models of aggregate ranking. Ranking methods are often criticized for being subjective and ignoring some factors or emphasizing others. In the ranking scheme here subjective considerations can be easily incorporated while their contributions to the overall ranking are made explicit.

The *inverse equal paths* problem is introduced here, and is shown to be tightly linked to the problem of aggregate ranking "optimally". This framework is useful in making an optimization framework available and by introducing specific performance measures for the quality of the aggregate ranking as per its deviations from the input rankings provided. Presented as inverse equal paths problem we devise for the aggregate ranking problem polynomial time combinatorial algorithms for convex penalty functions of the deviations; and show the NP-hardness of some forms of nonlinear penalty functions. Interestingly, the algorithmic setup of the problem is that of a network flow problem.

We compare the equal paths scheme here to the eigenvector method, Google PageRank for ranking web sites, and the academic citation method for ranking academic papers.

Keywords: Network flow, aggregate ranking, inverse problems.

1 Introduction

We consider here an aggregate ranking scenario whereby the input to the ranking process is in the form of pairwise comparisons. This form of input is typical in the ranking of web pages, in the citation-based ranking of academic papers, and

* Research supported in part by NSF award No. DMI-0620677.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 307–318, 2006.
© Springer-Verlag Berlin Heidelberg 2006

in sports teams' ranking. In team rankings the strength or rank of a team is determined by the scores of games it has played, and possibly also the identities of the teams with which each game is played. In the ranking of web pages, the pairwise comparison is in the form of a link from one page to another, that ,as we show, is analogous to one page "losing the game" to the page it points to. In academic ranking of papers the citation index counts the number of times that a paper has been cited in other papers. Each citation is again a form of pairwise comparison between the citing page and the cited page.

The problem of ranking competitors based on an incomplete set of pairwise comparisons is well-studied in the context of football and other sports, and also in general, [11]. There are numerous ranking schemes each with its uniquely emphasized factors and each with its advantages and shortcomings. The shortcomings, particularly in sports teams ranking where passions run high, bring about the ire of some people. The generic criticism is that certain games' outcomes have not been adequately incorporated, or have had an excessive impact on the aggregate ranking.

It is important to note that the "correctness" of a ranking is subjective. A recent method for ranking of academic papers by Chen et al. [9,5] is an illustration of this subjectivity in that its improvement is based on the prevailing opinion that certain papers are more important than indicated by their academic citations count rank. More details on this issue are discussed in Section 4.3.

One aspect that all existing schemes have in common is that all pairwise comparisons are considered equal in their impact on the final outcome. This causes biases such as counting a win against a weak team equally to a win against a strong team. In other schemes it might be preferable to not play a game at all if it is against a weak team, as such a game played by a strong team can actually reduce its rank, [15]. This uniformity of consideration of each pairwise comparison is one reason for the inclusion of *human polls* in, e.g., college footfall ranking. Human judgement has the advantage that it can take into account the quality of the game played, rather than the score quantifier alone, which is often attributed to some degree to chance. These human polls in turn, are often criticized for lack of transparency, as the factors that go into human ranking are not made explicit.

The model suggested here is different from the existing ones in that it is non-uniform with respect to the inputs and it permits the explicit inclusion of subjective factors. Any form of input from knowledgeable sources can be incorporated, and each input is associated with a degree of confidence deemed appropriate for the particular source and the source's expertise in a specific pairwise evaluation. The degree of confidence assigned can in itself be subjective, but it can be made according to a specific protocol and set of rules agreed upon in advance (e.g. based on past performance of the source). This allows to differentiate the importance of different games and calibrate their impact on the final ranking. The inclusion of human polls is still possible, and it can be further refined by having the assessments by different sources assigned varying degrees of confidence.

Another feature of the model here is that it provides a *performance measure* that can be used to evaluate the quality of aggregate rankings. Most of the literature on aggregate ranking has no performance measures on the quality of the attained consistent ranking, with the exception in Kemeny's model [16] based on inputs in the form of permutations corresponding to *ordinal rankings* seeking an aggregate ranking minimizing the number of reversed permutations. This model is limited by not allowing for partial lists (each permutation must be complete); there is no differentiation between the violation of rankings of different inputs; and the model is NP-hard to solve optimally.

Our ranking model can be viewed within the *inverse problem* paradigm. In an inverse problem one is given problem parameters that must, but do not, satisfy certain necessary conditions. The goal is to modify those parameters so the necessary conditions are satisfied, subject to a penalty function on the modification, and so that the total penalty is minimum. In the context of rankings we say that the inputs are inconsistent if they are conflicting with respect to any underlying ranking. For instance, if each team loses at least once, then a top team that ranks number one has its ranking inconsistent with the game(s) it has lost. So any aggregate ranking is going to conflict with some inputs, except in rare cases where each input is precisely consistent with one underlying ranking. The necessary condition that the comparisons have to satisfy is that of *consistency*. This concept is formally described in Section 2.

As an inverse problem, aggregate ranking has the scores of the games played and any other form of judgement and pairwise comparisons as the input parameters. These invariably are inconsistent and any aggregate ranking will modify these comparisons. The problem is to come up, for each pair, with a pairwise comparison that is consistent with some underlying ranking and that deviates as little as possible from the given inputs. The penalty for deviating from the inputs is measured in terms of *penalty functions* that are monotone increasing in the size of the deviation. These penalty functions are assigned to each input separately. So the penalty for deviating from the comparison assessment of a less reliable source can take lower values than the penalty for deviating from the assessment of a high confidence source.

We introduce here, for the first time, the inverse equal paths problem, and show that for convex penalty functions the problem is solvable with polynomial flow-based algorithms. We demonstrate how the inverse equal paths problem is equivalent to aggregate ranking with pairwise comparisons inputs. We then compare our new ranking technique to leading methodologies that include the eigenvector method, the Google PageRank algorithm and the citation index method for academic papers' ranking.

2 Fundamental Concepts and Preliminaries

2.1 The Inverse Equal Paths Problem

The *inverse problem paradigm* is as follows: Given observations and parameter values that do not conform with physical or feasibility requirements, adjust the parameter values so as to satisfy the requirements. The adjustment is made so as to minimize the cost of the adjustment in the form of penalty functions. A prominent application is to find the inverse shortest paths that conform with the reading of the speed of seismic waves. There one seeks a minimum penalty for deviation from existing estimates on lengths of arcs, so as to conform to the observation that a shortest path is of a given length, or of a given sequence of nodes. Variants of this inverse shortest paths problem were studied by Burton and Toint [6], [7], by Zhang, Ma and Yang [25] and by Ahuja and Orlin [2].

The input to the *inverse equal paths* problem is a non-simple connected graph $G = (V, A)$ where for each $(i, j) \in A$ there is a set of arcs R_{ij}, so that for $r \in R_{ij}$ there is an arc of weight w^r_{ij} from i to j and an arc in the opposite direction (j, i) of weight $-w^r_{ij}$ (anti-symmetric weights). Another input is a set of penalty functions $f_{w^r_{ij}}()$ for each arc $(i, j) \in A$ and $r \in R_{ij}$. A feasible solution to the problem is a set of anti-symmetric weights w^*_{ij} for all pairs $i, j \in V$ satisfying that for any pair of nodes $s, t \in V$ all the directed paths from s to t (and from t to s) with arc weights $\mathbf{w}^*$ are of the same length. A weight vector $\mathbf{w}^*$ is optimal if among all possible weight vectors $\mathbf{w}$ it minimizes the total sum of the penalty functions $\sum_{(i,j)\in A, r\in R_{ij}} f_{w^r_{ij}}(w_{ij})$.

In Section 3 we provide a formulation and algorithms for the inverse equal paths problem, EP.

2.2 Consistency of Rankings

A fundamental notion related to ranking is that of *consistency.* Pairwise comparisons can be expressed in terms of ordinal preference, or in terms of cardinal preference. An example of ordinal preferences is permutation ranking [16]. We consider here the cardinal preferences where each pairwise ranking is accompanied by a level of *intensity.* Intensity is a quantifier expressing the extent to which one is preferred to the other. For intensity rankings there are two forms of consistency: multiplicative and additive consistency. The notion of consistency in a multiplicative sense, (used e.g. by Saaty [20,21]) is that for a triple i, j, k, $a_{ij} \cdot a_{jk} = a_{ik}$. This is equivalent to the existence of an n-dimensional vector $\mathbf{w}$ so that $a_{ij} = \frac{w_i}{w_j}$. Such set of weights, called a *priority vector*, is not unique as for any consistent set of weights $w_1, \ldots, w_n$ and a scalar c the set $cw_1, \ldots, cw_n$ is also a priority vector. So we can anchor arbitrarily $w_1 = 1$ to ensure a unique set of weights corresponding to a consistent intensity ranking. The second definition of consistency in the additive sense (e.g. [1]) has for each triplet i, j, k, $a_{ij} + a_{jk} = a_{ik}$. We call this condition *triangle equality* or TE for short. If TE is satisfied for every triplet then there is an n-dimensional vector $\mathbf{w}$ so that $a_{ij} = w_i - w_j$. Again the vector of weights is not unique as the vector $\mathbf{w} + c$ for c a constant defines the same set of differences. Here we anchor the weights uniquely by setting $w_1 = 0$. Both definitions of consistency are obviously equivalent since the logarithms of the a_{ij} that are consistent in the multiplicative sense, are consistent in the additive sense, and vice versa.

One implication of this notion is that for consistent rankings matrix a single column or a single row contains the full information on the entire matrix: Given the ith column $(a_{i1}, a_{i2}, \ldots, a_{in})$ of a consistent matrix in the multiplicative sense and setting $w_1 = 1$, one can generate all pairwise rankings as $a_{kj} = a_{ki} \cdot a_{ij} = \frac{a_{ij}}{a_{ik}}$.

We comment that for *preference rankings* where preferences are expressed only in the ordinal sense, the notion of consistency is equivalent to the *transitivity* of valid rankings. That is, if i is preferred to j and j is preferred to k then i is preferred to k. The rankings of a set of projects $V = \{1, \ldots, n\}$ can be formalized as a graph on the set of nodes V with a set of arcs – or ordered pairs – A so that the ordered pair $(i, j) \in A$ if i is preferred to j. The consistency of the preferences is equivalent to the property of *acyclicity* of the corresponding directed graph $G = (V, A)$ – a graph that does not contain a directed cycle. It is well known that an acyclic directed graph admits a *topological ordering* which is an assignment of distinct indices from $\{1, \ldots, n\}$ to the n nodes (representing the projects) so that for every arc (i, j) in the graph $i > j$. So the value of the indices of the topological ordering can serve as the underlying weights of the respective objects. It should be noted that acyclic graphs do not typically represent a *full order*, unless they contain a Hamiltonian path. So some projects may not be comparable to others in the consistent preferences, and for such pairs the difference of weights' values is not meaningful.

3 The Inverse Equal Paths Model for Aggregate Ranking

A consistent aggregate ranking of a set of objects implies a setting of the pairwise comparisons so that they satisfy the triangle equality.

Lemma 1. *If triangle equality is satisfied for all triplets in a graph with anti-symmetric weights, then all paths between every pair of nodes are of equal length.*

Therefore, the requirement of equal lengths of the paths is equivalent to the requirement of consistency.

The input to both the equal paths and the aggregate ranking problems includes a penalty function for deviating from each pairwise comparison, or arc weight. Let the penalty function for the pair (i, j) be $F_{ij}(z_{ij})$ where z_{ij} is the ranking intensity (additive) of i compared to j in the aggregate ranking. For a set of R_{ij} of pairwise arcs comparing one pair (i, j), the penalty deviation function is $F_{ij}(z_{ij}) = \sum_{r \in R_{ij}} f^r_{ij}(z_{ij} - w^r_{ij})$. This function typically takes the value 0 for an argument of 0 (0 deviation). It is allowed to be non-symmetric for positive and negative arguments.

The next lemma establishes that all paths are equal, or the preferences are consistent, if and only if there is an underlying set of weights associated with the nodes, *node potentials*, which are the priority weights.

Lemma 2. *If* $\mathbf{z}$ *is a set of weights for which graph* G *has all equal paths, then there exists a set of values* x_j*, for all* $j \in V$*, such that* $x_i - x_j = z_{ij}$*.*

Including in the formulation a set of variables, x_j, for the node potentials, or priority weights, is redundant, but has the advantage that the properties of the problem become transparent. We use here the anchoring of $x_1 = 0$. The inverse equal paths problem EP is then,

$$\begin{array}{lll} \text{(EP)} \quad \text{Min} & \sum_{i<j} F_{ij}(z_{ij}) & \\ \text{subject to} & x_i - x_j = z_{ij} & \text{for } i < j \\ & x_1 = 0 & \\ & \ell_j \le x_j \le u_j & j = 1, \ldots, n. \end{array}$$

It is easy to see that for $C = \max_{r,(i,j)} w^r_{ij}$, $-nC \le x_j \le nC$. We thus let $\ell_j = -nC$ and $u_j = nC$ for all $j = 1, \ldots, n$. In the aggregate ranking problem it would be reasonable to require a set of weights with some finite resolution. A set of integer weights in the interval $[-n, n]$ is sufficient to guarantee that there are enough distinct ranks to assign to each node (or team). Therefore, we would replace the lower and upper bound constraints on $\mathbf{x}$ by,

$$-n \le x_j \le n \qquad \text{integer, for all } j \in V.$$

In case of rank ties, one might want to increase the resolution of the weights. The proximity algorithm of Hochbaum and Shanthikumar guarantees that an optimal solution in integers for one resolution level is close enough to an optimal solution on a finer grid, [13]. The EP model has the *fixed point* property. That is, if the input intensity preferences are consistent with some underlying ranking, then the optimal solution will be that underlying ranking.

3.1 Algorithms for the Inverse Equal Paths Problem

Observe that the constraint matrix of EP is totally unimodular. Therefore, when the objective function is convex, it follows immediately that the problem is solvable in polynomial time, [13]. Furthermore, the convex EP is a special case of convex dual of minimum cost network flow studied in [3].

We summarize below the complexity and algorithms for solving EP. Here $U = \max_j\{u_j - \ell_j\}$, and $T(n, m)$ is the running time required to solve the minimum s,t-cut problem on a graph with n nodes and m arcs.

1. For $F()$ convex functions the problem EP is solvable in polynomial time. An algorithm that runs in $\log U$ calls to a minimum cut procedure with complexity $O(\log U \cdot T(n^2, mn))$ is reported in [4]. Another, more efficient, algorithm for this problem runs in $O(mn \log n \log nU)$, [3]. Both these algorithms have been devised for the more general problem of the convex dual of minimum cost network flow (DMNCF).
2. For $F_{ij}(z_{ij}) = a^+_{ij} \max\{z_{ij}, 0\} + a^-_{ij} \max\{-z_{ij}, 0\}$ (that is, $F_{ij}()$ are linear for positive deviation and for negative deviation), the algorithm reported in [14] has complexity of $O(T(n, m) + n \log U)$, which is best possible.
3. For $F()$ arbitrary functions the problem is NP-hard – it can be shown to be only harder than the multi-way cut problem which is known to be NP-hard. This case is known more commonly as the metric labeling problem and the functions $F()$ are usually δ functions equal to 0 if the argument is 0 and a positive constant otherwise. For these problems there is a large body of research on approximation algorithms, e.g. [17].

Since for EP $U = O(n)$, the run times of the polynomial algorithms for the convex case are all strongly polynomial.

4 Leading Ranking Methods and Score-Based Algorithms

The simplest ranking algorithm is based on sorting according to total weight, or citation count, or in-degree of a web page counting the number of pages pointing to it. The total weight sorting algorithm is used to rank sports teams by counting the number of wins, losses (and draws). This is the method used for example to determine division winners in baseball. (To provide an incentive for goal-richer soccer games higher weights are assigned to stronger wins.) It is known that giving a weight that is inversely proportional to the out-degree (number of games won) of a node creates biases where it is possible that a team wins a game against a weaker team and this win actually decreases the team's rank, [15].

College football teams are ranked according to a weighted composite score called the BCS ranking that combines a number of algorithms with polls of expert human judges. The 2004 version included three components - the AP sportswriters' poll, the USA Today/ESPN coaches poll, and six computer rankings algorithms - all weighing equally. There is a great deal of criticism of the inclusion of human polls for their lack of transparency. We quote from http://spirit.tau.ac.il/public/gandal/bcs.htm

> Despite the criticism of computer rankings, they are the only ones that can be transparent and based on measurable criteria, which is to say, impartial. The computer ratings can also be improved. The computer ratings used by the BCS should be consistent (this has a formal mathematical meaning) with an endogenous strength of schedule.
>
> Additionally, all computer rankings should be required to publish their methodology. This insures transparency and will enable experts to evaluate them. For example, one could evaluate the rankings by using them to predict bowl game outcomes. This could create competition among the computer rankings themselves. Currently six computer ranking systems are used by the BCS. But there are many other ranking systems out there. As of December 4, Kenneth Massey (who produces a computer ranking for the BCS) lists 100 rankings on his comparison page: http://www.masseyratings.com/cf/compare.htm

All the computer rankings in BCS translate the scores of the games into relative strength of each of the competing teams. One reason for including human polls is that the scores alone do not fully reflect the strength of each team. For instance, the score does not capture whether a game is played in poor weather conditions, or a major player is sick on the day of the game, or a soccer team plays with fewer than 11 players, In those cases the significance of the score may need modifying. However, there no previously existing ranking system allowed to incorporate such contingencies.

4.1 The Principal Eigenvector Technique

The principal eigenvector technique has been known to apply to ranking since the 1950s. This method is reviewed e.g. in a study addressing the rankings of football teams by Keener, [15]. Consider intensity rankings that quantify by how much team i is stronger than team j by a positive number a_{ij} – a multiplicative intensity preference. (There is a great deal of research on how to determine the values of a_{ij} as a function of the score of a game, and Keener's study proposes one mapping between the score of the game and the value of a_{ij}.) Let n_i be the number of games played by team i. Then, r_i, the ultimate ranking of team i, is reasonably presumed to be proportional to the calibrated rank,

$$\frac{1}{n_i}\sum_{j=1}^{n} a_{ij} r_j .$$

Thus $r_i = \frac{1}{\lambda}\sum_{j=1}^{n} \frac{a_{ij}}{n_i} r_j$, or $A\mathbf{r} = \lambda\mathbf{r}$ for $A = (\frac{a_{ij}}{n_i})$. The solution to this system of equations – the principal eigenvector – plays an important role in the Analytic Hierarchical Process, [20], and in the Google PageRank.

Perron-Frobenius theorem states that for a nonnegative nontrivial matrix A there exists a nonnegative eigenvector $\mathbf{r}$ corresponding to a unique eigenvalue λ. If A is *irreducible* then $\mathbf{r}$ is strictly positive, unique and simple and λ is the largest eigenvalue.

The notion of irreducibility has an algebraic definition. We prefer to discuss it as a graph property: Firstly the concept of *deduced ranking* is important. One can deduce the relative ranking of a pair of teams indirectly from the outcomes of a sequence games played. The relative ranking of teams i and j can be deduced, even if the two teams did not play directly, if there is a sequence of games $[i, i_1]$, $[i_1, i_2], \ldots, [i_k, j]$ for $k \geq 1$. The ranking of a direct pairwise comparison can be viewed as such sequence for $k = 1$. Now the concept of irreducibility is equivalent to having all pairs of teams comparable by deduced ranking. In graph terms this means that there is a path between each pair of nodes – namely, the graph is connected. (Notice that although the graph is directed there are two symmetric arcs between pairs that are directly linked, so there is a directed path if and only if there is an undirected path.)

Some properties of the principal eigenvector method are:

1. Unlike the weight sorting algorithm, the eigenvector method takes into consideration not only the count of how many times one object is stronger than others, but also which objects it is compared to. So winning against a strong team counts more than winning against a weak one.
2. "Missing games" still must correspond to entries in the matrix, as the matrix must be full. The standard approach is to include such games as a draw. This however tends to skew the overall ranking.
3. All games contribute uniformly to the aggregate ranking and no subjective evaluation of a score of a game can be included. This is also a feature in the total weight sorting algorithm used for web page ranking or for academic

citation ranking both of which do not differentiate between citations of between pointers. So a negative citation stating that a result in a related paper is wrong, counts the same as a citation referring to a paper as seminal. On web pages there are sometimes pointers that companies are buying in order to increase their web page rank, and these pointers are often unrelated to the content of the web page. The principal eigenvector method as well as other existing models do not discriminate however between citations as per their quality and significance.

4. If there are multiple games between teams, it is not clear how to measure the aggregate effect of the games that have different, and often contradictory outcomes. In a simple example, if one team wins against the other in one game, and loses in a second game, then the often used average counts the same as if the two teams played a game resulting in a draw, or not having played at all.

Suppose the matrix of comparisons is consistent and the vector of weights is $\mathbf{w} = (w_i)_{i=1}^n$. Then $a_{ij} = \frac{w_i}{w_j}$. Summing up over all j, we obtain, $\sum_{j=1}^n a_{ij} w_j = n w_i$. Therefore, the vector of weights $\mathbf{w}$ satisfies, $\mathbf{Aw} = n\mathbf{w}$, and is thus an eigenvector specifying the weights assigned to each project or each criterion under the multiplicative model. In that the principal eigenvector satisfies the fixed point property. If the matrix is not consistent then the eigenvector approximates the preference weights. One measure of approximation for an asymmetric inconsistent matrix was defined by Saaty [21] is the *consistency index* C.I., $C.I. = \frac{\lambda_{max} - n}{n-1}$. where λ_{max} is the maximum eigenvalue of the matrix. A matrix is said to be consistent if and only if C.I is zero.

In terms of complexity, the computation of the principal eigenvector $\mathbf{w}^*$ is not practical for large values of n. It is common to calculate it instead with the power method, [23]: For a given initial assessment of ranks $\mathbf{w}_0$ (typically, assuming all ranks are equal), this is a recursive procedure based on,

$$\lim_{k \to \infty} \frac{A^k \mathbf{w}_0}{|A^k \mathbf{w}_0|} = \mathbf{w}^*. \tag{1}$$

4.2 Finding "Close" Consistent Rankings

Several approaches other than the eigenvector method have been proposed in the literature to generate a consistent matrix that is in some sense "close" to the given matrix. Most of these are based on minimizing some measure of distance of the generated consistent matrix from the given matrix. Regression-based approaches have been proposed (see [18] for a review and formal treatment of these methods) that assume the a_{ij}'s to be random variables with known distribution centered around a consistent comparison matrix. Least-squares and logarithmic least squares regression are the most popular of these techniques, and Saaty and Vargas [22] give a comparison of these methods to the eigenvector method. Techniques based on linear programming (Chandran et. al [8]), nonlinear programming (Wang et. al [24]), and goal programming have also been proposed.

4.3 The Google PageRank Algorithm

The Google PageRank algorithm is a finite approximation of the limit (1) using a small number of iterations. The following recursive formula that is used for Google PageRank can be shown to approximate in the limit the principal eigenvector of the respective matrix if $d = 0$:

$$G_i = (1 - d) \sum_{(j,i) \in A} \frac{G_j}{k_j} + \frac{d}{N}$$

where N is the number of objects in the universe, G_i is the google number or *strength* of object i, k_j is the out-degree of node j and d is a parameter.

The ranking of academic papers based on citation count has raised some interest and criticism recently, [9,5]. Citation-ranking of academic papers are determined by the citation count of a paper. Setting a citation of article i to j as an arc (i, j) is a graph $G = (V, A)$ with a node corresponding to each academic paper, this is equivalent to ranking each paper by its in-degree. Chen et al. [9] and Buchanan [5] point out that the traditional citation count brings about results that are contradictory to perceived importance of certain papers. In their study Chen et al. give some examples. One is a 1929 paper by Slater that ranks 1853rd in terms of citation count although there is a universal agreement among physicists that 'Slater determinant' introduced in that paper is a fundamental concept that is considered classic and therefore the citation count rank undervalues Slater's paper.

Chen et al. used instead the "Google PageRank Algorithm" noting that the ranking model of web pages is analogous to the academic citations model where pointing to a web page is equivalent to a citation. Chen et al. [9] computed the rank of Slater's paper with Google PageRank and showed it turns out 10th. This, and the improved rank of other 'classic' papers served as evidence that Google rank is a better measure of impact than the traditional citation count.

5 Using EP for Sports Ranking, Web Page Ranking and Academic Citations

Both applications of academic citations and web page rankings are unique among general aggregate ranking problems in that the "evaluators" are also the objects being evaluated. In sports team ranking the evaluators are the games and their outcomes provide a comparison of the relative strengths of the pairs of teams that played each game. In spite of this apparent difference, the models are analogous as citing a paper j by i is analogous to j winning a game against i. (This makes an unpleasant corollary that for a paper to retain a high citation count it should cite as few papers as possible. If using the PageRank for ranking of papers it is desirable to cite only recognized "strong" papers.)

The EP model can be used in several ways. Every citation or pointer from i to j are considered to be a pairwise evaluation of the relative strength of i and j in which j is stronger than i. The amount of this extra strength can be calibrated by the type of pointer or citation. The confidence level (or the steepness of the penalty function) can be determined by the quality of the journal in which the citing paper appears, or by the type of citation (positive, negative or neutral.)

The EP model shares the advantage of the principal eigenvector (and thus to some extent the Google pageRank that approximates it) in that it weighs more heavily comparative strength against strong objects than strength against weak ones. It does add however the flexibility of incorporating additional sources of information that are currently excluded from ranking schemes. Furthermore, it can use as a starting point the current ranking, regardless of the method that led to it, and adjust it based on additional pairwise comparisons. People and organizations can individualize the ranking using their own sources of information, and to the degree that they trust those sources.

One important issue is the evolution of rankings over time, as additional links and comparisons become available, [10]. The goal is not to recreate the ranking every time that new information becomes available. The total weight algorithm is obviously the simplest to adjust to new links - simply add to the count and shift the modified weight object in the sorted list. The principal eigenvector, Google pageRank and equal paths are however global in nature. Chien et al. [10] showed that for Google pageRank it is sufficient to apply the recursive formula within a limited "radius" from the modified link. Here another advantage of the EP model is that the relative rank of any selected subsets of objects can be retained unchanged, by fixing a reference point in the subset and all the relative rankings are then fixed with respect to that single weight. The position of the entire subset in the ranking may be shifted with comparisons that include objects in the subset, but the relative ranking remains the same. This makes it computationally easier to evolve the ranking weights as new comparisons become available. The same approach can be used on large data bases where within certain clusters the relative rankings are required to be unmodified. It remains to study rigorously the size of the neighborhood on which the impact of an added link is significant.

References

1. Ali I., Cook W.D. and Kress M.: Ordinal ranking and intensity of preference: a linear programming approach. Management Science **32** (1986) 1642–1647.
2. Ahuja R. K., and Orlin J. B.: Inverse optimization. Operations Research **49** 771-783, 2001.
3. Ahuja R. K., Hochbaum D. S. and Orlin J. B.: Solving the convex cost integer dual of minimum cost network flow problem. Management Science **49** (2003) 950–964.
4. Ahuja R. K., Hochbaum D. S. and Orlin J. B.: A cut based algorithm for the nonlinear dual of the minimum cost network flow problem. Algorithmica **39** (2004) 189–208.
5. Buchanan M.: Top rank. Nature Physics **2** (2006) 361.
6. Burton D. and Toint Ph. L.: On an instance of the inverse shortest paths problem. Mathematical Programming **53** (1992) 45–61.
7. Burton D. and Toint Ph. L.: On the use of an inverse shortest paths algorithm for recovering linearly correlated costs. Mathematical Programming **63** (1994) 1–22.

8. Chandran B., Golden B. and Wasil E.: Linear programming models for estimating weights in the analytic hierarchy process. Computers and Operations Research **32** (2005) 2213–2234.
9. Chen P., Xie H., Maslov S. and Redner S.: Finding scientific gems with Google. arXuv:physics/0604130 **1:18** (2006).
10. Chien S., Dwork C., Kumar R., Simon D.R. and Sivakumar D.: Link Evolution: Analysis and Algorithms. Internet Mathematics **1** (2004) 277–304.
11. David H. A.: The Method of Paired Comparisons. Charles Griffin and Company Ltd. and Oxford University Press, London, 2nd ed. (1988).
12. Hochbaum D. S. and Levin A.: Methodologies and algorithms for group rankings decision. Management Science (to appear).
13. Hochbaum D. S. and Shanthikumar J. G.: Convex Separable Optimization is not Much Harder Than Linear Optimization. Journal of the ACM **37** (1990) 843–862.
14. Hochbaum D. S.: An efficient algorithm for image segmentation, Markov Random Fields and related problems. Journal of the ACM **48** (2001) 686–701.
15. Keener J. P.: The Perron-Frobenius theorem and the ranking of football teams. SIAM Review **35** (1993) 80-93.
16. Kemeny J. G. and Snell J.L.: Preference ranking: An axiomatic approach. In Mathematical models in the social sciences, Boston, Ginn, (1962) 9–23.
17. Kleinberg J. and Tardos J.: Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (1999) 14–23.
18. Laininen P. and Hämääinen R.P.: Analyzing AHP-matrices by regression. European Journal of Operational Research **148** (2003) 514–524.
19. Park J. and Newman M.E.J.: A network-based ranking system for US college football. arXiv:physics/0505169 **4:31** Oct (2005).
20. Saaty T.: A scaling method for priorities in hierarchical structures. Journal of Math. Psychology **15** (1977) 234–281.
21. Saaty T.: The Analytic Hierarchy Process. McGraw-Hill, New York, (1980).
22. Saaty T. and Vargas L. G.: Comparison of eigenvalue, logarithmic least squares and least squares methods in estimating ratios. Journal of Mathematical Modeling **5** (1984) 309–324.
23. Vargas R. S.: Matrix Iterative Analysis. Prentice Hall, Englewood Cliffs, NJ, (1962).
24. Wang Y.M., Yang J.B. and Xu D.L.: Mathematical Programming Methods for Generating weights from Interval Comparison Matrices. Internet http://www.sm.umist.ac.uk/wp/abstract/wp0205.htm, retrieved July (2004).
25. J. Zhang, Z. Ma and C. Yang. A column generation method for inverse shortest paths problems. J. ZOR Mathematical Methods of Operations Research **41** (1995) 347–358.

Price of Anarchy for Polynomial Wardrop Games*

Dominic Dumrauf and Martin Gairing

Faculty of Computer Science, Electrical Engineering and Mathematics,
University of Paderborn, Fürstenallee 11, 33102 Paderborn, Germany
{masa, gairing}@uni-paderborn.de

Abstract. In this work, we consider *Wardrop games* where traffic has to be routed through a shared *network*. Traffic is allowed to be split into arbitrary pieces and can be modeled as network flow. For each edge in the network there is a latency function that specifies the time needed to traverse the edge given its congestion. In a *Wardrop equilibrium*, all used paths between a given source-destination pair have equal and minimal latency.

In this paper, we allow for *polynomial latency functions* with an upper bound d and a lower bound s on the degree of all monomials that appear in the polynomials. For this environment, we prove upper and lower bounds on the *price of anarchy.*

1 Introduction

Motivation and Framework. The *price of anarchy,* also known as *coordination ratio,* has been defined in the seminal work by Koutsoupias and Papadimitriou [14] as a measure of the extent to which competition approximates cooperation. In general, the price of anarchy is the worst-case ratio between the value of a social objective function, usually coined as *social cost,* in some equilibrium state of a system, and that of some social optimum. Usually, the equilibrium state has been taken to be that of a *Nash equilibrium* [16] – a state in which no *user* wishes to unilaterally leave its own strategy in order to improve the value of its private objective function, also known as *individual cost.* So, the price of anarchy represents a rendezvous of Nash equilibrium, a concept fundamental to Game Theory, with *approximation,* an ubiquitous concept in Theoretical Computer Science today (see, e.g., [22]).

The *Wardrop model* has already been studied in the context of road traffic systems by Pigou [17] in the 1920's and later by Wardrop [23], and by Beckmann, McGuire and Winsten [3] in the 1950's. For a survey of the early work on this model see [4]. In the Wardrop model, traffic has to be sent through a shared network and traffic is allowed to be split into arbitrary pieces. In this

* This work has been partially supported by the DFG-SFB 376 and by the European Union within the Integrated Project IST-15964 "Algorithmic Principles for Building Efficient Overlay Computers" (AEOLUS).

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 319–330, 2006.
© Springer-Verlag Berlin Heidelberg 2006

environment, unregulated traffic is modeled as network flow. Wardrop [23] introduced the concept of *Wardrop equilibrium* to describe user behavior in this kind of traffic networks. Given an arbitrary network with edge latency functions, Wardrop equilibria have been classified as flows with all flow paths used between a given source-destination pair having equal latency. A Wardrop equilibrium can be interpreted as a Nash equilibrium in a game with infinitely many users, each carrying an infinitesimal amount of traffic from a source to a destination.

Inspired by the arisen interest in the price of anarchy, Roughgarden and Tardos [21] re-investigated the Wardrop model and used the *total latency* as their social objective function. The total latency is a measure for the total travel time. In this context, the exact value for the price of anarchy was shown for linear latency functions by Roughgarden and Tardos [21] and for arbitrary polynomial latency functions with nonnegative coefficients and maximum degree d by Roughgarden [19]. In his book [20, Chapter 3], Roughgarden gives the following rule of thumb:

The price of anarchy is small unless cost functions are extremely steep.

In this work, we examine this rule of thumb closer by re-considering the price of anarchy for polynomial latency functions of maximum degree d. However, in contrast to the latency functions considered by Roughgarden [19], our latency functions have also a minimum degree of s. For large d these latency functions are *extremely* steep, however, we show that in many cases the price of anarchy remains small.

Related Work. The price of anarchy was introduced by Koutsoupias and Papadimitriou [14] and received a lot of attention in various routing games (see e.g. [1,2,5,6,7,8,9,10,12,13,15,19,21]).

Early work on the Wardrop model has been done in the context of road traffic systems [3,4,17,23]. Beckmann et al. [3] showed that a Wardrop equilibrium always exists and that it is essentially unique. These results were based on the observation that a Wardrop equilibrium is a solution to a related convex program.

