K.L. Katsifarakis

Hydrology, Hydraulics and Water Resources Management

A Heuristic Optimization Approach

WIT PRESS

Hydrology, Hydraulics and Water Resources Management

A Heuristic Optimization Approach

WIT*PRESS*
WIT Press publishes leading books in Science and Technology.
Visit our website for the current list of titles.
www.witpress.com

WIT*eLibrary*
Home of the Transactions of the Wessex Institute, the WIT electronic-library provides the international scientific community with immediate and permanent access to individual papers presented at WIT conferences.
Visit the WIT eLibrary athttp://library.witpress.com

This page intentionally left blank

Hydrology, Hydraulics and Water Resources Management

A Heuristic Optimization Approach

Edited by

K.L. Katsifarakis
Aristotle University of Thessaloniki, Greece

WIT PRESS Southampton, Boston

K.L. Katsifarakis
Aristotle University of Thessaloniki, Greece

Published by

WIT Press
Ashurst Lodge, Ashurst, Southampton, SO40 7AA, UK
Tel: 44 (0) 238 029 3223; Fax: 44 (0) 238 029 2853
E-Mail: witpress@witpress.com
http://www.witpress.com

For USA, Canada and Mexico

WIT Press
25 Bridge Street, Billerica, MA 01821, USA
Tel: 978 667 5841; Fax: 978 667 7582
E-Mail: infousa@witpress.com
http://www.witpress.com

British Library Cataloguing-in-Publication Data
A Catalogue record for this book is available
from the British Library

ISBN: 978-1-84564-664-6
eISBN: 978-1-84564-665-3

Library of Congress Catalog Card Number: 2012934918

Printed by Lightning Source, UK.

Contents

Preface

With the population of our planet exceeding seven billion, funds for infrastructure works being limited worldwide and climate change affecting water resources, their optimal development and management is literally vital.

This volume deals with application of some non-traditional optimization techniques to hydraulics, hydrology and water resources management.

Chapter 1 is a brief introduction to optimization and its application to water resources management. Emphasis is given to the definition of optimization problems and to the role of flow simulation models.

Chapter 2 is dedicated to genetic algorithms. It starts with a concise description of the method followed by an instructive example. Then a number of indicative applications is presented, mainly on groundwater resources management (including coastal aquifers and water quality issues), the inverse problem of groundwater hydraulics, hydrology and reservoir operation.

Chapter 3 also deals with genetic algorithms. It focuses, though, on applications to hydraulic networks, mainly to irrigation ones. A real-coded genetic algorithm to minimize energy and pipeline investment cost is presented, its original features are described and its application to collective on-demand irrigation networks is explained by means of two application examples.

Chapter 4 is dedicated to simulated annealing. The theoretical foundation of the method is presented, followed by a description of general and specific aspects of the respective algorithm. Three applications, namely on aquifer systems, on water distribution and on wastewater systems conclude the chapter.

The particle swarm method (PSO) is discussed in Chapter 5. The basics of

the method are provided first, followed by three modifications that considerably improve the performance of the standard algorithm. Then application of PSO to multi-objective optimization problems is discussed. Finally, a number of applications is presented, with emphasis on optimal design of water distribution networks.

Tabu search comes next. In Chapter 6 the basic concepts and features of the method are presented and its coupling with other heuristic optimizers such as genetic algorithm and simulated annealing is discussed. Applications of such hybrid optimizers on groundwater optimization problems conclude the chapter.

Chapter 7 is dedicated to the Harmony Search method. First, the concept and the basic characteristics of the harmony search algorithm are comprehensively presented. Then, some variations aiming at improving its efficiency are outlined and evaluated by means of two benchmark problems. Finally an application on a classic multi-reservoir system management problem is presented.

Finally, Chapter 8 deals with the Outer Approximation method. In its first part the mathematics of the method are presented and its main characteristic, namely the minimization of a concave objective function over a convex or concave feasible domain, is discussed. The second part of the chapter includes applications of the outer approximation method to groundwater optimal design problems.

Hopefully, this volume will help scientists dealing with water resources issues to reach the best decisions, at least from the technical point of view. One should keep in mind, though, that overall quality of the results depends also on the definition of the optimization goals, namely the so-called objective function. These, in turn, as in all engineering and scientific problems, are subordinate to principles and ethics that should always be taken into consideration.

K.L. Katsifarakis
Department of Civil Engineering
Aristotle University of Thessaloniki
Greece

CHAPTER 1

Optimization and water resources management

K.L. Katsifarakis
Department of Civil Engineering, Aristotle University of Thessaloniki, Greece.

Abstract

This introductory chapter includes some thoughts on the essence of optimization and on its role in water resources management. First, the relevance of the notion of optimization to most scientific fields is pointed out, using the example of Aristotle's Ethics. Then the importance of optimal management of water resources is stressed and certain issues related to the definition of optimization problems and to the role of flow simulation models are selectively discussed.
Keywords: ethics, flow simulation, optimization, water resources management.

1 The concept of optimization

Optimization could be defined rather simply as culmination of an improvement process. Optimization problems are encountered more or less frequently, in almost every scientific field (e.g., Floudas and Pardalos [1]). Schwefel [2], quoted by Michalewicz [3], puts it in the following way:

> "There is scarcely a modern journal, whether on engineering, economics, management, mathematics, physics, or the social sciences, in which the concept 'optimization' is missing from the subject index, If one abstracts from all specialist points of view, the recurring problem is to select a better or best (according to Leibniz, optimal) alternative from among a number of possible states of affairs."

Even when the term is not explicitly used, the notion of optimization can be traced in very diverse scientific fields. A prominent example can be found in Aristotle's Ethics [4], where ethical virtue is defined as "μεσότης" (usually translated in English as mean or mean state), between two extremes: a deficiency and an excess. Tassios, though, has proposed that optimization is a more appropriate term [5]. Most probably this interpretation is closer to the philosopher's mind, for the following reasons:

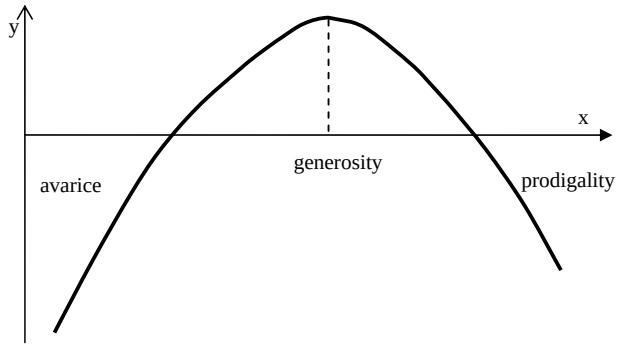


Figure 1: "Mean state," deficiency and excess, regarding wealth management.

Aristotle cautions us not to consider each virtue as similar to an arithmetic mean between the respective extremes. Moreover, he stresses that it is an extreme, with respect to excellence and rightness.

Aristotle also mentions that one extreme could be worse than the other and that the respective virtue could be closer to one of them, which could be even mistaken for virtue. An example is bravery and its excess.

Moreover, he states that one could err in many ways, while the correct behavior is unique, as if speaking for a function with one extreme value only. At the same time, he stresses that we do not blame those who deviate from the optimum only slightly.

Based on these remarks, we could construct the diagram of Fig. 1, regarding wealth management. Rightness is shown on the *y*-axis, whereas disposition toward getting and giving money on the *x*-axis.

By the way, Aristotle proposes an interesting way to arrive to the optimum: (a) avoid first the extreme which is the more opposed to the mean and (b) notice the errors to which one is more prone and move as far as possible toward the opposite direction.

As optimization penetrates very different scientific areas, its character (qualitative or quantitative) and its precision depend on the features of the particular application field. It is worth mentioning, though, that even when we concentrate to engineering problems, we may have to deal with costs and benefits that cannot be easily expressed in monetary units. This is the case of the so-called intangibles, such as the aesthetic value of a spring or a river, or the social cost of the displacement of the inhabitants of a village in order to build a dam and an artificial lake.

1.1 Optimization techniques

There are already many optimization methods, starting from simple differentiation (if the features of the problem permit its application) to sophisticated heuristic techniques. As the "no free lunch" theorem states, no method is superior to all others for every problem (e.g., Wolpert & Macready [6]); or, as Reeves & Raw [7] put it,

"there is no royal road to optimization." Moreover, there is no general rule for selecting the most suitable method, not even the best features of a certain method. Experience, intuition, and use of more than one optimization approaches may help with this process.

2 Optimal use of water resources

Fresh water availability is a basic prerequisite for the development of human activities. Unfortunately, their distribution is uneven in space and time. Moreover, human activities are often concentrated in areas with poor water resources, such as coastal areas, which are also vulnerable to salt water intrusion. Meanwhile: (a) water resources availability may tend to decrease in many parts of the world, due to unfavorable climatic change (i.e., reduction of precipitation or change of its pattern); (b) population increases continuously, particularly in poor countries; and (c) per capita water demand grows, following quality of life standards. Under these circumstances, the call for optimal management of both the supply and the demand side of fresh water balance is urgent.

2.1 Definition of an optimization problem

The first step for solving an optimization problem is to accurately define it. When one deals with water resources management, this might prove a difficult task, involving a set of assumptions. A holistic approach requires consideration of both sides of the water balance. Nevertheless, dealing with water demand is inherently and explicitly a very approximate process. It requires accurate data on and projections of population growth, per capita water demand, agricultural, tourism, and industrial growth. So, in many cases some rough estimates are used as data instead of variables in the optimization problem and elasticity of the demand is ignored.

The supply side of the water balance is generally considered a more accurate and precise engineering exercise. Optimal management usually requires combined use of surface and groundwater resources. Optimization of each resource, though, is in many cases separately considered, despite hydraulic interaction between the two. The benefit of this separation is simplification of the problem. The pitfall, on the other hand, is that the global optimum may not coincide with the sum, or combination, of partial optima. It is safe to optimize separately "closed" subsystems only.

In many cases, water quantity cannot be considered separately from water quality. This is quite clear in coastal aquifers, for instance, where saltwater intrusion sets the limit to water extraction. Moreover, development priorities may be based on water quality considerations. Usually, groundwater resources have better quality, since they are naturally more protected. It should be mentioned though, that if they are polluted, their restoration is more difficult.

Energy use is an important issue, too. In some cases, it enters as a discrete variable in water resources management optimization problems, for example, when desalination is considered as a water resource option. Moreover, energy production

is usually part of the objective function in reservoir management problems. Even when not explicitly considered, though, energy may be the decisive factor. Pumping cost, for instance, is actually energy cost.

During problem formulation, it is possible to leave out a secondary factor, because its mathematical expression is difficult, or because its inclusion leads to disproportionate increase of the overall computational load. In these cases, use of optimization techniques that end up with more than one good solutions is advantageous; they allow a final choice that will take into account the omitted factor. This point is also discussed in Chapter 2 of this volume, in connection with the method of genetic algorithms.

2.2 Optimization and flow simulation

An important part of most water resources optimization problems is flow simulation. It results are used either in the objective function or in constraint-related calculations. If water quality is involved, mass transport should be simulated, too. So, accuracy of the results of optimization models depends on the choice of the flow simulation models that are involved.

The accuracy of flow simulation models depends on the validity of underlying assumptions (e.g., steady-state flow, constant aquifer properties, etc.). Computational load increases with the complexity of the model. Total computational volume may be a restricting factor in model choice, in particular when evolutionary optimization techniques are used. In such cases, use of surrogate (simplified) models might be the best choice for the overall accuracy of the optimization process. This point is further discussed in Chapter 2 of this volume and, more extensively, in [8].

Moreover, accuracy of the results obtained even by the most sophisticated simulation models, depends on the quantity and quality of input data. Data collection presupposes the existence of a suitable monitoring network, which might have substantial cost. For this reason, input data are quite often insufficient. In such cases, a good solution, that is rather insensitive to changes of input parameters values, might be a better choice than an “optimal” solution, which is very sensitive to such changes.