For the Wardrop model, with social cost as total latency, Roughgarden and Tardos [21] showed that the price of anarchy is exactly $\frac{4}{3}$ in case of linear latency functions. For the case of polynomial latency functions of maximum degree d, Roughgarden [19] showed that the price of anarchy is $\frac{(d+1)\sqrt[d]{d+1}}{(d+1)\sqrt[d]{d+1}-d}$. Interestingly, in both cases, the price of anarchy is independent of the network topology, as it is achieved on the simple network of two parallel links [19,21]. Correa et al. [6] improved the bounds from [19] on the price of anarchy for the special case of polynomial latency functions without constant term and for $d \leq 4$. The price of anarchy was also studied for latency functions that arise as delay functions of $\mathsf{M/M/1}$ queues [19]. For arbitrary nondecreasing latency functions Roughgarden and Tardos [21] showed that the total latency in a Wardrop equilibrium is upper bounded by the optimum total latency for the instance where all traffic demands are doubled.

Related to Wardrop games are (weighted) congestion games as introduced by Rosenthal [18]. In a congestion game, there is a set of resources and players can choose as their strategy a set of resources from a given set of subsets of resources. Awerbuch et al. [2] and Christodoulou and Koutsoupias [5] were the first to study the price of anarchy for congestion games. They showed asymptotic tight bounds on the price of anarchy for congestion games with polynomial latency functions in case of unweighted [2,5] and in case of weighted player demands [2]. With a more careful analysis, Aland et al. [1] were able to derive the *exact* value for the price of anarchy in both cases. For a survey on weighted congestion games, we refer to [11].

Contribution. In this paper, we study the price of anarchy for Wardrop games with polynomial latency functions in more detail. In particular, we consider polynomials that consist of monomials of maximum degree d and minimum degree s. All our latency functions have nonnegative coefficients. We will call such polynomials (d,s)*-polynomials*.

As our first result, we show that for general (d,s)-polynomials, the price of anarchy (PoA(d,s)) is upper bounded by

$$\mathsf{PoA}(d,s) \leq \frac{\left(\frac{d}{d+1}\right)^d}{(d+1)\cdot\left(\frac{\left(\frac{d}{d+1}\right)^d(s+1)}{\left(\frac{s}{s+1}\right)^s(d+1)}\right)^{\frac{d}{d-s}}\cdot\left(1-\left(\frac{\left(\frac{d}{d+1}\right)^d(s+1)}{\left(\frac{s}{s+1}\right)^s(d+1)}\right)^{\frac{1}{d-s}}\right)}.$$

To achieve this result, we adopt a technique that was already used in [1] and that is again based on a technique from [5]. The core of our analysis is to determine parameters c_1 and c_2, such that

$$y \cdot f(z) \leq c_1 \cdot z \cdot f(z) + c_2 \cdot y \cdot f(y)$$

for all (d,s)-polynomials f and for all reals $y, z \geq 0$. Table 1 shows numerical values for the upper bound for all (d,s)-polynomials with $d \leq 10$. The values for

Table 1. Example values for our upper bound on the price of anarchy

$s \backslash d$	1	2	3	4	5	6	7	8	9	10
0	$\frac{4}{3}$	1.62575	1.89563	2.15050	2.39438	2.62971	2.85814	3.08084	3.29856	3.51206
1		1.03551	1.09820	1.16756	1.23859	1.30962	1.38002	1.44954	1.51811	1.58575
2			1.01466	1.04498	1.08174	1.12147	1.16262	1.20439	1.24638	1.28834
3				1.00805	1.02614	1.04938	1.07547	1.10324	1.13199	1.16131
4					1.00510	1.01717	1.03329	1.05192	1.07217	1.09348
5						1.00352	1.01215	1.02404	1.03808	1.05358
6							1.00257	1.00907	1.01821	1.02918
7								1.00197	1.00703	1.01428
8									1.00155	1.00561
9										1.00125

$s = 0$ have already been shown by Roughgarden [19]. Observe that the values for $s = 1$ and $d \leq 4$ improve the upper and match the lower bounds from [6].

We then prove monotonicity results on our upper bound. In particular, we show that the upper bound on $\mathsf{PoA}(d, s)$ is monotone increasing in d and decreasing in s. Furthermore, we show that if $s = \frac{d}{a}$ is a constant fraction of d, then the upper bound on $\mathsf{PoA}\left(d, \frac{d}{a}\right)$ is still monotone increasing in d. Equipped with these results, we apply the limit for $d \to \infty$ to prove that for any $a > 1$, $\mathsf{PoA}\left(d, \left\lceil \frac{d}{a} \right\rceil\right)$ is upper bounded by a constant. More precisely, we show that

$$\mathsf{PoA}\left(d, \left\lceil \frac{d}{a} \right\rceil\right) \leq \frac{a^{\frac{1}{a-1}} \cdot (a-1)}{\mathsf{e} \cdot \ln(a)}.$$

For instance, this gives upper bounds of 1.0614756 and 1.159983 for $a = 2$ and $a = 3$, respectively.

We close our paper with a discussion on lower bounds on the price of anarchy for Wardrop games with (d, s)*-polynomials*. Here, we use the very simple network of two parallel links. So far, we could not show that our general upper bound yields the exact value for the price of anarchy; however, numerical analysis for all (d, s)-polynomials with $d \leq 30$ gives a strong indication that this is the case.

For sufficiently large d and for the cases $s = \frac{d}{2}$ and $s = \frac{d}{3}$, we give almost matching lower bounds on $\mathsf{PoA}(d, s)$.

Roadmap. The rest of this paper is organized as follows. Section 2 introduces the Wardrop model. Section 3.1 presents the upper bound on the price of anarchy, whereas Section 3.2 discusses lower bounds. We conclude in Section 4 with a summary of our results and some open problems. Due to lack of space, we omit some proofs. They can be found in the appendix.

2 Notation

For all $k \in \mathbb{N}$ denote $[k] = \{1, \ldots, k\}$.

Routing with Splittable Traffic. A *Wardrop game* is a tuple $\Gamma = (n, G, \mathsf{w}, \mathcal{P}, \mathbf{f})$. Here, n is the number of *players* and $G = (V, E)$ is an undirected *(multi)graph*. The vector $\mathsf{w} = (w_1, \ldots, w_n)$ defines for every player $i \in [n]$ its *traffic* $w_i \in \mathbb{R}^+$. For each player $i \in [n]$ the set $\mathcal{P}_i \subset 2^E$ consists of all possible routing paths in $G = (V, E)$ from some node $s_i \in V$ to some other node $t_i \in V$. Denote $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$. Denote by $\mathsf{f} = \{f_e \mid e \in E\}$ the set of differentiable, monotone increasing and nonnegative *edge latency functions*.

In this paper, we allow for polynomial latency functions with nonnegative coefficients, where monomials of degree less than s are missing; that is, latency functions are of the form $f_e(x) = \sum_{i=s}^{d} a_{ie} x^i$ with $a_{ie} \geq 0$ for all integers $s \leq i \leq d$ and all edges $e \in E$. We will call such latency functions (d, s)*-polynomials*.

Strategies and Strategy Profiles. A player $i \in [n]$ can split its traffic w_i over the paths in $\mathcal{P}_i$. A *strategy* for player $i \in [n]$ is a tuple $\mathsf{x}_i = (x_{iP_i})_{P_i \in \mathcal{P}_i}$ with

$\sum_{P_i \in \mathcal{P}_i} x_{iP_i} = w_i$ and $x_{iP_i} \geq 0$ for all $P_i \in \mathcal{P}_i$. Denote by $\mathcal{X}_i = \{\mathsf{x}_i \mid \mathsf{x}_i$ is a strategy for player $i\}$ the set of all strategies for player i. A *strategy profile* $\mathsf{x} = (\mathsf{x}_1, \ldots, \mathsf{x}_n)$ is an n-tuple of strategies for the players. Define $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ as the set of all possible strategy profiles.

Wardrop Equilibria. For a strategy profile x, the load $l_e(\mathsf{x})$ on an edge $e \in E$ is given by $l_e(\mathsf{x}) = \sum_{i \in [n]} \sum_{P_i \in \mathcal{P}_i, P_i \ni e} x_{iP_i}$. A strategy profile x is a *Wardrop equilibrium*, if for every player $i \in [n]$, and every $P_i, P_i' \in \mathcal{P}_i$ with $x_{iP_i} > 0$ it holds that

$$\sum_{e \in P_i} f_e(l_e(\mathsf{x})) \leq \sum_{e \in P_i'} f_e(l_e(\mathsf{x})).$$

Observe that in a Wardrop equilibrium all flow paths of a player have equal latency. We can regard each player $i \in [n]$ as a service provider who has many clients each handling a negligible small amount of traffic. In a Wardrop equilibrium, each service provider satisfies all his clients because none of them can improve its experienced latency.

Social Cost and Price of Anarchy. For a strategy profile x, define the *social cost* $\mathsf{SC}(\mathsf{x})$ as the *total latency*; thus,

$$\mathsf{SC}(\mathsf{x}) = \sum_{i \in [n]} \sum_{P_i \in \mathcal{P}_i} x_{iP_i} \sum_{e \in P_i} f_e(l_e(\mathsf{x})).$$

This social cost is motivated by the interpretation as a game with infinitely many players with negligible demand and models the sum of the players latencies. The *optimum* associated with a game is defined by $\mathsf{OPT} = \min_{\mathsf{x} \in \mathcal{X}} \mathsf{SC}(\mathsf{x})$. The *price of anarchy*, also called *coordination ratio* and denoted PoA, is the maximum value, over all instances and Wardrop equilibria x, of the ratio $\frac{\mathsf{SC}(\mathsf{x})}{\mathsf{OPT}}$. For the class of Wardrop games, where all latency functions are (d,s)-polynomials, denote by $\mathsf{PoA}(d,s)$ the price of anarchy with respect to d and s.

3 Price of Anarchy

3.1 Upper Bound

Before proving a general upper bound on the price of anarchy for Wardrop games with (d,s)-polynomial latency functions, we have to prove the following technical lemma:

Lemma 1. *Let $s, d \in \mathbb{N}$ with $s \leq d$. Choose $c_1, c_2 \in \mathbb{R}_{\geq 0}$ such that*

$$y \cdot z^s \leq c_1 \cdot z^{s+1} + c_2 \cdot y^{s+1} \qquad \forall y, z \in \mathbb{R}_{>0} \tag{1}$$

$$\text{and} \qquad y \cdot z^d \leq c_1 \cdot z^{d+1} + c_2 \cdot y^{d+1} \qquad \forall y, z \in \mathbb{R}_{>0}. \tag{2}$$

Then, it follows that

$$y \cdot z^i \leq c_1 \cdot z^{i+1} + c_2 \cdot y^{i+1} \qquad \forall i \in \mathbb{N} : s \leq i \leq d \ \forall y, z \in \mathbb{R}_{>0}.$$

Proof. Since

$$y \cdot z^i \leq c_1 \cdot z^{i+1} + c_2 \cdot y^{i+1} \qquad \forall i \in \mathbb{N} : s \leq i \leq d \; \forall y, z \in \mathbb{R}_{>0}$$

is equivalent to

$$\left(\frac{z}{y}\right)^i \leq c_1 \cdot \left(\frac{z}{y}\right)^{i+1} + c_2 \qquad \forall i \in \mathbb{N} : s \leq i \leq d \; \forall y, z \in \mathbb{R}_{>0}$$

it suffices to show that

$$z^i \leq c_1 \cdot z^{i+1} + c_2 \qquad \forall i \in \mathbb{N} : s \leq i \leq d \; \forall z \in \mathbb{R}_{>0}.$$

This follows by replacing $\frac{z}{y} \in \mathbb{R}_{>0}$ with a new $z \in \mathbb{R}_{>0}$. Furthermore, it follows from (1) and (2) that

$$\begin{aligned} & z^s \leq c_1 \cdot z^{s+1} + c_2 \qquad \forall z \in \mathbb{R}_{>0} \\ \text{and} \quad & z^d \leq c_1 \cdot z^{d+1} + c_2 \qquad \forall z \in \mathbb{R}_{>0}. \end{aligned}$$

Fix an arbitrary $i \in \mathbb{N}$ with $s \leq i \leq d$. We proceed by case study dependent on $z \in \mathbb{R}_{>0}$.
First assume that $z \leq 1$. Let $i = s + j$, then $0 \leq j \leq d - s$. We get

$$z^i = z^{s+j} \leq z^j \left(c_1 \cdot z^{s+1} + c_2\right) = c_1 z^{i+1} + z^j \cdot c_2 \leq c_1 z^{i+1} + c_2,$$

since $z \leq 1$.
Now assume that $z \geq 1$. Let $i = d - j$, then $0 \leq j \leq d - s$. We get

$$z^i = z^{d-j} \leq \frac{1}{z^j} \left(c_1 \cdot z^{d+1} + c_2\right) = c_1 z^{i+1} + \frac{1}{z^j} \cdot c_2 \leq c_1 z^{i+1} + c_2,$$

since $z \geq 1$.
In any case $z^i \leq c_1 \cdot z^{i+1} + c_2$. This completes the proof of the lemma. □

We are now ready to prove our general upper bound on the price of anarchy.

Theorem 1. *For Wardrop games with (d, s)-polynomial latency functions, we have*

$$\mathsf{PoA}(d, s) \leq \frac{\left(\frac{d}{d+1}\right)^d}{(d+1) \cdot \left(\frac{\left(\frac{d}{d+1}\right)^d (s+1)}{\left(\frac{s}{s+1}\right)^s (d+1)}\right)^{\frac{d}{d-s}} \cdot \left(1 - \left(\frac{\left(\frac{d}{d+1}\right)^d (s+1)}{\left(\frac{s}{s+1}\right)^s (d+1)}\right)^{\frac{1}{d-s}}\right)}.$$

Proof. Observe, that for $s = 0$ our upper bound on the price of anarchy reduces to the exact value on the price of anarchy that was proved by Roughgarden [19] for this case. So, in the following, we assume that $s \geq 1$. Let $\mathsf{x} = (x_{1P_1}, \ldots, x_{nP_n})$ be a Wardrop equilibrium and let $\mathsf{x}^* = (x^*_{1P_1}, \ldots, x^*_{nP_n})$ be a strategy profile with optimum social cost. Since x is a Wardrop equilibrium, it follows by definition of a Wardrop equilibrium that

$$\begin{aligned} \mathsf{SC}(\mathsf{x}) &= \sum_{i \in [n]} \sum_{P_i \in \mathcal{P}_i} x_{iP_i} \sum_{e \in P_i} f_e(l_e(\mathsf{x})) \leq \sum_{i \in [n]} \sum_{P_i \in \mathcal{P}_i} x^*_{iP_i} \sum_{e \in P_i} f_e(l_e(\mathsf{x})) \\ &= \sum_{e \in E} \underbrace{l_e(\mathsf{x}^*)}_{:=y} \cdot \underbrace{f_e(l_e(\mathsf{x}))}_{:=x}. \end{aligned}$$

Now, since $l_e(\mathsf{x}^*)$ and $l_e(\mathsf{x})$ are both positive real numbers, assume that c_1 and c_2 are such that

$$y \cdot f(z) \le c_1 \cdot z \cdot f(z) + c_2 \cdot y \cdot f(y) \qquad \forall x, y \in \mathbb{R}_{\ge 0} \tag{3}$$

for all polynomials f with minimum degree s and maximum degree d, having nonnegative coefficients. Then,

$$\begin{aligned}\mathsf{SC}(\mathsf{x}) &\le c_1 \cdot \sum_{e \in E} l_e(\mathsf{x}) \cdot f_e(l_e(\mathsf{x})) + c_2 \cdot \sum_{e \in E} l_e(\mathsf{x}^*) \cdot f_e(l_e(\mathsf{x}^*)) \\ &= c_1 \cdot \mathsf{SC}(\mathsf{x}) + c_2 \cdot \mathsf{SC}(\mathsf{x}^*)\end{aligned}$$

and with $0 < c_1 < 1$ it follows that

$$\frac{\mathsf{SC}(\mathsf{x})}{\mathsf{SC}(\mathsf{x}^*)} \le \frac{c_2}{1 - c_1}.$$

Since x is an arbitrary Wardrop equilibrium, we get

$$\mathsf{PoA}(d, s) \le \frac{c_2}{1 - c_1}. \tag{4}$$

We will now show how to determine c_1 and c_2 such that inequality (3) holds and that the resulting upper bound is minimal. In case $y = 0$ equation (3) follows immediately from $c_1 \ge 0$ and $z \ge 0$. In case $z = 0$ the left hand side yields $y \cdot f(0) = 0$, since by the degree of the lowest monomial being $s \ge 1$ there are no additive constants in the latency functions. This is always less or equal than $c_2 \cdot y \cdot f(y)$, since latency functions are monotone increasing, $y \ge 0$ and $c_2 \ge 0$. So, in the following we assume that $y > 0$ and $z > 0$. In order to show that (3) holds, it suffices to show that (3) holds for all monomials of degree $i \in [n]$ with $s \le i \le d$, since polynomials are a linear combination of monomials. This implies that (3) then holds also for the considered polynomials. By Lemma 1 it suffices to show this for the monomials of degree s and d. Consider inequality (3) for a single monomial $f(z) = a_i \cdot z^i$, which we divide by $a_i \cdot y^{i+1}$ yielding

$$\left(\frac{z}{y}\right)^i \le c_1 \cdot \left(\frac{z}{y}\right)^{i+1} + c_2 \qquad \forall s \le i \le d \;\forall y, z \in \mathbb{R}_{>0}.$$

Set $\hat{z} := \frac{z}{y}$. Then $\hat{z} \in \mathbb{R}_{>0}$ and (3) reduces to

$$\hat{z}^i \le c_1 \cdot \hat{z}^{i+1} + c_2 \qquad \forall s \le i \le d \;\forall \hat{z} \in \mathbb{R}_{>0}. \tag{5}$$

We now view (5) as a function in i, c_1 and $\hat{z}$, since we want to determine the maximum $\hat{z}$ such that inequality (3) holds. Thus, we have the following function

$$c_2(i, c_1, \hat{z}) := \hat{z}^i - c_1 \cdot \hat{z}^{i+1}, \tag{6}$$

which we partially differentiate in $\hat{z}$ in order to retrieve the minimum c_2 such that (5) holds, yielding

$$\frac{\partial}{\partial \hat{z}} c_2(i, c_1, \hat{z}) = i \cdot \hat{z}^{i-1} - (i+1) \cdot c_1 \cdot \hat{z}^i.$$

The $\hat{z}$ for which c_2 is maximum can now be easily determined to be $\hat{z}^{max} := \frac{i}{c_1 \cdot (i+1)}$. Simple insertion in (6) yields

$$c_2(i, c_1, \hat{z}^{max}) = \left(\frac{i}{c_1 \cdot (i+1)}\right)^i - c_1 \cdot \left(\frac{i}{c_1 \cdot (i+1)}\right)^{i+1} = \frac{(i+1) \cdot i^i - i^{i+1}}{(i+1)^{i+1} \cdot c_1^i}$$
$$= \frac{\left(\frac{i}{i+1}\right)^i}{c_1^i \cdot (i+1)}.$$

We define

$$c_2(i, c_1) := \frac{\left(\frac{i}{i+1}\right)^i}{c_1^i \cdot (i+1)}. \tag{7}$$

Lemma 1 states that it suffices to focus on the monomials of degree s and d in order for (3) to hold. We therefore determine c_1 as a solution to the equation of $c_2(d, c_1) = c_2(s, c_1)$. Thus, we have

$$\frac{\left(\frac{d}{d+1}\right)^d}{c_1^d \cdot (d+1)} = \frac{\left(\frac{s}{s+1}\right)^s}{c_1^s \cdot (s+1)}$$

or equivalently

$$c_1 = \left(\frac{(s+1) \cdot \left(\frac{d}{d+1}\right)^d}{(d+1) \cdot \left(\frac{s}{s+1}\right)^s}\right)^{\frac{1}{d-s}}.$$

Having calculated c_1, we can retrieve c_2 by simple insertion in (7) using the maximum degree d of the monomials yielding

$$c_2 := c_2(d, c_1) = \frac{\left(\frac{d}{d+1}\right)^d}{\left(\frac{(s+1)\cdot\left(\frac{d}{d+1}\right)^d}{(d+1)\left(\frac{s}{s+1}\right)^s}\right)^{\frac{d}{d-s}} \cdot (d+1)}.$$

We get with (4) that

$$\mathsf{PoA}(d, s) \leq \frac{c_2}{1 - c_1} = \frac{\frac{\left(\frac{d}{d+1}\right)^d}{\left(\frac{(s+1)\cdot\left(\frac{d}{d+1}\right)^d}{(d+1)\left(\frac{s}{s+1}\right)^s}\right)^{\frac{d}{d-s}} \cdot (d+1)}}{1 - \left[\frac{(s+1)\cdot\left(\frac{d}{d+1}\right)^d}{(d+1)\cdot\left(\frac{s}{s+1}\right)^s}\right]^{\frac{1}{d-s}}}$$
$$= \frac{\left(\frac{d}{d+1}\right)^d}{(d+1) \cdot \left(\frac{\left(\frac{d}{d+1}\right)^d (s+1)}{\left(\frac{s}{s+1}\right)^s (d+1)}\right)^{\frac{d}{d-s}} \cdot \left(1 - \left(\frac{\left(\frac{d}{d+1}\right)^d (s+1)}{\left(\frac{s}{s+1}\right)^s (d+1)}\right)^{\frac{1}{d-s}}\right)},$$

which completes the proof of the theorem. □

Having proved the general upper bound, we now investigate the case $s = \left\lceil \frac{d}{a} \right\rceil$ with $a \in \mathbb{R}$ and $1 \leq a \leq d$. For the case $a = 1$, we have that $\mathsf{PoA}(d, d) = 1$

as shown in [21] for $d = 1$ and Theorem 1 shows that this also holds for the case $d \geq 2$. In order to prove an upper bound on $\mathsf{PoA}\left(d, \lceil \frac{d}{a} \rceil\right)$, we first show monotonicity results for the upper bound from Theorem 1.

Lemma 2. *The upper bound on* $\mathit{PoA}(d, s)$ *from Theorem 1 is monotone decreasing in* s.

Lemma 3. *The upper bound on* $\mathit{PoA}\left(d, \frac{d}{a}\right)$ *from Theorem 1 is monotone increasing in* d.

Combining the last two lemmas yields the corollary that $\mathsf{PoA}(d, s)$ is monotone increasing in d.

Corollary 1. *The upper bound on* $\mathit{PoA}(d, s)$ *from Theorem 1 is monotone increasing in* d.

By Lemma 2, we can neglect the ceilings and replace s by $\frac{d}{a}$ in the upper bound from Theorem 1 to get an upper bound on $\mathsf{PoA}\left(d, \lceil \frac{d}{a} \rceil\right)$. Furthermore, by Lemma 3, this upper bound has the largest value for $d \to \infty$. By computing this limit, we get:

Theorem 2. *For Wardrop games with* (d, s)*-polynomial latency functions where* $s = \lceil \frac{d}{a} \rceil$, *we have*

$$\mathit{PoA}\left(d, \left\lceil \frac{d}{a} \right\rceil\right) \leq \frac{a^{\frac{1}{a-1}} \cdot (a-1)}{\mathsf{e} \cdot \ln(a)}. \tag{8}$$

3.2 Lower Bound

For the lower bound, we consider an instance of a Wardrop game with $n = 1$ player of traffic $w_1 = 1$. The network consists of two parallel edges u and ℓ from node s_1 to node t_1. The latency functions are $f_u(x) = \alpha \cdot x^s$ and $f_\ell(x) = x^d$, where $\alpha \in \mathbb{R}_{>0}$ will be determined later. With a slight abuse of notation, let $\mathsf{z} = (z, 1-z)$ be a Wardrop equilibrium and let $\widehat{\mathsf{z}} = (\widehat{z}, 1-\widehat{z})$ be the optimum strategy profile, where z (resp. $\widehat{z}$) is the amount of traffic that is assigned to link u in the Wardrop equilibrium (resp. optimum).

In the Wardrop equilibrium, the latency on both links is the same, so z is the only positive solution to

$$\alpha \cdot z^s = (1-z)^d. \tag{9}$$

On the other hand, the optimum is defined by

$$\widehat{z} := \arg \min_{x \in [0,1]} \left\{ \alpha \cdot x^{s+1} + (1-x)^{d+1} \right\},$$

which yields that $\widehat{z}$ is the only positive solution to

$$\alpha \cdot \frac{s+1}{d+1} \cdot \widehat{z}^s - (1-\widehat{z})^d = 0. \tag{10}$$

Observe, that z and $\widehat{z}$ are both dependent on α, s and d. If we can compute z and $\widehat{z}$, then we can give a lower bound on the price of anarchy

$$\mathsf{PoA}(d,s) \geq \frac{\mathsf{SC(z)}}{\mathsf{SC}(\widehat{\mathsf{z}})} = \frac{\alpha \cdot z^{s+1} + (1-z)^{d+1}}{\alpha \cdot \widehat{z}^{s+1} + (1-\widehat{z})^{d+1}}.$$

We can now further optimize this lower bound by choosing the best possible α.

The problem is that to determine z and $\widehat{z}$, as we have to compute the root for polynomials with arbitrary degree as demanded for the equations (9) and (10). Numerical tests for all (d,s)-polynomials with $d \leq 30$ gives lower bounds that match the upper bounds from Theorem 1 up to some numeric precision. We have given example values for the factor α for polynomial latency functions up to a degree of 9 in Table 2, where reasonable. This is a strong indication that our lower bound might be matching for all $d \in \mathbb{N}$ and $s \in \mathbb{N}$ with $s \leq d$.

Table 2. Example values for α, such that the lower bound is matching

$s \backslash d$	2	3	4	5	6	7	8	9
1	$\frac{25}{24}$	1.06452	1.07793	1.08599	1.09074	1.09335	1.09450	1.09462
2		1.02181	1.03361	1.03954	1.04180	1.04166	1.03991	1.03705
3			1.01087	1.01544	1.01603	1.01405	1.01022	1.00524
4				1.00399	1.00365	1.00045	0.99531	0.98885
5					0.99927	0.99540	0.98939	0.98191
6						0.99583	0.98929	0.98112
7							0.99321	0.98459
8								0.99115

An Almost Matching Lower Bound for a Special Case

We now show a lower bound on the price of anarchy for the case that $s = \frac{d}{2}$.

Set the constant factor $\alpha = 1$. Then, the traffic on u in the Wardrop equilibrium is the solution of the equation

$$z^{\frac{d}{2}} = (1-z)^d,$$

which yields

$$z = \frac{3}{2} - \frac{1}{2}\sqrt{5}.$$

We get that the social cost in the Wardrop equilibrium is

$$\mathsf{SC(z)} = z^{\frac{d}{2}+1} + (1-z)^{d+1} = z^{\frac{d}{2}} = \left(\frac{3}{2} - \frac{1}{2}\sqrt{5}\right)^{\frac{d}{2}}.$$

On the other hand, for the optimum we get that $\widehat{z}$ is the only positive root of

$$\left(\frac{d}{2}+1\right)\widehat{z}^{\frac{d}{2}} - (d+1)(1-\widehat{z})^d.$$

This root calculates to

$$\widehat{z} = 1 + \frac{1 - \sqrt{1 + 4\left(\frac{2d+2}{d+2}\right)^{\frac{2}{d}}}}{2\left(\frac{2d+2}{d+2}\right)^{\frac{2}{d}}}.$$

The social cost in the optimum is

$$\mathsf{SC}(\widehat{\mathsf{z}}) = \widehat{z}^{\frac{d}{2}+1} + (1-\widehat{z})^{d+1} = \widehat{z}^{\frac{d}{2}} \cdot \frac{\frac{d}{2}+1+\frac{d}{2}\cdot\widehat{z}}{d+1}.$$

We get

$$\frac{\mathsf{SC}(\mathsf{z})}{\mathsf{SC}(\widehat{\mathsf{z}})} = \left(\frac{z}{\widehat{z}}\right)^{\frac{d}{2}} \cdot \frac{d+1}{\frac{d}{2}+1+\frac{d}{2}\cdot\widehat{z}},$$

with limit

$$\lim_{d\to\infty} \frac{\mathsf{SC}(\mathsf{z})}{\mathsf{SC}(\widehat{\mathsf{z}})} = \frac{1}{2^{\frac{1}{\sqrt{5}}}} \cdot \frac{4}{5-\sqrt{5}} > 1.0614704.$$

Thus, for d large enough, we have $\mathsf{PoA}(d, \frac{d}{2}) \geq 1.0614704$. This is slightly below the upper bound of $\mathsf{PoA}(d, \frac{d}{2}) \leq 1.0614756$. The same computations for $a = 3$ yield a lower bound of $\mathsf{PoA}(d, \frac{d}{3}) \geq 1.159949$, which is again slightly below the upper bound of $\mathsf{PoA}(d, \frac{d}{3}) \leq 1.159983$. Note that for $a \geq 5$, we are again confronted with the problem of computing a general root.

4 Conclusion

In this paper, we have shown a general upper bound on the price of anarchy for Wardrop games with (d, s)-polynomial latency functions. We then proved monotonicity results on this upper bound and applied these to show that the price of anarchy is upper bounded by a constant, if s is a constant fraction of d. As an example, for $s = \frac{d}{2}$ this upper bound is 1.0614756. This implies that the price of anarchy does not only depend on the "steepness" of the latency functions, but rather on the presence of the lower monomials as the price of anarchy increases with the presence of lower monomials. Our discussion on lower bounds strongly indicates that our upper bound yields the exact value for the price of anarchy. However, the problem of finding a matching general lower bound that holds for all (d, s)-polynomials remains tantalizingly open.

References

1. S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact Price of Anarchy for Polynomial Congestion Games. *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science (STACS'06)*, LNCS 3884, Springer Verlag, pages 218–229, 2006.
2. B. Awerbuch, Y. Azar, and A. Epstein. The Price of Routing Unsplittable Flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC'05)*, pages 57–66, 2005.
3. M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation.* Yale University Press, 1956.
4. M. J. Beckmann. On the Theory of Traffic Flow in Networks. *Traffic Quarterly*, 21:109–116, 1967.
5. G. Christodoulou and E. Koutsoupias. The Price of Anarchy of Finite Congestion Games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC'05)*, pages 67–73, 2005.

6. J. R. Correa, and A. S. Schulz and N. E. Stier-Moses. On the Inefficiency of Equilibria in Nonatomic Congestion Games. In *Proceedings of the 11th Conference on Integer Programming and Combinatorial Optimization (IPCO'05)*, LNCS 3509, Springer Verlag, pages 167–181, 2005.
7. R. Cominetti, J. R. Correa, and N. E. Stier-Moses. Network Games With Atomic Players. In *Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP'06)*, LNCS 4051, Springer Verlag, pages 525–536, 2006.
8. A. Czumaj and B. Vöcking. Tight Bounds for Worst-Case Equilibria. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02)*, pages 413–420, 2002. Also accepted to *Journal of Algorithms* as Special Issue of SODA'02.
9. M. Gairing, T. Lücking, M. Mavronicolas, and B. Monien. The Price of Anarchy for Polynomial Social Cost. *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS'04)*, LNCS 3153, Springer Verlag, pages 574–585, 2004.
10. M. Gairing, T. Lücking, M. Mavronicolas, and B. Monien. The Price of Anarchy for Restricted Parallel Links. *Parallel Processing Letters*, 16(1):117–131, 2006.
11. M. Gairing, T. Lücking, B. Monien, and K. Tiemann. Nash Equilibria, the Price of Anarchy and the Fully Mixed Nash Equilibrium Conjecture. In *Proceedings of the 32nd International Colloquium on Automata, Languages, and Programming (ICALP'05)*, LNCS 3580, Springer Verlag, pages 51–65, 2005.
12. M. Gairing, B. Monien, and K. Tiemann. Selfish Routing with Incomplete Information. In *Proceedings of the 17th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'05)*, pages 203–212, 2005.
13. M. Gairing, B. Monien, and K. Tiemann. Routing (Un-) Splittable Flow in Games with Player-Specific Linear Latency Functions. In *Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP'06)*, LNCS 4051, Springer Verlag, pages 501–512, 2006.
14. E. Koutsoupias and C. H. Papadimitriou. Worst-Case Equilibria. *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS'99)*, LNCS 1563, Springer Verlag, pages 404–413, 1999.
15. M. Mavronicolas and P. Spirakis. The Price of Selfish Routing. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC'01)*, pages 510–519, 2001.
16. J. F. Nash. Non-Cooperative Games. *Annals of Mathematics*, 54(2):286–295, 1951.
17. A. C. Pigou. *The Economics of Welfare*. Macmillan and Company, 1920.
18. R. W. Rosenthal. A Class of Games Possessing Pure-Strategy Nash Equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
19. T. Roughgarden. The Price of Anarchy is Independent of the Network Topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.
20. T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
21. T. Roughgarden and É. Tardos. How Bad Is Selfish Routing? *Journal of the ACM*, 49(2):236–259, 2002.
22. V. Vazirani. *Approximation Algorithms*. Springer Verlag, 2001.
23. J. G. Wardrop. Some Theoretical Aspects of Road Traffic Research. In *Proceedings of the Institute of Civil Engineers, Pt. II, Vol. 1*, pages 325–378, 1952.

Wardrop Equilibria and Price of Stability for Bottleneck Games with Splittable Traffic*

Vladimir Mazalov[1], Burkhard Monien[2], Florian Schoppmann[2], and Karsten Tiemann[2]

[1] Institute of Applied Mathematical Research, Karelian Research Center, Russian Academy of Sciences, 11, Pushkinskaya st., Petrozavodsk, Russia, 185910
vmazalov@krc.karelia.ru

[2] Faculty of Computer Science, Electrical Engineering and Mathematics, University of Paderborn, Fürstenallee 11, 33102 Paderborn, Germany
{bm, fschopp, tiemann}@uni-paderborn.de

Abstract. We look at the scenario of having to route a continuous rate of traffic from a source node to a sink node in a network, where the objective is to maximize throughput. This is of interest, e.g., for providers of streaming content in communication networks. The overall path latency, which was relevant in other non-cooperative network routing games such as the classic Wardrop model, is of lesser concern here.

To that end, we define bottleneck games with splittable traffic where the throughput on a path is inversely proportional to the maximum latency of an edge on that very path—the bottleneck latency. Therefore, we define a Wardrop equilibrium as a traffic distribution where this bottleneck latency is at minimum on all used paths. As a measure for the overall system well-being—called social cost—we take the weighted sum of the bottleneck latencies of all paths.

Our main findings are as follows: First, we prove social cost of Wardrop equilibria on series parallel graphs to be unique. Even more, for any graph whose subgraph induced by all simple start-destination paths is not series parallel, there exist games having equilibria with different social cost. For the price of stability, we give an independence result with regard to the network topology. Finally, our main result is giving a new exact price of stability for Wardrop/bottleneck games on parallel links with M/M/1 latency functions. This result is at the same time the exact price of stability for bottleneck games on general graphs.

1 Introduction

Motivation and Framework. In recent years, the Wardrop model—which was already introduced in the 1950's (see, e.g., [4,25])—received a lot of attention with regard to analyzing the price of anarchy, i.e., quantifying the worst-case system loss due to selfish behavior of its participants. A Wardrop game can be understood as a game with infinitely many players each in control of a negligible fraction of the total traffic in a

* This work was partially supported by the IST Program of the European Union under contract number IST-15964 (AEOLUS). The research of Florian Schoppmann and Karsten Tiemann was supported by the International Graduate School of Dynamic Intelligent Systems (Paderborn, Germany).

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 331–342, 2006.
© Springer-Verlag Berlin Heidelberg 2006

network. Players choose a path according to their respective start and destination node, with the assumption that they act purely selfishly and therefore each takes the fastest path under current network conditions. A situation in which none of the selfish players has an incentive to switch to another path is called a *Wardrop equilibrium*.

While the Wardrop model has been successfully applied for researching road traffic, its basic assumption of drivers minimizing just their own travel time (more generally called path latency) is not appropriate in all networks. For instance, in communication networks such as the Internet, providers of streaming content would strive to maximize the *throughput* to their clients whereas the transmission time is of lesser concern. More generally, for a routing path in this network, one would be interested in the maximum latency of all its edges—in other words, the *latency of the bottleneck*—as it is inversely proportional to the achievable throughput on that very path. From a purely mathematical perspective, the bottleneck latency of a path corresponds to the ∞-norm of the (finite) vector of edge latencies whereas the sum of edge latencies—that was of interest in the Wardrop model—equals the 1-norm. For a broader discussion of when the ∞-norm should be used, confer also Banner and Orda [3]. They mention, for instance, that the ∞-norm is appropriate to model wireless networks where each node has a limited transmission energy. As another motivation, a celebrated result by Leighton et al. [15] implies that the bottleneck latency is also of interest in settings with individual traffic: Their result states that individual packets can be routed in time $\mathcal{O}(\text{congestion} + \text{dilation})$ when the paths for the packets are given in advance. Here dilation denotes the maximum length of a path and congestion denotes the maximum number of paths sharing a common edge.

We address the scenario described in the last paragraphs by studying what we call *bottleneck games with splittable traffic*. Similar to Wardrop games, one could think of infinitely many selfish players each controlling a negligible amount of traffic. However, their objective is now to choose a path such that their experienced bottleneck latency is at minimum. Likewise, we define a Wardrop equilibrium in a game of our new model as a traffic distribution where the bottleneck latency is at minimum on all used paths. As a measure for the overall system well-being—called *social cost*—we take the weighted sum of the bottleneck latencies of all paths. A similar weighted sum over the path latencies was used as social cost for Wardrop games (see e.g. [22]).

In this work, we will consider the degradation of social welfare due to selfish behavior of the players. To that end, we will determine the so called *price of stability*, a term that was coined by Anshelevich et al. [2] and denotes the worst-case ratio, over all instances, between the social cost of the *best* equilibrium and optimum social cost. Roughly speaking, it describes the worst-case inefficiency of optimum stable states, in which no player wants to unilaterally deviate, compared to an overall optimum solution. In contrast, the *price of anarchy*, which was introduced by Koutsoupias and Papadimitriou [14], denotes the worst-case ratio, over all instances, between the social cost of *any* equilibrium and that of social optima.

Related Work. Wardrop Games: Inspired by the arisen interest in the price of anarchy, Roughgarden and Tardos [22] re-investigated the Wardrop games and used the *total latency* as social cost. In this context the price of anarchy was shown to be $\frac{4}{3}$ for linear latency functions [22] and $\Theta(\frac{d}{\ln d})$ for polynomials of degree at most d with non-negative coefficients [21]. Roughgarden [19] proved that the price of anarchy is independent of the network topology if a class $\mathcal{F}$ of latency functions is considered that only fulfills relatively weak assumptions. Instead, it only depends on the so called "anarchy value" $\alpha(\mathcal{F})$ of $\mathcal{F}$, and the worst-case ratio is already achieved on parallel links. Roughgarden [21] also considered networks with *M/M/1 latency functions*. When r is the amount of traffic and $c_{\min} > r$ is the minimum capacity among all edge capacities in the network, an upper bound on the price of anarchy is given by $\frac{1}{2} \cdot (1 + \sqrt{c_{\min}/(c_{\min} - r)})$. Observe that this expression approaches ∞ as the amount of traffic r approaches $c_{\min}$. The upper bound is asymptotically tight even for games on so-called union of paths graphs, i.e., on graphs that consist of many disjoint paths from s to t only having the two nodes s and t in common. Note that the results on the price of stability for games with M/M/1 latency functions that we give in this paper even apply if $c_{\min} \leq r$.

In a recent paper, Cole et al. [8] studied Wardrop-like games where the latency of a path is defined as the p-norm, $1 < p \leq \infty$, of the vector of its edge latencies. In this context, they also looked at "elastic traffic", i.e., some share of the participants might be better off by not traveling at all. When $p = \infty$ and in the case of inelastic traffic, their games are equal to our bottleneck games with splittable traffic. However, they looked at a subclass of Wardrop equilibria that they define as "subpath-optimal", with the reason for their restricting being that otherwise the price of anarchy is infinite even if latency functions are just linear. They showed that the anarchy value is an upper bound on the price of anarchy for subpath optimal equilibria and hence also an upper bound on the price of stability.

There is also some work that focused on the original Wardrop games but did not use total latency to measure the social cost [10,9,20].

Finite Splittable Routing Games: In this setting, a finite number of players with non-negligible effect on each other is given who have to split their traffics over the available paths with the objective to minimize their private costs. Two papers [13,16] studied such games with certain player-specific private cost functions that are based on M/M/1 latency functions. Korilis et al. [13] studied what happens to the private costs of the players if new capacity is added to the network or if existing capacity is reallocated. Orda et al. [16] considered the (non-) uniqueness of Nash equilibria. Banner and Orda [3] studied finite splittable routing games where the private cost of a player is defined as the maximum among all latencies of edges to which this player assigns a non-zero flow, whereas social cost is given by the maximum edge latency in the network. Banner and Orda proved the existence and non-uniqueness of equilibria. They were also able to show that the price of anarchy is unbounded.

Finite Unsplittable Routing Games: Again, there are finitely many players with non-negligible effect on each other and each having to route all its traffic on the same path (see [11] for a survey). Two recent papers [6,7] studied the routing of unsplittable traffics where the private cost of a player is defined as the maximum latency of any edge on its path, i.e., the bottleneck latency in our words. Caragiannis et al. [7] allowed different amounts of traffic for the players, whereas in the setting of Busch and Magdon-Ismail

[6] all players control traffic of unit size. Both studied the price of anarchy with respect to social cost defined as the maximum latency of any edge in the network.

M/M/1 Latency Functions: M/M/1 latency functions arise in queuing theory as the expected latency of queues with a Poisson arrival process and an exponentially distributed service time [12,18]. They are used in networking theory to model packet-switched networks. Here, a packet that starts at its entry node in the network or arrives at an intermediate node on its way to the destination is stored in a queue. It can leave the queue as soon as the next link on the path of the packet becomes available [5,24].

Contribution. In this work, we define and study *bottleneck games with splittable traffic*. The ingredients of such a game are a graph $G = (V, E)$, whose edges $e \in E$ are each endowed with a latency function $f_e : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \cup \{\infty\}$, and distinct source and sink nodes $s, t \in V$ between which an arbitrarily splittable traffic $r > 0$ has to be routed. We will give special attention to instances having M/M/1 latency functions, i.e., functions of the form $f_e(u) = \frac{1}{c_e - u}$ where $c_e > 0$ denotes the *capacity* of edge $e \in E$.

Our investigations are two-fold: First, we study general properties of bottleneck games with splittable traffic such as existence and uniqueness of Wardrop equilibria and dependence of both the price of anarchy and stability of the network topology. Most of our results here are based on properties of maximum flows and minimum cuts. In the second part we prove an exact expression for the price of stability for bottleneck games on parallel links with splittable traffic and M/M/1 latency functions. We view this result as the main result of our paper. Especially the proof of the upper bound requires a very careful analysis. In detail, our main findings are:

– General results for bottleneck games with splittable traffic:
  - We define the notion of *capacity* of a network and then show that a bottleneck game with splittable traffic has a Wardrop equilibrium with finite social cost if the traffic is smaller than the network capacity.
  - For games on *series parallel graphs* with arbitrary latency functions we prove the social cost of Wardrop equilibria to be unique. On the other hand, we also show that for any graph whose subgraph induced by all simple start-destination paths is not series parallel, there exist games having equilibria with different social cost.
  - We show that the price of stability for bottleneck games with splittable traffic is independent of the network topology, i.e., the worst-case ratio, over all instances, between the best Nash equilibrium and an optimum is attained on *parallel links*. (See Section 3.3 for a comparison with a similar result by Cole et al. [8].)

– Bottleneck games with splittable traffic and M/M/1 latency functions:
  We prove that the expression

$$\frac{m \cdot \frac{r}{c_{\min}}}{\frac{r}{c_{\min}} + 2 \cdot (m-1) \cdot \left(\sqrt{\frac{r}{c_{\min}} + 1} - 1\right)} \tag{1}$$

describes the *exact* price of stability for games on m parallel links with M/M/1 latency functions, minimum edge capacity $c_{\min}$, and traffic r. Furthermore, the expression is increasing in both m and r and it converges to m for large r.

Interestingly, series parallel graphs form exactly the class of graphs where the prices of anarchy and stability conside for every class of latency functions. Parallel links are special series parallel graphs. Furthermore, on parallel link graphs bottleneck games and Wardrop games coincide. Our results imply that for every class of latency functions bounds for the price of stability for parallel link graphs also hold for the price of stability of bottleneck games on arbitrary graphs. This can be used when latency functions are restricted to polynomials where results of Roughgarden [19] can be used, and also for the class of M/M/1 latency functions where the expression (1) describes the price of stability for bottleneck games on arbitrary graphs.

Road Map. The rest of the paper is organized as follows. In Section 2 we give exact definitions for our bottleneck games with splittable traffic. We study general games in Section 3, whereas we restrict ourselves to M/M/1 latency functions in Section 4. Due to lack of space we have to omit most of the proofs.

2 Notation

For all $k \in \mathbb{N}$ denote $[k] = \{1, \ldots, k\}$.

Latency Function, Network, Instance. A *latency function* $f : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \cup \{\infty\}$ is a nonnegative, continuous, and nondecreasing function. Here, $\mathbb{R}_0^+ \cup \{\infty\}$ is meant to be endowed with the topology of the one-point compactification of $\mathbb{R}_0^+$. (This basically means that f has no jump discontinuities, not even to ∞.) A *network* is a tuple $(G, s, t, (f_e)_{e \in E})$, where $G = (V, E)$ is a directed *multigraph*, $s, t \in V$ are distinct *source* and *sink* (target) nodes, and the f_e are latency functions. A *bottleneck game with splittable traffic* with *general latency functions* is a tuple $\Gamma = (G, s, t, (f_e)_{e \in E}, r)$ where $(G, s, t, (f_e)_{e \in E})$ is a network in which a *traffic* of $r \in \mathbb{R}^+$ has to be routed from s to t. When obvious, we will usually refer to our bottlenecks games with splittable traffic only as bottleneck games or even just games.

M/M/1 latency function. For ease of notation, we write $\Gamma = (G, s, t, (c_e)_{e \in E}, r)$ for a bottleneck game with splittable traffic and *M/M/1 latency functions* where $c_e > 0$ is the *capacity* for edge $e \in E$. The M/M/1 latency functions f_e, $e \in E$, are implicitly defined by

$$f_e(u) = \begin{cases} \frac{1}{c_e - u} & \text{if } u < c_e \\ \infty & \text{otherwise.} \end{cases}$$

Observe that the latency $f_e(u)$ approaches ∞ as the load u approaches c_e. We denote by $\mathcal{M}$ the set of all M/M/1 latency functions and by $\mathcal{M}_{\geq c} \subset \mathcal{M}$ the functions with a capacity of at least c where $c > 0$.

Strategy Profiles, Wardrop Equilibria, and Social Cost. The traffic r can split arbitrarily over the set $\mathcal{P}_{st}$ of all possible *simple* paths from s to t. A *strategy profile* is a vector $\mathbf{x} = (x_P)_{P \in \mathcal{P}_{st}}$ where $\sum_{P \in \mathcal{P}_{st}} x_P = r$ and $x_P \geq 0$ for all $P \in \mathcal{P}_{st}$. The *load* δ_e on an edge $e \in E$ is given by $\delta_e(\mathbf{x}) = \sum_{P \in \mathcal{P}_{st}, P \ni e} x_P$. A strategy profile is a *Wardrop equilibrium* if the latency of each used path is not larger than the latency of any other path, i.e., if for all $P, R \in \mathcal{P}_{st}$

$$x_P > 0 \quad \Rightarrow \quad \max_{e \in P} f_e(\delta_e(\mathbf{x})) \leq \max_{e \in R} f_e(\delta_e(\mathbf{x})).$$

The *social cost* of a strategy profile $\mathbf{x}$ is defined as the "canonically" weighted sum of all path latencies, i.e.,

$$\mathsf{SC}(\Gamma, \mathbf{x}) = \sum_{P \in \mathcal{P}_{st}} x_P \cdot \max_{e \in P} f_e(\delta_e(\mathbf{x})).$$

If $\mathbf{x}$ is a Wardrop equilibrium, $l(\mathbf{x}) = \frac{\mathsf{SC}(\Gamma, \mathbf{x})}{r}$ denotes the unique latency of all paths with non-zero flow. The *optimum* associated with a bottleneck game with splittable traffic Γ is the minimum social cost of any strategy profile: $\mathsf{OPT}(\Gamma) = \min_{\mathbf{x}} \mathsf{SC}(\Gamma, \mathbf{x})$. The *price of anarchy* (PoA) and *price of stability* (PoS) for a set $\mathscr{G}$ of games are defined as

$$\mathsf{PoA}(\mathscr{G}) := \sup_{\substack{\Gamma \in \mathscr{G} \\ \mathbf{x} \text{ Wardr. Equ. in } \Gamma}} \frac{\mathsf{SC}(\Gamma, \mathbf{x})}{\mathsf{OPT}(\Gamma)} \quad \text{and} \quad \mathsf{PoS}(\mathscr{G}) := \sup_{\Gamma \in \mathscr{G}} \inf_{\mathbf{x} \text{ Wardr. Equ. in } \Gamma} \frac{\mathsf{SC}(\Gamma, \mathbf{x})}{\mathsf{OPT}(\Gamma)}$$

where by definition $\infty/\infty := 1$ and $0/0 := 1$. Furthermore, $u/0 := \infty$ if $u > 0$. For a given network $(G, s, t, (f_e)_{e \in E})$ its *capacity* is given by

$$C(G, s, t, (f_e)_{e \in E}) = \sup \left\{ r \in \mathbb{R}_0^+ \;\middle|\; \begin{array}{l} \exists \text{ strategy profile } \mathbf{x} \text{ with } \mathsf{SC}(\Gamma, \mathbf{x}) < \infty \\ \text{for } (G, s, t, (f_e)_{e \in E}, r) \end{array} \right\} \cup \{0\}.$$

Series Parallel Graphs. (Sometimes also called two terminal series parallel.) *Series parallel* is a recursively defined property: As the base case, the graph that only consists of two nodes s, t and a single edge (s, t) is series parallel with terminals (s, t). An arbitrary multigraph G is series parallel with terminals (s, t) if it can be constructed from two series parallel graphs with terminals (s_1, t_1) and (s_2, t_2) connected either in series or in parallel. In a series connection, $t_1 = s_2$, $s = s_1$, and $t = t_2$. In a parallel connection, $s = s_1 = s_2$ and $t = t_1 = t_2$.

Parallel Links. A graph of *parallel links* is a multigraph $G = (V, E)$ consisting of two nodes $V = \{s, t\}$ and m edges $E = \{1, 2, \ldots, m\}$ from s to t. Whenever a bottleneck game with splittable traffic and M/M/1 latency functions on parallel links is considered we assume that $c_1 \geq \ldots \geq c_m$ and denote $C := \sum_{i=1}^m c_i$ and $C^{\leq i} := \sum_{k=1}^i c_k$. Clearly, C is just the capacity of the network.

For a non-empty set $\mathcal{F}$ of latency functions we define $\mathscr{G}(\mathcal{F})$ as the set of all bottleneck games with latency functions drawn from $\mathcal{F}$. The subset $\mathscr{P}(\mathcal{F}) \subset \mathscr{G}(\mathcal{F})$ consists of all games in $\mathscr{G}(\mathcal{F})$ that are defined on a graph of parallel links. To further differentiate we denote by $\mathscr{G}(\mathcal{F}, m, r) \subset \mathscr{G}(\mathcal{F})$ the set of games with at most m edges and a traffic of at most r. Likewise, $\mathscr{P}(\mathcal{F}, m, r) := \mathscr{G}(\mathcal{F}, m, r) \cap \mathscr{P}(\mathcal{F})$.

3 General Results for Bottleneck Games with Splittable Traffic

For general bottleneck games with splittable traffic we will prove the existence of Wardrop equilibria (Section 3.1), study the (non-)uniqueness of equilibria social cost (Section 3.2), and show that the price of stability is independent of the the network topology (Section 3.3).

3.1 Existence of Wardrop Equilibria

Existence of Wardrop equilibria in bottleneck games with splittable traffic can be established by employing the general result of [23] (for a proof using more elementary maths, see [17]). To illustrate the connection to maximum flows, however, we start by giving a proof that makes use of the max-flow min-cut theorem (see, e.g., [1]). The construction described in the proof will also be used in the proof of Theorem 5. Obviously, the only interesting case is the traffic to be routed being smaller than the capacity of the network.

Theorem 1. *Let $\Gamma = (G, s, t, (f_e)_{e\in E}, r)$ be a bottleneck game with splittable traffic where $r < C(G, s, t, (f_e)_{e\in E})$. Then Γ possesses a Wardrop equilibrium of finite social cost.*

3.2 Uniqueness Results About Social Cost of Equilibria

We will show in this section that different equilibria for a bottleneck game with splittable traffic on a series parallel graph have the same social cost. The proof for this result employs a technique based on what we define as *strong cuts*.

Definition 1. *Let Γ be a bottleneck game with splittable traffic on a series parallel graph $G = (V, E)$ and let $\mathbf{x}$ be a Wardrop equilibrium for Γ. Then $D \subseteq E$ is called* strong cut *with respect to Γ and $\mathbf{x}$ if*

1. *each path $P \in \mathcal{P}_{st}$ contains exactly one edge that belongs to D, and*
2. *$f_e(\delta_e(\mathbf{x})) \geq l(\mathbf{x})$ for all edges $e \in D$.*

Observe that, given a strong cut D with respect to Γ and an equilibrium $\mathbf{x}$, all edges $e \in D$ with $\delta_e(\mathbf{x}) > 0$ have latency $l(\mathbf{x})$ whereas all other edges $e \in D$ with $\delta_e(\mathbf{x}) = 0$ have latency at least $l(\mathbf{x})$. Before making use of the crucial properties of strong cuts, we need to ensure their existence.

Theorem 2. *Let Γ be a bottleneck game with splittable traffic on a series parallel graph and let $\mathbf{x}$ be a Wardrop equilibrium for Γ. Then a strong cut with respect to Γ and $\mathbf{x}$ exists.*

Proof. The proof is by structural induction over all series parallel graphs. Our *induction hypothesis* is that every series parallel graph G with terminals (s, t) has the following property: For any bottleneck game Γ on G and all Wardrop equilibria of Γ, there is a strong cut.

The only *base case* to verify consists of the graph with two nodes s, t solely connected by the edge e. Obviously, for any game Γ on this graph, $\{e\}$ is a strong cut with respect to Γ and its trivial equilibrium. For the *induction step*, consider any arbitrary graph $G = (V, E)$ with terminals (s, t). Furthermore, assume that G is a series parallel connection of two series parallel graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with terminals (s_1, t_1) and (s_2, t_2), respectively, and both G_1 and G_2 fulfill the induction hypothesis. To prove the induction step, we then have to show that G fulfills the induction hypothesis, too. Thus, let $\Gamma = (G, s, t, (f_e)_{e\in E}, r)$ be an arbitrary game on G and $\mathbf{x}$ be an arbitrary Wardrop equilibrium for Γ and consider the two cases:

Parallel Connection: Set $r_1 = \sum_{P\in\mathcal{P}_{s_1t_1}} x_P$ and $r_2 = \sum_{P\in\mathcal{P}_{s_2t_2}} x_P$ where $\mathcal{P}_{s_1t_1}$ and $\mathcal{P}_{s_2t_2}$ are meant to only contain paths from G_1 and G_2, respectively. Obviously, $r_1 + r_2 = r$ and the games $\Gamma_1 = (G_1, s_1, t_1, (f_e)_{e\in E_1}, r_1)$ and $\Gamma_2 = (G_2, s_2, t_2, (f_e)_{e\in E_2}, r_2)$ have Wardrop equilibria $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ where $x_P^{(1)} = x_P$ for all paths P with edges in E_1 and $x_P^{(2)} = x_P$ for all paths P with edges in E_2. It follows by the induction hypothesis that there are strong cuts D_1 and D_2 with respect to $G_1, \mathbf{x}^{(1)}$ and $G_2, \mathbf{x}^{(2)}$ that we can use to get a strong cut $D := D_1 \cup D_2$ for Γ and its equilibrium $\mathbf{x}$.

Series Connection: Consider the games $\Gamma_1 = (G_1, s_1, t_1, (f_e)_{e\in E_1}, r)$ and $\Gamma_2 = (G_2, s_2, t_2, (f_e)_{e\in E_2}, r)$. Obviously, $\mathbf{x}$ induces strategy profiles $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ for Γ_1 and Γ_2, respectively. At least one of $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ is a Wardrop equilibrium. Otherwise, there would be a path $P \in \mathcal{P}_{st}$ with non-zero flow on which the latency is larger than on another path $R \in \mathcal{P}_{st}$, and $\mathbf{x}$ cannot be a Wardrop equilibrium. If $\mathbf{x}^{(1)}$ is an equilibrium we set $D := D_1$ and $D := D_2$ otherwise. In either case, D is a strong cut for Γ and its equilibrium $\mathbf{x}$. □

We now use strong cuts in the proof of the next theorem to show that all Wardrop equilibria of a bottleneck game on a series parallel graph have the same social cost. Obviously, this implies that the price of stability does not differ from the price of anarchy for this class of games, i.e., $\mathsf{PoA}(\mathscr{S}) = \mathsf{PoS}(\mathscr{S})$ for any set $\mathscr{S}$ of bottleneck games on series parallel graphs.

Theorem 3. *Let Γ be a bottleneck game with splittable traffic on a series parallel graph, let $\hat{\mathbf{x}}$ and $\mathbf{x}$ be two Wardrop equilibria for Γ. Then* $\mathsf{SC}(\Gamma, \hat{\mathbf{x}}) = \mathsf{SC}(\Gamma, \mathbf{x})$.

Proof. The proof is by contradiction. Assume that two different Wardrop equilibria $\hat{\mathbf{x}}$ and $\mathbf{x}$ for $\Gamma = (G, s, t, (f_e)_{e\in E}, r)$ are given such that $\mathsf{SC}(\Gamma, \hat{\mathbf{x}}) < \mathsf{SC}(\Gamma, \mathbf{x})$. Clearly, $l(\hat{\mathbf{x}}) < l(\mathbf{x})$. Let D be a strong cut with respect to Γ and $\mathbf{x}$. Consider first of all an edge $e \in D$ with $\delta_e(\hat{\mathbf{x}}) > 0$. Since $\hat{\mathbf{x}}$ is a Wardrop equilibrium and e is an edge of the strong cut D with respect to Γ and $\mathbf{x}$, we get that $f_e(\delta_e(\hat{\mathbf{x}})) \le l(\hat{\mathbf{x}}) < l(\mathbf{x}) \le f_e(\delta_e(\mathbf{x}))$, which implies $\delta_e(\hat{\mathbf{x}}) < \delta_e(\mathbf{x})$ because f_e is nondecreasing. If instead an edge $e \in D$ with $\delta_e(\hat{\mathbf{x}}) = 0$ is considered we trivially obtain that $\delta_e(\hat{\mathbf{x}}) \le \delta_e(\mathbf{x})$. Together we get that

$$r = \sum_{e\in D} \delta_e(\hat{\mathbf{x}}) \; < \; \sum_{e\in D} \delta_e(\mathbf{x}) = r,$$

which is a contradiction. □

We will now consider general graphs. If traffic is sent through a graph G then only edges that are on a simple path from s to t can be used. So the same equilibria are obtained when playing the game not on G but on the maximum subgraph of G containing only edges that are on a simple path from s to t, i.e., the subgraph induced by all paths from s to t. This idea is captured by the following definition.

Definition 2. *A directed multigraph $G = (V, E)$ without isolated vertices where $s, t \in V$, $s \neq t$, is called* strongly (s, t)-connected *if every edge $e \in E$ is contained in a simple path from s to t.*

Theorem 4. *Let G be a strongly (s, t)-connected graph that is not series parallel. Then there exists a bottleneck game $\Gamma = (G, s, t, (f_e)_{e \in E}, r)$ possessing Wardrop equilibria of different social cost.*

3.3 Price of Stability

In this section we will show that the price of stability for bottleneck games with splittable traffic and latency functions from an arbitrary non-empty set of nonnegative, continuous, and nondecreasing functions $\mathcal{F}$ is the same on general graphs as on parallel links, i.e., $\mathsf{PoS}(\mathscr{G}(\mathcal{F})) = \mathsf{PoS}(\mathscr{P}(\mathcal{F}))$. To do so, we will show that given a game Γ on a general graph with latency functions from $\mathcal{F}$ there exists a game Γ' on parallel links with latency functions from $\mathcal{F}$ and Wardrop equilibria $\mathbf{x}$ for Γ and $\hat{\mathbf{x}}$ for Γ' such that $\frac{\mathsf{SC}(\Gamma,\mathbf{x})}{\mathsf{OPT}(\Gamma)} \leq \frac{\mathsf{SC}(\Gamma',\hat{\mathbf{x}})}{\mathsf{OPT}(\Gamma')}$.

We assume that Cole et al. [8] proved a very similar result to establish their Theorem 4.6 (whose proof they had to omit due to lack of space). Since we need a rather technical formulation for our main result on the price of stability for games with M/M/1 latency functions, we give the following Theorem 5.

Theorem 5. *Let $\Gamma = (G, s, t, (f_e)_{e \in E}, r)$, $G = (V, E)$, be a bottleneck game with splittable traffic where $r < C(G, s, t, (f_e)_{e \in E})$. Then there exist*

- *a bottleneck game with splittable traffic on parallel links $\Gamma' = (G', s', t', (f'_e)_{e' \in E'}, r)$, $G' = (V', E')$, where $|E'| \leq |E|$ and for each $e' \in E'$ there is an edge $e \in E$ such that $f'_{e'} = f_e$ and*
- *Wardrop equilibria $\mathbf{x}$ for Γ and $\hat{\mathbf{x}}$ for Γ',*

such that $\frac{\mathsf{SC}(\Gamma,\mathbf{x})}{\mathsf{OPT}(\Gamma)} \leq \frac{\mathsf{SC}(\Gamma',\hat{\mathbf{x}})}{\mathsf{OPT}(\Gamma')}$.

Recall that in the case of parallel links bottleneck games do not differ from Wardrop games and hence the prices of stability coincide. This, together with Theorem 5 implies that the price of stability for bottleneck games on arbitrary graphs corresponds to the price of stability (or anarchy) for Wardrop games on parallel links. Consequently, the results by Roughgarden [19] on the price of anarchy for Wardrop games lead to the following corollary.

Corollary 1. *Let $\Gamma = (G, s, t, (f_e)_{e \in E}, r)$ be a bottleneck game with splittable traffic where all functions f_e, $e \in E$, are polynomials of degree at most d with non-negative coefficients. Then there exists a Wardrop equilibrium $\mathbf{x}$ where*

$$\frac{\mathsf{SC}(\Gamma, \mathbf{x})}{\mathsf{OPT}(\Gamma)} \leq \frac{(d+1) \cdot \sqrt[d]{d+1}}{(d+1) \cdot \sqrt[d]{d+1} - d} .$$

For readers who are familiar with the anarchy value defined in [19] we would like to mention that it is possible to draw a more general conclusion that is also the result of Cole et al. [8, Theorem 4.6]: If the anarchy value $\alpha(\mathcal{F})$ exists for a set of functions $\mathcal{F}$ this value $\alpha(\mathcal{F})$ is an upper bound on the price of stability for bottleneck games on general graphs and with latency functions from $\mathcal{F}$, i.e., $\mathsf{PoS}(\mathscr{G}(\mathcal{F})) \leq \alpha(\mathcal{F})$. Under some moderate assumptions being made on $\mathcal{F}$ even equality holds. This result, however, cannot be used to prove our main finding, i.e., the price of stability for bottleneck games with M/M/1 latency functions, since we will include other game properties in the sets of games under consideration. Therefore, Theorem 5 is essential for the generalization from parallel links to arbitrary graphs in the M/M/1 case.

4 Bottleneck Games with Splittable Traffic and M/M/1 Functions

In the rest of this paper we focus on bottleneck games with splittable traffic and M/M/1 latency functions. We want to remark that in this setting there are instances for which social cost of Wardrop equilibria may be arbitrarily worse than those of an optimum. This justifies looking at the price of stability instead. Unfortunately, also $\mathsf{PoS}(\mathscr{G}(\mathcal{M})) = \mathsf{PoS}(\mathscr{P}(\mathcal{M})) = \infty$, which will be a trivial consequence of Theorem 8. Hence, we need to consider other game properties, too, in order to get a meaningful result for the price of stability. To achieve this goal we will derive the exact value for $\mathsf{PoS}(\mathscr{P}(\mathcal{M}_{\geq c}, m, r))$ where $m \in \mathbb{N}$, $c > 0$, $r > 0$, and then argue that it is the same as $\mathsf{PoS}(\mathscr{G}(\mathcal{M}_{\geq c}, m, r))$. By our notation, m is meant here to denote the maximum number of edges, c the minimum edge capacity, and r the maximum amount of traffic.

4.1 Social Cost of Equilibria and Optimum Solutions in the Parallel Link Case

For our later proofs on the price of stability we need some insight into the social cost of Wardrop equilibria and optimum solutions. Thus we now give the exact social cost of Wardrop equilibria.

Theorem 6. *Let $\Gamma = (G, s, t, (c_e)_{e\in E}, r)$ be a bottleneck game with splittable traffic and M/M/1 latency functions on m parallel links where $r < C$, and let $\mathbf{x}$ be a Wardrop equilibrium. Furthermore, let $s = |\{i \in [m] \mid x_i > 0\}|$ denote the number of links used in $\mathbf{x}$. Then*

$$s = \max\left\{i \in [m] \mid r + i \cdot c_i > C^{\leq i}\right\} \quad \textit{and} \quad \mathsf{SC}(\Gamma, \mathbf{x}) = \frac{s \cdot r}{C^{\leq s} - r}.$$

We will now derive an expression that describes the social cost of an optimum solution.

Theorem 7. *Let $\Gamma = (G, s, t, (c_e)_{e\in E}, r)$ be a bottleneck game with splittable traffic and M/M/1 latency functions on m parallel links where $r < C$, and let $\mathbf{x}$ be a strategy profile with optimal social cost. Furthermore, let $t = |\{i \in [m] \mid x_i > 0\}|$ denote the number of links used in $\mathbf{x}$. Then:*

$$t = \max\left\{i \in [m] \mid r + \sqrt{c_i} \cdot \sum_{k=1}^{i} \sqrt{c_k} > C^{\leq i}\right\} \textit{ and } \mathsf{OPT}(\Gamma) = \frac{\left(\sum_{i=1}^{t} \sqrt{c_i}\right)^2}{C^{\leq t} - r} - t.$$

4.2 Price of Stability for Games on Parallel Links

Combining our knowledge about the social cost of Wardrop equilibria and optimum solutions, we will now give the exact price of stability for games on $m \in \mathbb{N}$ parallel links routing a traffic of $r > 0$. To that end, we require the capacities $c_1, \ldots, c_m$ and the traffic r to be normalized such that $c_m = 1$, i.e., we will derive an exact expression for $\mathsf{PoS}(\mathscr{P}(\mathcal{M}_{\geq 1}, m, r))$. This is not a restriction as for an $\alpha > 0$ the bijective mapping $\Gamma = (G, s, t, (c_e)_{e \in E}, r) \mapsto \Gamma_\alpha := (G, s, t, (\alpha \cdot c_e)_{e \in E}, \alpha \cdot r)$ associates both the Wardrop equilibrium and optimum in Γ with the respective equilibrium and optimum in Γ_α. Note that, if $(x_P)_{P \in \mathcal{P}_{st}}$ is a strategy profile in Γ, $(\alpha \cdot x_P)_{P \in \mathcal{P}_{st}}$ is a strategy profile in Γ_α with social costs α times as much as that of $(x_P)_{P \in \mathcal{P}_{st}}$ in Γ. Hence, $\mathsf{PoS}(\mathscr{P}(\mathcal{M}_{\geq c}, m, r)) = \mathsf{PoS}(\mathscr{P}(\mathcal{M}_{\geq 1}, m, \frac{r}{c}))$, where again $m \in \mathbb{N}$ denotes the maximum number of edges, $c > 0$ the minimum edge capacity, and $r > 0$ the maximum amount of traffic.