3 Concluding remarks

With population of our planet exceeding seven billion already, funds for infrastructure works being limited worldwide, and climate change affecting water resources, their optimal development and management is literally vital. The development of optimization tools and methods and their use in water resources management could be very helpful in meeting the aforementioned challenge. The rest of this book is dedicated to them. Hopefully, this introductory chapter has pointed out two, superficially contradictory, things: (a) that optimization methods used in water resources can be transferred successfully to quite different scientific fields and (b) that optimization techniques are mere tools, and understanding their limitations (i.e., their validation) is a prerequisite for their correct use.

References

[1] Floudas, C.A. & Pardalos, P.M. (eds), *Encyclopedia of Optimization*, 2nd edn. Springer: Berlin, 2008.
[2] Michalewicz, Z., *Genetic algorithms + Data structures = Evolution programs*, Springer-Verlag: Berlin Heidelberg, 1996.
[3] Schwefel, H.-P., *Numerical Optimization for Computer Models*, John Wiley & Sons: Chichester, UK, 1981.
[4] Aristotle, *Nicomachean Ethics (Book 2)*, 1106–1109.
[5] Tassios, Th., A different approach to Aristotle's mean state. *Unpublished document*, 2005.
[6] Wolpert, D.H. & Macready, W.G., No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, **1(1)**, pp. 67–82, 1997.
[7] Reeves, C.R. & Raw, J.E., *Genetic Algorthms – Principles and Perspectives*, Kluwer Academic Publishers: Dordrecht, the Netherlands, 2003.
[8] Katsifarakis, K.L., Flow simulation and optimal management of groundwater resources. The balance between accuracy and computational efficiency (Chapter 11). *Groundwater: Modelling, Management and Contamination*, eds L.F. Konig & J.L. Weiss, Nova Science Publishers: Commack, NY, ISBN 978-1-60456-832-5, pp. 291–308, 2009.

This page intentionally left blank

CHAPTER 2

Genetic algorithms and water resources management: an established, yet evolving, relationship

K.L. Katsifarakis[1] & D.K. Karpouzos[2]
[1]*Department of Civil Engineering, Aristotle University of Thessaloniki, Greece.*
[2]*Faculty of Agriculture, Aristotle University of Thessaloniki, Greece.*

Abstract

This chapter is dedicated to the method of genetic algorithms and its application to water resources management problems. It consists of two parts. The first one includes a concise overview of the method, namely: (a) Description of its basic features, (b) Presentation of the basic genetic operators (selection, crossover, and mutation), and of some additional ones, too, (c) Outline of techniques for handling constraints and of conditions for the termination of the optimization process, and (d) Discussion on the use of surrogate flow models. The overview is followed by an application example to a simple groundwater management problem, namely optimal distribution of a given total flow rate to a number of wells. It aims at clarifying the basic notions and the use of genetic algorithms (from chromosome length to constraint handling).
The second part of the chapter includes brief presentations of indicative applications of genetic algorithms to: (a) groundwater resources management (with special subsections on coastal aquifers and on groundwater remediation and pollution control), (b) the inverse problem of groundwater hydraulics, (c) surface water hydrology, (d) reservoir management, and (e) related optimization problems. These applications demonstrate the wide range of applicability, the usefulness, and the versatility of the method. The chapter ends with brief concluding remarks.
Keywords: genetic algorithms, groundwater resources management, hydrology, inverse problem, optimization, reservoir operation, water pollution.

1 Introduction

Genetic algorithms are a mathematical tool with a very wide range of applications. They are particularly efficient in optimization problems, especially when the respective objective functions exhibit many local optima or discontinuous derivatives. This means that they could be very useful in difficult problems, where more conventional local search methods fail.

Genetic algorithms were initially introduced by Holland [1]. The respective literature is quite rich, including books exclusively or partially dedicated to them, e.g. Goldberg [2], Rawlins [3], Michalewicz [4], Coley [5], Reeves and Raw [6], Sivanandam and Deepa [7], and Yu and Gen [8]. Therefore, their theoretical basis has been extensively discussed. This chapter does not aim at going deep into theory, but rather at serving as a practical guide for those who would like to enjoy the use of this powerful optimization tool, in particular in the fields of hydrology, hydraulics, and water resources management.

2 Basic features

As their name implies, genetic algorithms are essentially a simplified mathematical imitation of a biological process, namely that of evolution of species. The influence of biology is ubiquitous in the respective terminology. Each potential solution of a problem is an individual, which is represented by a chromosome. In a pumping cost minimization problem, for instance, involving pumping from 10 wells, a chromosome would represent a set of 10 well flow rate values, concatenated with each other.

Classical genetic algorithms are based on binary encoding, namely each chromosome is a binary string, e.g. [01001110], that can be interpreted as a set of binary (or Gray) numbers. Each digit of the string is called gene (or character) and the different values that it can take are called alleles. Some authors, though, define a gene as a subset of the chromosome that represents a decision variable.

One more loan from biology is the set of terms genotype and phenotype. The former means the form (type) of the individual at the gene level, namely the respective encoded structure that permits application of genetic operators. Phenotype, derived from the Greek words φαίνομαι (phenomai = appear, look like) and type, means the respective decoded set of decision variables.

The chromosome length (CL), namely the number of its genes, depends on the range of values of the encoded parameters and the accuracy that we seek for the solution. If, for instance, a parameter can vary between the real values $a1$ and $b1$ and an accuracy of d decimal points is sought, the required number of binary digits is the smallest integer m that fulfils the following inequality:

$$2^m - 1 \geq (b1 - a1)\ 10^d \tag{1}$$

If very high accuracy is required, binary CL may become excessive. Moreover, binary representation may not be convenient for a number of problems. In such cases, real-coded genetic algorithms, where each variable is represented by one gene, are a better choice. Actually, their use is gaining momentum over the years. Octal, hexadecimal, and other encodings have also been used.

To solve a problem, genetic algorithms start with a set of chromosomes, in most cases randomly generated at any area of the search space (namely the set of all

possible solutions). Yet, one could take the advantage of information on probable locations of the optimal solution. The chromosomes of this set constitute the initial population, or the first generation. Their number, called population size (PS), is defined by the user. In most cases, PS remains constant throughout the optimization process. Still, genetic algorithms with variable PS seem to offer some advantages in certain cases (e.g. Koumousis and Katsaras [9]).

Each chromosome of the first generation undergoes evaluation, by means of a pertinent function or process. This process depends entirely on the specific application of genetic algorithms and aims at attributing a fitness value VB(I) to each chromosome I. Then the next generation is produced, by means of certain operators, which are inspired by biological processes and apply to the chromosomes of the current generation.

The main genetic operators are (a) selection, (b) crossover, and (c) mutation. Many other operators have been also proposed and used.

2.1 Selection

Selection is used to form the 'intermediate' population between two generations and it is essentially a mathematical imitation of Darwin's theory of survival of the fittest. It is accomplished in a predetermined 'random' way. All chromosomes are assigned a survival probability, depending on their fitness, which has been calculated during the chromosome evaluation process. Therefore, the comparatively better chromosomes have more chances to be selected for the intermediate population. The most common selection processes are (a) the biased roulette wheel and (b) the tournament method.

The first process is based on the idea of a roulette wheel having PS unequal slots, as shown in Fig. 1. The size of each slot is proportional to the fitness of the respective chromosome. The roulette wheel is 'spinned' PS times. In every round, the selection probability $p(I)$ of a chromosome I (in maximization problems) is equal to:

$$p(I) = \frac{\mathrm{VB}(I)}{\mathrm{SVB}} \tag{2}$$

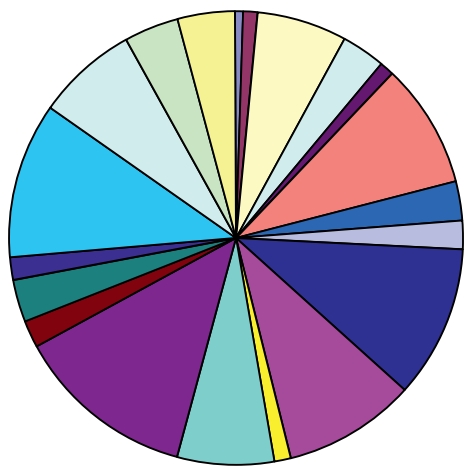


Figure 1: Biased roulette wheel (PS = 20).

where SVB is the sum of the fitness values of all chromosomes of current generation. Its computational application is quite simple: First, we calculate SVB. At the same time, the fitness value VB(J) of chromosome J is replaced by the sum of the VB(I) values for $I=1$ to J. Then a number XXX between 0 and SVB is randomly selected. If VB($K-1$) < XXX < VB(K), then a copy of chromosome K is added to the intermediate population. This process is repeated PS times. In this way, the intermediate population is formed, including (statistically) more copies of the best chromosomes, which eventually replace some of the worst ones. It should be mentioned, though, that even the worst chromosome has a tiny chance to 'survive', namely to be copied to the intermediate population.

Roulette wheel can be used in minimization problems, too, after transforming them to maximization ones. This can be achieved by the subtraction of the fitness value of each chromosome from a rather large number (certainly larger than the maximum fitness value) and use of the remainder to calculate the size of the slots of the roulette wheel. Moreover, most minimization problems such as pumping cost minimization, can be stated as maximization ones, simply by considering the inverse of the cost function.

If the initial population contains few chromosomes with very large fitness values (compared to the rest), the roulette wheel selection may lead to an intermediate population that contains copies of these privileged individuals only. This phenomenon, which is called 'premature convergence', is not desirable since it shrinks the search around these strong solutions (probably representing local optima). One way to avoid premature convergence is to scale down the range of fitness values in the first generations. Another way is to use the ranking of the solutions in order to construct the slots of the roulette wheel, instead of the actual fitness values.

Tournament method depends on the ranking of the chromosomes only. It starts with determination of the respective selection constant KK, which usually takes values between 3 and 5. Then KK chromosomes are randomly selected from the population of the current generation, and their fitness values are compared to each other. A copy of the chromosome with the best (largest or smallest) fitness value is added to the intermediate population. This process is repeated PS times. In this way, the intermediate population is formed.

It should be mentioned that the value of the selection constant KK depends on the PS and the desired selective pressure, namely the degree to which best chromosomes are favored. Increase in KK value reduces the possibility of selection of weak solutions, namely it increases selective pressure.

A comparative evaluation of selection processes can be found in Goldberg and Deb [10].

2.1.1 The elitist approach

The aforementioned selection processes do not guarantee that a copy of the best chromosome of a generation will be included in the next one, too. For this reason, many codes adopt the so-called elitist approach, namely they include an additional procedure of passing at least one copy of the best chromosome to the next generation.

2.2 Crossover and mutation

Selection is followed by application of other genetic operators to randomly selected members of the intermediate population, in order to produce better chromosomes. Nonselected members are passed unaltered to the next generation.

An outline of the main operators, namely crossover and mutation, follows.

2.2.1 Crossover

Crossover is a recombination process. It applies, with a given probability, to pairs of chromosomes, which are called parents. In some crossover versions, parents can be successive chromosomes only, whereas in others, they are randomly selected from the intermediate population. An integer number XX, between 0 and (CL-1), is also randomly selected. Then each parent binary string is cut into two pieces, immediately after gene XX. The first piece of each parent is combined with the second piece of the other parent, as shown in Fig. 2. In this way, two new chromosomes are formed that are called offspring and substitute their parents in the next generation. If the chromosome represents a set of concatenated variables, as is usually the case, crossover may be allowed only at positions separating those variables.

Crossover aims at combining the best features of the parents to one offspring, at least. There is no guarantee, though, that the aim is accomplished. In one version of the process, offspring replace parents, only if they have a better fitness value. In a more complicated approach, each offspring replaces a member of the intermediate population that is genotypically similar to it. This approach is called crowding.

Noteworthy extensions of the basic crossover process are two- and multiple-point crossover. As their name implies, parent chromosomes are cut in more than two pieces, which they exchange in order to produce the offspring. A two-point crossover allows passing useful information, included both at the head and at the tail of a parent chromosome, to one offspring. There is no guarantee, though that these more complicated crossover versions perform better than the original simpler one.

Additional randomness is introduced by uniform crossover: the parents are again cut in *N* pieces, but offspring formation is achieved through selection between respective parental pieces with a predetermined probability.

Another extension is three-parent crossover. As the name implies, genetic material of three parent chromosomes is recombined to form each offspring. Operators with more than two offspring per parent couple have been also proposed (e.g. Deb *et al.* [11]). In this case, the best two make it to the next generation.

The crossover operator applies in an analogous way to real-coded genetic algorithms, too. It could also be performed by equating each gene of the offspring to

Parent A	**1000101110/10011100**
Parent B	**0101101100/10101011**
Offspring A	**1000101110101011**
Offspring B	**0101101100011100**

Figure 2: Crossover after the 10th character (gene).

the arithmetic or the geometric mean of the respective parent genes. An interesting variant is blend crossover (BLX-α), where offspring are produced in the following way: A real positive constant α is selected ($\alpha = 0.5$ is a common choice). Let gene i of parent A is equal to A_i, while that of parent B equal to B_i. The quantity $c = |A_i - B_i|$ is calculated and a number d ranging in the interval $[\min(A_i - B_i) - \alpha c, \max(A_i - B_i) + \alpha c]$ is selected. Finally, gene i of the offspring is set equal to d. This process is repeated for all genes of the parent chromosomes.

A more extensive review of the crossover operator for real-coded genetic algorithms can be found in Herrera *et al.* [12].

2.2.2 Mutation

Mutation introduces new 'genetic' material. It applies to genes, which form the chromosomes. In binary genetic algorithms, the gene that is selected for mutation is changed from 0 to 1 and vice versa. This process aims at: (a) extending the search to more areas of the solution space (mainly in the first generations) and (b) helping local refinement of good solutions (mainly in the last generations).

The mutation probability MP is equal for all genes of all chromosomes and it is much smaller than the respective crossover probability, because the latter refers to chromosomes, not to genes. In many applications, a value close to 1/CL has been used. The idea is to have, statistically, one mutation per chromosome. In any case, too small MP values may fail to hinder premature convergence, whereas too large ones render the search random. Some researchers have indicated that variable MP may lead to better results.

In real-coded genetic algorithms, mutation is performed by adding or subtracting a small, randomly generated, amount from the selected gene or by multiplying it by a number close to 1. In 'nonuniform' mutation, the magnitude of the added (or subtracted) amount is allowed to be larger in the first generations and then it is gradually reduced. Replacement by a random number (in the range of variation of the selected gene) has also been proposed.

Obviously, crossover and mutation can lead to better or to worse solutions. It is their combination with selection that results in general improvement of population fitness.

2.3 Other operators

Because of the versatility of genetic algorithms, many other operators have been proposed and used. Some of them are problem-specific, whereas others are of general use. A few of them are outlined in the following paragraphs, as an indication of the possibilities offered by the method.

2.3.1 Niche, speciation, and migration

The idea behind these procedures is to divide the total population in independent or semiindependent subpopulations (species), which evolve separately in niches, namely distinct subdomains of the solution space. This approach is very useful in locating a number of good extrema, namely in multimodal problems. One way

to create and sustain subpopulations is 'sharing'. Sharing is a process of reducing a chromosome's fitness, if many individuals are similar to it or, in others words, if many individuals are found in the same neighborhood of the solution space. It is performed in the following way: First, a sharing function Sim(I,J) is defined, based on similarity, and the respective values are calculated for each chromosome I. Sim(I,J) ranges between 0 and 1 with Sim(I,I) = 1 and Sim(I,J) = 0 for similarity less than a threshold value. Then, for each I, the share factor, namely the sum of the respective Sim(I,J) values, is calculated and its fitness value VB(I) is divided by this sum.

Also the crowding process, which has been mentioned in Section 2.2.1, favors niching.

To allow information exchange, a migration rate can be introduced, allowing crossover between chromosomes belonging to different subpopulations. Migration is allowed to occur with a predefined schedule, e.g. every 10 generations, or if evolution of subpopulations is stalled.

2.3.2 Antimetathesis

This operator has been first described by Katsifarakis and Karpouzos [13]. It applies to pairs of successive positions (genes) of a chromosome. Any position (except for the last one) can be selected, with equal probability p_a. If the value of the selected gene equals to 1, it is set to 0, whereas that of the following gene is set to 1. The opposite happens if the value of the selected gene is 0. More explicitly, the following happens with regard to gene pairs:

- 11 becomes 01
- 00 becomes 10
- 10 becomes 01
- 01 becomes 10

In the first two cases, the operator is equivalent to mutation at the selected position. In the last two though, it is equivalent to mutation of both genes. It has been anticipated that antimetathesis is useful both in refinement of good solutions and in exploring different areas of search space.

Effectiveness of antimetathesis in refining a solution can be seen through the following example:

A genetic algorithm is used to find the optimum value of function $F(x)$, x being an integer from 0 to 1000. Let us assume that this optimum occurs for $x = 82$, and that a good approach, i.e. $x = 81$, has been obtained. In binary form, we have:

$$82 = [0001010010]_2$$
$$81 = [0001010001]_2$$

Comparing the two chromosomes, we can see that mutation alone cannot improve the solution. Antimetathesis, though, can find the optimum, if applied to the ninth position of the chromosome. On the contrary, mutation can lead to the

optimum, starting from $x = 83$. It can be argued then that mutation and antimetathesis are complementary to each other in such cases.

Antimetathesis is also complementary to mutation in leading search to different solutions, for the following reason: The jump, or change caused by mutation, is always equal to a power of 2. The change introduced by antimetathesis, applied at position i, is equal to $2^i - 2^{i-1}$. Thus, the solution space can be searched more thoroughly, if the two operators are used interchangeably (e.g. in the even and odd generations, respectively).

The operator has been called antimetathesis, based on its function (when different from mutation). This name is in line with tradition in genetic algorithm terminology, which calls for terms of Greek origin.

2.4 Termination of the process

The whole process, i.e. evaluation–selection–crossover–mutation–other operators, is repeated for a predetermined number of generations, or until a certain termination criterion is fulfilled. Such criteria are as following:

- No better solution is found over a predetermined number of generations.
- The variation of the average fitness value of the population is below a threshold over a predetermined number of generations.
- A predetermined desired value of the objective function is reached by the best individual.
- The median fitness of the population is better than a predetermined value. With this approach a number of satisfactory solutions will be found.

It is anticipated that, at least in the last generation, a chromosome will prevail that represents a suboptimal (if not the global optimal) solution of the problem. It is also possible to end up with a number of different chromosomes with almost equal high fitness values. Identification of a number of different good solutions is an asset of the method, at least when it is used by experts on the particular optimization problem, as explained in following text.

2.5 Handling constraints

In many applications, optimization is subject to constraints. This means that chromosomes, which result from genetic operators, may represent infeasible solutions. In certain cases, it is difficult to find even one feasible solution, to start with. Constraints have usually the form of lower and/or upper limits. For instance, upper limits to hydraulic head level drawdown may be imposed in groundwater resources management problems.

Constraints can be handled in a number of ways. The most obvious is rejection of chromosomes that violate them. This is equivalent to application of death penalty, even for small crimes. But this Draconian approach is not efficient, in particular when the percentage of infeasible chromosomes is large.

Another approach is chromosome repair, namely application of specific algorithms that correct infeasible chromosomes, in order to observe the constraints. Applicability of this method depends on the additional computational load induced by the repair algorithms. Modification of the genetic operators, in order to produce feasible solutions only, has similar restrictions.

The most usual approach to constraint handling is to include penalty functions in the evaluation process that affect the fitness value of chromosomes that violate constraints, increasing it in minimization problems and decreasing it in maximization ones.

Penalties can be (a) constant, (b) proportional to the number of violated constraints, and (c) increasing (linearly, quadratically, logarithmically, etc.) with the degree of violation of each constraint. The latter ones are usually the most efficient (e.g. [14]).

In any case, the values of the penalty function should be large enough in order to guarantee observance of the constraints. Excessive values, though, may level out differences in the evaluation function and may render optimization more difficult. Moreover, if there are disjoint feasible regions in the search space, it would be difficult for the genetic algorithm to move from one of them to another.

The magnitude of the penalty function depends also on the elasticity of the constraints. If small violations can be tolerated, the penalty can be rather small. On the contrary, if the constraint is strict, the respective penalty should at least guarantee that the global optimum of the penalty-including evaluation function corresponds to a feasible solution.

Variants of the process include: (a) predetermined penalty increase with the generation number and (b) penalty increase if the best solution is infeasible and penalty reduction if the best solution is feasible for a number of generations.

Finally, the approach of Coello and Cortés [15] who introduced an artificial immune system code to handle constraints is worth mentioning.

2.6 Steady-state genetic algorithms

Steady-state genetic algorithms do not follow the generation pattern, which has been described so far. Instead, new chromosomes resulting from reproduction operators (i.e. crossover and mutation) replace immediately an equal number of existing ones. Chromosomes that are replaced are usually randomly selected. Replacement of weak individuals is an option, too, but it might result in quick reduction of population diversity and in premature convergence.

2.7 Use of surrogate models: the balance between accuracy and computational efficiency

In most water resources optimization models, which use genetic algorithms (or other evolutionary techniques) as the optimization tool, the flow (and mass transport)

simulation model constitutes the main part of the chromosome evaluation procedure. It should be solved, therefore, for every chromosome of each generation. The result is that, with rather conservative values of genetic algorithm parameters, such as PS = 40 and number of generations NG = 100, the flow simulation model should be solved 4000 times. When the total computational volume becomes a restricting factor, two options may be present: a) to reduce the number of generations and/or the PS, thus reducing the efficiency of the optimization process and (b) to use simplified groundwater flow (or mass transport) simulation models, namely to describe the physical problem less accurately. In fact, one should find the optimal compromise, namely the combination that offers the greatest overall accuracy, before tackling the actual optimization problem [16]. When simplified simulation models are used, sources of reduced accuracy in the overall process are more evident. Moreover, it is possible to check the results obtained by running a more detailed simulation model once only.

2.8 A simple example

Application of genetic algorithms is illustrated through the following simple example:

Eight wells pump a total flow rate TQ = 250 l/s from an 'infinite', confined, and homogeneous aquifer, with transmissivity $T = 0.001$ m^2/s. Find the combination of the eight well flow rates that can minimize the respective pumping cost. Well coordinates appear in Table 1.

2.8.1 Chromosome construction and length

Each chromosome represents a solution to the problem. Therefore, in this case, it represents a set of eight well flow rates. Let us suppose that we allow each of them to vary from 0 to 250. In binary form, $250 = [11111010]_2$. We need then 8 digits for each flow rate (which actually can vary from 0 to 255) and the CL = 64.

Table 1: Well coordinates (data) and flow rates (typical best results).

Well	x_i	y_i	Q_i(l/s)	s_i(m)
1	0	0	41.69	40.64
2	620	580	23.31	40.61
3	700	600	22.16	40.59
4	1100	1550	34.96	40.64
5	1200	800	26.26	40.78
6	420	1400	35.29	40.54
7	1300	0	37.92	40.59
8	1300	1050	28.40	40.62

2.8.2 The evaluation function

The evaluation function should depend on the pumping cost that is defined as:

$$C_1 = A \cdot \sum_{J=1}^{N} Q_J \cdot h_J \tag{3}$$

where N is the number of wells ($N = 8$ in our case), Q_J is the flow rate of well J, h_J is the distance between water level at well J and a predefined level (e.g. highest ground elevation), and A is a constant, depending on energy cost. Using A as constant implies that pump efficiencies are also considered as constants and equal to each other.

Since flow is due to the system of wells only, the initial hydraulic head level is horizontal. Then, eqn (3) can be written as:

$$C_1 = A \cdot \sum_{J=1}^{N} Q_J \cdot (s_J + \delta) \tag{4}$$

where s_J is the drawdown of the hydraulic head at well J, which is due to the operation of the system of the wells, and δ is the distance between the initial horizontal level of the hydraulic head and the predefined reference level. Since δ is same everywhere, the chromosome evaluation function EV could be equal to the variable part of cost C, namely:

$$\mathrm{EV} = C = \sum_{J=1}^{N} Q_J \cdot s_J \tag{5}$$

Since steady flow is considered, s_J for a system of N wells is given as:

$$s_J = -\frac{1}{2\pi \mathrm{T}} \cdot \sum_{\mathrm{I}=1}^{\mathrm{N}} \mathrm{Q_I} \cdot \ln \frac{\mathrm{r_{IJ}}}{\mathrm{R}} \tag{6}$$

where T is aquifer's transmissivity, R the radius of influence of the system of wells, and r_{IJ} the distance between wells I and J (calculated from the respective coordinates). The value of r_{JJ} in particular is equal to the radius of well J, denoted by r_0 for all Js. In our application example, we have set $R = 2500$ m and $r_0 = 0.25$ m. Given the well flow rate values, the calculation of the chromosome fitness value by means of eqns (6) and (5) is straightforward.