Theorem 8. *For bottleneck games $\Gamma = (G, s, t, (c_e)_{e \in E}, r)$ with splittable traffic and M/M/1 latency functions on m parallel links where $c_m = 1$ the price of stability is exactly*

$$\mathsf{PoS}(\mathscr{P}(\mathcal{M}_{\geq 1}, m, r)) = \frac{m \cdot r}{r + 2 \cdot (m-1) \cdot (\sqrt{r+1} - 1)} =: \Psi(m, r).$$

Note the following properties of $\Psi(m, r)$:

- $\Psi(m, r)$ is strictly increasing in both $m \in \mathbb{N}$ and $r > 0$. To see the latter it can similarly be shown with standard methods that $\frac{\partial}{\partial r}\Psi(m, r) > 0$.
- $\lim_{m \to \infty} \Psi(m, r) = \frac{r}{2\sqrt{r+1}+2}$ and $\lim_{r \to \infty} \Psi(m, r) = m$, hence we always have the bound $\Psi(m, r) \leq m$.

We conclude this section by the following corollary which is a direct consequence of the preceding result together with Theorem 5.

Corollary 2. *The price of stability for general bottleneck games with M/M/1 latency functions on a graph with no more than $m \in \mathbb{N}$ edges each with a minimum capacity of at least $c > 0$ and traffic at most $r > 0$ is the same as in the parallel links case, i.e.,* $\mathsf{PoS}(\mathscr{G}(\mathcal{M}_{\geq c}, m, r)) = \mathsf{PoS}(\mathscr{P}(\mathcal{M}_{\geq c}, m, r))$.

Acknowledgment. We would like to thank Marios Mavronicolas for many helpful comments.

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
2. E. Anshelevich, A. Dasgupta, J. Kleinberg, T. Roughgarden, É. Tardos, and T. Wexler. The Price of Stability for Network Design with Fair Cost Allocation. In *Proc. of the 45th Annual Symposium on Foundations of Computer Science (FOCS'04)*, 2004.
3. R. Banner and A. Orda. Bottleneck Routing Games in Communication Networks. In *Proc. of the 25th Conference on Computer Communications (INFOCOM'06)*, to appear 2006.

4. M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
5. D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1987.
6. C. Busch and M. Magdon-Ismail. Atomic Routing Games on Maximum Congestion. In *Proc. of the 2nd International Conference on Algorithmic Aspects in Information and Management (AAIM'06)*, to appear 2006.
7. I. Caragiannis, C. Galdi, and C. Kaklamanis. Network Load Games. In X. Deng and D. Du, editors, *Proc. of the 16th Annual International Symposium on Algorithms and Computation (ISAAC'05)*, LNCS Vol. 3827, Springer Verlag, pages 809–818, 2005.
8. R. Cole, Y. Dodis, and T. Roughgarden. Bottleneck Links, Variable Demand, and the Tragedy of the Commons. In *Proc. of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*, pages 668–677, 2006.
9. J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. Computational Complexity, Fairness, and the Price of Anarchy of the Maximum Latency Problem. In D. Bienstock and G. Nemhauser, editors, *Proc. of the 10th Conference on Integer Programming and Combinatorial Optimization (IPCO'04)*, LNCS Vol. 3064, Springer Verlag, pages 59–73, 2004. An extended version will appear in *Operations Research*.
10. A. Czumaj, P. Krysta, and B. Vöcking. Selfish Traffic Allocation for Server Farms. In *Proc. of the 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 287–296, 2002.
11. M. Gairing, T. Lücking, B. Monien, and K. Tiemann. Nash Equilibria, the Price of Anarchy and the Fully Mixed Nash Equilibrium Conjecture. In *Proc. of the 32nd International Colloquium on Automata, Languages, and Programming (ICALP'05)*, LNCS Vol. 3580, Springer Verlag, pages 51–65, 2005.
12. L. Kleinrock. *Queueing Systems, Volume 1: Theory*. Wiley-Interscience, 1975.
13. Y. A. Korilis, A. A. Lazar, and A. Orda. Capacity Allocation Under Noncooperative Routing. *IEEE Transactions on Automatic Control*, 42(3):309–325, 1997.
14. E. Koutsoupias and C. H. Papadimitriou. Worst-Case Equilibria. In C. Meinel and S. Tison, editors, *Proc. of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS'99)*, LNCS Vol. 1563, Springer Verlag, pages 404–413, 1999.
15. F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet Routing and Job-Scheduling in O(Congestion + Dilation) steps. *Combinatroica*, 14(2):167–186, 1994.
16. A. Orda, R. Rom, and N. Shimkin. Competitive Routing in Multiuser Communication Networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.
17. K. P. Rath. A direct proof of the existence of pure strategy equilibria in games with a continuum of players. *Economic Theory*, 2(3):427–433, 1992.
18. T. G. Robertazzi. *Computer Networks and Systems: Queuing Theory and Performance Evaluation*. Springer-Verlag, 2000.
19. T. Roughgarden. The Price of Anarchy is Independent of the Network Topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.
20. T. Roughgarden. The Maximum Latency of Selfish Routing. In *Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'04)*, pages 980–981, 2004.
21. T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
22. T. Roughgarden and É. Tardos. How Bad Is Selfish Routing? *Journal of the ACM*, 49(2):236–259, 2002.
23. D. Schmeidler. Equilibrium points of nonatomic games. *Journal of Statistical Physics*, 7(4):295–300, 1973.
24. M. Schwarz. *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison-Wesley, 1987.
25. J. G. Wardrop. Some Theoretical Aspects of Road Traffic Research. In *Proc. of the Institute of Civil Engineers, Pt. II, Vol. 1*, pages 325–378, 1952.

A Worm Propagation Model Based on People's Email Acquaintance Profiles*

T. Komninos[3], Y.C. Stamatiou[2,3], and G. Vavitsas[1,3]

[1] University of Patras, Department of Computer Engineering, 26500, Rio, Patras, Greece

[2] Mathematics Department, 451 10, Ioannina, Greece
istamat@cc.uoi.gr

[3] Research and Academic Computer Technology Institute, N. Kazantzaki, University of Patras, 26500, Rio, Patras, Greece
komninos@cti.gr, istamat@cc.uoi.gr, vavitsas@ceid.upatras.gr

Abstract. One frequently employed way of propagation exploited by worms is through the victim's contact book. The contact book, which reflects the acquaintance profiles of people, is used as a "hit-list", to which the worm can send itself in order to spread fast. In this paper we propose a discrete worm propagation model that relies upon a combined email and Instant Messaging (IM) communication behaviour of users. We also model user reaction against infected email as well as the rate at which antivirus software is installed. User acquaintance is perceived as a "network" connecting users based on their contact book links. We then propose a worm propagation formulation based on a token propagation algorithm, further analyzed with a use of a system of continuous differential equations, as dictated by Wormald's theorem on approximating "well-behaving" random processes with deterministic functions.

1 Introduction

A *worm* is a self-contained malicious code that is able to spread itself in computer networks. Propagation, usually, occurs through the exploitation of network connections, shared storage, email, Instant Messengers or Peer to Peer (P2P) file sharing networks. Simple Mail Transfer Protocol (SMTP), for instance, is one of the most common malicious code propagation vehicles. To spread by email, a worm can propagate as an email attachment or embed itself into html code within the email body. Then it obtains email addresses from the victim's computer in order to propagate. Worm propagation modelling has attracted the attention through a series of incidents such as the CodeRed [14] worm, Nimda [2] worm, Slammer worm [9], Sobig [3], W32/Bagle and W32/Novarg [1], Sober. X, Netsky. P and Mytob. ED [11].

* Partially supported by the IST Programme of the European Union under contact number IST-2005-15964 (AEOLUS) and the INTAS Programme under contract with Ref. No 04-77-7173 (Data Flow Systems: Algorithms and Complexity (DFS-AC)).

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 343–352, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Recently, worms have appeared that are able to propagate using another social-like popular communication method such as Instant Messengers (IM) or Peer-to-Peer (P2P) file sharing networks [5]. IM networks provide the ability not only to transfer text messages, but also files supporting peer-to-peer file sharing, leading to the immediate spread of files that are infected. Worms use social engineering to trick people into downloading and execute malicious code [4]. Using IM, worms spread faster as locating potential victims does not require scanning attempts to possibly unknown or unused IP addresses. What they need is simply online users' contact list. However, there were some IM worms which have exploited the processing vulnerabilities described in [8] to allow automatic execution of code. This worms are much faster than any other that requires user intervention and, thus, causes significant devastation. As more users adopt IM services, new worms will spread combining different propagation vectors, not only using email but also IM and P2P links.

While many researchers deal with the development of new techniques for the detection and elimination of worms, there seems to be, relatively, little activity in the theoretical modelling of viral code replication and propagation. Less research effort has been expended on modelling worms that use IM and email simultaneously. In [12] Wang *et al.* study a worm propagation model based on a clustered and a tree-like hierarchic topologies. In their model, copies of the worm propagate at a constant rate without needing user interactions. The lack of a user model coupled with the clustered and tree-like topologies make it unsuitable for modelling the propagation of email and IM worms/viruses over the Internet. Zou et al. studied Code Red worm propagation based on the classical epidemic Kermack-Mckendrick model [14]. Newman et al. derived the analytical solution of the percolation threshold of small world topology [6,12]. Albert et al. were the first to explain the vulnerability of power law networks under attacks [9]. The authors conclude that the power law topology is vulnerable under deliberate attack. Wang, Knight *et al.* study the effect of immunization on worm propagation [12]. They compare the effect of random immunization and selective immunization. They show that immunizing nodes with highest degrees has better effect than random immunization. This is different from reality where the immunization is randomly applied to hosts by users or administrators.

In [14] Zou *et al.* an email model is given as an undirected graph of relationships between people. It is assumed that each user opens an incoming worm attachment with a certain probability, depending on the user and not on time. This, however, does not describe well the typical user behaviour. Indeed, as the new worm starts spreading there is no user alertness, who tend to open the contaminated attached file. As news about the worm are circulated, users become more cautious. Thus users' behaviour should depend on time. The authors consider a "reinfection" model, where a user sends out copies of the worm each time an infected attachment is reopened, but this does not add to the infected population as long as the host is either already infected or it has been immunized by an antivirus. An interesting conclusion can be drawn from this study: the overall spread rate of worms increases as the variability of users' email checking times increases. Thus a worm is more vicious as a better social engineering technique is applied. Mannan and van Oorschot [7] review selected IM worms and summarize their main characteristics, motivating a brief overview of the network formed by IM contact lists, and a discussion of theoretical consequences of worms in such networks.

In our work, we view email, IM and P2P networks as forming a kind of "social" network. These networks can be macroscopically considered as an interconnection of a number of Autonomous Systems (AS). An AS is a subnetwork (usually a Domain Network) administered by a single authority. In this paper, we describe a worm spread model based on the social structures induced by users' communication habits. The model incorporates users' behaviour by including the probability of a user opening the message. We also propose a more realistic model of the progressive immunization of systems using the probability that an updated antivirus software is installed on a host. We finally model the acquaintance network using a formalism stemming from the domain of Constraint Satisfaction Problems (CSP). Using this model we can determine the impact of a worm spreading without having proper antivirus or informed users.

2 Acquaintance Networks: Motivation and Formalism

An *acquaintance network* consists of several hypernodes, where each hypernode represents a specific *domain* or LAN (e.g. a university or a company network). Each hypernode contains several nodes which represent personal computers or users' contact information (e.g. email or IM addresses). We assume that with probability p_{intacq} a node of a hypernode contains in its contact book the contact address of another node of the same hypernode. Also, with p_{extacq} a node of a hypernode is associated with a node (user) in a different hypernode. Finally, with probability p_{hyper} we consider that there is a connection between two hypernodes (which means that at least one user of one hypernode is associated with at least one user of the other hypernode). The connections between hypernodes forms the *network acquaintance graph* while the connections between nodes forms the *person acquaintance graph*. Our focus is on modelling a worm outbreak which starts at some random set of nodes and propagates along the acquaintance links.

More formally, an *acquaintance network* consists of a set of hypernodes $X_1, \ldots, X_n$ containing node sets $D_1, \ldots, D_n$ respectively, and a set of acquaintance relations $\mathcal{C}$. An edge $R_{i_1,i_2} \in \mathcal{C}$ is a subset of $D_{i_1} \times D_{i_2}$, with i_1, i_2 distinct. We say that R_{i_1,i_2} bounds hypernodes nodes X_{i_1}, X_{i_2} to mutual acquaintance, because of mutual acquaintances stemming from the nodes they contain. The *person acquaintance hypergraph* of a network acquaintance graph is an n-partite graph. Its ith part corresponds to hypernode X_i and it has exactly $|D_i|$ vertices, one for each node in D_i. There exists an eedge $\{v_{i_1}, v_{i_2}\}$ if and only if the corresponding nodes $d_{i_1} \in D_{i_1}$ and $d_{i_2} \in D_{i_2}$ belong to some acquaintance relation that bounds the corresponding variables.

Email, IM and P2P contacts form a kind of social network. Modelling these networks as graphs, with each node representing a host, is clearly unfeasible. On the other hand, they can be macroscopically considered as an interconnection of a number of Autonomous Systems (AS). An AS is a subnetwork usually a Domain Network which is administered by a single authority. For this reason we propose a hypernode based model with a hyper-node representing a Domain Network.

According to the above formalism (which, actually, stems from the formalism of the Constraint Satisfaction Problem (CSP)), the network acquaintance graph represents the structure of email/IM acquaintances across network domains (or LANs). The set of hypernodes $X_1, \ldots, X_n$ represent *Autonomous Systems (AS) or Domains* (e.g. universities) of the acquaintance network, while the node sets $D_1, \ldots, D_n$ represent distinct contact addresses of each domain. These addresses comprise an email or IM acquaintances network. We can safely assume, without loss of generality, that every distinct email or IM address is associated with one host computer which is associated with a single user. The connections between the hypernodes form the *network acquaintance graph*, while the connections among the nodes of each distinct email and IM contact address, form the *person acquaintance graph*. Also, in our model the quantity $B(i)$ represents the number of the infected nodes at step i while $W(i)$ represents the number of immunized nodes, i.e. nodes on which updated antivirus software is already installed. The quantity $nd - B(i) - W(i)$ represents the number of *susceptible* nodes, that is the number of nodes that have no defence against the new worm.

We will now define some probabilities related to our model: p_{hyper} is the probability that two hypernodes (AS-Domains) are connected. Then p_{extacq} is the probability with which a user (or a node) has a contact with a specific user of another hypernode. Also, p_{intacq} is the probability that a node has a contact with a specific node belonging in same hypernode. Moreover, p_{antv} represents the percentage of the nodes protected by updated antivirus software. We can model this probability using a function of time and network size that is gradually increasing with time and decreasing with the size of the network. In particular we set $p_{\text{antiv}}(n, t) = \frac{g(t)}{n}$ where $g(t)$ is a monotonicaly increasing function of t and n is the network size which is the number of hypernodes in our model, and p_{openm} is the probability that a user opens his/her email or IM message. We also model this probability as a function of time, $p_{\text{openm}}(t) = f(t)$, which is monotonicaly decreasing with time, since as time passes and information of the worm outbreak circulates people are more cautious in opening suspicious email messages. The pair $I = (p_{\text{antv}}, p_{\text{openm}})$ is called an *attack reaction pair* since it characterizes a new worm that has started propagating within a network.

3 Randomly Generated Acquaintance Graphs

According to the notation given in the previous section, an acquaintance graph is defined by the following four parameters: (a) The number of hypernodes n, (b) the size of each hypernode d, (c) The network acquaintance graph, and (d) The person acquaintance graph. We define the following process that generates random acquaintance graph instances, for given n and d: Construct the network acquaintance graph by having each of the possible $\binom{n}{2}$ edges selected uniformly and independently with probability p_{hyper}. Then construct the person acquaintance graph by having each of the possible d^2 edges that may exist between two hypernodes that are adjacent in the network acquaintance graph selected uniformly and independently with probability p_{extacq} and by having each of the $\binom{d}{2}$ edges that may exist between two nodes of the same hypernode selected uniformly and independently with probability p_{intacq}. If no edges are introduced we repeat the edge formation process. We will denote by $G(p_{\text{hyper}}, p_{\text{extacq}}, p_{\text{intacq}}, d, n)$ be the random acquaintance network generated according to the process described above.

4 Virus Propagation Model

We assume that a worm spreads itself by attaching its malicious code to an email, a file transferred or a URL to an infected link and sending it to all contact addresses it finds on a users' computer. IM contact lists enable users to track the presence status of their contacts. To a worm, an online contact list provides an instant hit-list. Note that most email clients provide an address book which does not reveal any online status of the users thus the propagation is slowed down by the time the user interacts opening his email. A host is infected when the user opens the attached or transferred file or when the client previews it or the exploited vulnerability makes it to execute automatically. According to the theory above we will now refer to the model that the worm uses in order to propagate between the hypernodes and the nodes accordingly to the existing connections. We assume that a worm randomly infects a node v of a hypernode. By exploiting the address book of the node the worm starts to propagate. Initially, all the nodes are susceptible to infection. At step zero a randomly chosen set of nodes becomes infected. Then the infection spreads as follows: at every infection cycle i the nodes that are infected turn into black and start to infect other susceptible nodes by sending infected messages to the hit-list they have. The messages that are been sent by the infected nodes follow the edges of the person acquaintance graph. We assume that all generated messages, infectious or not, are send sequentially, as IM and email servers are receiving and sending messages sequentially. Step i is completed after the ith message is ready to be dispatched. Every user that receives an infected message opens this message with probability p_{openm}. The user's computer becomes infected if there is no updated antivirus program installed at the computer. A susceptible or infected (black) node becomes white (immunized) with probability p_{antv} if an updated antivirus program is installed at the node.

5 Theoretical Analysis and Model Evaluation

We will now analyze theoretically the proposed worm propagation model by applying the differential equations method (for details see [13]). This theorem says is that if we have a number of co-evolving discrete random variables (associated with some discrete random process) that satisfy a Lipschitz condition and their expected fluctuation at each time step is known, then the value of these variables at each time step can be approximated using the solution of a system of differential equations. Furthermore, the system of differential equations results directly from the expressions for the expected fluctuation of the random variables describing the random process.

The process we will analyze in what follows involves two jointly evolving random variables: $B(i)$, the number of black nodes at step i of the worm spread process, and $W(i)$, the number of immune nodes at step i of the process. According to our discussion in Section 4, step i is completed after the ith message (according to some global ordering) is ready to be dispatched from one node (i.e. personal computer) belonging to some hypernode (a domain). For our model, the following holds:

Theorem 1. *Let $G(p_{hyper}, p_{extacq}, p_{intacq}, d, n)$ be an acquaintance graph and $I(p_{antv}, p_{openm})$ be an attack reaction pair. Assume a transition from step i to step $i+1$ with the dispatch of an email from the lth hypernode. Let $W_l(i)$ and $B_l(i)$ be the number of immune and infected nodes respectively within this hypernode. Then the expected increase in the random variables $B(i)$ and $W(i)$ in a transition from step i to step $i+1$ is the following:*

$$\begin{aligned}\mathbf{E}[B(i+1) - B(i)] &= \frac{B(i)}{dn} p_{hyper} p_{extacq} p_{openm} \\ &\quad \cdot [dn - W(i) - B(i) - (d - W_l(i) - B_l(i))] \\ &\quad + p_{intacq} p_{openm} \frac{B(i)}{dn} \cdot [d - W_l(i) - B_l(i)] \\ \mathbf{E}[W(i+1) - W(i)] &= p_{antv}[dn - W(i)].\end{aligned}$$

Proof. We will first consider the expected fluctuation of the number of black nodes. Let $B(i)$ be the number of black nodes at step i and l the hypernode in which the currently considered mail-dispatching node u belongs. Let v be one of the $dn - W(i) - B(i) - (d - W_l(i) - B_l(i))$ susceptible nodes which do not belong to this hypernode. Then v can become black due to the emailing activity of u if the following, independent, events hold simultaneously: (1) There is a network acquaintance edge between the hypernodes in which u and v belong. This event holds with probability p_{hyper}. (2) There is a person acquaintance edge between u and v. This event holds with probability p_{extacq}. (3) Node u is black. This event holds with probability $\frac{B(i)}{dn}$ assuming a uniform distribution of black nodes among all the dn network nodes. (4) Node v eventually opens the email, when it arrives at v's mailbox. This holds with probability p_{opemn}. Thus, the expected number of black nodes produced by nodes not belonging in the same hypernode as u, is equal to

$$p_{\mathrm{hyper}} p_{\mathrm{extacq}} \frac{B(i)}{dn} p_{\mathrm{openm}} \cdot [dn - W(i) - B(i) - (d - W_l(i) - B_l(i))]. \quad (1)$$

Similarly, for the $d - W_l(i) - B_l(i)$ susceptible nodes that belong to the same hypernode as u, we have an expected contribution to $B(i)$ equal to

$$p_{\text{intacq}}p_{\text{openm}}\frac{B(i)}{dn}\cdot[d-W_l(i)-B_l(i)]. \tag{2}$$

Adding (1) and (2) we obtain the expression for $\mathbf{E}[B(i+1)-B(i)]$.

For the white (i.e. immune) nodes we simply observe that at each step, a new white node is produced which was either black or not infected previously (there are $nd-W(i)$ such nodes) with probability p_{antiv} which is the probability of installing one antivirus package at the ith step. Thus, $\mathbf{E}[W(i+1)-W(i)] = p_{\text{antiv}}[dn-W(i)]$ completing the proof. □

In computing the probability that a node is black at step i we tacitly assumed a uniform spread of the infection throughout the network. This allows as to set this probability equal to $\frac{B(i)}{dn}$. Although this may not true in general (especially in the time instances close to the beginning of the infection), it can be a close approximation in the "long run" of the infection process. Based on this assumption, applied within each hypernode, we can approximate $W_l(i)$ and $B_l(i)$ as follows: $W_l(i) = \frac{W(i)}{n}, B_l(i) = \frac{B(i)}{n}$. Thus, we obtain the following corollary, from Theorem 1:

Corollary 1 (assuming uniform spread of the worm).

$$\begin{aligned}
\mathbf{E}[B(i+1)-B(i)] &= \frac{B(i)}{dn}p_{hyper}p_{extacq}p_{openm}\\
&\cdot\ [dn-W(i)-B(i)-(d-\frac{W(i)}{n}-\frac{B(i)}{n})]\\
&+p_{intacq}p_{openm}\frac{B(i)}{dn}\cdot[d-\frac{W(i)}{n}-\frac{B(i)}{n}]\\
\mathbf{E}[W(i+1)-W(i)] &= p_{antv}[dn-W(i)].
\end{aligned}$$

We will now obtain the system of differential equations.

Theorem 2. *Let $p_{hyper}=\frac{c}{n}$, with c a constant, p_{extacq} a constant independent of n and t, $p_{antiv}(n,t)=\frac{g(t)}{n}$, p_{intacq} a constant independent of n,t, and $p_{openm}(t)=f(t)$. Then the system of differential equations that results from the application of Wormald's theorem to the evolution of the random variables $B(i)$ and $W(i)$, with observation window $T=rdn$, is the following (in the limit, letting $n\to\infty$):*

$$\begin{aligned}
\frac{db(t)}{dt} &= [b(t)cp_{extacq}f(t)d+p_{intacq}f(t)b(t)d][1-w(t)-b(t)]\\
\frac{dw(t)}{dt} &= g(t)d[1-w(t)].
\end{aligned}$$

Proof. We will use the equations for the expected fluctuations in $W(i)$ and $B(i)$ as given in Corollary 1. Then, we scale the time steps by T (we divide i by T) and the quantities $W(i)$ and $B(i)$ by dn (we divide by dn), replacing the expectations on the left hand-side of the equations in Corollary 1 by first derivatives. The differential equations given in the statement of the theorem are then obtained by simple algebraic manipulations, replacing the factor $\frac{1}{n}(d - \frac{W(i)}{n} - \frac{B(i)}{n})$ that appears with 0, as $n \to \infty$, since $d = O(1), W(i) = O(n)$ and $B(i) = O(n)$. □

The differential equation for $w(t)$ is easy to solve and replacing $w(t)$ in the differential equation for $b(t)$ with this solution gives us a differential equation of Bernoulli type. This is a special type of the Ricatti differential equation and can be solved using for example the method outlined in [10].

Theorem 3. *Let* $p_{hyper} = \frac{c}{n}$*, with* c *a constant,* p_{extacq} *a constant independent of* n *and* t*,* $p_{antiv}(n,t) = \frac{g(t)}{n}$*,* p_{intacq} *a constant independent of* n, t*, and* $p_{openm}(t) = f(t)$*. Let, also,*

$$h = [c\, p_{extacq}\, w(0) - p_{extacq}\, c + p_{intacq}\, w(0) - p_{intacq}]$$
$$u(x) = e^{\left[-hd \int_0^x \mathrm{f}(y)\, e^{\left(-d \int_0^y \mathrm{g}(z)\, dz\right)} dx\right]}. \tag{3}$$

Then the solution to the system of differential equations given in Theorem 2 is the following:

$$w(t) = \exp\left[d \int_0^t g(z)dz + w(0) - 1\right] \exp\left[-d \int_0^t g(z)dz\right]$$

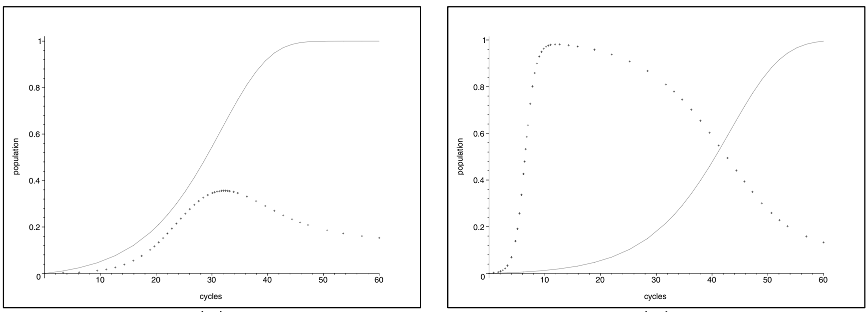
$$\mathrm{b}(t) = b(0) \frac{u(t)}{b(0)d(p_{extacq}\, c + p_{intacq}) \int_0^t u(s)\mathrm{f}(s)ds + 1}.$$

We will now plot these solutions, for various values of the parameters, in order to see the interaction between the numbers of black and white nodes. In our model, the main parameters that affect this interaction are p_{antv} and p_{openm}. In addition, as we have already argued above, the probabilities p_{antv} and p_{openm} should depend on the time parameter. In particular, we have set $p_{\mathrm{antv}} = \frac{1}{1+ae^{(-\beta t+\gamma)}}$ and $p_{\mathrm{openm}} = \frac{0.9}{\delta+e^{(+\zeta t-\theta)}}$, where $a, \beta, \gamma, \delta, \zeta$ and θ are constants. The chosen function for p_{antv} is, initially, monotonicaly increasing with a small rate while afterwards it increases at a faster rate. This has the interpretation that after a new worm has been analyzed, as time goes by, more people start downloading and installing defense software against it while at first only few antivirus installations take place. The chosen function for p_{openm} has the opposite behaviour. At first, this function is monotonicaly decreasing at a slow rate, reflecting the fact that people tend to open emails without a second thought. Then the function is decreasing with a faster rate reflecting the fact that information about the worm becomes available and people become more cautious with opening their email.

In the figures that follow, we plot the percentage of black and white nodes, as a function of time, for two values of d (hypernode or local network size). We can see the effect that p_{antv} and p_{openm} have to the relative sizes of the populations of white and black nodes for the two values of d, all other parameters being fixed. First we observe that the effect of the worm in Figures 2.(a) and 2.(b) is more severe than in the Figures 1.(a) and 1.(b). This is due to the fact that in the former figures $d = 20$ while in the latter $d = 100$. This shows that the infection within the hypernode with 100 nodes can create fast multiple infections outside the hypernode due to the spread of the worm to many of these 100 nodes which, in turn, spread the worm to more nodes outside the hypernode.

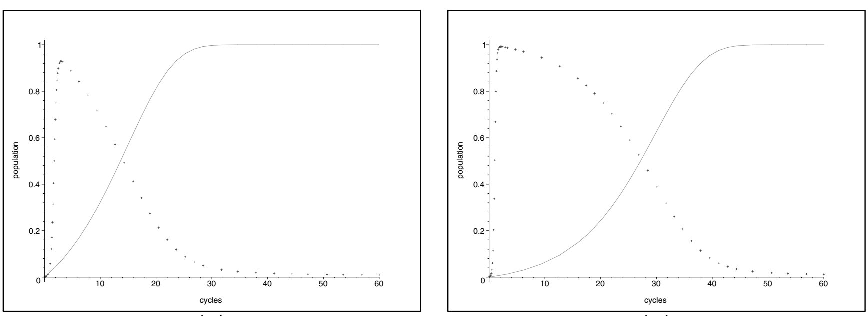
With regard to the effect of the antivirus installation rate as well as the users' opening mail easiness, in Figure 1.(a) we have higher installation rate and less easiness, in comparison with Figure 1.(b). Thus, the spread in Figure 1.(a) is less devastating than it is in Figure 1.(b). This is also true for Figures 2.(a) and 2.(b) with the only difference the value for d.

One can also tune the other parameters of the model and, thus, construct "what-if" scenarios for various worm propagation patterns and and acquaintance network structures.



(a) (b)

Fig. 1. (a) d=20, large p_{antv}, small p_{openm}, (b) d=20, small p_{antv}, large p_{openm} (black nodes dotted, white nodes continuous)



(a) (b)

Fig. 2. (a) d=100, large p_{antv}, small p_{openm}, (b) d=100, small p_{antv}, large p_{openm} (black nodes dotted, white nodes continuous)

6 Conclusions

In this paper we have proposed a model for users' acquaintance profiles, as they result from their email address books or their IM communication habits. This model can be used for the study of worm propagation as a function of the antivirus installation rate as well as users' easiness in opening their email attachments. We showed that the theoretical analysis of this model leads to a system of differential equations that result from the application of Wormald's theorem to the analysis of the expected fluctuations of infected as well as immunized nodes. These equations can be analytically solved, offering a practical means of conducting "what-if" scenarios by tuning the parameters of the model. We believe that our model can be used as a basis for extensions by including other factors which may affect virus propagation (e.g. network link speed) having, at the same time, a straightforward theoretical analysis with the aid of Wormald's theorem.

References

1. CERT advisory CA-2004-02.
2. CERT advisory CA-2001-26 Nimda Worm.
3. CERT incident note IN-2003-03.
4. A. Gostev, "Malware Evolution: January - March 2005," Kaspersky Lab Report 4.
5. IMlogic Threat Center, Symantec Corporation. `http://www.imlogic.com/im_threat_center/index.asp`
6. J.O. Kephart and S.R. White, "Measuring and Modeling Computer Virus Prevalence," in *Proc. 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, California, 1993.
7. M. Mannan and P. Oorschot,"On Instant Messaging Worms, Analysis and Countermeasures," in *Proc. of the 2005 ACM workshop on Rapid malcode (WORM'05).*
8. Microsoft, "How to update your computer with the JPEG processing (GDI+) security update". `http://www.microsoft.com/athome/security/update/bulletins/200409_jpeg_tool.mspx`
9. D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE security and privacy*, **1(4)**,33–39, July 2003.
10. G.M. Murphy, *Ordinary Differential Equations and their Solutions*, D. Van Nostrand Company Inc., 1960.
11. Symantec Internet Security Threat Report Trends for January 05-December 05, Volume VIII and IX, 2005.
12. C. Wang, J. Knight, and M. Elder, "On computer viral infection and the effect of immunization," in *Proc. of the 16th annual computer security applications conference (ACSAC 00)* , New Orleans, LA, Dec. 2000.
13. N.C. Wormald, "The differential equation method for random graph processes and greedy algorithms," *Ann. Appl. Probab.* **5**, 1217–1235, 1995.
14. C.C Zou, W. Gong, and D. Towsley,"Code-red worm propagation modeling and analysis.," in *Proc. of the 9th ACM conference on Computer and Communications Security*, ACM Press, pp. 138–147, 2002.

Mixed Strategies in Combinatorial Agency
(Extended Abstract)*

Moshe Babaioff[1], Michal Feldman[2], and Noam Nisan[2]

[1] UC Berkeley School of Information
moshe@sims.berkeley.edu
[2] School of Computer Science, Hebrew University of Jerusalem
mfeldman@cs.huji.ac.il, noam@cs.huji.ac.il

Abstract. We study a setting where a principal needs to motivate a team of agents whose combination of hidden efforts stochastically determines an outcome. In a companion paper we devise and study a basic "combinatorial agency" model for this setting, where the principal is restricted to inducing a pure Nash equilibrium. Here, we show that the principal may possibly gain from inducing a mixed equilibrium, but this gain can be bounded for various families of technologies (in particular if a technology has symmetric combinatorial structure). In addition, we present a sufficient condition under which mixed strategies yield no gain to the principal.

1 Introduction

1.1 Background: Combinatorial Agency

The well studied principal-agent problem deals with how a "principal" can motivate a rational "agent" to exert costly effort towards the welfare of the principal. The difficulty in this model is that the agent's action (i.e. whether he exerts effort or not) is invisible to the principal and only the final outcome, which is probabilistic and also influenced by other factors, is visible. "Invisible" here is meant in a wide sense that includes "not precisely measurable", "costly to determine", or "non-contractible" (meaning that it can not be upheld in "a court of law"). This problem is well studied in many contexts in classical economic theory and we refer the readers to introductory texts on economic theory such as [7] Chapter 14. The solution is based on the observation that a properly designed contract, in which the payments are contingent upon the final outcome, can influence a rational agent to exert the required effort.