2.8.3 Handling the constraint

The eight well flow rates, which result from chromosome decoding, should sum up to TQ, namely they should fulfill the following relationship:

$$\sum_{i=1}^{8} Q_i = \mathrm{TQ} = 250 \tag{7}$$

Since each Q_i can vary between 0 and 250, the constraint is practically never fulfilled. For this reason, its observance cannot be enforced using a penalty

function and the best approach is chromosome repair. This can be easily achieved in the following way: first, the sum SQ of the eight Q_i is calculated and then each of them is multiplied by TQ/SQ. These corrected well flow rate values are used in eqns (6) and (5) to calculate the s_j values and the chromosome fitness value, respectively.

The aforementioned chromosome correction procedure preserves the relative size of the initial well flow rate values.

2.8.4 Results

Typical best results (namely Q_I values) appear in Table 1, together with the respective s_I values. They have been obtained using the following set of genetic algorithm parameters: PS = 50, number of generations NG = 200, crossover probability CRP = 0.60, mutation (and antimetathesis) probability MP = 0.016 (approximately equal to 1/CL), and selection constant KK = 3.

The respective fitness value is 10156.1. The exact global minimum, which can be calculated using the procedure outlined in Section 3.1, equals 10150.94. Therefore, the genetic algorithm has almost exactly hit the global optimum (with an error less than 0.06%).

2.8.5 Constraint on hydraulic head level drawdown

For the best solution, which is shown in Table 1, the hydraulic head level drawdown s is approximately 40.6 m at all wells. Now suppose that s should not exceed 30 m for one part of the field area, namely for $x > 750$. According to the well coordinates, appearing in Table 1, this constraint applies only to wells 4, 5, 7, and 8.

To handle this set of constraints, a penalty function has been included in the evaluation function, having the following general form:

$$\mathrm{SPEN} = \sum_{I=1}^{8} \mathrm{PEN}(I) \tag{8}$$

where

$$\mathrm{PEN}(I) = A + B(s_I - s_{\max})^{\Gamma} \quad \text{if } s_I > s_{\max} \quad \text{and} \quad I = 4, 5, 7, \text{ or } 8 \tag{9a}$$

$$\mathrm{PEN}(I) = 0 \text{ otherwise} \tag{9b}$$

The values of coefficients A, B, and Γ depend on (a) the magnitude of the other part of the evaluation function, which is of the order of 10,000 in our case and (b) the 'elasticity' of the constraint. If small violations of the constraint can be tolerated, the penalty value could be smaller.

Results for different combinations of A, B, and Γ appear in Table 2. Column R_1 includes the s_I values for the optimal solution of the unconstrained case. Sets of typical best results using penalty functions proportional to the number of violated constraints only, namely for $B = 0$, appear next. For $A = 100$ (column R_2), all constraints are violated and s_I values are similar to those of the unconstrained case. This proves that the value $A = 100$ is too small, even

Table 2: Hydraulic level drawdowns s_I for different penalty functions.

	R_1	R_2	R_{3A}	R_{3B}	R_{4A}	R_{4B}	R_5	R_7	R_8
A	–	100	1000	1000	10	10	10	1000	10
B	–	0	0	0	100	100	100	100	100
Γ	–	–	–	–	1	1	2	2	2
s_1	40.61	40.63	54.25	48.69	53.67	53.16	55.58	55.02	55.19
s_2	40.61	40.67	50.25	44.96	50.29	49.40	47.70	48.98	48.98
s_3	40.60	40.53	49.38	45.69	48.25	49.18	48.04	51.09	49.27
s_4	40.60	40.60	30.00	30.00	29.99	29.92	30.27	29.97	29.98
s_5	40.60	40.64	29.70	29.99	29.66	30.00	29.99	29.99	29.97
s_6	40.61	40.62	51.35	44.95	51.88	53.04	51.30	50.04	51.21
s_7	40.60	40.65	29.88	47.14	30.66	29.98	30.34	29.92	30.00
s_8	40.61	40.64	29.95	29.99	30.00	29.99	30.00	29.88	29.47
C	10,151	10,156	11,401	10,827	11,353	11,389	11,340	11,399	11,403
SPEN	–	400	0	1000	76	0	39	0	0
EV	10,151	10,556	11,401	11,827	11,429	11,389	11,379	11,399	11,403

to influence the solution. If $A = 1000$, then the minimum EV value corresponds to solutions that observe the constraints. Many times, however, the genetic algorithm code ends up to solutions that violate one constraint. This is due to the rather inefficient form (stepwise change) of the penalty function. Typical results observing and violating the constraints appear in columns R_{3A} and R_{3B}, respectively.

Typical best results for penalty functions increasing with the magnitude of constraint violation are presented next. Setting $A = 10$, $B = 100$, and $\Gamma = 1$ leads to small or zero violation of the constraints, as shown in columns R_{4A} and R_{4B}, respectively. Typical best results for the coefficient combination $A = 10$, $B = 100$, and $\Gamma = 2$ appear next (column R_5). It always leads to small constraint violations. If such violations can be tolerated, these two forms of the penalty function are suitable, since they lead to somewhat smaller pumping cost.

The constraints are always observed, if we set $A = 1000$, $B = 100$, and $\Gamma = 2$. Typical best results appear in column R_7. Observance of the constraints is also guaranteed for $A = 10$, $B = 100$, and $\Gamma = 2$, namely with the coefficients of column R_5, if PEN(I) is given as:

$$\text{PEN}(I) = A + B(s_I - s_{\max} + 1)^{\Gamma} \quad \text{if } s_I > s_{\max} \text{ and } I = 4, 5, 7, \text{ or } 8 \qquad (10a)$$

$$\text{PEN}(I) = 0 \text{ otherwise} \qquad (10b)$$

With this small change, the term raised to the second power is always larger than one. Typical best results appear in column R_8. Such penalty functions should be used if the constraint is ‘rigid’.

2.9 Advantages and disadvantages of genetic algorithms

Genetic algorithms are very efficient in solving difficult optimization problems. Their main comparative advantages are as following:

1. The search for the global optimum starts from many points of the search space, whereas traditional methods use one initial point only.
2. They use data from the objective function only. For this reason, their application is not restricted by continuity or differentiability requirements.
3. They do not involve complex mathematics.
4. There is no need to linearize the problem.
5. They are very flexible, regarding the construction of the evaluation function.
6. They are highly adapted to parallel processing.
7. Many times, they end up with more than one good solutions (plural) of the problem. This is an asset of the method because it offers the possibility of selecting the most suitable solution, based on:
   a. Additional criteria, which have not been included in the evaluation process of the genetic algorithm, because their mathematical expression is difficult, or because their inclusion leads to disproportionate increase of the computational load, or for some other reason.
   b. Experience or scientific intuition.
   c. Solution stability to changes of the values of input parameters; in particular, when these values are not accurately known. This stability could be checked by means of sensitivity analysis.

Genetic algorithms have some some limitations also. The most important are as following:

1. In most cases, there is no guarantee that the global optimum has been reached.
2. The fitness value has to be calculated for every chromosome of each generation. For this reason, the total computational volume increases rapidly with the complexity of the evaluation function, the number of generations, and the PS.
3. The efficiency of the method depends on the form of the evaluation function. With step functions, for instance, the performance of genetic algorithms is mediocre.

3 Applications

Applications of genetic algorithms in the fields of water resources management, hydrology, and hydraulics are presented in the following sections (except for irrigation and water supply networks, which are discussed in Chapter 3 of this volume). Due to the large number of papers on these topics, the references that are included are indicative only. Still, they show quite clearly the wide range of applicability, the usefulness, and the versatility of the method.

3.1 Groundwater resources management

Genetic algorithms have been used extensively in optimal management of groundwater resources (e.g. Ouazar and Cheng [17]). The main problems are of two types: (a) minimization of pumping cost of a given flow rate from a system of wells and (b) maximization of pumping rate under certain constraints. In most cases, the chromosome evaluation process includes solution of the respective groundwater flow problem.

The difficulty of the first problem depends on the characteristics of the flow field, the type of flow (steady or transient), and the constraints (if any). Moreover, additional cost items, such as amortization of well construction, might be part of the optimization process. There are few analytical solutions for simple flow fields. For steady flow in infinite aquifers, as well as in semi-infinite ones to which the method of images applies, the following proposition has been proved by Katsifarakis [18]: The cost to pump a given total flow rate Q_T from any number and layout of wells in a confined aquifer is minimized, when hydraulic head levels at all wells are equal to each other, as long as flow is due to that system of wells only. This result has been extended to aquifers with two zones of different transmissivities, to which the method of images applies [19]. Moreover, it has been proved that if there is an additional regional flow, pumping cost is minimized when final differences between hydraulic head values at the locations of the wells, resulting from superposition of the regional flow and the operation of the system of the wells, are equal to the half of those resulting from the regional flow only. It shows that the optimal distribution of well flow rates can be calculated analytically, by solving a linear system of N equations and N unknowns, N being the number of the wells.

Even with such simple flow fields, the optimal solution might be not known, if finding optimal locations of the wells is part of the optimization procedure or if additional cost items are considered. Genetic algorithms have been used extensively in difficult groundwater management problems, including complex flow fields, many cost items, or different sets of constraints. Some indicative applications are outlined in the following paragraphs. The case of coastal aquifers is discussed separately in Section 3.1.1, whereas that of groundwater remediation and pollution control is discussed in Section 3.1.2.

Hsiao and Chang [20] combined genetic algorithms with constrained differential dynamic programming (CDDP) to determine the optimal solution of a groundwater resources management problem, taking into account fixed well costs and stepwise time-varying pumping cost. A simple genetic algorithm code was used to determine the number and location of pumping wells (from a predefined set of possible well locations). CDDP was then used in order to define optimal pumping rates. A finite element code was used for groundwater flow simulation. The authors stressed the importance of incorporating fixed costs in the optimization procedure.

Sidiropoulos and Tolikas [21] aimed at pumping cost minimization of a given total flow rate from a number of wells. They considered well locations as members of a discrete set of possible positions instead of treating them as continuous

variables ranging over the extent of the aquifer. Thus, they reduced significantly the CL that includes the well flow rates q_i and one more variable only, representing the well configuration. Moreover, they scaled up the objective function, in order to overcome its flatness. Finally, they introduced a transformation of the q_i values, in order to observe the respective constraint. The same authors [22] extended their work by combining the genetic algorithm code with a cellular automaton, which represents the well configuration.

Koutsourelakis *et al.* [23] used genetic algorithms to optimize groundwater resources management in semi-infinite aquifers, with two rectilinear constant head boundaries. Their objective was to minimize the total cost due to: (a) pumping of a given flow rate from a system of wells and (b) amortization of the construction of the pipe network, which connects these wells with a central water tank. To simulate the flow, they introduced approximate forms of the method of images, in order to reduce the required computational volume. They concluded, based on application examples, that their approach achieves computational efficiency, without seriously compromising the quality of the results. The possibility of using surrogate models, based on approximate forms of the method of images, has been further investigated by Tselepidou [24].

3.1.1 Coastal aquifers

Sustainable management of coastal aquifers poses a very challenging optimization problem that could be stated in the following general way: Maximize total pumping rate of fresh water from a system of wells, without allowing seawater to reach and pollute them, for a predetermined number of years.

The evaluation process, repeated for each chromosome of every generation, includes solution of the respective two-phase flow problem, which is complicated, even when the sharp interface assumption is adopted. To alleviate the total computational volume, many researchers have introduced surrogate models (e.g. Sreekanth and Datta [25] and [26]) or have used low PS and number of generations. Some indicative applications are discussed in the following paragraphs.