In [2] we initiated a general study of handling *combinations* of agents rather than a single agent. While much work was previously done on motivating teams of agents [5,9,6,3], our emphasis is on dealing with the complex combinatorial structure of dependencies between agents' actions. In the general case, each

* A full version of this paper is available at the authors' web sites. All proofs and some additional results appear in the full version.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 353–364, 2006.
© Springer-Verlag Berlin Heidelberg 2006

combination of efforts exerted by the n different agents may result in a different expected gain for the principal. The general question asks, given an exact specification of the expected utility of the principal for each combination of agents' actions, which conditional payments should the principal offer to which agents as to maximize his net utility? We view this problem of hidden actions in computational settings as a complementary problem to the problem of hidden information that is the heart of the field of Algorithmic Mechanism Design [8]. An example that was discussed in [4] is Quality of Service routing in a network: every intermediate link or router may exert a different amount of "effort" (priority, bandwidth, ...) when attempting to forward a packet of information. While the final outcome of whether a packet reached its destination is clearly visible, it is rarely feasible to monitor the exact amount of effort exerted by each intermediate link – how can we ensure that they really do exert the appropriate amount of effort? For example, in Internet routing, IP routers may delay or drop packets, and in mobile ad hoc networks, devices may strategically drop packets to conserve their constrained energy resources.

In the general model presented in [2], each of n agents has a set of possible *actions*, the combination of actions by the players results in some *outcome*, where this happens probabilistically. The main part of the specification of a problem in this model is a function ("the technology") that specifies this distribution for each n-tuple of agents' actions. Additionally, the problem specifies the principal's utility for each possible outcome, and for each agent, the agent's cost for each possible action. The principal motivates the agents by offering to each of them a *contract* that specifies a payment for each possible outcome of the whole project. Key here is that the actions of the players are non-observable ("hidden-actions") and thus the contract cannot make the payments directly contingent on the actions of the players, but rather only on the outcome of the whole project.

Given a set of contracts, each agent optimizes his own utility; i.e., chooses the action that maximizes his expected payment minus the cost of the action. Since the outcome depends on the actions of all players together, the agents are put in a game here and are assumed to reach a Nash Equilibrium (NE). The principal's problem is that of designing the *optimal contract*: i.e. the vector of contracts to the different agents that induce an equilibrium that will optimize his expected utility from the outcome minus his expected total payment. The main difficulty is that of determining the required Nash equilibrium point.

Our interest in this paper (as in [2]), is focused on the binary case: each agent has only two possible actions "exert effort" and "shirk" and there are only two possible outcomes "success" and "failure". Our motivating examples comes from the following more restricted and concrete "structured" subclass of problem instances: Every agent i performs a subtask which succeeds with a low probability γ_i if the agent does not exert effort and with a higher probability $\delta_i > \gamma_i$, if the agent does exert effort. The whole project succeeds as a deterministic Boolean function of the success of the subtasks.

1.2 This Paper: Mixed Equilibria

In [2] we studied the notion of Nash-equilibrium in pure strategies: we did not allow the principal to attempt inducing an equilibrium where agents have mixed strategies over their actions. In the observable-actions case (where the principal can condition the payments on the agents' individual actions) the restriction to pure strategies is without loss of generality: mixed actions can never help since they simply provide a convex combination of what would be obtained by pure actions. Yet, surprisingly, we show this is not the case for the hidden-actions case which we are studying: in some cases, a Mixed-Nash equilibrium can provide better expected utility to the principal than what he can obtain by equilibrium in pure strategies. In particular, this already happens for the "OR" function with two players, with a certain (quite restricted) range of parameters (see Section 3).

Our main goal is to quantify the principal's gain from inducing mixed equilibrium, rather then pure. To do that, we analyze the worst ratio (over all principal's values) between the principal's optimal utility with mixed equilibrium, and his optimal utility with pure equilibrium. We term this ratio "the price of purity" (POP) of the instance under study. We prove that for a class of instances, those with "increasing returns to scale", which contains in particular the AND function, the price of purity is trivial (i.e., $POP = 1$). Moreover, we show that for any other Boolean function, there is an assignment of the parameters (agents' individual success probabilities) for which the obtained structured technology has non trivial POP (i.e., $POP > 1$). (Section 4).

While the price of purity may be strictly greater than 1, we obtain quite a large number of results bounding this ratio (Section 5). These bounds range from very weak ones (e.g., $POP \leq n$ for any anonymous or DRS technology) to better ones for restricted cases (e.g., $POP \leq 1.154...$ for a family of anonymous OR technologies, and $POP \leq 2$ for any technology with 2 agents). We conjecture that there exists a universal constant C that bounds the POP for any technology, thus shrinking the large gaps between our conjecture and the obtained bounds is the main open problem of this paper.

Conjecture 1. There exists a constant $C > 1$, such that for any technology t, $POP(t) \leq C$.

A more extreme form of the conjecture states that a non-anonymous OR technology with 2 agents is the most extreme case and yields the highest possible POP for any structured technology.

Additionally, we study some other properties of mixed equilibrium. We show that mixed Nash equilibria are more delicate than pure ones. In particular, we show that unlike the pure case, in which the optimal contract is also a "strong equilibrium" [1] (i.e., resilient to deviations by coalitions), an optimal mixed contract (in which at least two agents truly mix) never satisfies the requirements of a strong equilibrium (Section 6).

Finally, we study the computational hardness of the optimal mixed Nash equilibrium, and show that the hardness results from the pure case hold for the mixed case as well (Section 7).

2 Model and Preliminaries

We focus on the simple “binary action, binary outcome” scenario where each agent has two possible actions ("exert effort" or "shirk") and there are two possible outcomes ("failure", "success"). We begin by presenting the model with pure actions (which is a generalization of [10]), and then move to the mixed case. A principal employs a set of agents N of size n. Each agent $i \in N$ has a set of two possible actions $A_i = \{0, 1\}$ (binary action), the low effort action (0) has a cost of 0 ($c_i(0) = 0$), while the high effort action (1) as a cost of $c_i > 0$ ($c_i(1) = c_i$). The played profile of actions determine, in a probabilistic way, a “contractible” outcome, $o \in \{0, 1\}$, where the outcomes 0 and 1 denote project failure and success, respectively (binary-outcome). The outcome is determined according to a success function $t : A_1 \times \ldots \times A_n \to [0, 1]$, where $t(a_1, \ldots, a_n)$ denotes the probability of project success where players play with the action profile $a = (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n = A$. We use the notation $(t, \boldsymbol{c})$ to denote a technology (a success function and a vector of costs, one for each agent).

The principal's value of a successful project is given by a scalar $v > 0$, where he gains no value from a project failure. In this hidden-actions model the actions of the players are invisible, but the final outcome is visible to him and to others, and he may design enforceable contracts based on this outcome. We assume that the principal can pay the agents but not fine them (known as the *limited liability* constraint). The contract to agent i is thus given by a scalar value $p_i \geq 0$ that denotes the payment that i gets in case of project success. If the project fails, the agent gets no money (this is in contrast to the “observable-actions” model in which payment to an agent can be contingent on his action).

Given this setting, the agents have been put in a game, where the utility of agent i under the profile of actions $a = (a_1, \ldots, a_n) \in A$ is given by $u_i(a) = p_i \cdot t(a) - c_i(a_i)$. As usual, we denote by $a_{-i} \in A_{-i}$ the $(n-1)$-dimensional vector of the actions of all agents excluding agent i. i.e., $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. The agents will be assumed to reach Nash equilibrium, if such an equilibrium exists. The principal's problem (which is our problem in this paper) is how to design the contracts p_i as to maximize his own expected utility $u(a, v) = t(a) \cdot (v - \sum_{i \in N} p_i)$, where the actions $a_1, \ldots, a_n$ are at Nash-equilibrium. In the case of multiple Nash equilibria, in our model we let the principal choose the desired one, and “suggest” it to the agents, thus focusing on the “best” Nash equilibrium.

As we wish to concentrate on motivating agents, rather than on the coordination between agents, we assume that more effort by an agent always leads to a better probability of success. Formally, $\forall i \in N, \forall a_{-i} \in A_{-i}$ we have that $t(1, a_{-i}) > t(0, a_{-i})$. We also assume that $t(a) > 0$ for any $a \in A$.

We next consider the extended game in which an agent can mix between exerting effort and shirking (randomize over the two possible pure actions). Let q_i denote the probability that agent i exerts effort, and let q_{-i} denote the $(n-1)$-dimensional vector of investment probabilities of all agents except for agent i. We can extend the definition of the success function t to the range of mixed strategies, by taking the expectation.

$$t(q_1, \ldots, q_n) = \sum_{a \in \{0,1\}^n} (\prod_{i=1}^{n} q_i^{a_i} \cdot (1 - q_i)^{(1-a_i)}) t(a_1, \ldots, a_n)$$

Note that for any agent i and any (q_i, q_{-i}) it holds that $t(q_i, q_{-i}) = q_i \cdot t(1, q_{-i}) + (1 - q_i) \cdot t(0, q_{-i})$. A mixed equilibrium profile in which at least one agent mixes with probability $p_i \in [0, 1]$ is called a *non-degenerate* mixed equilibrium.

In pure strategies, the marginal contribution of agent i, given $a_{-i} \in A_{-i}$, is defined to be: $\Delta_i(a_{-i}) = t(1, a_{-i}) - t(0, a_{-i})$. For the mixed case we define the marginal contribution of agent i, given q_{-i} to be: $\Delta_i(q_{-i}) = t(1, q_{-i}) - t(0, q_{-i})$. Since t is monotone, Δ_i is a positive function.

We next characterize what payment can result in an agent mixing between exerting effort and shirking.

Claim. Agent i's best response is to mix between exerting effort and shirking with probability $q_i \in (0, 1)$ only if he is indifferent between $a_i = 1$ and $a_i = 0$. Thus, given a profile of strategies q_{-i}, agent i mixes only if:

$$p_i = \frac{c_i}{\Delta_i(q_{-i})} = \frac{c_i}{t(1, q_{-i}) - t(0, q_{-i})}$$

which is the payment that makes him indifferent between exerting effort and shirking. The expected utility of agent i, who exerts effort with probability q_i is: $u_i(q) = c_i \cdot \left(\frac{t(q)}{\Delta_i(q_{-i})} - q_i \right)$.

A profile of mixed strategies $q = (q_1, \ldots, q_n)$ is a *Mixed Nash equilibrium* if for any agent i, q_i is agent i's best response, given q_{-i}.

The principal's expected utility under the mixed Nash profile q is given by $u(q, v) = (v - P) \cdot t(q)$, where P is the total payment in case of success, given by $P = \sum_{i|q_i>0} \frac{c_i}{\Delta_i(q_{-i})}$. An *optimal mixed contract* for the principal is an equilibrium mixed strategy profile $q^*(v)$ that maximizes the principal's utility at the value v. In [2] we show a similar characterization of optimal pure contract $a \in A$. An agent that exerts effort is paid $\frac{c_i}{\Delta_i(a_{-i})}$, and the utilities are the same as the above, when given the pure profile. In the pure Nash case, given a value v, an *optimal pure contract* for the principal is a set of agents $S^*(v)$ that exert effort in equilibrium, and this set maximizes the principal's utility at the value v.

A simple but crucial observation, generalizing a similar one in [2] for the pure Nash case, shows that the optimal mixed contract exhibits some monotonicity properties in the value.

Lemma 1 (Monotonicity lemma). *For any technology $(t, \boldsymbol{c})$ the expected utility of the principal at the optimal mixed contract, the success probability of the optimal mixed contract, and the expected payment of the optimal mixed contract, are all monotonically non-decreasing with the value.*

The proof shows that the same monotonicity holds in the observable-actions case as well. Additionally, the lemma holds in more general settings, where each agent has an arbitrary action set (not restricted to the binary-actions model considered here).

We wish to quantify the gain by inducing mixed Nash equilibrium, over inducing pure Nash. We define the *price of purity* as the worse ratio (over v) between the maximum utilities that are obtained in mixed and pure strategies.

Definition 1. *The* price of purity $POP(t, \boldsymbol{c})$ *of a technology* $(t, \boldsymbol{c})$ *is defined as the worse ratio, over* v*, between the principal's optimal utility in the mixed case and his optimal utility in the pure case. Formally,*

$$POP(t, \boldsymbol{c}) = Sup_{v>0} \frac{t(q^*(v)) \left(v - \sum_{i|q_i^*(v)>0} \frac{c_i}{\Delta_i(q^*_{-i}(v))} \right)}{t(S^*(v)) \left(v - \sum_{i \in S^*(v)} \frac{c_i}{\Delta_i(a_{-i})} \right)}$$

where $S^*(v)$ denotes an optimal pure contract and $q^*(v)$ denotes an optimal mixed contract, for the value v.

The price of purity is at least 1, and may be greater than 1, as we later show. Additionally, it is obtained at some value that is a transition point of the pure case (a point in which the principal is indifferent between two optimal pure contracts).

Lemma 2. *For any technology $(t, \boldsymbol{c})$, the price of purity is obtained at a finite v that is a transition point between two optimal pure contracts.*

2.1 Structured Technology Functions

In order to be more concrete, we next present technology functions whose structure can be described easily as being derived from independent agent tasks – we call these *structured technology functions*. This subclass gives us some natural examples of technology functions, and also provides a succinct and natural way to represent technology success functions.

In a structured technology function, each individual succeeds or fails in his own "task" independently. The project's success or failure deterministically depends, maybe in a complex way, on the set of successful sub-tasks. Thus we will assume a monotone Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which indicates whether the project succeeds as a function of the success of the n agents' tasks.

A structured technology function t is defined by $t(a_1, \ldots, a_n)$ being the probability that $f(x_1, \ldots, x_n) = 1$ where the bits $x_1, \ldots, x_n$ are chosen according to the following distribution: if $a_i = 0$ then $x_i = 1$ with probability $\gamma_i \in [0, 1)$ (and $x_i = 0$ with probability $1 - \gamma_i$); otherwise, i.e. if $a_i = 1$, then $x_i = 1$ with probability $\delta_i > \gamma_i$ (and $x_i = 0$ with probability $1 - \delta_i$). Thus, a structured technology is defined by n, f and the parameters $\{\delta_i, \gamma_i\}_{i \in N}$.

Let us consider two simple structured technology functions, "AND" and "OR". First consider the "AND" technology: $f(x_1, \ldots, x_n)$ is the logical conjunction of x_i ($f(x) = \bigwedge_{i \in N} x_i$). Thus the project succeeds only if all agents succeed in their tasks. For this technology, the probability of success is the product of the individual success probabilities. Agent i succeeds with probability $\delta_i^{a_i} \cdot \gamma_i^{1-a_i}$, thus $t(a) = \prod_{i \in N} \delta_i^{a_i} \cdot \gamma_i^{1-a_i}$.

Next, consider the "OR" technology: $f(x_1, \ldots, x_n)$ is the logical disjunction of x_i ($f(x) = \bigvee_{i \in N} x_i$). Thus the project succeeds if at least one of the agents succeed in their tasks. For this technology, the probability of success is 1 minus the probability that all of them fail. Agent i fails with probability $(1-\delta_i)^{a_i} \cdot (1-\gamma_i)^{1-a_i}$, thus $t(a) = 1 - \prod_{i \in N}(1-\delta_i)^{a_i} \cdot (1-\gamma_i)^{1-a_i}$.

These are just two simple examples. One can consider other more interesting examples as the Majority function (the project succeed if the majority of the agents are successful), or the OR-Of-ANDs technology, which is a disjunction over conjunctions (several teams, the project succeed if all the agents in any one of the teams are successful). For additional examples see [2].

A success function t is called *anonymous* if it is symmetric with respect to the players. I.e. $t(a_1, \ldots, a_n)$ depends only on $\sum_i a_i$. For example, in an anonymous OR technology there are parameters $1 > \delta > \gamma > 0$ such that each agent i succeed with probability γ with no effort, and with probability $\delta > \gamma$ with effort. If m agents exert effort, the success probability is $1 - (1-\delta)^m \cdot (1-\gamma)^{n-m}$.

A technology has *identical costs* if there exists a c such that for any agent i, $c_i = c$. As in the case of identical costs the POP is independent of c, we use $POP(t)$ to denote the POP for technology t with identical costs. We abuse notation and denote a technology with identical costs by its success function t. Throughout the paper, unless explicitly stated otherwise, we assume identical costs. A technology t with identical costs is *anonymous* if t is anonymous.

3 Example: Mixed Nash Outperforms Pure Nash!

If the actions are observable (henceforth, the observable-actions case), then an agent that exerts effort is paid exactly his cost, and the principal's utility equals the social welfare. In this case, the social welfare in mixed strategies is a convex combination of the social welfare in pure strategies; thus, it is clear that the optimal utility is always obtained in pure strategies. However, surprisingly enough, in the hidden-actions case, the principal might gain higher utility when mixed strategies are allowed. This is demonstrated in the following example:

Example 1. Consider an anonymous OR technology with two agents, where $c = 1$, $\gamma = \gamma_1 = \gamma_2 = 1 - \delta_1 = 1 - \delta_2 = 0.09$ and $v = 348$. The mixed strategies $q_1 = q_2 = 0.92$ achieve a utility of 324.27, while the optimal contract with pure strategies is obtained when both agents exert effort and achieves a utility of 318.3. This implies that by moving from pure strategies to mix strategies, one gains at least $324.27/318.3 > 1.0187$ factor improvement (which is approximately 1.8%).

A worse ratio exists for the more general case (in which it does not necessarily hold that $\delta = 1 - \gamma$) of $\gamma = 0.0001$, $\delta = 0.9$ and $v = 233$. For this case we get that the optimal pure contract is with one agent, gives utility of 208.7, while the mixed contract $q_1 = q_2 = 0.92$ gives utility of 213.569, and the ratio is at least 1.0233 (approximately 2.3%).

To complete the example, Diagram 1 presents the optimal contract for *OR* of 2 agents, as a function of γ (when $\delta = 1 - \gamma$) and v. It shows that for some parameters of γ and v, the optimal contract is obtained when both agents exert effort with equal probabilities.

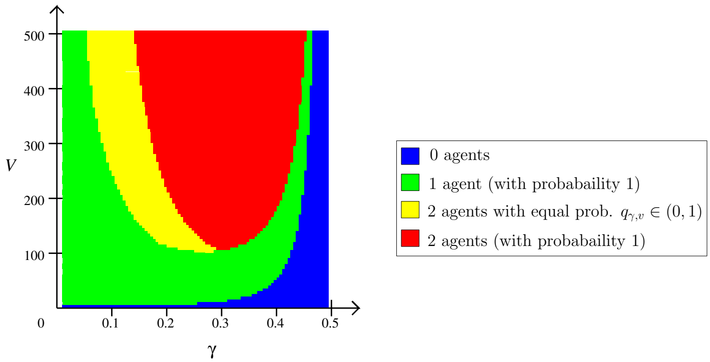


Fig. 1. Optimal mixed contracts in *OR* technologies with 2 agents. The light area corresponds to both agents exert effort with the same non-trivial probability, $q_{\gamma,v}$. For any fixed γ, $q_{\gamma,v}$ increases in v.

The following lemma shows that optimal mixed contracts in any anonymous OR technology have this specific structure. That is, all agents that do not shirk, mix with exactly the same probability.

Lemma 3. *For any anonymous OR technology (any $\delta > \gamma, c, n$) and value v, either the optimal mixed contract is a pure contract, or, in the optimal mixed contract $k \in \{2, \ldots n\}$ agents exert effort with equal probabilities $q_1 = \ldots = q_k \in$* $(0,1)$*, and the rest of the agents exert no effort.*

4 When Is Pure Nash Good Enough?

Next, we identify a class of technologies for which the price of purity is 1; that is, the principal cannot improve his utility by moving from pure Nash equilibrium to mixed Nash equilibrium. These are technologies for which the marginal contribution of any agent is non-decreasing in the effort of the other agents. Formally, for two pure action profiles $a, b \in A$ we denote $b \succeq a$ if for all j, $b_j \succeq_j a_j$ (effort b_j is at least as high as the effort a_j).

Definition 2. *A technology success function t exhibits (weakly)* **increasing returns to scale (IRS)** *if for every i, and every pure profiles $b \succeq a$*

$$t(b_i, b_{-i}) - t(a_i, b_{-i}) \geq t(b_i, a_{-i}) - t(a_i, a_{-i})$$

Any AND technology exhibits IRS [10,2]. For IRS technologies we show that POP=1.

Theorem 1. *Assume that t exhibits increasing returns to scale (IRS). For any cost vector $\boldsymbol{c}$, $POP(t, \boldsymbol{c}) = 1$. Moreover, a non-degenerate mixed contract is never optimal.*

We show that AND (on some subset of bits) is the only function such that any structured technology based on this function exhibits IRS, that is, this is the only function such that for any choices of parameters (any n and any $\{\delta_i, \gamma_i\}_{i \in N}$), the structured technology exhibits IRS. For any other Boolean function, there is an assignment for the parameters such that the created structured technology is essentially OR over 2 inputs (see lemma in full version), thus it has non-trivial POP (recall Example 1).

Theorem 2. *Let f be any monotone Boolean function with $n \geq 2$ inputs, that is not constant and not a conjunction of some subset of the input bits. Then there exist parameters $\{\gamma_i, \delta_i\}_{i=1}^{n}$ such that the POP of the structured technology with the above parameters (and identical cost $c = 1$) is greater than* 1.0233.

Thus, our goal now is to give upper bounds on the POP for various technologies.

5 Quantifying the Gain by Mixing

5.1 POP for General Technologies

We first show that the POP can be bounded by the principal's price of unaccountability [2]. The principal's price of unaccountability ($POU_P(t)$) is the worse ratio (over all v), of the utility of the principal in the observable-actions case, and the utility of the principal in the hidden-actions case (see formal definition of POU_P in the full version).

Theorem 3. *For any technology t is holds that $POU_P(t) \geq POP(t)$.*

However, this bound is rather weak. To best see this, note that the principal's price of unaccountability for AND might be unbounded (see [2]). Yet, as shown in section 1, $POP(AND) = 1$.

In this section we provide better bounds on technologies with identical costs. We begin by characterizing the payments for a mixed contract. We show that under a mixed profile, each agent in the support of the contract is paid at least the minimal payment to a single agent under a pure profile with the same support, and at most the maximal payment.

For a mixed profile $q = (q_1, q_2, \ldots, q_n)$, let $S(q)$ be the support of q, that is, $i \in S(q)$ if and only if $q_i > 0$. Similarly, for a pure profile $a = (a_1, a_2, \ldots, a_n)$ let $S(a)$ be the support a. Under the mixed profile q, agent $i \in S(q)$ is being paid $p_i(q_{-i}) = \frac{c_i}{t(1,q_{-i}) - t(0,q_{-i})}$. Similarly, under the pure profile a, agent $i \in S(a)$ is being paid $p_i(S(a) \setminus \{i\}) = p_i(a_{-i}) = \frac{c_i}{t(S(a)) - t(S(a) \setminus \{i\})}$, where $t(T)$ is the success probability when $a_j = 1$ for $j \in T$, and $a_j = 0$ for $j \notin T$.

Lemma 4. *For a mixed profile $q = (q_1, q_2, \ldots, q_n)$, and for any agent $i \in S(q)$ let $S_{-i} = S(q) \setminus \{i\}$ be the support of q excluding i. It holds that*

$$max_{T \subseteq S_{-i}} \; p_i(T) \geq p_i(q_{-i}) \geq min_{T \subseteq S_{-i}} \; p_i(T)$$

In what follows, we consider two general families of technologies with n agents: anonymous technologies and technologies that exhibit decreasing returns to scale (DRS). DRS technologies are technologies with decreasing marginal contribution (more effort by others decrease the contribution of an agent). For both families we present a bound of n on the POP.

We begin with a formal definition of DRS technologies.

Definition 3. *A technology success function t exhibits (weakly)* **decreasing returns to scale (DRS)** *if for every i, and every $b \succeq a$*

$$t(b_i, b_{-i}) - t(a_i, b_{-i}) \leq t(b_i, a_{-i}) - t(a_i, a_{-i})$$

Theorem 4. *For any anonymous technology or a (non-anonymous) technology that exhibits DRS, it holds that $POP(t) \leq n$.*

We also prove a bound on the POP for any technology with 2 agents (even not anonymous), and an improved bound for the anonymous case.

Theorem 5. *For any technology t (even non-anonymous) with 2 agents, it holds that $POP(t) \leq 2$. If t is anonymous then $POP(t) \leq 3/2$.*

We do not provide bounds for any non-anonymous technology, this is left as an open problem for future research.

Open Problem 1. *Provide an upper bound on the POP for general technologies.*

As mentioned in the introduction, we believe that the obtained bounds are very weak. In particular, we conjecture that there exists a constant C that bounds the POP for any technology. Moreover, we believe that a non-anonymous OR technology with 2 agents yields the highest possible POP. This motivates us to explore the POP for the OR technology in more detail.

5.2 POP for the OR Technology

As any OR technology (even non-anonymous) exhibits DRS (see claim in the full version), this implies a bound of n on the POP of the OR technology. Yet, for anonymous OR technology we present improved bounds on the POP. In particular, if $\gamma = 1 - \delta < 1/2$ we can bound the POP by 1.154....

Theorem 6. *For any anonymous OR technology with n agents:*

1. *If $1 > \delta > \gamma > 0$:*
   (a) *$POP \leq \frac{1-(1-\delta)^n}{\delta} \leq n - (n-1)\delta$.*
   (b) *POP goes to 1 as n goes to ∞ (for any fixed δ) or when δ goes to 1 (for any fixed $n \geq 2$).*

2. If $\frac{1}{2} > \gamma = 1 - \delta > 0$*:*
   (a) $POP \leq \frac{2(3-2\sqrt{3})}{3(\sqrt{3}-2)}(= 1.154..)$*.*
   (b) POP goes to 1 *as* γ *goes to* 0 *or as* γ *goes to* $\frac{1}{2}$ *(for any fixed* $n \geq 2$*).*

While the bounds for anonymous OR technologies for the case in which $\delta = 1-\gamma$ are much better than the general bounds, they are still not tight. The highest POP we were able to obtain by simulations was of 1.0233 for $\delta > \gamma$, and 1.0187 for $\delta = 1 - \gamma$ (see Section 3), but deriving the exact bound analytically is left as an open problem.

Open Problem 2. *What is the POP for an anonymous OR technology? what is it for a non-anonymous OR technology?*

6 The Robustness of Mixed Nash Equilibria

In order to induce an agent i to truly mix between exerting effort and shirking, p_i must be equal exactly to $c_i/\Delta_i(q_{-i})$ (see claim 2). Even under an increase of ϵ in p_i, agent i is no longer indifferent between $a_i = 0$ and $a_i = 1$, and the equilibrium falls apart. This is in contrast to the pure case, in which any $p_i \geq \frac{c_i}{\Delta_i(a_{-i})}$ will maintain the required equilibrium. This delicacy exhibits itself through the robustness of the obtained equilibrium to deviations in *coalitions* (as opposed to the unilateral deviations as in Nash). A "strong equilibrium" [1] requires that no subgroup of players (henceforth *coalition*) can coordinate a joint deviation such that every member of the coalition strictly improves his utility.

Definition 4. *A mixed strategy profile* $q \in [0,1]^n$ *is a* strong equilibrium (SE) *if there does not exist any coalition* $\Gamma \subseteq N$ *and a strategy profile* $q'_\Gamma \in \times_{i\in\Gamma}[0,1]$ *such that for any* $i \in \Gamma$*,* $u_i(q'_{-\Gamma}, q_\Gamma) > u_i(q)$*.*

In [2] we show that under the payments that induce the pure strategy profile S^* as the best pure Nash equilibrium (i.e., the pure Nash equilibrium that maximizes the principal's utility), S^* is also a *strong equilibrium.* In contrast to the pure case, we next show that any non-degenerate mixed Nash equilibrium q in which there exist at least two agents that truly mix (i.e., $\exists i \neq j$ *s.t.* $q_i, q_j \in (0,1)$), can never be a strong equilibrium. This is because if the coalition $\Gamma = \{i|q_i \in (0,1)\}$ deviate to q'_Γ in which each $i \in \Gamma$ exerts effort with probability 1, each agent $i \in \Gamma$ strictly improves his utility.

Theorem 7. *If the mixed optimal contract* q *includes at least two agents that truly mix (*$\exists i \neq j$ *s.t.* $q_i, q_j \in (0,1)$*), then* q *is not a strong equilibrium.*

In any OR technology, for example, it holds that in any non-degenerate mixed equilibrium at least two agents truly mix (see lemma 3). Therefore, no non-degenerate contract in the OR technology can be a strong equilibrium.

As generically a mixed Nash contract is not a strong equilibrium while a pure Nash contract always is, if the pricipal wishes to induce a strong Nash equilibrium (e.g., when the agents can coordinate their moves), he can restrict himself to inducing a pure Nash equilibrium, and his loss from doing so is bounded by the POP (see section 5).

7 Algorithmic Aspects

The computational hardness of finding the optimal mixed contract depends on the representation of the technology and how it is being accessed. For a black-box access and for the special case of read-once networks, we generalize our hardness results of the pure case [2] to the mixed case. The main open question is whether it is possible to find the optimal mixed contract in polynomial time, given a table representation of the technology (the optimal pure contract can be found in polynomial time in this case). Our generalization theorems follow.

Theorem 8. *Given as input a black box for a success function t (when the costs are identical), and a value v, the number of queries that is needed, in the worst case, to find the optimal mixed contract is exponential in n.*

Even if the technology is a structured technology and further restricted to be the source-pair reliability of a read-once network (see [2]), computing the optimal mixed contract is hard.

Theorem 9. *The optimal mixed contract problem for read once networks is #P-hard (under Turing reductions).*

Acknowledgments. The authors thank Robert Kleinberg for his help creating the figure. This work is supported by the Israel Science Foundation, the USA-Israel Binational Science Foundation, the Lady Davis Fellowship Trust, and by a National Science Foundation grant number ANI-0331659.

References

1. R. Aumann. Acceptable Points in General Cooperative n-Person Games. In *Contributions to the Theory of Games*, volume 4, 1959.
2. M. Babaioff, M. Feldman, and N. Nisan. Combinatorial agency. In *the 7th ACM conference on Electronic Commerce*, pages 18–28, 2006.
3. Y. K. Che and S. W. Yoo. Optimal Incentives in Teams. *American Economic Review*, 91:525–541, 2001.
4. M. Feldman, J. Chuang, I. Stoica, and S. Shenker. Hidden-Action in Multi-Hop Routing. In *ACM EC'05*, pages 117-126, 2005.
5. B. Holmstrom. Moral Hazard in Teams. *Bell Journal of Economics*, 13:324–340, 1982.
6. H. Itoh. Incentives to Help Multi-Agent Situations. *Econometrica*, 59:611–636, 1991.
7. A. Mass-Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.
8. N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behaviour*, 35:166 – 196, 2001. A preliminary version appeared in STOC 1999.
9. R. Strausz. Moral hazard in sequential teams. *Departmental Working Paper. Free University of Berlin*, 1996.
10. E. Winter. Incentives and Discrimination. *American Economic Review*, 94:764–773, 2004.

The Sound of Silence: Mining Implicit Feedbacks to Compute Reputation

Mikołaj Morzy[1] and Adam Wierzbicki[2,⋆]

[1] Institute of Computing Science
Poznań University of Technology
Piotrowo 2, 60-965 Poznań, Poland
Mikolaj.Morzy@put.poznan.pl
[2] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warszawa, Poland
adamw@pjwstk.edu.pl

Abstract. A reliable mechanism for scoring the reputation of sellers is crucial for the development of a successful environment for customer-to-customer e-commerce. Unfortunately, most C2C environments utilize simple feedback-based reputation systems, that not only do not offer sufficient protection from fraud, but tend to overestimate the reputation of sellers by introducing a strong bias toward maximizing the volume of sales at the expense of the quality of service.

In this paper we present a method that avoids the unfavorable phenomenon of overestimating the reputation of sellers by using implicit feedbacks. We introduce the notion of an implicit feedback and we propose two strategies for discovering implicit feedbacks. We perform a twofold evaluation of our proposal. To demonstrate the existence of the implicit feedback and to propose an advanced method of implicit feedback discovery we conduct experiments on a large volume of real-world data acquired from an online auction site. Next, a game-theoretic approach is presented that uses simulation to show that the use of the implicit feedback can improve a simple reputation system such as used by eBay. Both the results of the simulation and the results of experiments prove the validity and importance of using implicit feedbacks in reputation scoring.

1 Introduction

Internet economy is doing very well. According to eMarketer, the e-commerce market is steadily growing with annual gains reaching 25% in 2004 and 21% in 2005. The growth is broad-based and distributes almost equally among all categories of retail, travel and entertainment. Projection for the future is optimistic: annual gains will continue to grow at a double-digit level reaching retail revenues

⋆ The research was supported by the Polish Ministry of Science and Higher Education under grant 3T11C 005 27 "Models and Algorithms for Efficient and Fair Resource Allocation in Complex Systems."

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 365–376, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of $139 billion in 2008 in the United States alone. Online auctions are among the most popular and important e-commerce services. It is estimated that over 15% of all e-commerce sales can be attributed to online auctions. EBay, the global leader in online auctions, has over 56 million active users (and over 95 million registered users). Annual transactions on eBay surpass $23 billion, with approximately 12 million different items posted simultaneously on eBay at any point in time and slightly less than 1 million transactions committed daily. This immense marketplace for customer-to-customer e-commerce provides means for anonymous and geographically dispersed users to seal retail transactions. But an important question arises: how does one estimate the reputation of an anonymous business partner and how does one develop trust when there is hardly any history of business contacts between any two partners?

A reliable reputation system is crucial for enabling a fair and credible environment for e-commerce activities. The quality of the reputation system directly affects the credibility of an online auction service and impacts the amount of fraud present on the online auction market. Unfortunately, fraud is still the main reason hindering further development of online auctions. According to National Fraud Information Center, online auctions account for 42%[1] of all registered complaints with an average loss of $1,155. The number of complaints grows quickly (12,315 complaints in 2005 compared to 10,794 in 2004) as well as the total loss ($13,863,003 in 2005 compared to $5,787,170 reportly lost in 2004). Online auction fraud definitely outranks other popular types of scams. Therefore, the reputation system used by an online auction site must be robust enough to safeguard the online community of auction participants against fraudsters.

Devising a robust and fraud-free reputation system is difficult for various reasons. Most importantly, the reputation system must take into consideration high asymmetry between buyers and sellers in online auctions. These two classes of auction participants are exposed to different types of risk. Sellers are almost never threatened financially, because they can postpone the shipment of the merchandise until the payment is delivered. Therefore, sellers are generally not concerned with the reputation of their business partners. On the other hand, buyers decide upon participation in an auction solely based on the reputation of a seller. Furthermore, after delivering the payment buyers are still in hazard of receiving no merchandise, or receiving merchandise of lower quality and inconsistent with the initial offer. From a buyer's point of view, a credible estimation of seller's reputation is indispensable for secure and successful trade.

Most online auction sites use a simple feedback-based reputation system [10]. Typically, parties involved in a transaction mutually post feedbacks after the transaction is committed. Each transaction can be judged as 'positive', 'neutral', or 'negative'. The reputation of a user is simply the number of distinct partners providing positive feedbacks minus the number of distinct partners providing negative feedbacks. As pointed out in [5], such simple reputation system suffers from numerous deficiencies, including the subjective nature of feedbacks and the

[1] This number is grossly underestimated due to eBay's reluctance to cooperate with NFIC. NFIC estimates that the real number is closer to 70%.

lack of transactional and social contexts. We identify yet another drawback of feedback-based reputation systems: these systems do not account for psychological motivation of users. Many users refrain from posting a neutral or negative feedback in fear of retaliation, thus biasing the system into assigning overestimated reputation scores. This phenomenon is manifested by high asymmetry in feedbacks collected after auctions and, equally importantly, by high number of auctions with no feedback provided. We believe that many of these missing feedbacks convey implicit and unvoiced assessments of poor seller's performance which must be included in the computation of seller's reputation to provide an unbiased estimation of seller's reliability.

In this paper we introduce the concept of an implicit feedback. Implicit feedback is a useful, actionable pattern hidden in large amounts of online auction data. We mine the history of user feedbacks to discover missing feedbacks that were left out purposely and we include these implicit feedbacks in reputation scoring. We present an efficient and flexible strategy for identifying implicit feedbacks and we compare it to a simple majority voting strategy. We present a twofold experimental evaluation of our proposal using both game-theoretic simulation and experiments involving a large set of real-world data. The results of conducted experiments clearly indicate an important impact of using implicit feedbacks in reputation scoring. The paper is organized as follows. In Sect. 2 we present the related work on the subject. The existence and computational feasibility of implicit feedback are presented in Sect. 3. Two strategies for discovering implicit feedbacks are also presented. Section 4 contains the results of the experimental evaluation of our proposal. In this section we show how a simple reputation algorithm, such as used by eBay, can be greatly enhanced by using the implicit feedback. The paper concludes in Sect. 5 with a summary and future work agenda.

2 Related Work

Reputation systems [11] are trust management systems that are used to enable trust between anonymous, heterogeneous, and geographically dispersed business partners. One of the most important tasks of reputation systems is to provide an economical incentive to behave honestly [2] and to reward participants for fair behavior [7]. Among many definitions of trust, the one that is the most useful in the context of online auctions is that trust is a subjective expectation that other agents (i.e., buyers and sellers) will behave fairly [3,9]. In this context, fair behavior is defined as carrying out the agreed transaction to the best of agent's ability. This definition of trust relates reputation systems to the theory of justice that can be used to evaluate reputation systems. A reputation system works well if all agents that behave fairly receive just payoffs.

One of the most well-known method to evaluate justice is the use of the Lorenz curve and the Gini coefficient [4,6]. The Lorenz curve is a graphical representation of the ordered cumulative distribution function of a distribution of goods (income, resources, etc.). Consider a set of n agents that receive shares of goods denoted by $x_1, \ldots, x_n$. First, we need to sort the shares of agents incrementally, receiving a permutation of shares $x_{i_1}, \ldots, x_{i_n}$. From the resulting permutation we take cumulative sums $\theta_i = \sum_{j=1}^{i} x_{i_j}$. These sums are sometimes normalized by the sum of all shares, θ_n. The values of θ_i plotted against i form the Lorenz curve. Note that if all shares x_i are equal, then the Lorenz curve will be a straight line. For unequal distributions, the Lorenz curve is convex. An idealized perfect distribution of goods is given by $\theta_i = (i * \theta_n/n)$. The area between the Lorenz curve and this perfect distribution line (normalized by $2\theta_n$) is known as the Gini coefficient. We shall use the Gini coefficient as the measure of justice for the evaluation of reputation system effectiveness.

3 Existence of Implicit Feedback

A close investigation of the distribution of feedbacks reveals a striking deviation. The examined dataset contains data on a sample group of 10 000 buyers collected over the period of six months. There are 656 376 committed auctions and 890 876 mutual feedbacks. Table 1 summarizes data statistics.

Table 1. Distribution of feedbacks

	negative	%	neutral	%	positive	%	$\sum$	%
buyer	4318	1%	2877	0.6%	445 723	98.4%	452 918	69%
seller	2558	0.6%	553	0.1%	434 847	99.3%	437 948	66%

Buyers provided 452 918 feedbacks, which accounts for 69% of all examined auctions. Note that over 30% of all auctions are not sealed with a feedback. Almost all registered feedbacks are positive (98.4%), with only 1% of negative and 0.6% neutral feedbacks. Similar characteristics can be observed for feedbacks provided by sellers, although sellers are slightly less eager to provide a feedback in general. Similar results are reported in [12], so we believe that such distribution is quite typical for online auction sites. Table 1 presents a grossly optimistic view of the quality of service offered by participants. There are two interesting points to make. First, neutral feedback is missing, the scope for positive feedback ranges from an open praise to the acknowledgement of a correct auction (but nothing more), and negative feedback appears only when the quality of service becomes totally unacceptable. Second, more than 30% of auctions did not finalize with a feedback. In many of these auctions sellers conducted poorly, but the quality of service was either bearable, or the buyer was intimidated and afraid of a retaliatory negative feedback. In both cases the reputation of a seller should be affected negatively. We refer to these purposely omitted feedbacks as implicit feedbacks that indicate a seller's performance that is unsatisfactory and not deserving a praise, yet passable. Listening to these silent, unvoiced feedbacks makes the reputation estimation more credible. Alas, current reputation systems are not aware of the existence of implicit feedbacks and do not incorporate implicit feedbacks into reputation scoring.

Not every missing feedback should be regarded as an implicit assessment of user's performance. A feedback might be missing for various reasons, e.g., one of the trading parties might be an unexperienced user who does not know how to post a feedback. One simple strategy is to check the history of user's feedback and compute the ratio of user's auctions for which a given user has posted a feedback. If the majority of user's auctions have been sealed with a feedback, a missing feedback for a given auction might indicate a purposeful omission of the feedback, i.e., an implicit feedback. We call this strategy the *majority voting strategy*. We also devise a more complex and flexible *cosine strategy* presented next. Let $F(u_i) = \langle f_1, f_2, \ldots, f_n \rangle, f_i \in \{0,1\}$ be a chronologically ordered list of feedback flags posted by the user u_i, where $f_k = 0$ denotes the fact that the user u_i did not provide a feedback for her k-th auction, and $f_k = 1$ denotes the fact that the user u_i explicitly provided a feedback for her k-th auction. We arbitrarily assume that the effect of each auction experience (either positive or negative) influences the next two auctions of a given user[2]. $F(u_i)$ can be transformed into an ordered list of trigrams $T(u_i) = \langle t_1, t_2, \ldots, t_{n-2} \rangle$, where $t_i = f_i f_{i+1} f_{i+2}$ is a binary concatenation of feedback flags for the i-th auction with feedback flags for the consecutive two auctions. There are $2^3 = 8$ possible trigrams represented by binary numbers ranging from 000 (three consecutive auctions do not have a feedback) to 111 (three consecutive auctions have a feedback). Thus, $T(u_i)$ can be represented as a vector $\bar{T}(u_i) = \left[t_i^0, \ldots, t_i^7\right]$, where t_i^n is the number of occurrences of the n-th trigram in $T(u_i)$. We perceive $\bar{T}(u_i)$ as a condensed representation of feedback habits of the user u_i. Having transformed the original history of user feedbacks into an 8-dimensional vector we can compare this vector to a template vector representing a user who almost never provides a feedback for her auctions (in our experiments we have used the template vector $\bar{T}(0) = [1, 0.1, 0.1, 0.01, 0.1, 0.01, 0.01, 0]$, where three consecutive auctions without a feedback have the weight 1, two missing feedbacks have the weight 0.1, and one missing feedback has the weight 0.01). Let k-th auction of the user u_i does not have a feedback. First, we build $F(u_i) = \langle f_1, f_2, \ldots, f_k \rangle$, which is transformed into $T(u_i) = \langle t_1, t_2, \ldots, t_{k-2} \rangle$, and the resulting list $T(u_i)$ is transformed into the vector $\bar{T}(u_i)$. Next, we compute the Ochini coefficient (the cosine similarity function) between $\bar{T}(u_i)$ and $\bar{T}(0)$ as follows

$$Ochini(\bar{T}(u_i), \bar{T}(0)) = \frac{\sum_{k=0}^{7} t_i^k * t_0^k}{\sqrt{\sum_{k=0}^{7} (t_i^k)^2 * \sum_{k=0}^{7} (t_0^k)^2}}$$

If $Ochini(\bar{T}(u_i), \bar{T}(0)) < \beta$, where β is a user-defined threshold, we conclude that the two vectors are similar and the omission of a feedback should not be regarded as an implicit feedback.

Example 1. Let us assume a user u with the following list of feedback flags: $F(u) = \langle 0, 1, 0, 1, 1, 0 \rangle$. The user u participated in six auctions and did not pro-

[2] The authors acknowledge that the choice of three consecutive auctions as the range of psychological influence of an auction outcome is arbitrary and the correctness of this assumption remains open for discussion.

vide feedback for three of them. We want to know if the last missing feedback is a purposeful omission. First, the list of feedback flags $F(u)$ is transformed into a list of trigrams $T(u) = \langle 010, 101, 011, 110 \rangle$. Next, the list of trigrams is transformed into a compact vector representation $\bar{T}(u) = [0, 0, 1, 1, 0, 1, 1, 0]$. The final result is $Ochini(\bar{T}(u), \bar{T}(0)) = 0.09$. After a certain period of time the user u participates in more auctions and gains experience. Let us assume that, after a while, the list of feedback flags for the user u is $F(u) = \langle 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0 \rangle$. We want to decide on the last missing feedback as being an implicit feedback. The list of trigrams is $T(u) = \langle 010, 101, 011, 110, 101, 011, 111, 110, 101, 011, 110 \rangle$ and the vector representation is $\bar{T}(u) = [0, 0, 1, 3, 0, 3, 3, 1]$. Now the computation of the Ochini coefficient yields $Ochini(\bar{T}(u), \bar{T}(0)) = 0.035$. As can be seen, this procedure is flexible and allows for temporal changes in feedback habits.

To prove the existence of the implicit feedback we begin by investigating the distribution of the number of missing feedbacks per user (in this experiment we include only buyers). The results of the experiment are depicted in Figure 1. Interestingly, there are only a few buyers with more than 20 missing feedbacks. This might indicate that most of the missing feedbacks are in fact purposeful omissions, thus turning the missing feedbacks into implicit feedbacks. When we have constrained our search to buyers who had participated in at least 10 auctions, the average percentage of missing feedbacks dropped to 11.6%, which indicates that experienced users are even less likely to omit a feedback.

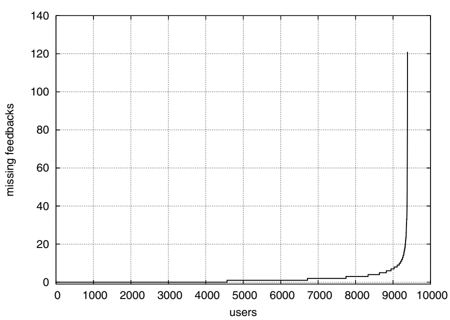


Fig. 1. Missing feedback distribution

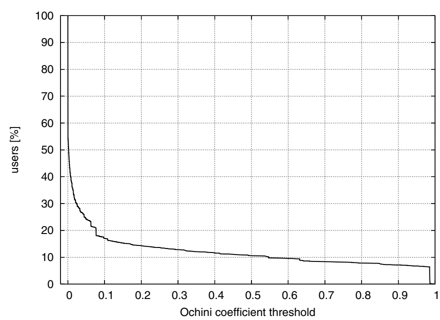


Fig. 2. Ochini coefficient selectivity

Figure 2 presents the user selectivity depending on the value of the Ochini coefficient. Recall that the Ochini coefficient represents the similarity between a given user's feedback vector and the template vector of a hypothetical 'I-don't-do-feedbacks' user, with the values closer to 1 representing high similarity and the values closer to 0 representing high dissimilarity. The figure presents the percentage of users who would be considered as generally not providing feedbacks, given the value of the Ochini coefficient threshold. For reasonable values of the Ochini coefficient threshold (i.e., 0.5 and above) less than 10% of buyers are regarded as reluctant to provide feedbacks, which means that their missing feedback would not be considered as implicit feedback. Again, this result proves that for the majority of buyers a missing feedback is an important, yet unvoiced, assessment of business partner's performance.

4 Effectiveness of Using Implicit Feedback

To evaluate a reputation system it is necessary to find out how this system affects the behavior of users and the outcome of user transactions. Ideally, we would like to know whether a reputation system enables trust: all honest users should trust other honest users and should be treated fairly by other honest users. On the other hand, all dishonest users should not be trusted and therefore should not participate in transactions. In this section we compare a simple reputation algorithm, such as used by eBay, to a more complex algorithm that uses implicit feedback. Section 4.1 presents the design of the simulator of online auctions. The results of conducted simulations are reported in Sect. 4.2. The impact of implicit feedback on real-world data is presented in Sect. 4.3.

4.1 The Simulator

Prior to starting the simulation we had to make a decision about a sufficiently realistic, yet not too complex model of the auctioning system, of user behavior, and of the reputation system. We choose to simulate the reputation system almost totally faithfully, the only simplification is that we use only positive and negative feedbacks. The behavior of a user is also realistic: the user takes into account the reputation when choosing a business partner. The user also decides whether she wishes to report or not, depending on the type of report. Users can cheat in reports, as well as in transactions. Users may also use transaction strategies that depend on the history of their individual interactions with other participants.

The auctioning system, on the other hand, has been simplified. We reflect that the simulation of the entire auction process is unnecessary. Rather, we simulate the selection of users using a random choice of a set of potential sellers. The choosing user (i.e., the buyer) selects one of the sellers from among users with the highest reputation in the set. The auction itself has also been simplified. We use a popular game-theoretic model of an auction, namely, the iterated Prisoner's Dilemma [1].

In the simulator, a number of agents that represent users interact with each other. Each interaction represents an auction between a seller and a buyer. The reputation system is maintained by a reputation server that is also used to summarize the outcomes of agent interactions. Each agent is characterized by the following parameters: r^+, the probability that an agent will send a report if it is positive, r^-, the probability that an agent will send a report if it is not positive, the chosen game strategy, the reputation threshold ρ_{min} that is used by some strategies, and the probability of cheating c, that is used by some strategies. We can specify the number of agents and every agent can have distinct parameters. However, we usually partition all agents into two sets that have the same parameters, called the honest and dishonest agents.

The two game strategies used in the simulations are: to cheat with the probability c or to play Tit-for-Tat with a reputation threshold ρ_{min}. Tit-for-Tat is a famous strategy for the iterated Prisoner's Dilemma game. This strategy works simply by repeating the move made by the other agent in the previous encounter. If two agents meet for the first time, the classic Tit-for-Tat strategy forces the agents to cooperate, thus allowing the agents to start an unending pattern of honest transactions. We modify Tit-for-Tat to use a reputation threshold: if two agents meet for the first time and the second agents' reputation is below ρ_{min}, the first agent defects.

The server computes reputation scores using available feedbacks and using any implemented algorithm. The results of the simulation include: reputation scores of individual agents and the total payoffs (from all auctions) of every agent. The payoffs are affected by the way the reputation system works. For example, if agents post very few feedbacks, reputation scores will be generally random, and the payoffs of good agents would drop. The simulator allows to check whether the implemented reputation algorithm is effective. To verify the concept of implicit feedbacks, we simulate the behavior of a simple reputation algorithm that uses implicit feedbacks.

Consider a user u with the history of n auctions. Let us assume that only $m \leq n$ of these auctions have a feedback. Out of these m feedbacks m^+ are positive feedbacks, while $m^- = m - m^+$ are all other feedbacks. Thus, $m^+ \leq m \leq n$. The reputation ρ_u of the user u will be calculated as follows

$$\rho_u = \frac{m^+}{\alpha * m^- + m}$$

where $0 \leq \alpha \leq 1$. Thus, if $\alpha = 0$, the above reputation score becomes a simple ratio of the number of positive feedbacks received by the user u. In the case when the user u has had no auctions, the above formula is undefined. In such case we set the reputation ρ_u to an initial value, ρ_0. To be precise, in our simulations we use a slightly more complex version of the above algorithm. Since agents in the simulator choose whom they want to interact with on the basis of reputation scores, it is necessary to avoid that the reputation would drop suddenly to a low level. This can happen in the initial phase of the simulation, when the reputation score has not yet stabilized (initially, a single negative feedback could decrease the initial reputation by a large degree). Therefore, we use a moving average to smooth reputation changes. The smoothed reputation $\rho_u^{ma}(t) = 0.5\rho_u^{ma}(t-1) + \rho_u(t)$, where t is time, and $\rho_u^{ma}(0) = \rho_0$ - the smoothed reputation is initialized by the initial reputation value. Note that over time, the estimate converges to the formula for ρ_u (since the impact of the initial reputation decreases exponentially).

4.2 Evalutation by Simulation

We have tested the algorithm described above using the following simulation scenario. First, we have divided all 300 agents into two sets, the *good agents* and the *bad agents*. Good agents were 66% of all agents, the remaining agents were bad agents. A good agent used the Tit-for-Tat strategy with the reputation threshold of $\rho_{min} = 0.5$. A bad agent used a strategy of random cheating with probability $c = 0.6$. All agents had the same behavior with respect to feedbacks. This behavior was a further parameter of the simulation scenarios. We used two posting behaviors: *perfect feedbacks*, where all agents always posted feedback truthfully, and *poor feedbacks*, where if the feedback was positive, an agent would post it with probability $r^+ = 0.66$, and if the feedback was not positive, an agent would post it with probability $r^- = 0.05$. All feedbacks were always true, if they were sent. The parameters of the poor feedbacks were derived from the analysis of traces obtained from the real-world data. In all simulations, 40 000 auctions were simulated between the agents.

Together, there are three significant simulation scenarios: perfect reports with reputation calculated using a simple ratio of positive feedbacks (a reputation algorithm like described in the previous section, only with $\alpha = 0$); poor reports with a simple ratio; and poor reports with the reputation algorithm that uses implicit feedbacks, with different settings for α.

All experiments were conducted using the Monte-Carlo method. We present average results from 10 simulation runs, together with 95% confidence intervals of results. The outcomes of the experiments were the payoffs of every agent. We evaluate the effectiveness of a reputation system using the following criteria: the average payoff of a good agent, the average payoff of a bad agent, and the Gini coefficient of the payoffs of the good agents. The last criterion was introduced as a way of evaluating the effectiveness of the reputation system in providing fairness of the treatment of good agents.

The results of the simulations are summarized in Table 2. The first row in the table corresponds to the perfect feedback scenario, where all agents always post truthful feedbacks, and reputation is calculated using a simple ratio of positive feedbacks. In this idealized scenario, the average payoff of good agents is the highest, at 101.76 (the values of payoffs for a single auction were derived from the payoff table of the Prisoner's dilemma game). The average payoff of a bad agent is much lower, at 22.66. This indicates that the reputation mechanism

Table 2. Impact of implicit feedback reputation algorithm on agent payoffs and justice

Scenario	AVGP+[a]	AVGP-[b]	GC+[c]	GCI[d]	AGPCI[e]
Perfect reports	101.76	22.66	0.70	0.63–0.76	100–103,5
Poor reports, $\alpha = 0$	96.45	54.20	0.51	0.45–0.58	93.6–99.3
Poor reports, $\alpha = 0.05$	99.08	23.03	0.75	0.66–0.85	96.1–102
Poor reports, $\alpha = 0.1$	100.40	23.52	0.67	0.58–0.76	98.8–101.9
Poor reports, $\alpha = 0.2$	100.74	22.64	0.74	0.66–0.82	98–103.4

[a] Average payoff of a good agent.
[b] Average payoff of a bad agent.
[c] Gini coefficient of good agents.
[d] Gini confidence interval (95%).
[e] Average good payoff confidence interval (95%).

is working because cheating agents get punished by lower reputation. In these simulations, the final reputation value of bad agents was almost always 0. The Gini coefficient at about 0.7 will be treated as a reference level for further experiments and values of the Gini coefficient above this level will be considered unacceptable. The 95% confidence intervals for both the Gini coefficient and the average payoff are quite narrow. The second row of the Table 2 shows the results of the second simulation scenario. In this scenario, agents provided feedbacks realistically, and the effect of this is an increase in the payoff of bad agents by almost 150%. In many simulations, bad agents managed to keep a high reputation value, leading the good agents to trust them. This enabled bad agents to cheat more good agents. As a result, the average payoff of good agents also decreased significantly. This decrease is also visible in the confidence interval of payoffs of good agents.

Further rows of the table show the impact of using implicit feedbacks. The rows correspond to using the reputation algorithm described in the previous section with different values of α. For all considered values of α, the payoffs of bad agents dropped sharply, almost to the level achieved when agents reported perfectly. This is the main argument for using implicit feedback: as our simulations indicate, the use of implicit feedbacks is efficient in preventing cheating. The payoffs of good agents also increased to a varying degree, but for all values of α, the average payoff of a good agent was higher than when a simple ratio of positive feedbacks was used as the reputation algorithm. On the basis of the performed experiments, it seems that the value of $\alpha = 0.1$ gave the best results. For $\alpha = 0.2$, the average payoffs of the good agents were higher, and the average payoffs of bad agents were lower than for $\alpha = 0.1$. However, the average Gini coefficient was also higher. The reason for this may be that in the simulations, good agents sent positive feedbacks randomly with a probability of 66%. It was possible that a good agent would repeatedly get no positive feedback for her cooperation with another good agent. This could result in decreasing the reputation of the good agent, especially for higher values of α. The poor performance of $\alpha = 0.05$ can be explained by the fact that with such a low setting of α, the reputation of bad agents did not decrease quickly enough. While our simulations do not allow to choose the value of α that would be applicable in a real-world scenario, they are sufficient to indicate that there should exist an optimal value of α that is neither too high nor too low.

4.3 Evaluation by Mining

We conduct the experiments on a large body of real-world data acquired from `www.allegro.pl`, the leading provider of online auctions in Poland. The examined dataset consists of 656 376 auctions collected over the period of six months. The reputation of users is determined based on 890 876 mutual feedbacks provided by users. The dataset has been created by gathering all auctions of a seed set of 10 000 buyers during a fixed period of six months.

Figure 3 presents the influence of implicit feedbacks discovered using the majority voting strategy on the reputation score. Users are ordered according to

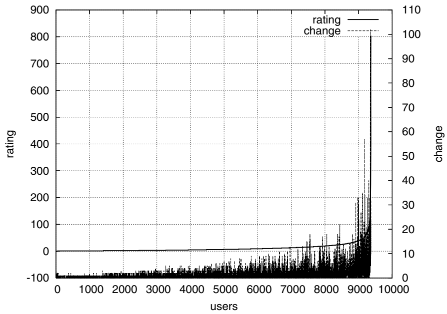


Fig. 3. Influence of missing feedbacks (majority voting strategy)

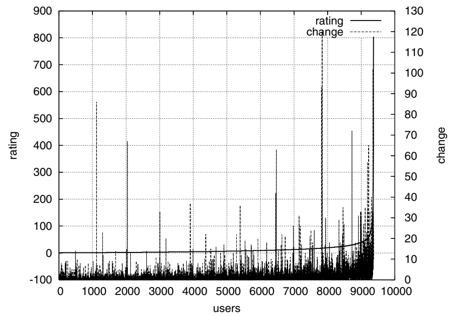


Fig. 4. Influence of missing feedbacks (cosine strategy

their rating computed traditionally and reported on the first y-axis. The second y-axis represents the change in user's reputation if implicit feedbacks are taken into consideration. We give an implicit feedback the weight of 0.2 of the weight of a negative feedback. On average, the reputation of users drops by 15% when implicit feedbacks are included in reputation scoring. For many users the change in reputation score is negligible, but there are users for whom the change is significant. We believe that traditional reputation systems grossly overestimate the reputation of these buyers and only incorporating the implicit feedback reveals their poor performance. The results of a similar experiment are depicted in Figure 4. This time we use the cosine strategy with the Ochini coefficient threshold $\beta = 0.4$. As can be easily noticed, the cosine strategy identifies more missing feedbacks as implicit feedbacks, thus affecting the reputation scoring stronger than the majority voting strategy. On average, the reputation of users drops by 17% when implicit feedbacks are included in reputation scoring. Of course, the impact factor depends on the value of the Ochini coefficient threshold. It remains to be seen which value of the threshold produces the most accurate and credible identification of implicit feedbacks. The results of both experiments affirm the practical usability and importance of using implicit feedbacks.

5 Conclusions

In this paper we have introduced the notion of the implicit feedback. To the best of authors' knowledge this is the first proposal to mine online auction data in search of unvoiced assessments of other user performance. We have been able to show that the use of implicit feedbacks in a reputation system can be effective. A simple reputation algorithm that used implicit feedbacks outperformed the reputation algorithm used by eBay. Using implicit feedbacks prevents the overestimation of reputation of dishonest auction participants, because negative opinion about their performance, otherwise concealed by intimidated users, can be used in reputation scoring. In this paper we have presented our initial findings on the impact of using implicit feedback with a simple reputation system. We plan to extend our experiments on more complex reputation algorithms [8].

References

1. Robert M. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.
2. Sulin Ba, Andrew B. Whinston, and Han Zhang. Building trust in online auction markets through an economic incentive mechanism. *Decision Support Systems*, 35:273–286, 2003.
3. Piotr Gmytrasiewicz and E. Durfee. Toward a theory of honesty and trust among communicating autonomous agents. *Group Decision and Negotiation*, 2:237–258, 1993.
4. Michael M. Kostreva, Wlodzimierz Ogryczak, and Adam Wierzbicki. Equitable aggregations and multiple criteria analysis. *European Journal of Operational Research*, 158(2):362–377, October 2004.
5. R. A. Malaga. Web-based reputation management systems: Problems and suggested solutions. *Electronic Commerce Research*, 4(1), 2001.
6. Albert W. Marshall and Ingram Olkin. *Inequalities: Theory of Majorization and Its Applications*. Academic Press, 1979.
7. Mikhail I. Melnik and James Alm. Does a seller's ecommerce reputation matter? evidence from ebay auctions. *The Journal of Industrial Economics*, L(3), September 2002.
8. Mikołaj Morzy. New algorithms for mining the reputation of participants of online auctions. In *WINE 2005, 1st Workshop on Internet and Network Economics, 15-17 December 2005, Hong Kong*. Springer Verlag, 2005.
9. Lik Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD thesis, Massachusetts Institute of Technology, 2003.
10. Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Advances in Applied Microeconomics*, 11, 2002.
11. Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12), 2000.
12. Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The value of reputation on ebay: a controlled experiment. Technical report, School of Information, University of Michigan, 2004.

Strongly Polynomial-Time Truthful Mechanisms in One Shot*

Paolo Penna[1], Guido Proietti[2], and Peter Widmayer[3]

[1] Dipartimento di Informatica ed Applicazioni "Renato M. Capocelli", Università di Salerno, Via **S. Allende** 2, 84081 Baronissi (SA), Italy
`penna@dia.unisa.it`

[2] Dipartimento di Informatica, Università di L'Aquila, Via Vetoio, 67010 L'Aquila, Italy, and Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", CNR, Viale Manzoni 30, 00185 Roma, Italy
`proietti@di.univaq.it`

[3] Institut für Theoretische Informatik, ETH Zürich, CAB H 15 Universitätstrasse 6, 8092 Zürich, Switzerland
`widmayer@inf.ethz.ch`

Abstract. One of the main challenges in *algorithmic mechanism design* is to turn (existing) efficient algorithmic solutions into efficient *truthful mechanisms*. Building a truthful mechanism is indeed a difficult process since the underlying algorithm must obey certain "monotonicity" properties and suitable *payment functions* need to be computed (this task usually represents the bottleneck in the overall time complexity).

We provide a general technique for building truthful mechanisms that provide optimal solutions in *strongly polynomial time*. We show that the *entire* mechanism can be obtained if one is able to express/write a strongly polynomial-time algorithm (for the corresponding optimization problem) as a "suitable combination" of simpler algorithms. This approach applies to a wide class of *mechanism design graph problems*, where each selfish agent corresponds to a weighted edge in a graph (the weight of the edge is the cost of using that edge). Our technique can be applied to several optimization problems which prior results cannot handle (e.g., MIN-MAX optimization problems).

As an application, we design the first (strongly polynomial-time) truthful mechanism for the *minimum diameter spanning tree* problem, by obtaining it directly from an existing algorithm for solving this problem. For this non-utilitarian MIN-MAX problem, no truthful mechanism was known, even considering those running in exponential time (indeed, exact algorithms do not necessarily yield truthful mechanisms). Also, standard techniques for payment computations may result in a running time which is not polynomial in the size of the input graph. The overall

* Work partially supported by the Research Project GRID.IT, funded by the Italian Ministry of Education, University and Research, by the European Project IST-15964 "Algorithmic Principles for Building Efficient Overlay Computers" (AEOLUS), by the European Union under COST 295 (DYNAMO), and by the Swiss BBW. Part of this work has been developed while the first and the second author were visiting ETH.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 377–388, 2006.
© Springer-Verlag Berlin Heidelberg 2006

running time of our mechanism, instead, is polynomial in the number n of nodes and m of edges, and it is only a factor $O(n\,\alpha(n,n))$ away from the best known canonical centralized algorithm.

1 Introduction

The emergence of the Internet as the platform for distributed computing has posed interesting questions on how to design efficient solutions which account for the lack of a "central authority" [11,15,16]. This aspect is certainly a key ingredient for the success of the Internet and, probably, of any "popular" system that one can envision (peer-to-peer systems are a notable example of this type of anarchic systems). In their seminal works, Koutsoupias and Papadimitriou [11] and Nisan and Ronen [15], suggest a *game-theoretic* approach in which the various "components" of the system are modeled as *selfish agents*: each agent performs a "strategy" which results in the highest *utility* for him/her-self. For instance, each agent may control a link of a communication network and each link has a cost for transmitting (i.e., for using it). A protocol that wishes to establish a minimum-cost path between two nodes would have to ask the agents for the cost of the corresponding link [15,2]. An agent may thus find it to be in his/her interest to lie about his/her costs (e.g., an agent might untruthfully report a very high cost in order to induce the protocol to use an alternative link, and thus no cost for the agent). Nisan and Ronen [15] propose a *mechanism design* approach that combines an underlying algorithm (e.g., a shortest-path algorithm) with a suitable payment function (e.g., how much we pay an agent for using his/her link). The idea is to come up with a so called *truthful mechanism*, that is, a combination of an algorithm with payments which guarantee that no agent can improve his/her own utility by misreporting his/her piece of private information (e.g., the cost of his/her link). Unfortunately, the design of truthful mechanisms is far from trivial and known results, originally developed in the microeconomics field [21,3,5,13], pose new algorithmic challenges which are the main subject of *algorithmic mechanism design* (see e.g. [4]).

Some interesting classes of problems (including a family of mechanism design graph problems considered here and in a number of works [15,7,6,10]) require the underlying algorithm to be *monotone* (e.g., if the algorithm selects an edge then it cannot drop this edge if its cost gets smaller and everything else remains the same). Though this condition suffices for the existence of a truthful mechanism [13,2], it is not clear how to guarantee this property nor how the corresponding payment functions can be efficiently computed (see e.g. [9,14]).

Mu'Alem and Nisan [12] were the first to propose a general method for constructing monotone algorithms (and thus truthful mechanisms). Basically, their approach consists of a set of "rules" to combine monotone algorithms so that the final combination results in a monotone algorithm as well. As observed by Kao *et al.* [10], the method in [12] does not provide an *efficient* way of computing the payments. Kao *et al.* [10] then extend some of the techniques in [12] and provide an efficient way for computing the corresponding payment functions.

Kao *et al.*'s approach [10] represents a significant progress towards a general technique which accounts for computational issues, though it cannot be applied to some very basic graph problems (e.g., a problem recently tackled in [19] – we discuss this issue more in detail below).

1.1 Our Contribution

In this work, we turn one of the main results in [12] into a general technique for building optimal truthful mechanisms running in *strongly polynomial time* (optimality refers to the quality of the computed solution). We show that the *entire* mechanism can be obtained if one is able to express/write an algorithm (for the corresponding optimization problem) as a "suitable combination" of simpler ones (see Section 2 and Theorem 1 therein). Obviously, the resulting mechanism is optimal and/or runs in strongly polynomial time if the algorithm does. However, neither of these conditions is required by our technique to guarantee truthfulness. This approach applies to a wide class of *mechanism design graph problems*, where each selfish agent corresponds to a weighted edge in a graph (the weight of the edge is the cost of using that edge). Our technique can deal with problems in which the cost function "underlying" the algorithm(s) is any *monotonically non-decreasing* function in the edge weights of the graph (i.e., in the costs of the agents). Since this includes several *non-utilitarian*[1] problems (e.g., MIN-MAX optimization functions), the results in [12] extend "only partially", that is, truthfulness can be guaranteed but the payments computation cannot be done "directly" by computing the "alternative" solution in which an agent is removed from the input (see e.g. [15,9]). We indeed observe that, for the problems considered in this work (see the discussion in Example 1), the payments computation is more complex than the case of *monotonically increasing* optimization functions, which are assumed in both [12] (where the problem is utilitarian) and in [10] (this assumption precedes Theorem 10 in [10] and the applications therein consist exclusively of utilitarian graph problems).