Cheng *et al.* [27] studied semi-infinite homogeneous aquifers under steady-state flow conditions, based on the assumption of sharp fresh water–saltwater interface. First, they present two analytical optimal solutions: (a) for the case of two wells, located at equal distances from the coast and having equal pumping rates and (b) for the case of one well and a recharge canal parallel to the coast. Then they use a messy genetic algorithm to seek the optimal solution for a large number of existing wells. The evaluation procedure includes calculation of the abscissa x_{toe} of salt wedge toe at the location of each well and a penalty is applied if x_{toe} exceeds the respective well coordinate x_i (the coastline coincides with the y-axis).

Park and Aral [28] studied semi-infinite homogeneous aquifers under steady-state flow conditions, based on the assumption of sharp fresh water–saltwater interface. Their code, however, allows for optimization of well locations, too. Moreover, the evaluation function includes two conflicting targets: (a) maximization of total pumping rate and (b) minimization of the distances between the wells and the coastline. The relative importance of the two objectives is defined

by the user, through the relative value of two constants. A penalty is imposed when x_{toe} for any well i exceeds the respective x_c, namely the coordinate of the respective stagnation point. It is also worth mentioning that, instead of pumping rates and well coordinates, the authors used the respective perturbations as decision variables. Finally, the authors compared their results with those of Cheng *et al.* [27] (for fixed well locations) and concluded that their code performs slightly better.

Mantoglou *et al.* [29] studied a coastal aquifer of complex shape under steady-state flow conditions. They used a simulation model based on the assumption of sharp interface, which they solved numerically, using a finite differences scheme. They used both sequential quadratic programming (SQP) and genetic algorithms as optimization tool. The objective function of the genetic algorithm code included the sum of the well flow rates and a constant large penalty for constraint violation. They concluded that genetic algorithms are superior to SQP, but require more computing time.

Qalman *et al.* [30] studied a rectangular coastal aquifer with four pumping wells, using a 3-D density-dependent advective–dispersive solute transport model, solved by a finite element technique. Since this model should be solved for every chromosome of each generation, the respective computational volume is very large. For this reason, they had to use a small PS in the genetic algorithm code, namely PS = 5. Four optimization tasks have been examined:

1. Maximization of total pumping rate, with constraint on the salinity of pumped water.
2. Maximization of profit from selling the pumped water, which depends on the price of fresh water, the pumping cost, and the water treatment cost, which is a function of water salinity.
3. Maximization of total pumping rate and minimization of the amount of salt produced. In this case, the relative importance of the two objectives is defined by the user, through the relative value of two constants.
4. The previous case with an additional constraint on the hydraulic head level drawdown, which is taken into account in the evaluation function by means of an additional penalty term.

Bhattacharjya and Datta [31] used a real-coded genetic algorithm to maximize total water abstraction form a coastal aquifer, while maintaining salt concentration of pumped water under a specified permissible limit. To simulate flow and mass transport, they used an artificial neural network (ANN) that was trained by means of a 3-D advective–dispersive transient flow and mass transport model. This approach allowed them to use large PS (PS = 800) and number of generations. They applied their model to a rectangular aquifer with eight fixed wells. Rather surprisingly, wells closer to the coast appeared to have larger flow rates in the optimal solution.

Katsifarakis and Petala [14] combined genetic algorithms with a boundary element code. They expressed the optimization goal as maximization of the following objective function:

$$F = \sum_{i=1}^{N} Q_i - \mathrm{PEN} \tag{11}$$

where N is the number of the wells, Q_i is the flow rate of well i, and PEN is a penalty, imposed if the pumping scheme induces seawater intrusion to the wells. This is checked by means of the water flow simulation tool. To reduce the computational volume, they adopted the following idea: Under steady-state flow conditions, seawater, eventually intruding the wells, enters the aquifer through coastline sections with a net water inflow. If no such sections exist, there is no seawater intrusion to the wells. It is adequate then to ignore density differences and apply a typical two-dimensional model for the average flow.

Actually, the aforementioned model should only produce velocities vertical to the coastline. This is equivalent to calculation of $q = \partial\varphi/\partial n$ on the respective flow field boundary (where φ is the hydraulic head and n the vertical direction). The boundary element method (BEM) is a very suitable choice for this task, since it requires discretization of the flow field boundaries only and produces the respective φ and q values directly. Thus, unnecessary computations are minimized and the resulting computational procedure is very efficient. Moreover, wells are described very accurately as concentrated 'loads', i.e. without distributing well flow rates to grid elements.

Using this surrogate flow model, the authors were able to address flows in zoned aquifers with complex shape, without reducing the PS and the number of generations of the genetic algorithm code.

Tsai [32] studied an aquifer in Baton Rouge area, Louisiana, aiming at managing saltwater intrusion by means of a hydraulic barrier and a saltwater extraction well system. The total management period was divided into a number of time steps, during which the function of the well system remained constant. The specific remediation targets were (a) to keep salt concentration below a predefined limit in existing fresh water production wells during all time steps and (b) to reduce salt concentration below that limit in the intruded aquifer area at the end of the management period. The author used a genetic algorithm to minimize total injected and pumped water quantities, which were required for the remediation process. Therefore, the chromosome fitness function consisted of the sum of these water quantities plus two penalty terms, to take into account failures to achieve the aforementioned remediation targets. The chromosome consisted of (a) one value of injection and one of production well flow rate for each time step (common for the respective subset of wells) and (b) one gene for every injection or pumping well, to define whether it is active during each time step. It is worth mentioning that the evaluation procedure included solution of multiple simulation models, to take into account uncertainty in their parameters.

3.1.2 Groundwater remediation and pollution control

The main problem can be stated as: Minimize the cost of remediation of a contaminated aquifer or minimize the cost of hydraulic control of a contamination plume. Some indicative applications are discussed in the following paragraphs.

McKinney and Lin [33] used genetic algorithms in three common groundwater resources management problems, in order to prove their efficiency, compared to traditional optimization techniques. The objective of the first problem was

to maximize total pumping rate by means of 10 fixed wells from a rectangular aquifer under steady-state flow conditions. Constraints included maximum pumping rate of each well and minimum hydraulic head values. The objective of the second problem was to minimize the cost of obtaining a given total flow rate from the same well system under the same set of constraints. The evaluation function included well construction amortization, together with the pumping cost. The objective of third problem was to minimize the cost of a pump-and-treat system, using air stripping treatment technology and five pumping or injection wells, the latter to return treated water to the same aquifer. Then the objective function included three terms, namely pumping, injection, and treatment cost, the latter based on an air-stripping performance equation. The solution should fulfill the following constraints: (a) upper bounds on contaminant concentration at selected aquifer points at the end of the remediation period, (b) an upper bound on contaminant concentration of the effluent of the treatment plant, (c) lower and upper bounds on hydraulic heads at various aquifer locations, (d) upper and lower bounds on extraction and injection well flow rates, and (e) total well injection rate equal to total well pumping rate. The authors concluded that genetic algorithms performed as well as traditional methods in simple problems and inferred that they may be better in more complicated ones.

Chan Hilton and Culver [34] addressed two aquifer remediation problems, in order to study the efficiency of two penalty forms for constraint violation. The objective of the first problem was to minimize an aquifer remediation cost. The latter included pumping cost from 18 wells and operating cost of a granular activated carbon treatment facility, together with capital cost of its adsorbers. The constraints included upper bound on concentration at 13 observation wells at the end of the remediation period. The second problem was the one described by McKinney and Lin [33] as outlined in previous paragraphs. The authors examined two constraint handling methods: (a) to add a penalty proportional to the square of constraint violation to the actual cost (multiplied by a proper weight factor) and (b) to multiply the cost by a factor proportional to constraint violation and a proper weight factor. They concluded that penalties of the second type were more efficient, in particular in finding strictly feasible solutions, while using a range of weights.

Kalwij and Peralta [35] combined genetic algorithms and tabu search (see Chapter 6, this volume) to find optimal solution to a groundwater remediation problem. They used MODFLOW and MT3DMS, namely commonly used codes, to simulate groundwater flow and contaminant transport, respectively. The objective was to minimize total pumping (from extraction and injection wells), whereas the constraints included: (a) upper and lower bounds on hydraulic head and (b) an upper bound on contaminant concentration that should not be exceeded at predefined times at each field zone. The authors applied the proposed model with success to an aquifer contaminated by volatile hydrocarbons and explosives.

Katsifarakis *et al*. [36] used a particle tracking code to simulate advective mass transport in contaminated aquifers, in order to alleviate the computational load of the chromosome evaluation process. They applied their code in two cases: In the

first case, they sought to maximize total clean water pumping rate Q_s from three production wells, situated in the same restricted area with two wells that inject polluted water. The penalty used had the following form:

$$\mathrm{PEN} = \sum (100 + 10(\mathrm{TT1} - \mathrm{Tb}_i - 1) \tag{12}$$

where TT1 is the total number of time steps (covering the pollutant deactivation period) and Tb_i is the time step at which a moving point i arrived at a production well. Of course, summation extends to moving points with $\mathrm{Tb}_i < \mathrm{TT1}$ only.

In the second case, the total flow rate of the production wells was fixed and the minimum treatment cost of pumped water was sought. Best solutions resulting from a number of runs were classified in three different patterns with almost equal fitness values, which were comparatively evaluated, using additional criteria. Some remarks on taking into account dispersive mass transport conclude this chapter.

3.2 The inverse problem of groundwater hydraulics

To simulate flow and mass transport in aquifers, one should know the pertinent flow parameters such as transmissivity and dispersivity. The difficulty in obtaining these parameters depends on the features of the flow field. Some indicative applications of genetic algorithms are outlined in the following paragraphs.

Wagner [37] studied the design of a sampling network aiming at reducing uncertainty of estimates of aquifer parameters, which are used in flow and mass transport simulation models. He aimed at minimizing a properly selected uncertainty measure under a budget constraint. He applied a rather simple genetic algorithm code to a synthetic contaminated rectangular aquifer with two zones of different hydraulic conductivities. Estimation of hydraulic conductivities, together with longitudinal and transverse transmissivities, was based on hydraulic head and contaminant concentration 'measurements'. The author compared the results with the global optimal values, obtained by means of the branch-and-bound method. He concluded that 'the genetic algorithm can identify near-optimal solutions for a small fraction of the computational effort needed to identify the globally optimal branch-and-bound solution'.

El Harrouni *et al.* [38] studied the efficiency of genetic algorithms in obtaining transmissivity values in a heterogeneous aquifer, based on point measurements of hydraulic heads and using the dual reciprocity BEM to simulate flow. They applied the combined simulation–optimization tool to a synthetic square aquifer with transmissivity varying in a known way. First, they calculated hydraulic head values h_{mi} at N points, based on the known transmissivity distribution and used them as 'measured' values in the objective function, which had the following form:

$$F = \sum_{i=1}^{N} w_i (h_i - h_{mi})^2 \tag{13}$$

where h_i is the calculated value by means of the simulation–optimization tool and w_i is the predefined weight. The authors concluded that the genetic algorithms perform very well in such problems.

Katsifarakis *et al.* [39] combined genetic algorithms with boundary elements to study three problems related to groundwater resources management, namely: (a) determination of transmissivities in zoned aquifers, based on a restricted number of field measurements, (b) minimization of pumping cost from any number of wells under various constraints, and (c) hydrodynamic control of a contaminant plume, by means of pumping and injection wells. They worked on a synthetic aquifer of irregular shape and four zones of different transmissivities. To solve the first problem, each chromosome represented a set of four transmissivity values, whereas the evaluation function had the following form:

$$F = \sum_{i=1}^{N} \left| h_i - h_{mi} \right| \tag{14}$$

where h_i is the calculated value and h_{mi} virtual field measurement, derived by solving the direct problem with the correct transmissivity values. They concluded that(a) the results are very satisfactory, as long as the measurement points are close to the pumping wells, and (b) transmissivity estimates are not affected seriously by reasonably small-field measurements errors.

In the work of Karpouzos *et al.* [40], the inverse problem of groundwater flow was treated with an automatic method that can produce several alternative solutions at once. During their joint optimization, these solutions can exchange information in order to maintain some diversity and thus avoid a systematic premature convergence toward a single local minimum. For that purpose, a multipopulation genetic algorithm (MPGA), specially adapted to the resolution of the inverse problem in hydrogeology, was developed. A Gray logarithmic coding for transmissivity values was proposed that favors the order of magnitude adjustment and promotes bits economy. Genetic operators consisted of elitist tournament selection, crossover, mutation, gradual update (expressed by the probability that a new solution will be retained in the next generation, which was inspired by the acceptance probability of a Metropolis algorithm), and migration. The aforementioned set of the first four successive operators was applied to each subpopulation and migration was then applied to allow exchange of individuals between subpopulations. The MPGA was tested on two synthetic cases of steady-state flow with transmissivity values T extending over four orders of magnitude. The first test was nonparametric and optimized as many parameters as those used to define the reference case, whereas the second one referred to a parametric case of an aquifer built of 600 blocks. The 600 values were identified from a restricted set of 28 parameters at point (elementary block) locations called pilot points. The MPGA handled individuals made of perturbation values of the initial T values at the 28 pilot points. The numerical definition of the 600 T values was obtained by interpolation of the pilot-point perturbation values and the interpolated perturbation was added, point by point, to an initial a priori field (initial solution) invariant during the optimization. A simple least squares form with a limited number of observed heads was used as the objective function. Based on the good quality of the results, the authors concluded that MPGA was able to simultaneously handle and identify a significant

number of alternative solutions to the problems studied. Furthermore, the combined use of the MPGA and the pilot point parameterization method seems to be a promising approach to solving large-scale inverse problems since the convergence of the objective function and the transmissivity identification were very successful. Finally, it was noted that the mapping of subpopulations on parallel processors can substantially decrease the computational cost.

Tsai *et al.* [41] studied heterogeneous aquifers and used genetic algorithms to estimate the optimal distribution of transmissivity values. The chromosomes, namely the solutions of the problem, consisted of the coordinates of a number of 'basis points' and the respective transmissivity values. Calculation of the latter at any field point was achieved using the natural neighbor method, thoroughly described in their paper. The objective function was based on the sum of the differences between measured and calculated hydraulic head values. Best solutions obtained by the genetic algorithm code were further refined by means of a gradient-based algorithm.

Recently, Lu *et al.* [42] used genetic algorithms to estimate six parameters, namely aquifer thickness, horizontal hydraulic conductivity, vertical hydraulic conductivity, storativity, specific yield, and delay coefficient of a karst aquifer based on a pumping test. The respective fitness function was the sum of square differences between calculated and observed drawdowns. They concluded that genetic algorithms require comparatively large computational time, but they can yield dependable results.

3.3 Surface water hydrology

Finding the most suitable rainfall–runoff relationship is a difficult task in many cases. Genetic algorithms have their share in optimizing parameters of hydrological models and on surface water management. Three indicative applications follow.

Wang [43] has used a genetic algorithm code to calibrate a conceptual rainfall–runoff model with nine parameters. First, he studied a hypothetical example, in which the true optimum set of parameter values was known by assumption, in order to examine whether the genetic algorithm was capable of finding that optimum. Then he studied its performance further, using real data from four catchments. He concluded that the genetic algorithm was always able to find an objective function value close to the global minimum.

Cheng *et al.* [44] have used real-coded genetic algorithms to calibrate the parameters of a conceptual rainfall–runoff model for flood forecasting. The procedure aimed at optimal forecast of: (a) peak discharge, (b) peak time, and (c) total runoff volume. The weight coefficients, included in the objective function, were equal to 0.45, 0.45, and 0.1 respectively. The authors have applied their code successfully to the drainage basin of a reservoir in Southern China.

Bekele and Nicklow [45] have used a 'nondominated sorted' genetic algorithm (Srinivas & Deb [46]), suitable for multiobjective optimization, to calibrate a physically based, semidistributed watershed model with a large number of parameters.

The authors selected 18 parameters, in order to calibrate the model for stream flow and sediment concentration, and used as objective function the root mean square error between observed and simulated model responses.

3.4 Reservoir management

Reservoir management poses a number of very interesting optimization problems. Some indicative applications of genetic algorithms are outlined in the following paragraphs.

Wardlaw and Sharif [47] have studied the four-reservoir problem (Fig. 3), in order to evaluate alternative formulations of genetic algorithms. The problem was initially proposed by Larson [48] and its global optimum is known. The objective is to maximize benefits from hydropower generation and irrigation, by regulating the reservoir releases R_{ij} over 12 two-hour operating periods. Therefore, each chromosome represents 48 (4×12) R_{ij} values. Constraints include lower and upper bounds on R_{ij} and on reservoir storage in all time steps and target final storages for all reservoirs. Moreover, the continuity constraint, regarding inflows, outflow, and storage at each reservoir, should be fulfilled in all time steps.

The authors compared binary, Gray, and real value encodings and a number of mutation and crossover schemes. They concluded that 'a real value representation, incorporating tournament selection, elitism, uniform crossover, and modified uniform mutation' operates most efficiently and produces the best results for the four-reservoir problem. Moreover, they studied a 10-reservoir problem and concluded that genetic algorithms can handle large problems, too.

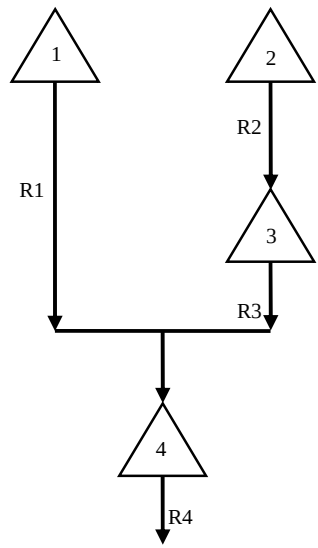


Figure 3: The four-reservoir problem (after Wardlaw and Sharif [47]).

Oliveira and Loucks [49] used a real-coded genetic algorithm to optimize performance of a multireservoir system, taking into account water and energy demands. Each chromosome represented an operating policy and it was evaluated by means of a function of water and energy deficit. Chromosomes were ranked according to their performance and a roulette wheel selection scheme was used to produce the intermediate population. Moreover, a copy of the best current solution was included in the next generation (elitist approach). Finally, to ensure feasibility of new solutions, crossover and mutation applied to groups of parameters, each defining an operating rule. A secondary conclusion of their research is that 'it is preferable to select a small population size and allow it to evolve for more generations than to select a larger population and let it evolve for fewer generations', if the total number of system simulations is restricted.

Cai *et al.* [50] combined genetic algorithms and linear programming to address two large-scale problems: (a) a reservoir operation model with nonlinear hydropower generation equations and nonlinear reservoir topologic equations and (b) a long-term dynamic river basin planning model with a large number of nonlinear relationships, applied to the Syr Darya basin in Central Asia. The objective function included benefits to irrigation and hydropower generation, together with ecological water use over a multiyear period.

Huang *et al.* [51] combined genetic algorithms with stochastic dynamic programming to study a multireservoir system. The latter was used to optimize the operating policy of each single reservoir, while the genetic algorithm defined the release allocation for each water user of the system. The objective was to minimize the annual demand deficit. The code was used to optimize a two-parallel-reservoir water supply system in northern Taiwan.

Ahmed and Sharma [52] used genetic algorithms to define best operating policies for a multipurpose reservoir, located on the river Pagdalia, India. As input data they used synthetic monthly stream flow series, produced by means of an ANN. The objective function was the square deficit from demand, whereas constraints included lower and upper storage volumes. They concluded that operating policies derived by genetic algorithms are comparable to those derived by stochastic dynamic programming

Chen [53] has applied a real-coded genetic algorithm to optimize the rule curves of a multipurpose reservoir for flood control, power production, and water supply. Each chromosome represented 12 decision variables, defining the 'lower limit' and the 'critical' curves, which in turn defined the water quantities that should be released for power generation and water supply. The objective function, aiming at minimizing water supply deficit was based on the shortage index, introduced by the US Army Corps of Engineers. The author tested a number of versions of selection and crossover operators and concluded the BLX-0.5 crossover produced the best results. He also concluded that genetic algorithms have advantages in the context of reservoir systems optimization.

Chen and Chang [54] introduced a multipopulation genetic algorithm to overcome problems of premature convergence. They tested it in the four-reservoir benchmark problem, shown in Fig. 3, and then they applied it to a three-reservoir

system (two existing and one planned) in Taiwan. The objective was to minimize deviations from water supply and hydropower production targets. The authors concluded that the proposed multipopulation code outperforms the conventional genetic algorithm.

Chang *et al.* [55] used genetic algorithms to optimize the operating rule curves (upper, lower, and critical) of a reservoir, which was used for flood control, hydroelectric power generation, water supply, and recreation. As objective function, the authors used the aforementioned shortage index. Constraints included: (a) satisfaction of the continuity equation for all the system components, such as the reservoir, irrigation areas, and merger points, (b) upper and lower water level at the reservoir, and (c) bounds on water requirement and water release for each waterwork. The authors tried binary and real-coded genetic algorithms; they concluded that both led to better results than the existing operating rule curves and that the real-coded version was more efficient.

Chang *et al.* [56] extended the previous work by taking into account the requirement for ecological flow, in the form of an additional constraint. Moreover, they modified the objective function in order to avert continuous water-deficit periods. They presented also an interesting distinction between strict and soft constraints. Also Chang *et al.* [57] included genetic algorithms in their 'intelligent fuzzy flood control model' for real-time reservoir operation.

Shirangi *et al.* [58] combined a multiobjective genetic algorithm-based optimization model and a water-quality simulation model to determine a trade-off curve between objectives related to the allocated reservoir water quantity and quality. To reduce the run-time of the genetic algorithm-based optimization model, they decomposed to long-term and annual optimization models. The operating policies obtained using this long-term model provided the time series of the optimal reservoir water storages at the beginning and the end of each water year. In the next step, these optimal reservoir storage values were considered as constraints for water storage in the annual reservoir operation optimization models.

Liu *et al.* [59] stressed the assets of having multiple near optimal solutions in reservoir operation optimization and compared the efficiency of three techniques in deriving them. They concluded that genetic algorithms perform better than the near-shortest paths method but that they are less efficient than the Markov chain Monte Carlo method.

3.5 Other applications

Genetic algorithms have been used in more problems of water resources management, or implicitly related to water resources, such as optimization of ANNs that are used as surrogate models, e.g. to predict rainfall–runoff relationship. Some indicative applications are outlined in the following paragraphs.

Jain and Srinivasulu [60] used real-coded genetic algorithms to train an ANN rainfall–runoff model. They used daily rainfall and stream flow data from the Kentucky River watershed and found that the ANN ability to reproduce low-magnitude flows was improved.

Pump scheduling is the interface between groundwater resources and water supply network management and has attracted a lot of attention. Barán *et al.* [61], for instance, used six multiobjective evolutionary techniques, based on genetic algorithms, in order to optimize operation of a system of pumps that feed the reservoir of a water supply system. Four objectives were simultaneously sought, namely minimization of (a) electric energy cost, (b) maintenance cost, (c) maximum power peak, and (d) level variation in the reservoir.

Low-enthalpy geothermal aquifers constitute both energy and water resources. Tselepidou and Katsifarakis [62] used genetic algorithms to optimize the exploitation system of a hypothetical geothermal aquifer. With respect to water flow, the aquifer consisted of two zones of different transmissivities, whereas from the thermal point of view it could bear any number of zones with different temperatures. The objective function comprised the annual pumping cost of the required flow and the amortization cost of the pipe network, which carries the hot water from the wells to a central water tank, situated at the border of the geothermal field. Results showed that application of the proposed optimization procedure allows better planning of low-enthalpy geothermal heating systems, which may be crucial in cases of marginal financial viability.

In another application related to water resources management, Koutroulis and Kolokotsa [63] used genetic algorithms to optimize the design of desalination systems powered by photovoltaics and wind generators. The objective function was equal to sum of the capital and maintenance costs during the desalination system lifetime period, including those of the water storage tank.

4 Concluding remarks

Introduction of genetic algorithms in mid-seventies offered a new approach to optimization problems. More nature-inspired methods followed, allowing researchers a choice based on the features of the problem in hand or on their personal scientific background and taste.