In Section 3, we apply our technique to the *minimum diameter spanning tree* problem and obtain the first (strongly polynomial-time) mechanism for it. For this non-utilitarian MIN-MAX problem, no truthful mechanism was known, even considering those running in exponential time (indeed, exact algorithms do not necessarily yield truthful mechanisms – see [17]). Also, standard techniques for payment computations may result in a running time which is not polynomial in the size of the input graph (see discussion in Example 1). The overall running time of our mechanism is instead $O(mn^2\,\alpha(n,n))$, and thus is only a factor $O(n\,\alpha(n,n))$ away from the best known algorithm for this problem [8], where $\alpha(\cdot,\cdot)$ is the classic inverse of the Ackermann's function. For two-edge connected graphs we also guarantee the *voluntary participation* condition, that is, no truthful agent runs into a loss (see next section for a formal definition). The minimum

[1] An optimization problem is called utilitarian if the goal is to minimize the sum of all agents costs or, equivalently, to maximize the sum of all agents valuations. Utilitarian graph problems have been studied in [15,9,10].

diameter spanning tree has both theoretical and practical relevance (e.g., in a peer-to-peer system we may want to set up a loop-free logical network using the resources – links – of a physical network so that any two peers can communicate efficiently).

The results for the minimum diameter spanning tree are paradigmatic of what happens when considering certain non-utilitarian mechanism design problems (another case is the *minimum radius spanning tree* problem [19] described in Example 1). First, one has to determine whether an existing algorithm can be turned into a truthful mechanism, whether a new one is needed, or if none can serve for this purpose [15,2,19]. In case a suitable algorithm exists, one has to find out how to compute the corresponding payments efficiently, possibly without burdening the complexity of the chosen algorithm [9,19]. Our technique can be used to give a positive answer to both questions, and thus to obtain the efficient mechanism in "one shot" (see Theorem 1).

We discuss other possible extensions and applications of our technique in Section 4 (these include a mechanism for the *p-center* graph problem and an improvement in the running time of the mechanism for the *minimum radius* in [19]).

1.2 Mechanism Design Graph Problems

Consider problems in which we are given a graph $G = (V, E)$ and the set of feasible outcomes consists of a suitable set $\mathcal{O} = \mathcal{O}(G)$ which depends only on the combinatorial structure of the graph (e.g., it consists of certain subgraphs of G). We have one agent per edge and the type $t_e \in \mathbb{R}^+$ of agent e is nothing but the *weight* of edge $e \in E$. Each solution $Y \in \mathcal{O}$ *uses* a subset of the edges of G; in particular, if Y uses edge e, then agent e has a cost (for implementing this outcome) equal to t_e. This scenario is common to several problems considered in the algorithmic mechanism design community: shortest-path [15], minimum spanning tree [15], shortest-paths tree [7], minimum-radius spanning tree [19]. Consider an agent e and let $\mathbf{r}_{-e}$ denote the values reported by the other agents, that is, $\mathbf{r}_{-e} = (r_1, \ldots, r_{e-1}, r_{e+1}, \ldots, r_m)$. When agent e reports x and the other agents report $\mathbf{r}_{-e}$, algorithm A computes a feasible outcome $A(x, \mathbf{r}_{-e})$. (That is, the algorithm returns a solution on input the vector $(x, \mathbf{r}_{-e}) := (r_1, \ldots, r_{e-1}, x, r_{e+1}, \ldots, r_m)$.) We say that declaration x is a *winning declaration* if solution $A(x, \mathbf{r}_{-e})$ uses edge e. A *mechanism* $M = (A, P)$ associates a payment $P_e(x, \mathbf{r}_{-e})$ with every agent e whose declaration x is a winning declaration (given the other agents' declarations $\mathbf{r}_{-e}$). This determines the *utility* of agent e:

$$u_e^M(x, \mathbf{r}_{-e}) := \begin{cases} P_e(x, \mathbf{r}_{-e}) - t_e & \text{if } A(x, \mathbf{r}_{-e}) \text{ uses } e, \\ 0 & \text{otherwise.} \end{cases}$$

Mechanism M is a *truthful* mechanism (with dominant strategies) if every function $u_e^M(x, \mathbf{r}_{-e})$ is maximized for $x = t_e$, for all $\mathbf{r}_{-e}$. We are interested in truthful mechanisms which optimize some objective function $\mu(Y, \mathbf{t})$ depending on the agents types $\mathbf{t} = (t_1, \ldots, t_m)$. Notice that the mechanism will work on the reported types $\mathbf{r}$. Hence, truthfulness guarantees that, if the algorithm returns an optimal solution for the given input, then the mechanism outputs an optimal solution w.r.t. the true types. We will also consider mechanisms which satisfy the *voluntary participation*, that is, a truthful agent is guaranteed to have a non-negative utility (i.e., $u_e^M(t_e, \mathbf{r}_{-e}) \geq 0$). This property will be achieved whenever there exists an "alternative" solution that does not use edge e, i.e., $\mathcal{O}(G-e) \neq \emptyset$.

2 A Technique for Efficient Truthful Mechanisms

Our approach consists in defining an optimal algorithm A as a "suitable combination" of simpler ones. For minimization problems, we combine algorithms by means of the following 'MIN' operator, which is essentially the same as the 'MAX' operator by Mu'Alem and Nisan [12]:

MIN$_\mu$(A_1,A_2) operator

- compute $Y_1 = A_1(\mathbf{r})$ and $Y_2 = A_2(\mathbf{r})$;
- if $\mu(Y_1, \mathbf{r}) \leq \mu(Y_2, \mathbf{r})$ then return Y_1 else return Y_2.

We can recursively apply this operator to several algorithms and obtain a new one:

$$\mathrm{MIN}_\mu(A_1, \ldots, A_k) := \mathrm{MIN}_\mu(\mathrm{MIN}_\mu(A_1, \ldots, A_{k-1}), A_k).$$

Notice that the ordering among the algorithms specifies how the new algorithm breaks ties. Our main concern is to have a general technique for building truthful mechanisms which optimize $\mu(\cdot)$ and that are *computationally efficient.*

To this end, we will assume that each algorithm A_i satisfies a property (called *plateau-like*) which is slightly stronger than the one (called bitonic) used in [12]:

Definition 1 (plateau-like algorithm). *An algorithm A for a mechanism design graph problem is* monotone *if, for all agents e, and for all $\mathbf{r}_{-e}$ there exists a threshold $\theta_e(\mathbf{r}_{-e}) \in (\mathbb{R}^+ \cup \infty)$ such that (i) every $x \leq \theta_e(\mathbf{r}_{-e})$ is a winning declaration and (ii) every $x > \theta_e(\mathbf{r}_{-e})$ is not a winning declaration. A monotone algorithm A is* plateau-like *w.r.t. $\mu(\cdot)$ if, for all e, for all $\mathbf{r}_{-e}$, the function $g_A(x) := \mu(A(x, \mathbf{r}_{-e}), (x, \mathbf{r}_{-e}))$ is non-decreasing in x and constant for $x > \theta_e(\mathbf{r}_{-e})$.*

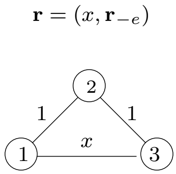
It is well known that an algorithm A can be turned into a truthful mechanism (A, P) if and only if A is monotone [13,2], in which case the payments are uniquely[2] determined by the thresholds:

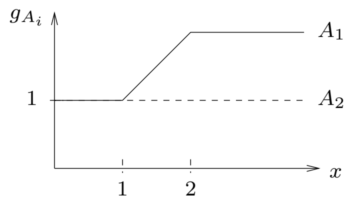
$$P_e(x, \mathbf{r}_{-e}) = \begin{cases} \theta_e(\mathbf{r}_{-e}) & \text{if } A(x, \mathbf{r}_{-e}) \text{ uses } e, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

[2] If $\theta_e(\mathbf{r}_{-e}) = \infty$, then we can set the payment of e to be any constant value and guarantee truthfulness. This case arises if edge e will be always included by A, e.g. for $\mathcal{O}(G-e) = \emptyset$, in which case voluntary participation cannot be guaranteed (unless we assume an upper bound on t_e). Otherwise, i.e. $\theta_e(\mathbf{r}_{-e}) < \infty$, the only payments which guarantee truthfulness are those in (1) [12] which then satisfy the voluntary participation condition.

Mu'Alem and Nisan [12] proved that, if all algorithms A_i are bitonic, then the algorithm $A = \mathrm{MIN}_\mu(A_1, \ldots, A_k)$ is monotone, and thus truthfulness can be guaranteed. Our main contribution here is a method for computing these payments efficiently if we assume that the algorithms are plateau-like. This task is non-trivial since the computation of the thresholds of a 'MIN' combination of algorithms can be rather involved if $\mu(\cdot)$ is not monotone increasing in x as in [12,10]:

Example 1 (Minimum Radius Spanning Tree (MRST)). Consider the problem of computing the *minimum radius spanning tree*, that is, a tree rooted at some node of the graph whose height is minimal. Consider the following simple graph (left):





If A_i outputs a shortest paths tree rooted at node i and $h(\cdot)$ denotes the height of any rooted tree, then both $A := \mathrm{MIN}_h(A_1, A_2)$ and $A' := \mathrm{MIN}_h(A_2, A_1)$ compute a MRST for this graph. However, the thresholds $\theta_e(\mathbf{r}_{-e})$ and $\theta'_e(\mathbf{r}_{-e})$ of the two algorithms are different. This is due to a different tie-breaking rule: For $0 \leq x \leq 1$, algorithms A_1 and A_2 have the same cost, i.e., $g_{A_1}(x) = g_{A_2}(x)$; hence,

$$A(x, \mathbf{r}_{-e}) = \begin{cases} A_1(x, \mathbf{r}_{-e}) \text{ if } x \leq 1 \\ A_2(x, \mathbf{r}_{-e}) \text{ otherwise} \end{cases}$$

while $A'(x, \mathbf{r}_{-e}) = A_2(x, \mathbf{r}_{-e})$. Since A_2 never uses edge e, while A_1 uses this edge for $x \leq 2$, it turns out that $\theta_e(\mathbf{r}_{-e}) = 1$ and $\theta'_e(\mathbf{r}_{-e}) = 0$.

Observe that, the threshold of algorithm $\mathrm{MIN}_h(A_1, A_2)$ is *different from the thresholds of the two algorithms.* Its computation depends on the way the functions g_{A_i} cross with each other, which in general can be quite involved (we have to consider how n "stairway" functions intersect pairwise [19] and the order in which we break ties). Finally, a binary search of this threshold may require a time which *depends on the edge weights* (namely, the logarithm of the largest reported type) and thus *not* strongly polynomial time, i.e., not polynomial in the number of nodes and edges. □

We reduce the computation of the payment $P_e(x, \mathbf{r}_{-e})$ to the task of computing, for every algorithm A_i, three thresholds $\theta^i = \theta^i_e(\mathbf{r}_{-e})$, $\hat{\theta}^i = \hat{\theta}^i_e(\mathbf{r}_{-e})$ and $\breve{\theta}^i = \breve{\theta}^i_e(\mathbf{r}_{-e})$. The value θ^i is the threshold in Def. 1 relative to algorithm A_i. The other two thresholds are defined as follows. Since A_i is plateau-like, $g_{A_i}(x) = \mu(A_i(x, \mathbf{r}_{-e}), (x, \mathbf{r}_{-e}))$ is constant for all $x > \theta^i$, where it also reaches its maximum. Let $\overline{g}_i$ be this maximum and let $g_{\min} := \min_i\{\overline{g}_i\}$. We let $\inf\{\emptyset\} = \infty$ and define $\hat{\theta}^i, \breve{\theta}^i \in (\mathbb{R}^+ \cup \infty)$ as follows:

$$\hat{\theta}^i := \inf\{x \mid g_{A_i}(x) \geq g_{\min}\}; \tag{2}$$

$$\breve{\theta}^i := \inf\{x \mid g_{A_i}(x) > g_{\min}\}. \tag{3}$$

Notice that the maximum $\overline{g}_i$ can be easily computed knowing θ_i. This is the main "additional" feature of plateau-like algorithms over bitonic ones. Intuitively speaking, $g_{\min}$ is the minimum cost if we do *not* use edge e. Thus, the solution of algorithm A_i will be selected only if its cost is better/not worse than this value (depending on the used tie-breaking rule). The two thresholds in (2-3) say what is the largest x for which this happens.

Our general approach for constructing computational efficient mechanisms consists in rewriting algorithms as suggested by the following:

Definition 2 (MIN-reducible algorithm). *An algorithm A is* MIN-reducible *if it can be written as the 'MIN' of plateau-like algorithms. That is, there exist k algorithms $A_1, \ldots, A_k$ such that $A = \mathrm{MIN}_\mu(A_1, \ldots, A_k)$ and each algorithm A_i is plateau-like w.r.t. $\mu(\cdot)$. Such an algorithm A is* MIN-reducible in τ time *if, for every input $\mathbf{r}$, it is possible to compute all thresholds $\theta_e^i(\mathbf{r}_{-e})$, $\hat{\theta}_e^i(\mathbf{r}_{-e})$ and $\breve{\theta}_e^i(\mathbf{r}_{-e})$ in at most τ time steps, for all $1 \leq i \leq k$ and for all edges e used by $A(\mathbf{r})$.*

The following result provides a powerful tool for designing efficient truthful mechanisms:

Theorem 1. *If algorithm A is MIN-reducible in $O(\tau)$ time, then there exist payment functions P such that (A, P) is a truthful mechanism and all payments $P_e(x, \mathbf{r}_{-e})$ can be computed in $O(\tau + k(\tau_\mu + N))$ time, where τ_μ is the time to compute $\mu(\cdot)$ and N is the number of used agents/edges.*

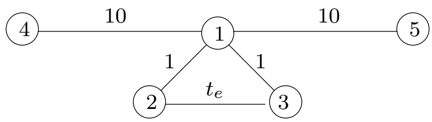
Proof Sketch. The first part of the theorem follows from a result by Mu'Alem and Nisan [12]. In order to prove the second part, we simply show that, given the values θ^i, $\hat{\theta}^i$ and $\breve{\theta}^i$, it is possible to compute $\theta_e(\mathbf{r}_{-e})$ in $O(k)$ time after the following preprocessing requiring $O(k \cdot \tau_\mu)$ time. First of all, we compute the index $i_{\min}$ of the first algorithm A_i such that $\overline{g}_i = g_{\min}$. This requires $O(k \cdot \tau_\mu)$ time for computing all $\overline{g}_i$, and from that the computation of $g_{\min}$ and $i_{\min}$ requires $O(k)$ time. (Recall that $\overline{g}_i = g_{A_i}(x)$ for any $x > \theta^i$.) Then we prove the following identity (see the full version [17] for the proof):

$$\theta_e(\mathbf{r}_{-e}) = \max\{\theta^{i_{\min}}, \max\{\hat{\theta}^i \mid i > i_{\min}\}, \max\{\breve{\theta}^i \mid i < i_{\min}\}\}. \tag{4}$$

Obviously, if we know $\hat{\theta}^i$, $\breve{\theta}^i$ and $i_{\min}$, then the above equality says that a single $\theta_e(\mathbf{r}_{-e})$ can be computed in time linear in k. From (1) we need to compute the payments only for the N edges used in $A(\mathbf{r})$. In this way, by Definition 2, the overall computation of all such $P_e(x, \mathbf{r}_{-e})$ takes $O(\tau + k \cdot \tau_\mu + k \cdot N)$ time.

3 The Minimum Diameter Spanning Tree Problem

In the *minimum diameter spanning tree* (MDST) problem we are given a weighted undirected graph and the goal is to find a spanning tree which minimizes the longest path between any two nodes in that tree (the length of a path is the sum of the weights of its edges). In this section we study the corresponding mechanism design graph problem. Formally, given a graph $G = (V, E)$, the set $\mathcal{O}(G)$ of feasible solutions consists of all spanning trees; the set of used edges naturally consists of all edges in the tree, and the goal is to find a tree T of minimum *diameter*, that is, a tree such that the length of a maximum-length simple path in T is minimum. We denote this value by $d(G, \mathbf{t})$. Consider the following graph:



For all $t_e \leq 9$, *any* spanning tree is a MDST since, according to the edge weights $\mathbf{t}$ in the picture, the maximum-length simple path is the upper one and this path appears in any spanning tree. Unfortunately, the fact that an algorithm is exact for the MDST problem is not sufficient for obtaining a truthful mechanism. A well-known result by Myerson [13] (see also Archer and Tardos [2]) states that, for our problem, truthfulness can be achieved only if the algorithm is monotone (see Def. 1). It is possible to show that exact algorithms need not lead to truthful mechanisms (see [17] for the details).

In the sequel we will show that there exists an efficient polynomial-time algorithm for the MDST problem which is monotone and such that the payments can be computed efficiently. Both results follow from our main technique (Theorem 1).

3.1 A MIN-Reducible Algorithm for the MDST Problem

The computation of a MDST of a given graph can be reduced to the computation of a shortest paths tree rooted at the *absolute 1-center* (simply *center*, in the following) of G [8]. Loosely speaking, the center of a graph is a point c located on an edge (or on one of its endpoints) such that the distance from c to the farthest node is minimized. In particular, all edges are rectifiable, meaning that any point c on edge $f = (u, v)$ can be specified as a pair $c = (f, \lambda)$ with $\lambda \in [0, 1]$; in this case, we obtain a new graph G_c where edge (u, v) is replaced by two edges (u, c) and (v, c); their weights are $\overline{uc} := \lambda t_e$ and $\overline{vc} := (1 - \lambda)t_e$, respectively. (Notice that we consider each edge as an ordered pair of vertices.) Given a point c on f, one can build a spanning tree T_c of G by computing a shortest paths tree of G_c rooted at c, and then by replacing edges incident to c with the edge (u, v). Trivially, the tree T_c has diameter at most $2h_f^\lambda(\mathbf{t})$.[3] We let $h_f^*(\mathbf{t})$ be the minimum height among all shortest paths trees rooted at some point on f, that is, $h_f^*(\mathbf{t}) := \min_{\lambda \in [0,1]} h_f^\lambda(\mathbf{t})$. Our building block is the following algorithm which computes the *relative center* of edge f for the reported input $\mathbf{r}$, namely, a point $c = (f, \lambda)$ minimizing $h_f^\lambda(\mathbf{r})$:

[3] Formally, the tree T_c is obtained by removing c from the shortest paths tree and by adding back edge (u, v), unless c is sitting on one of the endpoints of (u, v) and is not connected to the other endpoint.

Algorithm CENTER$_f$

• compute the minimum $\lambda \in [0,1]$ such that $h_f^{\lambda}(\mathbf{r}) = h_f^*(\mathbf{r})$;
• compute the tree T_c for $c = (f, \lambda)$ and edge weights $\mathbf{r}$;
• return $Y = (T_c, c)$. /* return the tree T_c associated with the SPT and the center */

Since it holds that $d(G, \mathbf{r})/2 = h^*(\mathbf{r}) := \min_{f \in E} h_f^*(\mathbf{r})$ [8], we can compute a MDST by searching through all relative centers of G for a best possible position of the center:

$$A_{\text{MDST}} := \text{MIN}_h(\text{CENTER}_{e_1}, \ldots, \text{CENTER}_{e_m}),$$

where $e_1, \ldots, e_m$ denote the edges of G in some arbitrary order (independent of the agents' bids). We stress that a MDST cannot be obtained by restricting the computation of the relative center to one of the endpoints of edge f, that is, by considering only the vertices as possible center locations. This will produce a minimum radius spanning tree, instead, and thus the mechanism in [19] cannot be used here.

The following result, combined with Theorem 1, implies the existence of a truthful mechanism for the MDST (see the full version [17] for the proof).

Theorem 2. *Algorithm A_{MDST} is MIN-reducible and, on input a graph G with edge weights $\mathbf{r}$, it returns a MDST and an absolute center for this input. This computation requires $O(mn\,\alpha(n,n))$ time.*

We need one more step to guarantee that payments can be computed in strongly polynomial time. One of our major technical contributions is to show that the "MIN-reduction" can be done efficiently:

Theorem 3. *Algorithm A_{MDST} is MIN-reducible in $O(mn^2\,\alpha(n,n))$ time.*

Efficient Computations via Upper/Lower Envelopes (Proof Idea of Theorem 3). Let $\delta_{u,v}(G, \mathbf{r})$ be the (shortest path) distance from node u to node v in a graph G with weights $\mathbf{r}$. We compute the distances $\delta_{u,v}(G, \mathbf{r})$ and $\delta_{u,v}(G - e, \mathbf{r}_{-e})$, for all nodes u and v, and for all edges e used by the computed solution. Using the $O(mn \log \alpha(m,n))$-time all-pairs shortest paths algorithm by Pettie and Ramachandran [18], this step takes $O(mn^2 \log \alpha(m,n))$ time. (We have n graphs in total since the computed solution uses $n-1$ edges.) This term is dominated by $O(mn^2\alpha(n,n))$.

In the remaining of this section, we fix an edge e, and $\mathbf{r}_{-e}$, and an algorithm $A_i = \text{CENTER}_f$, and we show how to compute the thresholds $\theta_e^i(\mathbf{r}_{-e})$, $\hat{\theta}_e^i(\mathbf{r}_{-e})$ and $\breve{\theta}_e^i(\mathbf{r}_{-e})$ in $O(n\,\alpha(n,n))$ time. This implies Theorem 3 since there are m algorithms and $n-1$ agents/edges e used by the computed solution.

At the heart of the proof is an efficient method for computing, for any edge $f = (u,v)$, the following function in $O(n\,\alpha(n,n))$ time:

$$\hat{F}(\ell) := \inf\{x \mid h_f^*(x, \mathbf{r}_{-e}) \geq \ell\}. \tag{5}$$

This value can be computed by considering the *lower envelope* of n functions $\hat{f}_z(\ell, \lambda)$, one for each node z, defined as follows. The value $\hat{f}_z(\ell, \lambda)$ is the infimum value x for r_e such that the distance from a fixed center $c = (f, \lambda)$ to node z is at least ℓ. (Recall that x is the weight of edge e.) Such distance is the minimum of the following two functions, one for the path through u and one for the path through v:

$$upath_\lambda(x) := \lambda r_f + \min\{x + \delta_{u,z}(G, (0, \mathbf{r}_{-e})), \delta_{u,z}(G - e, \mathbf{r}_{-e})\}; \quad (6)$$

$$vpath_\lambda(x) := (1 - \lambda) r_f + \min\{x + \delta_{v,z}(G, (0, \mathbf{r}_{-e})), \delta_{v,z}(G - e, \mathbf{r}_{-e})\}. \quad (7)$$

These two functions are of the form $\lambda r_f + \min\{x + a, b\}$ and $(1-\lambda) r_f + \min\{x + a', b'\}$, respectively (see Fig. 1(left)).

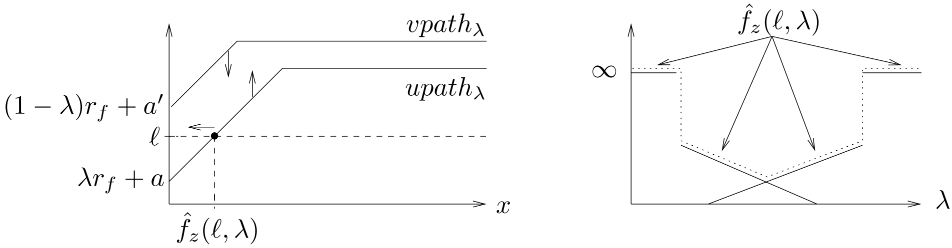


Fig. 1. From shortest paths distances to upper envelopes

We let $\hat{f}_z(\ell, \lambda) = \infty$ (respectively $\hat{f}_z(\ell, \lambda) = 0$) if, for all x, one function is below ℓ (respectively, both functions are not below ℓ). Otherwise, $\hat{f}_z(\ell, \lambda)$ is the x coordinate of the point where the *lowest* (i.e., smaller for $x = 0$) of the functions in (6-7) intersects with the limit ℓ. Notice that, when increasing λ by one unit, the two functions (6-7) move by r_f units as shown in Fig. 1(left). Hence, the point moves accordingly and thus the function $\hat{f}_z(\ell, \lambda)$ can be fully specified by two slanted segments as in Fig. 1(right). Each slanted segment is obtained by considering the intersection of each function in Fig. 1(left) with the limit ℓ. Hence, $\hat{f}_z(\ell, \lambda)$ is the dotted curve in Fig. 1(right) which is given by the upper envelope of the two solid curves in Fig. 1(right) (see [17] for the details).

In order to compute $\hat{F}(\ell)$, we consider $\hat{f}(\ell, \lambda) := \min_z\{\hat{f}_z(\ell, \lambda)\}$ and observe that $\hat{F}(\ell) = \sup_{\lambda \in [0,1]} \hat{f}(\ell, \lambda)$. The actual computation of $\hat{F}(\ell)$ consists in determining the *lower envelope* of all functions $\hat{f}_z(\ell, \cdot)$ and then finding its maximum for $\lambda \in [0, 1]$. Since the functions $\hat{f}_z(\ell, \cdot)$ intersect pairwise in at most one point, this requires $O(n\,\alpha(n, n))$ time using Agarwal and Sharir [1] approach, once the segments of each function have been computed. The latter can be obtained from the pre-computed distances using (6-7). Moreover, these distances allow us to compute the solution of algorithm $A_i = \mathrm{CENTER}_f$ and the values $\overline{g}_i$ and $g_{\min}$, still in $O(n\,\alpha(n, n))$ time. From the first step of CENTER_f and from (2) we obtain the following two identities, respectively: (i) $\theta^i_e(\mathbf{r}_{-e}) = \inf\{x |\ h^*_f(x, \mathbf{r}_{-e}) \geq \overline{g}_i\} = \hat{F}(\overline{g}_i)$; (ii) $\hat{\theta}^i_e(\mathbf{r}_{-e}) = \inf\{x |\ h^*_f(x, \mathbf{r}_{-e}) \geq \overline{g}_{\min}\} = \hat{F}(\overline{g}_{\min})$. Since the threshold $\breve{\theta}^i_e(\mathbf{r}_{-e})$ can be computed with a very similar approach, each of the $O(mn)$ thresholds can be computed in $O(n\,\alpha(n,n))$ time (see [17] for the details). Hence, Theorem 3 follows.

From Theorems 1, 2, and 3 we obtain the following:

Corollary 1. *There exists an $O(mn^2\,\alpha(n,n))$-time truthful mechanism for the MDST problem.*

4 Conclusions

We have described a general approach for building truthful mechanisms running in *strongly polynomial time* based on the 'MIN' operator defined by Mu'Alem and Nisan [12]. This is similar to what Kao *et al.* [10] propose, though their method for computing the payments assumes that each function $g_{A_i}(x)$ is *monotonically increasing* in $x < \theta_e(\mathbf{r}_{-e})$ (see the assumptions preceding Theorem 10 in [10]). This is too restrictive as the optimization functions used in the MRST and MDST do not fulfill this requirement and payments obtained from [10] do *not* guarantee truthfulness in these cases (in Example 1, their approach would ignore the tie-breaking rule among the algorithms).

Our technique has a very natural application to the MDST problem where the underlying algorithm in [8] optimizing the diameter $d(\cdot)$ can be rewritten as a 'MIN' combination of m algorithms optimizing a *different* function $h(\cdot)$, i.e., the height of a SPT rooted at the relative center of an edge. Although the results have been presented for mechanism design graph problems, they apply to a more general framework in which the agent valuations are either 0 or t_e, that is, to the *known single minded bidders* in [12] or, equivalently, to the *binary demand games* in [10]. The fact that we require plateau-like algorithms (instead of bitonic ones in [12]) does not directly prevent from optimal solutions (any bitonic algorithm minimizing the function $\mu(\cdot)$ is automatically plateau-like). Voluntary participation is guaranteed if optimal algorithms must drop an agent when its cost becomes too high. We can also obtain a strongly polynomial time truthful mechanism for the *p-center* graph problem [20] (in addition to the location of the p centers, we want to compute the associated trees), for any constant p. Notice that the problem is NP-hard for arbitrary p [20]. For the MRST, our method yields a mechanism which improves slightly the running time in [19]. (Details on both these problems are given in the full version [17].)

An interesting future direction is to apply our technique to NP-hard problems to obtain truthful approximation mechanisms (this was done in [10] for problems maximizing the *welfare*, i.e., the sum of all agents costs which obviously meet the "monotone increasing" requirement). According to Theorem 1, it suffices to show that an approximation algorithm is MIN-reducible in polynomial time. An interesting question here is whether the approximation ratio of the "best" approximation polynomial-time algorithm can be attained by some truthful polynomial-time mechanism.

Notice that our positive results cannot be extended to the case in which an agent owns several edges of a graph (these problems can model certain scheduling problems for which no exact truthful mechanism exists [15,2], while an extension of Theorem 1 would imply such an exact mechanism).

References

1. P.K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. *Cambridge University Press, New York*, 1995.
2. A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of 42nd FOCS*, pages 482–491, 2001.
3. E.H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, pages 17–33, 1971.
4. J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proceedings of the 6th DIALM*, pages 1–13. ACM Press, 2002.
5. T. Groves. Incentive in Teams. *Econometrica*, 41:617–631, 1973.
6. L. Gualà and G. Proietti. A truthful (2-2/k)-approximation mechanism for the Steiner tree problem with k terminals. In *Proc. of the 11th COCOON*, volume 3595 of *LNCS*, pages 90–400, 2005.
7. L. Gualà and G. Proietti. Efficient truthful mechanisms for the single-source shortest paths tree problem. In *Proc. of the 11th EURO-PAR*, volume 3648 of *LNCS*, pages 941–951, 2005.
8. R. Hassin and A. Tamir. On the minimum diameter spanning tree problem. *Info. Proc. Lett.*, 53(2):109–111, 1995.
9. J. Hershberger and S. Suri. Vickrey prices and shortest paths: what is an edge worth? In *Proc. of the 42nd FOCS*, pages 252–259, 2001.
10. M.-Y. Kao, X.-Y. Li, and W. Wang. Towards truthful mechanisms for binary demand games: A general framework. In *Proc. of ACM EC*, pages 213–222, 2005.
11. E. Koutsoupias and C.H. Papadimitriou. Worst-case equilibria. In *Proc. of STACS*, volume 1563 of *LNCS*, pages 404–413, 1999.
12. A. Mu'Alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proc. of 18th AAAI*, pages 379–384, 2002.
13. R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.
14. E. Nardelli, G. Proietti, and P. Widmayer. Finding the most vital node of a shortest path. *Theoretical Computer Science*, 296(1):167–177, 2003.
15. N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proc. of the 31st STOC*, pages 129–140, 1999.
16. C.H. Papadimitriou. Algorithms, games, and the Internet. In *Proc. of the 33rd STOC*, pages 749–753, 2001.
17. P. Penna, G. Proietti, and P. Widmayer. Strongly polynomial-time truthful mechanisms in one shot. Technical report, Università di Salerno, Available at http://ec.tcfs.it/Group/Selfish_Agents.html, 2006.
18. S. Pettie and V. Ramachandran. Computing shortest paths with comparisons and additions. In *Proc. of the 13th SODA*, pages 267–276, 2002.
19. G. Proietti and P. Widmayer. A truthful mechanism for the non-utilitarian minimum radius spanning tree problem. In *Proc. of SPAA*, pages 195–202, 2005.
20. B.C. Tansel, R.L. Francis, and T.J. Lowe. Location on networks: a survey. Part I: The p-center and p-median problems. *Management Sciences*, 29:482–497, 1983.
21. W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

Secretary Problems with Competing Employers

Nicole Immorlica[1], Robert Kleinberg[2,⋆], and Mohammad Mahdian[1]

[1] Microsoft Research, One Microsoft Way, Redmond, WA
{nickle, mahdian}@microsoft.com
[2] UC Berkeley Computer Science Division
and Cornell University Computer Science Department
rdk@cs.cornell.edu

Abstract. In many decentralized labor markets, job candidates are offered positions at very early stages in the hiring process. It has been argued that these early offers are an effect of the competition between employers for the best candidate. This work studies the timing of offers in a theoretical model based on the classical secretary problem. We consider a secretary problem with multiple employers and study the equilibria of the induced game. Our results confirm the observation of early offers in labor markets: for several classes of strategies based on optimal stopping theory, as the number of employers grows, the timing of the earliest offer decreases.

1 Introduction

An essential feature of many modern markets, particularly networked markets, is that they are *online*: information about agents, goods, and outcomes is revealed over time, and the agents must make irrevocable decisions before all of the information is revealed. A powerful tool for analyzing such scenarios is *optimal stopping theory*, the theory of problems which require optimizing an objective function over the space of stopping rules for a stochastic process. By combining optimal stopping theory with game theory, we can model the actions of rational agents applying competing stopping rules in an online market.

Perhaps the best-known optimal stopping problem is the *secretary problem*, also known as the *best-choice problem* or *marriage problem*. This problem was introduced in the 1960's as a model for studying online selection processes in the presence of a randomly ordered input. In the most basic version of the secretary problem, a decision-maker observes a sequence of elements of a totally-ordered set, presented in random order. At any time the decision-maker may stop the sequence and select the most recently presented element, with the objective of maximizing the probability of selecting the minimum element of the entire set. Dynkin [1] determined the optimal stopping rule for this problem and proved that its probability of success approaches $1/e$ as the size of the set tends to infinity. A myriad of subsequent papers have extended the problem by varying the objective function, varying the information available to the decision-maker, allowing for multiple choices, and so on, e.g. [2,3,4].