Genetic algorithms have been used very efficiently in most kinds of water resources management, hydrology, and hydraulic network problems. In earlier applications, the authors had to prove that genetic algorithms were at least as efficient as traditional optimization techniques. In a few years, however, the fact that they perform better in many complicated problems, e.g. when the respective objective functions exhibit many local optima or discontinuous derivatives, has become common wisdom.

In water resources management, genetic algorithms are often coupled with flow simulation models, which form the main part of the respective objective function. Availability of more powerful and parallel computers allows more sophisticated flow simulation approaches. Still the use of surrogate models is necessary in many applications.

The basic features of the genetic algorithms, outlined in this chapter, and the huge variety of their applications prove beyond doubt that they are a mature technique. Still, they do not restrict the creativity of the users who can invent new

operators or adapt existing ones to the features of the investigated problem. This is probably the most attractive feature of genetic algorithms.

References

[1] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press: Ann Arbor, 1975.

[2] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company: Reading, MA, USA, 1989.

[3] Rawlins, G.J.E., *Foundations of Genetic Algorithms*, Morgan Kaufmann: Publishers, San Mateo, CA, 1991.

[4] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag: Berlin Heidelberg, 1996.

[5] Coley, D.A., *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific: Singapore, 1999.

[6] Reeves, C.R. & Raw, J.E., *Genetic Algorthms-Principles and Perspectives*, Kluwer Academic Publishers: Dordrecht, the Netherlands, 2003.

[7] Sivanandam, S.N. & Deepa, S.N., *Introduction to Genetic Algorithms*, Springer: Berlin Heidelberg, 2008.

[8] Yu, X. & Gen, M., *Introduction to Evolutionary Algorithms*, Springer: London, Dordrecht, Heidelberg, New York, 2010.

[9] Koumousis, V.K. & Katsaras, C.P., A saw-tooth genetic algorithm combining the Effects of variable population size and reinitialization to enhance performance. *IEEE Transactions on Evolutionary Computation*, **10(1)**, pp. 19–28, 2006.

[10] Goldberg, D.E. & Deb, K., A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms, ed.* G.J.E. Rawlins, Morgan Kaufmann: San Mateo, CA, pp. 69–93, 1995.

[11] Deb, K., Anand, A. & Joshi, D., A computationally efficient evolutionary algorithm for real-parameter evolution. *Evolutionary Computation*, **10(4)**, pp. 371–395, 2002.

[12] Herrera, F., Lozano, M. & Sánchez, A.M., Hybrid crossover operators for real-coded genetic algorithms: an experimental study. *Soft Computing*, **9**, pp. 280–298, 2005.

[13] Katsifarakis K.L. & Karpouzos, D.K., Minimization of pumping cost in zoned aquifers by means of genetic algorithms. *Proc. Int. Conf. on Protection and Restoration of the Environment IV*, eds K.L. Katsifarakis, G.P. Korfiatis, Y.A. Mylopoulos & A.C. Demetracopoulos, Sani Greece, pp. 61–68, 1998.

[14] Katsifarakis, K.L. & Petala, Z., Combining genetic algorithms and boundary elements to optimize coastal aquifers' management. *Journal of Hydrology*, **327(1–2)**, pp. 200–207, 2006.

[15] Coello, C.A.C. & Cortés, N.C., Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, **36(5)**, pp. 607–634, 2004.

[16] Katsifarakis, K.L., Flow simulation and optimal management of groundwater resources. The balance between accuracy and computational efficiency (Chapter 11). *Groundwater: Modelling, Management and Contamination*, eds L.F. Konig & J.L. Weiss, Nova Science Publishers: Commack, NY, ISBN 978-1-60456-832-5, pp. 291–308, 2009.
[17] Ouazar, D. & Cheng, A.H.-D., Application of genetic algorithms in water resources (Chapter 7). *Groundwater Pollution Control*, ed. K.L. Katsifarakis, WIT Press: Southampton, pp. 293–316, 2000.
[18] Katsifarakis, K.L., Groundwater pumping cost minimization-An analytical approach. *Water Resources Management*, **22(8)**, pp. 1089–1099, 2008.
[19] Katsifarakis, K.L. & Tselepidou K., Pumping cost minimization in aquifers with regional flow and two zones of different transmissivities. *Journal of Hydrology*, **377(1–2)**, pp. 106–111, 2009.
[20] Hsiao, C.-T. & Chang, L.-C., Dynamic Optimal Groundwater Management with Inclusion of Fixed Costs. *Journal of Water Resources Planning and Management*, **128(1)**, pp. 57–65, 2002.
[21] Sidiropoulos, E. & Tolikas, P., Well locations and constraint handling in groundwater pumping cost minimization via genetic algorithms, *Water, Air & Soil Pollution: Focus*, **4(4–5)**, pp. 227–239, 2004.
[22] Sidiropoulos, E. & Tolikas, P., Genetic algorithms and cellular automata in aquifer management. *Applied Mathematical Modelling*, **32(4)**, pp. 617–640, 2008.
[23] Koutsourelakis, I.-G., Gardikou, V. & Katsifarakis, K.L., Use of approximate forms of the method of images in groundwater management problems, *E-Proceedings, International Conference on Protection and Restoration of the Environment VIII*, Chania, Kriti, 2006.
[24] Tselepidou, K., *Optimal Management of Groundwater and Geothermal Resources*, Ph.D. Thesis (in Greek), Department of Civil Engineering, Aristotle University of Thessaloniki, Greece, p. 567, 2008.
[25] Sreekanth J. &. Datta B., Multi-objective management of saltwater intrusion in coastal aquifers using genetic programming and modular neural network based surrogate models. *Journal of Hydrology*, **393(3–4)**, pp. 245–256, 2010.
[26] Sreekanth J. &. Datta B., Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate models and multiple-realization optimization. *Water Resources Research*, **47**, DOI: 10.1029/2010WR009683, 2011.
[27] Cheng, A.H.-D., Halhal, D., Naji, A. & Ouazar, D., Pumping optimization in saltwater-intruded coastal aquifers. *Water Resources Research*, **36(8)**, pp. 2155–2166, 2000.
[28] Park, C.H. & Aral, M.M., Multi-objective optimization of pumping rates and well placement in coastal aquifers. *Journal of Hydrology*, **290(1–2)**, pp. 80–93, 2004.
[29] Mantoglou, A., Papantoniou, M. & Giannoulopoulos, P., Management of coastal aquifers based on nonlinear optimization and evolutionary algorithms. *Journal of Hydrology*, **297**, pp. 209–228, 2004.

[30] Qalman, K., Larabi, A. Ouazar, D., Naji, A. & Cheng, A.H.-D., Optimal and sustainable extraction of groundwater in coastal aquifers. *Stochastic Environmental Research and Risk Assessment*, **19(2)**, pp. 99–110, 2005.
[31] Bhattacharjya, R.K. & Datta, B., Optimal management of coastal aquifers using linked simulation optimization approach. *Water Resources Management*, **19**, pp. 295–320, 2005.
[32] Tsai, F.T.-C., Bayesian model averaging assessment on groundwater management under model structure uncertainty. *Stochastic Environmental Research and Risk Assessment*, **24**, pp. 845–861, 2010.
[33] McKinney, D.C. & Lin, M.-D., Genetic algorithm solution of groundwater management models. *Water Resources Research* **30(6)**, pp. 1897–1906, 1994.
[34] Chan Hilton, A.B. & Culver, T.B., Constraint handling for genetic algorithms in optimal remediation design. *Journal of Water Resources Planning and Management ASCE*, **126(3)**, pp. 128–137, 2000.
[35] Kalwij, I.M. & Peralta, R.C., Non-adaptive and adaptive hybrid approaches for enhancing water quality management. *Journal of Hydrology*, **358(3–4)**, pp. 182–192, 2008.
[36] Katsifarakis, K.L., Mouti, M. & Ntrogkouli, K., Optimization of groundwater resources management in polluted aquifers. *Global Nest*, **11(3)**, pp. 283–290, 2009.
[37] Wagner, B.J., Sampling design methods for groundwater modeling under uncertainty. *Water Resources Research*, **31(10)**, pp. 2581–2591, 1995.
[38] El Harrouni, K., Ouazar, D., Walters, G.A. & Cheng, A.H.-D., Groundwater optimization and parameter estimation by genetic algorithm and dual reciprocity boundary element method. *Engineering Analysis with Boundary Elements*, **18**, pp. 287–296, 1996.
[39] Katsifarakis, K.L., Karpouzos, D.K. & Theodossiou, N., Combined use of BEM and genetic algorithms in groundwater flow and mass transport problems. *Engineering Analysis with Boundary Elements*, **23(7)**, pp. 555–565, 1999.
[40] Karpouzos, D.K., Delay, F., Katsifarakis, K.L. & de Marsily G.A., Multipopulation genetic algorithm to solve the inverse problem in hydrogeology. *Water Resources Research*, **37(9)**, pp. 2291–2302, 2001.
[41] Tsai, F.T.-C., Sun, N.-Z. & Yeh, W.W.-G., Geophysical parameterization and parameter structure using natural neighbors in groundwater inverse problems. *Journal of Hydrology*, **308**, pp. 269–283, 2005.
[42] Lu, C., Shu, L., Chen, X. & Cheng C., Parameter estimation for a karst aquifer with unknown thickness using the genetic algorithm method. *Environmental Earth Sciences*, **63**, pp. 797–807, 2011.
[43] Wang, Q.J., Using genetic algorithms to optimise model parameters. *Environmental Modelling & Software*, **12(1)**, pp. 27–34, 1997.
[44] Cheng, C.T., Oub, C.P. & Chau, K.W., Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall–runoff model calibration. *Journal of Hydrology*, **268**, pp. 72–86, 2002.

[45] Bekele, E.G. & Nicklow, J.W., Multi-objective automatic calibration of SWAT using NSGA-II. *Journal of Hydrology*, **341**, pp. 165–176, 2007.
[46] Srinivas, N. & Deb, K., Multiple objective optimizations using non-dominated sorting in genetic algorithms. *Evolutionary Computation* **2(2)**, pp. 221–248, 1994.
[47] Wardlaw, R. & Sharif, M., Evaluation of genetic algorithms for optimal reservoir system operation. *Journal of Water Resources Planning and Management ASCE*, **125(1)**, pp. 25–33, 1999.
[48] Larson, R.E., *State Increment Dynamic Programming*, Elsevier Science: New York, 1968.
[49] Oliveira, R. & Loucks, D.P., Operating Rules for Multireservoir Systems. *Water Resources Research*, **33(4)**, pp. 839–852, 1997.
[50] Cai, X., McKinney, D.C. & Lasdon, L.S., Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. *Advances in Water Resources*, **24(6)**, pp. 667–676, 2001.
[51] Huang, W.C., Yuan, L.C. & Lee, C.M., Linking genetic algorithms with stochastic dynamic programming to the long-term operation of a multireservoir system. *Water Resources Research*, **38(12)**, pp. 1304–1312, 2002.
[52] Ahmed, J.A. & Sharma, A.K., Genetic algorithm for optimal operating policy of a multipurpose reservoir. *Water Resources Management*, **19(2)**, pp. 145–161, 2005.
[53] Chen, L., Real-coded genetic algorithm optimization of long-term reservoir operation. *Journal of American Water Resources Association*, **39**, pp. 1157–1165, 2003.
[54] Chen L. & Chang, F.J., Applying a real-coded multi-population genetic algorithm to multi-reservoir operation. *Hydrological Processes*, **21(5)**, pp. 688–698, 2007.
[55] Chang, F.-J., Chen, L. & Chang, L.-C., Optimizing reservoir operating rule curves by genetic algorithms. *Hydrological Processes*, **19**, pp. 2277–2289, 2005.
[56] Chang, L.-C., Chang, F.-J., Wang K.-W. & Dai, S.-Y., Constrained genetic algorithms for optimizing multi-use reservoir operation. *Journal of Hydrology*, **390(1–2)**, pp. 66–74, 2010.
[57] Chang, L.-C., Chang, F.-J. & Hsu H.-C., Real-time reserevoir operation for flood control using artificial intelligent techniques. *International Journal of Nonlinear Sciences and Numerical Simulation*. **11(11)**, pp. 887–902, 2010.
[58] Shirangi, E., Kerachian, R. & Bajestan, M.S., A simplified model for reservoir operation considering the water quality issues: application of the Young conflict resolution theory. *Environmental Monitoring and Assessment*, **146**, pp. 77–89, 2008.
[59] Liu, P., Cai, X. & Guo, S., Deriving multiple near-optimal solutions to deterministic reservoir operation problems. *Water Resources Research*, **47**, W08506, doi:10.1029/2011WR010998, 2011.
[60] Jain, A. & Srinivasulu, S., Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic

algorithms and artificial neural network techniques. *Water Resources Research*, **40**, W04302, doi:10.1029/2003WR002355, 2004.