⋆ Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

P. Spirakis et al. (Eds.): WINE 2006, LNCS 4286, pp. 389–400, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Another line of work (see [5] for a survey) extends the secretary problem by studying scenarios in which two players compete to select the minimum element of a randomly-ordered sequence. If the original secretary problem can be thought of as the decision problem faced by an employer interviewing candidates for a job, then this variant should be thought of as the strategic problem faced by two employers interviewing the same sequence of candidates, when only one of them can hire the best candidate. Papers in this area differ in their assumptions about the game's payoff structure and about the ways in which conflicts (both players making an offer at the same time) are resolved. For example, Dynkin [6] proves a minimax theorem and characterizes the game value for a zero-sum version of the game in which conflicts are avoided by stipulating that in each step of the game, only one player is allowed to make an offer. Szajowski studies zero-sum stopping games with a tie-breaking rule which always gives priority to Player 1 [7] or which gives priority to a player chosen by a (possibly biased) random lottery [8]. A non-zero-sum version of the two-player game with unbiased random tie-breaking was studied by Fushimi [9], who observed that although the game is symmetric, the only pure Nash equilibria are asymmetric. Non-zero-sum stopping games in which Player 1 always receives priority [10,11] or in which priority is granted by a (possibly biased) random lottery [12] have also been studied. As far as we are aware, all of the prior work studies equilibria of the resulting games when just two employers compete.

In this paper, we first present our own variant of the two-employer game, providing a new derivation of the resulting symmetric two-player mixed-Nash equilibrium, and then proceed to study k-employer extensions of our game. One of the properties of any strategy for the employer game is its *threshold time*, defined as the largest fraction τ such that the strategy is guaranteed not to make an offer to any of the first τ candidates, no matter what the player observes nor what the other players do. We show that in any pure Nash equilibrium of the k-employer game, at least one of the players has a threshold time $\tau \leq 2/k$. This fact follows from a theorem which generalizes a striking feature of the one-player secretary problem: in any pure Nash equilibrium, the player with the earliest threshold time r_1 has probability exactly r_1 of winning the game. (In the one-player case, this specializes to the familiar fact that $1/e$ is both the probability of winning and the optimal threshold value.)

Next we consider a more realistic version of the game, in which the players are allowed to use *adaptive* strategies which base their decisions on the opponent's past actions as well as the public information revealed thus far. This defines a multi-player stochastic game. We describe the unique subgame perfect mixed Nash equilibrium of this game as the solution of a dynamic program. Using properties of the dynamic program, we prove that the timing of the first offer converges to zero as the number of players tends to infinity. More precisely, every player's threshold time is at most $1/k$.

In many labor markets, competition between employers often leads to an "unraveling" effect: employers wishing to attract the best candidates make early offers with short expirations. Such effects are quite pronounced in many markets.

In the market for law clerks, offers are made to candidates as early as the second year of law school, a full two years before graduations [13]. For a survey of markets which exhibit unraveling effects, see [14]. Our results can be interpreted as a theoretical justification for these unraveling effects. Namely, as the number of employers grows, our results confirm that the timing of the earliest offer in an equilibrium decreases.

2 Preliminaries

In this section we define a discrete-time and a continuous-time version of the game. The discrete-time version is conceptually simpler whereas the continuous-time version is more analytically tractable.

In the discrete-time game, we are given a totally-ordered set $\mathcal{U} = \{x_1 \prec x_2 \prec \ldots \prec x_n\}$, (representing the secretaries in decreasing order of value) and a set of $k \geq 1$ players (representing the employers). A random bijection $Z : \mathcal{U} \to [n]$ is chosen (representing the order in which the secretaries will be interviewed), but Z is not initially revealed to the players. (Here and throughout this paper, when m is a natural number we use $[m]$ to denote the set $\{1, 2, \ldots, m\}$.) As the game proceeds, each player is in a state which is either *active* or *inactive*; all players are initially active. At time $t = 1, 2, \ldots, n$, the relative ordering of the elements $Z^{-1}(1), Z^{-1}(2), \ldots, Z^{-1}(t)$ is revealed to the players. Each active player then chooses an action from the set $\{\mathsf{O}, \mathsf{P}\}$, whose elements are referred to as "offer" and "pass", respectively. If one or more active players chooses to offer, then one of these players (chosen uniformly at random) receives the element $x = Z^{-1}(t)$ and becomes inactive; this player is denoted by $\chi(x)$. The others remain active. If all active players choose to pass at time t, then all of them remain active and no player receives x. Each player is informed of the actions of all other players and of the identity of player $\chi(x)$, if defined. At the end of the game, all players receive a payoff of 0 except for $\chi(x_1)$ (if defined) who receives a payoff of 1.

The continuous-time variant of the game intuitively captures the limit of the discrete game as n tends to infinity. In most of this paper, we work on the continuous model, as this model hides details like integrality in our computations, hence making the computations cleaner, while still capturing the main ideas. It is not hard to generalize our results in the continuous model to the discrete model by incurring an additive error of $o(1)$; the details of this generalization is omitted from this extended abstract.

We now give an informal definition of the continuous-time model. The rigorous definition requires technical details and is deferred to the full version of the paper. In the continuous-time game, $\mathcal{U} = \mathbb{N}$ and $\prec$ denotes the usual ordering of $\mathbb{N}$ (hence the best element x_1 is 1). Each element x has an arrival time $Z(x)$ picked independently and uniformly at random from $[0, 1]$. Each player i has a (possibly randomized) strategy S_i, which at any time $t \in [0, 1]$ specifies whether or not player i makes an offer to the element arriving at time t (if any).[1] In general,

[1] Note that we can ignore zero-probability events such as the arrival of two elements at the exact same time.

$S_i(t)$ can depend on all the information revealed before time t (i.e., the ordering of the elements that have arrived by time t, and the offer/pass decisions made by other players before time t). We call such strategies *adaptive.* A simpler class of strategies, which correspond to stopping rules, are *non-adaptive* strategies and are defined as follows. For each $r \in [0, 1]$, the *non-adaptive strategy with threshold time* r is the strategy which makes an offer to every element observed after time r which outranks the best element arriving before time r, until it receives one of these elements and passes on all subsequent ones.

3 Analysis of Non-adaptive Strategies

3.1 Two Players

We begin our discussion with a warmup involving the computation of equilibria in the two-player game. We will work in the continuous-time model with $\mathcal{U} = \mathbb{N}$.

Pure Strategies. Our first calculation characterizes the set of pure strategy Nash equilibria. Our analysis is quite similar to that of Fushimi [9]. The difference in the results stems from the fact that our process continues until time $t = n$ unless both players make successful offers, whereas in Fushimi's model the process stops if both employers make an offer to a single element simultaneously.

Assume that players 1 and 2 use the non-adaptive pure strategies r, s, respectively. Let Y denote the following random subset of $\mathcal{U}$:

$$Y = \{y \in \mathcal{U} \mid Z(y) < Z(x_1)\},$$

and let $y_1 \prec y_2$ denote the two minimum elements of Y. We can then compute the resulting expected payoff to player 2 as follows.

Case 1 ($s > r$): Then player 2 wins if either

1. Player 1's earlier threshold caused him to make a sub-optimal offer, and player 2 made an offer to the best element: $Z(x_1) > s$ and $r < Z(y_1) \leq s$.
2. Both players made an offer to the best element, and player 2 won the coin toss: $Z(x_1) > s$ and $Z(y_1) \leq r$ and $\chi(x_1) = 2$.
3. Both players made offers to the same sub-optimal element, but player 2 lost the coin toss and then proceeded to make an offer to the best element: $Z(x_1) > s$ and $Z(y_1) > s$ and $Z(y_2) \leq r$ and $\chi(y_1) = 1$.

Conditioned on the arrival time $t = Z(x_1)$ of the best element, the random variables $Z(y_1), Z(y_2)$ are independently uniformly distributed in $[0, t]$. Therefore the probabilities of the three events listed above are respectively $\int_s^1 \frac{s-r}{t}\, dt$, $\frac{1}{2}\int_s^1 \frac{r}{t}\, dt$, and $\frac{1}{2}\int_s^1 \left(\frac{t-s}{t}\right)\left(\frac{r}{t}\right)\, dt$, and their sum integrates to $s \ln\left(\frac{1}{s}\right) - \frac{r(1-s)}{2}$.

Case 2 ($s \leq r$): Then player 2 wins if either

1. The best element arrives between the two thresholds and player 2 makes an offer: $s < Z(x_1) \leq r$ and $Z(y_1) \leq s$.

2. Both players make an offer to the best element, and player 2 wins the coin toss: $Z(x_1) > r$ and $Z(y_1) \leq s$ and $\chi(x_1) = 2$.
3. Both players made offers to the same sub-optimal element, but player 2 lost the coin toss and then proceeded to make an offer to the best element: $Z(x_1) > r$ and $Z(y_1) > r$ and $Z(y_2) \leq s$ and $\chi(y_1) = 1$.

The probabilities of the three events listed above are respectively $\int_s^r \frac{s}{t}\,dt$, $\frac{1}{2}\int_r^1 \frac{s}{t}\,dt$, and $\frac{1}{2}\int_r^1 \left(\frac{t-r}{t}\right)\left(\frac{s}{t}\right)dt$, and their sum integrates to $s\ln\left(\frac{1}{s}\right) - \frac{s(1-r)}{2}$.

Hence, when players 1 and 2 play the pure strategies r, s, respectively, the expected payoff to player 2 is

$$f(s) = \begin{cases} s\ln(1/s) - s(1-r)/2 \text{ if } s \leq r \\ s\ln(1/s) - r(1-s)/2 \text{ if } s \geq r \end{cases} \tag{1}$$

and the derivative of the expected payoff is

$$f'(s) = \begin{cases} \ln(1/s) - 3/2 + r/2 \text{ if } s < r \\ \ln(1/s) - 1 + r/2 \quad\;\; \text{if } s > r \end{cases} \tag{2}$$

Let $s_-(r), s_+(r)$ denote the best responses to r in the intervals $[0, r]$ and $[r, 1]$, respectively. It follows from (2) that

$$s_-(r) = \begin{cases} e^{r/2-3/2} \text{ if } \ln(1/r) - 3/2 + r/2 < 0 \\ r \qquad\quad\;\; \text{otherwise} \end{cases} \tag{3}$$

$$s_+(r) = \begin{cases} e^{r/2-1} \text{ if } \ln(1/r) - 1 + r/2 > 0 \\ r \qquad\;\;\; \text{otherwise} \end{cases} \tag{4}$$

By symmetry, player 1's best response function is identical. It follows easily from (3) and (4) that there is a pure Nash equilibrium (r, s) where $r = 0.27557...$ satisfies $2\ln(r) + 3 = \exp\left(\frac{r}{2} - 1\right)$ and $s = 0.42222...$ satisfies $2\ln(s) + 2 = \exp\left(\frac{s}{2} - \frac{3}{2}\right)$. In fact, (3) and (4) imply that the only two pure Nash equilibria are (r, s) and (s, r).

Mixed Strategies. The two-player game with non-adaptive strategies also has a symmetric mixed Nash equilibrium which we may compute explicitly. If player 1's choice of r is a random variable with density function $\nu(r)$ then we find, using (1), that player 2's expected payoff from playing strategy s is:

$$f(s) = \int_0^s \left[s\ln\left(\frac{1}{s}\right) - \frac{s(1-r)}{2}\right]\nu(r)\,dr + \int_s^1 \left[s\ln\left(\frac{1}{s}\right) - \frac{r(1-s)}{2}\right]\nu(r)\,dr,$$

and from (2) we obtain:

$$f'(s) = \ln\left(\frac{1}{s}\right) - 1 + \mathbb{E}\left(\frac{r}{2}\right) - \frac{1}{2}\Pr(r \geq s). \tag{5}$$

Let us assume that s has positive probability density in an interval (s_0, s_1) and zero probability of lying outside $[s_0, s_1]$. Then every $s \in [s_0, s_1]$ is a best response to player 1's mixed strategy, so $f'(s) = 0$ for $s \in [s_0, s_1]$. Since we are assuming a symmetric mixed Nash equilibrium, r also has zero probability of lying outside $[s_0, s_1]$, i.e. $\Pr(r \geq s_0) = 1$, $\Pr(r \geq s_1) = 0$. This implies, using (5) and the fact that $f'(s) = 0$ for $s \in [s_0, s_1]$, that:

$$\begin{aligned}
\ln(1/s_0) - 3/2 + \mathbb{E}(r/2) &= 0 \\
\ln(1/s_1) - 1 + \mathbb{E}(r/2) &= 0 \\
\ln(1/s_0) - \ln(1/s_1) &= 1/2 \\
s_1 &= \sqrt{e} \cdot s_0.
\end{aligned}$$

Taking the derivative of (5) we obtain:

$$f''(s) = -\frac{1}{s} + \frac{1}{2}\nu(s) = 0 \quad \text{for } s \in [s_0, s_1],$$

which implies $\nu(s) = 2/s$ for $s \in [s_0, s_1]$. Hence

$$\mathbb{E}(r/2) = \mathbb{E}(s/2) = \int_{s_0}^{s_1} (s/2)\,\nu(s)\,ds = \int_{s_0}^{s_1} ds = s_1 - s_0 = (\sqrt{e} - 1)s_0.$$

Recalling that $\ln(1/s_0) - 3/2 + \mathbb{E}(r/2) = 0$, we find that s_0 satisfies the equation

$$(\sqrt{e} - 1)s_0 = 3/2 + \ln(s_0),$$

i.e. $s_0 = 0.265...$, $s_1 = 0.437....$ Finally, we must verify that this is indeed a mixed Nash equilibrium by checking that the expected payoff function $f(s)$ is maximized when $s \in [s_0, s_1]$. To do so, it suffices to prove that $f'(s) > 0$ when $s < s_0$ and that $f'(s) < 0$ when $s > s_1$:

$$\boxed{s < s_0} \quad f'(s) = \ln(1/s) - 3/2 + \mathbb{E}(r/2) > \ln(1/s_0) - 3/2 + \mathbb{E}(r/2) = 0,$$
$$\boxed{s > s_1} \quad f'(s) = \ln(1/s) - 1 + \mathbb{E}(r/2) < \ln(1/s_1) - 1 + \mathbb{E}(r/2) = 0.$$

3.2 Multiple Players

In the previous section we saw that the two-player game has a pure Nash equilibrium $(r, s) = (0.27557..., 0.42222...)$. One striking feature of this equilibrium is that player 1's probability of winning is

$$f(r) = r\ln(1/r) - r(1 - s)/2 = 0.27557...$$

which is exactly equal to r. Recall that for the one-player game (i.e. the original secretary problem) the optimal strategy sets its threshold at time $1/e$ and has probability $1/e$ of winning. We begin this section with a theorem which shows that it is not coincidental that player 1's optimal threshold time and her probability of winning are exactly equal in both the one-player and two-player games. This phenomenon holds for every pure Nash equilibrium of the k-player game, for every k.

Theorem 1. *If* $(r_1, \ldots, r_k)$ *is a pure Nash equilibrium of the k-player non-adaptive game and* $r_1 \le r_2 \le \ldots \le r_k$*, then* $\Pr(\textit{player } 1 \textit{ wins}) = r_1$.

Proof. Fix the values of $r_2, r_3, \ldots, r_k$ and let $f(r)$ denote the probability that player 1 wins when the players use strategies $(r, r_2, r_3, \ldots, r_k)$. We will prove that there is a constant C, depending only on $r_2, \ldots, r_k$, such that $f(r) = r\ln(1/r) + Cr$ when $r \in (0, r_2]$. Since $r_1 = \arg\max f(r)$ and $r_1 \in (0, r_2]$ we have

$$0 = f'(r_1) = \ln(1/r_1) - 1 + C = \frac{f(r_1)}{r_1} - 1,$$

from which we conclude that $f(r_1) = r_1$ as claimed.

It remains to prove that $f(r) = r\ln(1/r) + Cr$ when $r \in (0, r_2]$. The probability that player 1 wins before time r_2 is

$$\int_r^{r_2} \frac{r}{t}\, dt = r\ln(1/r) - r\ln(1/r_2).$$

To compute the probability that player 1 wins after time r_2, let $Y = \{y \in \mathcal{U} \mid Z(y) < Z(x_1)\}$, and let $y_1 \prec y_2 \prec y_3 \prec \ldots$ be the elements of Y in sorted order. Note that, conditional on $Z(x_1) = t$, the random variables $\{Z(y)\}_{y \in Y}$ are independent and uniformly distributed in $[0, t)$. Let $A = \min\{a \mid Z(y_a) \le r\}$, and let $u_a = Z(y_a)$ for $a = 1, 2, \ldots, A$. If player 1 wins after time r_2 then it must be the case that $Z(y_a) > r_2$ for $a < A$. Let

$$g(u_1, u_2, \ldots, u_A) = \Pr(\chi(x_1) = 1 \parallel r, r_2, \ldots, r_k, u_1, u_2, \ldots, u_A).$$

Note that the value of g depends only on the relative ordering of the numbers in the set $S = \{r, r_2, \ldots, r_k, u_1, u_2, \ldots, u_A\}$. In particular, g is constant as u_A varies over the range $[0, r]$ because u_A, r are always the two smallest numbers in S. Now, letting $\mathcal{E}$ denote the event that player 1 wins after time r_2,

$$\begin{aligned}
\Pr(\mathcal{E}) &= \sum_{a=1}^{\infty} \Pr((A = a) \wedge \mathcal{E}) \\
&= \sum_{a=1}^{\infty} \int_{r_2}^{1} \left[\int_0^r \left(\int_{r_2}^1 \cdots \int_{r_2}^1 g(u_1, \ldots, u_a) t^{-a}\, du_1 \ldots du_{a-1} \right) du_a \right] dt \\
&= r \sum_{a=1}^{\infty} \int_{r_2}^{1} \left[\int_{r_2}^1 \cdots \int_{r_2}^1 g(u_1, \ldots, u_{a-1}, 0) t^{-a} du_1 \ldots du_{a-1} \right] dt \\
&= C' r,
\end{aligned}$$

where C' denotes the sum on the penultimate line of the equation above. Thus $f(r) = r\ln(1/r) + Cr$, with $C = C' - \ln(1/r_2)$.

Lemma 1. *In any pure Nash equilibrium of the k-player non-adaptive game, no player receives an expected payoff which is more than twice another player's expected payoff.*

Proof. Let p_i denote the expected payoff of player i. If $p_j > 2p_i$, then player i can deviate from the equilibrium by playing r_j instead of r_i. We will prove that this deviation yields an expected payoff of at least $p_j/2$ for player i, contradicting the assumption that $r_1, \ldots, r_k$ is a Nash equilibrium.

To prove that player i achieves an expected payoff of at least $p_j/2$ by changing her strategy to r_j, note first that players i and j have equal expected payoffs when they both play r_j. Now consider the change in player j's expected payoff in a series of two steps. First, player i changes her strategy from r_i to 1. (This is equivalent to player i leaving the game, since a player with a threshold time of 1 never makes an offer.) This change not decrease p_j. Second, player i changes her strategy from 1 to r_j. For every time $t \geq r_j$, this increases the number of active players at time t by at most one, so it decreases Pr(player j wins at time t) by at most a factor of 2. Thus, player j's expected payoff after this second change is at least $p_j/2$. This is equal to player i's expected payoff when she deviates by playing r_j, which concludes the proof of the lemma.

The following corollary confirms the observation of early offers in labor markets: it shows that in any pure Nash equilibrium of the k-player game, the earliest threshold time is $O(2/k)$. (An easy consequence of this is that the timing of the first offer converges to zero almost surely as $k \to \infty$.)

Corollary 1. *In any pure Nash equilibrium of the k-players non-adaptive game, if the players are numbered in order of increasing threshold times $r_1 \leq r_2 \leq \ldots \leq r_k$, then $r_1 \leq 2/k$.*

Proof. As before, let p_i denote the expected payoff of player i, for $i = 1, 2, \ldots, k$. Note that $\sum_i p_i \leq 1$ since the combined payoff of all players is at most 1. By Lemma 1 we have $p_i \geq p_1/2$ for all i. Hence $kp_1/2 \leq \sum_i p_i \leq 1$ which implies that $p_1 \leq 2/k$. By Theorem 1 we have $r_1 = p_1$, hence $r_1 \leq 2/k$.

4 Adaptive Strategies

When players are allowed to use adaptive strategies, it will be more convenient to adopt the discrete-time model of the game. This can be described as a stochastic game with state space $[n] \times [k]$. The interpretation of state (t, j) for $t \leq n$ is that there are j players active at time t, and an element x arrives at time t which is the best element observed so far. (More formally, there is an element $x \in \mathcal{U}$ with $Z(x) = t$ such that $x \prec y$ for all y with $Z(y) < t$.)

Proposition 1. *The adaptive k-player game has a unique symmetric subgame perfect equilibrium. This equilibrium can be described as follows: for each state $(t, j) \in [n] \times [k]$, there exist numbers $p(t, j), q(t, j)$ such that the equilibrium strategy of an active player in state (t, j) is to play* O *with probability $p(t, j)$,* P *with probability $q(t, j)$, regardless of the prior history of the game. Let $v(t, j)$ denote the probability that player i wins, given that the game is currently in state (t, j) and i is active. Then the values of $p(t, j), q(t, j), v(t, j)$ are correctly computed by the algorithm in Figure 1.*

```
/* Initialization */
for t = 1, 2, ..., n
    v(t, 0) ← 0;  w(t, 0) ← 0
end
for j = 1, ..., k
    p(n, j) ← 1;  q(n, j) ← 0;  w(n, j) ← 0;  v(n, j) ← 1/j
end

/* Dynamic program */
for j = 1, 2, ..., k
    for t = n − 1, n − 2, ..., 1
        w(t, j) ← w(t + 1, j) + v(t + 1, j)/(t^2 + t)
        A ← w(t, j − 1) − 1/n
        B ← w(t, j − 1) − w(t, j)
        if 1/n ≥ w(t, j − 1) /* Pure strategy O is optimal */
            q ← 0
        else if 1/n ≤ w(t, j) /* Pure strategy P is optimal */
            q ← 1
        else /* Mixed strategy is optimal */
            r ← the unique solution of 1 + x + x^2 + ... + x^(j−1) = jB/A in the interval (1, ∞)
            q ← 1/r
        p(t, j) ← 1 − q
        q(t, j) ← q
        v_O(t, j) ← tw(t, j − 1) − t(A/j)(1 + q + ... + q^(j−1))
        v_P(t, j) ← tw(t, j − 1) − tBq^(j−1)
        v(t, j) ← max{v_O(t, j), v_P(t, j)}
    end /* for t = n − 1, n − 2, ..., 1 */
end /* for j = 1, 2, ..., k */
```

Fig. 1. Dynamic program to compute the symmetric subgame perfect equilibrium of the adaptive game

Proof. See Appendix A.

Although the algorithm in Figure 1 is elaborate and does not yield a closed-form formula for the equilibrium strategy, it does enable us to draw some useful qualitative conclusions about this equilibrium. For example, the following theorem again confirms the observation of early offers in labor markets.

Theorem 2. *In the adaptive game with k players, assume all players use the symmetric equilibrium strategy described in Proposition 1. If there are j active players at time $t \geq n/j$ and the element x which arrives at time t is the best element observed so far, then all of the active players make an offer to this element.*

Proof. For all $j \geq 1$ and all $t \in [n]$, we have $v(t, j) \leq 1/j$ because each of the j active players in state (t, j) has probability $v(t, j)$ of winning, and these j events are mutually exclusive. It follows that

$$w(t, j-1) = \sum_{u=t+1}^{n} \frac{v(u, j-1)}{u(u-1)} \leq \frac{1}{j-1} \sum_{u=t+1}^{n} \frac{1}{u(u-1)} = \frac{1}{j-1} \left(\frac{1}{t} - \frac{1}{n} \right).$$

If $t \geq n/j$ then $1/t - 1/n \leq (j-1)/n$, so $w(t, j-1) \leq 1/n$. According to the algorithm in Figure 1, this implies that in equilibrium, all active players play O.

References

1. Dynkin, E.B.: The optimum choice of the instant for stopping a markov process. Sov. Math. Dokl. **4** (1963)
2. Ajtai, M., Megiddo, N., Waarts, O.: Improved algorithms and analysis for secretary problems and generalizations. SIAM J. Discrete Math. **14** (2001) 1–27
3. Freeman, P.: The secretary problem and its extensions: a review. Internat. Statist. Rev. **51** (1983) 189–206
4. Gilbert, J.P., Mosteller, F.: Recognizing the maximum of a sequence. J. Amer. Statist. Assoc. **61** (1966) 35–73
5. Sakaguchi, M.: Optimal stopping games: A review. Math. Japonica **42** (1995) 343–351
6. Dynkin, E.: Game variant of a problem on optimal stopping. Sov. Math. Dokl. **10** (1969) 270–274
7. Szajowski, K.: Double stopping by two decision-makers. Adv. in Appl. Probab. **25** (1993) 438–452
8. Szajowski, K.: Optimal stopping of a discrete markov process by two decision makers. SIAM J. Control Optim. **33** (1995) 1392–1410
9. Fushimi, M.: Secretary problem in a competitive situation. Journal of the Operations Research Society of Japan **24** (1981) 350–358
10. Ravindran, G., Szajowski, K.: Nonzero sum game with priority as Dynkin's game. Math. Japonica **37** (1992) 401–413
11. Szajowski, K.: On nonzero sum game with priority in the secretary problem. Math. Japon. **37** (1992) 415–426
12. Szajowski, K.: Markov stopping games with random priority. Z. Oper. Res. **39** (1994) 69–84
13. Avery, C., Jolls, C., Posner, R., Roth, A.: The market for federal judicial law clerks. University of Chicago Law Review **68** (2001) 793–902
14. Roth, A., Xing, X.: Jumping the gun: Imperfections and institutions related to the timing of market transactions. American Economic Review **84** (1994) 992–1044
15. Billingsley, P.: Probability and Measure. John Wiley (1995)

A Proof of Proposition 1

We prove the claims in the proposition by induction on states, in the order they are considered by the algorithm, i.e. downward induction on t and upward induction on j. It will be helpful to maintain an additional induction hypothesis that

$$w(t, j) = \sum_{u=t+1}^{n} \frac{v(u, j)}{u(u-1)},$$

where the right side is interpreted as 0 when $t = n$.

The base cases $j = 0, t = 1, 2, \ldots, n$ are trivial. (No players are active, so there is no need to compute the equilibrium strategies, only the values of $v(t,j)$ and $w(t,j)$.) The base cases $t = n, j = 1, \ldots, k$ are also trivial: if the n-th element is the best one observed so far, there is no reason for any active player to play P. All of them will play O, and each of them has probability $1/j$ of winning.

For the induction step, it is easy to check that $w(t,j)$ satisfies the induction hypothesis given that $w(t+1,j)$ does. To verify the induction hypothesis for $v(t,j), p(t,j), q(t,j)$ requires an elaborate calculation which we now explain. First let us define several events which will appear in the conditional probability expressions that define the transition probabilities of the stochastic game.

$$\begin{aligned}
\mathcal{E}(t,j) &= \{\text{the game visits state } (t,j)\} \\
\mathcal{E}_{\mathsf{P}}(t,j) &= \mathcal{E}(t,j) \cap \{\text{all players pass at time } t\} \\
\mathcal{E}_{\mathsf{O}}(t,j) &= \mathcal{E}(t,j) \cap \{\text{at least one player makes an offer at time } t\} \\
\mathcal{E}((t,j) \to (u,\ell)) &= \{\text{the game makes a state transition from } (t,j) \text{ to } (u,\ell)\} \\
\mathcal{E}((t,j) \to \bullet) &= \{\text{the game visits state } (t,j) \text{ and } Z^{-1}(t) = x_1.\}
\end{aligned}$$

The following conditional probabilities can be calculated using arguments analogous to the calculations of transition probabilities for Markov decision process representing the one-player secretary problem (e.g. [15]).

$$\begin{aligned}
\Pr(\mathcal{E}((t,j) \to \bullet) \parallel \mathcal{E}_{\mathsf{P}}(t,j)) &= t/n \\
\Pr(\mathcal{E}((t,j) \to \bullet) \parallel \mathcal{E}_{\mathsf{O}}(t,j)) &= t/n \\
\Pr(\mathcal{E}((t,j) \to (u,j)) \parallel \mathcal{E}_{\mathsf{P}}(t,j)) &= t/(u(u-1)) \\
\Pr(\mathcal{E}((t,j) \to (u,j-1)) \parallel \mathcal{E}_{\mathsf{O}}(t,j)) &= t/(u(u-1)).
\end{aligned}$$

Using these conditional probabilities, we wish to calculate the expected payoff to player i when playing O or P in state (t,j) given that the other players are all playing O with probability p, P with probability q. Let us denote the expected payoff to player i in these two cases by $v_{\mathsf{O}}(t,j), v_{\mathsf{P}}(t,j)$, respectively. A simple case analysis combined with the conditional probability formulas above yields:

$$v_{\mathsf{O}}(t,j) = \sum_{i=1}^{j} \binom{j-1}{i-1} p^{i-1} q^{j-i} \left[\frac{1}{i} \cdot \frac{t}{n} + \left(1 - \frac{1}{i}\right) \sum_{u=t+1}^{n} \frac{tv(u,j-1)}{u(u-1)} \right] \tag{6}$$

$$v_{\mathsf{P}}(t,j) = q^{j-1} \sum_{u=t+1}^{n} \frac{tv(u,j)}{u(u-1)} + \left(1 - q^{j-1}\right) \sum_{u=t+1}^{n} \frac{tv(u,j-1)}{u(u-1)} \tag{7}$$

Let $A = w(t,j-1) - 1/n$, $B = w(t,j-1) - w(t,j)$. We may simplify equations (6) and (7) considerably, obtaining:

$$v_{\mathsf{O}}(t,j) = tw(t,j-1) - (t/j)A \frac{1-q^j}{1-q} \tag{8}$$

$$v_{\mathsf{P}}(t,j) = tw(t,j-1) - q^{j-1} tB \tag{9}$$

$$\frac{1}{t}(v_{\mathsf{O}}(t,j) - v_{\mathsf{P}}(t,j)) = q^{j-1} B - (A/j) \frac{1-q^j}{1-q}. \tag{10}$$

From (10) we see that the set of best responses for player i is $\{\mathsf{O}\}, \{\mathsf{P}\}$, or $\{\mathsf{O}, \mathsf{P}\}$ according to whether the value of the function $f(q) = q^{j-1}B - (A/j)(1 + q + \ldots + q^{j-1})$ is positive, negative, or zero.

When $j = 1$ this computation derives the optimal policy for the ordinary (one-player) secretary problem. We have $A = -1/n$, $B = -w(t, 1)$, $f(q) = B - A = 1/n - w(t, 1)$. Let t_1 be the largest value of t such that $w(t, 1) > 1/n$. For all $t > t_1$, $f(q) > 0$ and the unique best strategy in state $(t, 1)$ is O; moreover for $t > t_1$ we have $v(t, 1) = t/n$ and $w(t, 1) = \frac{1}{n}\sum_{u=t}^{n-1}\frac{1}{u}$. From this formula for $w(t, 1)$ we deduce that $t_1 \sim n/e$.

When $j > 1$ and $t < n$, let us first observe that $B > 0$. To see this, note that $B = \sum_{u=t+1}^{n}(v(u, j-1) - v(u, j))/(u(u-1))$, that each term of this sum is non-negative because increasing the number of active players at time t can not increase player i's probability of winning after time t, and that the final term is strictly positive. If $A \leq 0$ then $f(q) > 0$ for all q, which implies that O is the unique best response for player i and therefore (since we are assuming a symmetric equilibrium) $q = \Pr(i \text{ plays } \mathsf{P}) = 0$. Conversely, if $q = 0$ then O is in player i's best response set, which implies that $f(q) = f(0) \geq 0$ and therefore $A \leq 0$. Recalling that $A = w(t, j-1) - 1/n$, we have derived:

$$q = 0 \iff w(t, j-1) \leq 1/n. \tag{11}$$

Now we come to the case $q > 0$. We know that in this case, $A > 0$. Note that $f(q)$ has the same sign as $q^{1-j}f(q)$ and that

$$q^{1-j}f(q) = B - (A/j)(1 + q^{-1} + \ldots + q^{-(j-1)}). \tag{12}$$

Letting $r = 1/q$, the right side of (12) is the polynomial $g(r) = B - (A/j)(1 + r + \ldots + r^{j-1})$ which is monotonically decreasing as a function of $r \geq 1$ because $A > 0$. If $g(1) = B - A \leq 0$ and $q < 1$, then $r > 1$ and $g(r) < 0$, which implies that player i's unique best response is P contradicting the fact that $q < 1$ and that we are in a symmetric equilibrium. Hence we see that $B - A \leq 0$ implies $q = 1$. Conversely, if $q = 1$ then P is in player i's best response set and $f(1) = g(1) = B - A \leq 0$. Recalling that $B - A = 1/n - w(t, j)$, we have derived:

$$q = 1 \iff 1/n \leq w(t, j). \tag{13}$$

Finally, if $0 < q < 1$, then player i's best response set is $\{\mathsf{O}, \mathsf{P}\}$ which implies that $g(r) = 0$, i.e. $1 + r + \ldots + r^{j-1} = jB/A$. Since g is monotonic in the interval $(1, \infty)$, the equation $g(r) = 0$ can have at most one solution in this interval. In fact there is a solution in this interval by the intermediate value theorem, since $g(1) = B - A > 0$ (by our assumption that $q < 1$) and $g(r)$ tends to $-\infty$ as $r \to \infty$ (by our assumption that $A > 0$, as follows from the fact that $q > 0$).

This concludes the verification that there is a unique symmetric mixed Nash equilibrium in state (t, j), and that the algorithm correctly computes this equilibrium. The verification that the algorithm correctly computes $v(t, j)$ — which is the final part of establishing the induction step — is a trivial consequence of formulas (8) and (9) above.

Author Index