[61] Barán, B., von Lűcken, C. & Sotelo, A., Multi-objective pump scheduling optimisation using evolutionary strategies. *Advances in Engineering Software*, **36**, pp. 39–47, 2005.

[62] Tselepidou, K. & Katsifarakis, K.L., Optimization of the exploitation system of a low enthalpy geothermal aquifer with zones of different transmissivities and temperatures. *Renewable Energy*, **35**, pp. 1408–1413, 2010.

[63] Koutroulis, E. & Kolokotsa, D., Design optimization of desalination systems power-supplied by PV and W/G energy sources. *Desalination*, **258**, pp. 171–181, 2010.

This page intentionally left blank

CHAPTER 3

Applications of genetic algorithms to water distribution networks: optimization of irrigation networks design

D.K. Karpouzos
Faculty of Agriculture, Aristotle University of Thessaloniki, Greece.

Abstract

This chapter deals with the application of genetic algorithms in the optimal design of water distribution networks, and especially those that are used for agricultural purposes. A genetic optimization approach is elaborated and presented in order to minimize the total cost of a pumping irrigation network, including the energy and pipeline investment cost. Specially adapted features of genetic algorithms are proposed to handle the hydraulic constraints (pressure and velocity requirements) and a discrete-value coding of variables is adopted to represent the available commercial pipe diameters. Pumping head at the upstream end of a pressurized irrigation network and pipe diameters distribution are determined simultaneously through an automatic optimization procedure that aims to obtain the least total cost network design. The proposed methodology is applied to two collective on-demand irrigation networks with different characteristics, and the performance of genetic algorithms is discussed.
Keywords: genetic algorithms, irrigation, least-cost design, on-demand operation, optimization, pumping, water networks.

1 Introduction

Nowadays water distribution networks are complex systems demanding appropriate design and a high level of investment for their efficient operation. Because of the escalating need to improve efficiency and to reduce maintenance and construction costs, many models have been developed to solve the least-cost design problem of water distribution systems. Thus, system component characteristics are explored in order to minimize the total system cost and improve their design and performance. Methods for the least-cost design include a broad spectrum of

models employing among others several evolutionary/metaheuristic techniques [1]. Over the past two decades, the employment of genetic algorithms (GAs) [2] substantially improved the optimization of water distribution systems [3].

GAs are metaheuristic independent global search techniques, typically characterized by the following elements: (a) generation of an initial population of potential solutions, each identified as a chromosome, (b) computation of the objective function value of each solution and subsequent ranking of chromosomes according to this metric, (c) some aspect of chromosome ranking and selection of candidate solutions to participate in a mating pool, where the information from two or more parent solutions are combined through crossover operator to create offspring solutions, and (d) mutation of each individual offspring to maintain diversity and prevent premature convergence to local optima. These elements are repeated in sequential generations until a suitable solution is obtained.

Simpson *et al*. [4] were the first to apply GAs for water distribution systems in least-cost design and compared the solution derived to enumeration and to nonlinear optimization using the network of Gessler [5]. Later, Savic and Walters [6] used GAs to solve and compare optimal results for the one-loading gravity systems of the two-loop network [7], the New York City Tunnels [8] and the Hanoi networks [9]. Dandy *et al*. [10] improved GAs by incorporating variable power scaling of the fitness function for the New York City Tunnels (NYCT) networks optimization. The potential of a fast messy GA [11] was explored by Wu and Simpson [12], who noted its usage for the optimization of large-scale water distribution systems. Vairavamoorthy and Ali [13] developed a GA framework for the least-cost pipe network that excludes regions of the search space in case impractical or infeasible solutions are likely to exist, in order to improve the GA search efficiency. Wu and Simpson [14] introduced a self-adaptive constraint handling mechanism that was later embodied in a multiobjective fast messy GA scheme [15]. The self-adaptation was introduced to handle the transformation from a constrained into a nonconstrained framework of the least-cost design and rehabilitation problems of a water distribution system.

The optimal design of water networks involves significant engineering issues beyond the least-cost design such as multiple operating conditions, reliability, redundancy, resilience, and time scheduling of investments [16]. Therefore, due to various decision variables, multiobjective optimization approaches have been developed and applied in water resources problems in order to evaluate the trade-offs of the least-cost design problem with other related design competing objectives [17–22]. However, Di Pierro *et al*. [21] note that multiobjective optimization techniques "suffer from the loss of efficiency when compared to its single-objective counterpart" since the total number of simulation runs is practically extended in real-world water distribution networks applications. GAs have been also used successfully to determine the optimum design of gravity-sustained pressurized irrigation systems under different flow delivery schemes [23].

Pump scheduling is of great importance for reducing energy costs and therefore operational expenditures of water distribution systems. Under these premises and according to modern approaches in water engineering, GAs followed the classical techniques in pump scheduling [24–28].

The application of pumping systems is of great importance in agricultural water management, since it is quite common in irrigation networks where topography does not allow water to flow at desired pressure and quantity. In pumped irrigation networks, the system cost includes the cost of pipes, pumps, energy (pumping), and operation and maintenance. The optimization of such systems is a critical issue due to high recurring energy cost. In the optimal design of pumping hydraulic networks, there is an economic trade-off between the pumping head and pipe diameters. Selection of pipe diameters (from a set of commercially available discrete-valued diameters) to construct a water supply network of minimum cost is a highly constrained problem and has been shown to be an NP-hard problem [29], [6]. The present chapter investigates the adaptation of GAs for the automatic optimization of pumping irrigation networks operating on-demand.

2 Methodology

A simulation-optimization procedure has been developed with the aim of optimizing the design of branched irrigation networks that are fed by a pumping station. The proposed model uses an adapted GA as optimization method. The model procedure is presented in Fig. 1. The input data refer to network layout data, available commercial diameters, irrigation, and GAs parameters. The genetic process is performed iteratively until a predefined maximum number of generations is reached (NG = 1 to MNG, where NG is the index number of generation and MNG the maximum number of generations). The basic features of the optimization algorithm are described in the following subsections.

2.1 Initial population–coding

First, an initial population of possible solutions (pipe diameters) is produced randomly according to a coding scheme. In heuristic methods, like GAs, coding is an important part of the algorithm formulation as it defines the magnitude of the domain space and the accuracy of discretization required. Typical GAs use a binary coding in order to represent the candidate solutions that are named as chromosomes. Real-coded GAs have also been suggested and applied recently with success in many cases. However, the choice of the appropriate coding scheme as well as the genetic structure of the algorithm is mainly based on the nature of the problem.

Since the design variables are the diameters that follow a predetermined discrete distribution based on pipe manufacturers' databases, an integer discrete coding [30] is adopted in this study.

Each chromosome is represented by an array of np integers, where np is the number of pipes, whereas each integer corresponds to a discrete diameter value among the commercial available ones. The design variables range from 1 to ND, where 1 is assigned to the smallest diameter and ND to the largest one of the predefined diameter list. In that way, an economy of bits is achieved

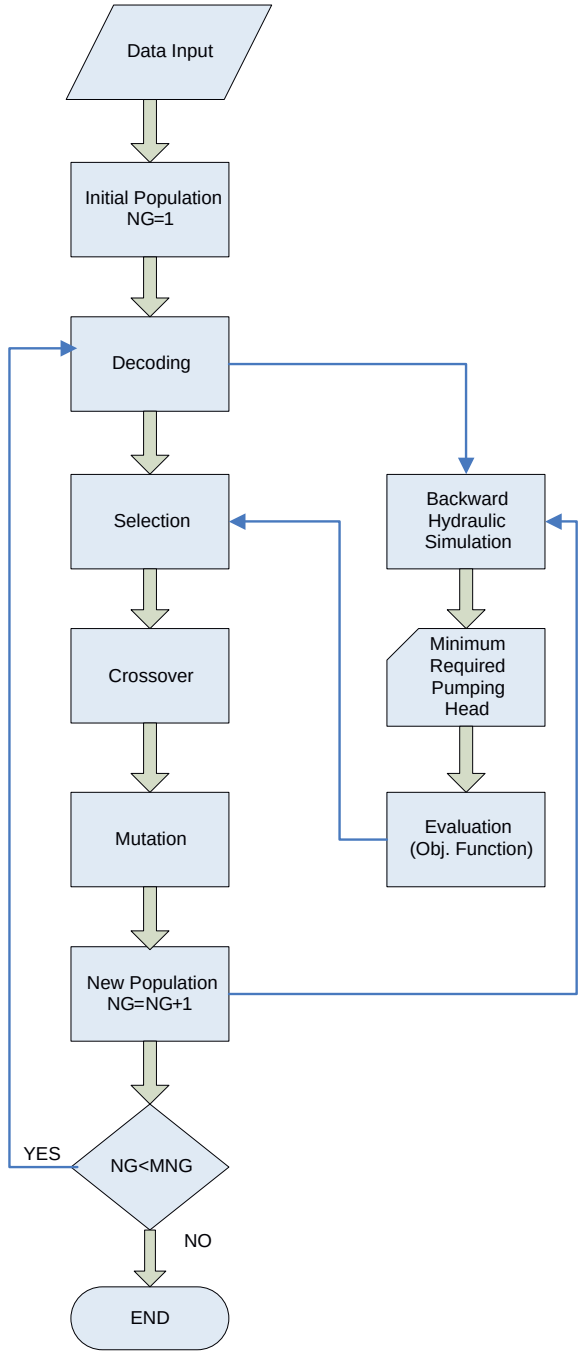


Figure 1: Optimization model's flowchart.

in the definition of the domain space, while there are no limitations on the number of possible diameter sizes that can be assigned to a specific pipe. Moreover, decoding can be effectuated in a simple and fast way, since there is no need of base conversions that are required in binary coding. Genetic operators are applied directly to a discrete defined domain rather than to a continuous search space, thus avoiding useless approximations that are required in real coding as variables should be rounded to the nearest discrete diameter value.

2.2 Backward hydraulic simulation

Once a solution, i.e. a possible set of diameters, is produced by the application of genetic operators (or randomly if it belongs to the initial population), a hydraulic simulation of the pumping irrigation network is performed. A backward procedure is adopted in order to find the minimum pumping head required at the head of the network to fulfill all the nodes pressure requirements.

Considering a fixed distribution of flow rates, as it is the case of on-demand irrigation networks, hydraulic losses can be calculated for each pipe since diameter is known. According to Hazen-Williams, hydraulic losses *DH* are given by the following equation:

$$DH = \omega \left(\frac{Q}{C_{HW}} \right)^{a} D^{-\beta} L \tag{2.1}$$

where ω is a numerical conversion constant, which depends on the units used, a is a coefficient equal to 1/0.54, b is a coefficient equal to 2.63/0.54, D is the pipe internal diameter, L is the length of the pipe, and C_{HW} is the Hazen–Williams roughness coefficient.

The hydraulic simulation is performed backwards, i.e. from terminal nodes to upstream nodes. A matrix G [np+1, Ml+1] describing the network geometry is created that stores the downstream nodes that are directly linked with any node, where Ml is the maximum number of adjacent links of a junction node for the irrigation network under study. So, matrix first column contains the indexes of nodes, whereas next columns contain the indexes of corresponding adjacent downstream nodes. At each node, piezometric heads are calculated using eqns (2) and (3):

$$H_j = \max\left\{ H_j^{\mathrm{req}}, \max\{H_{ij} + DH_{ij}\} \right\} \text{ for } \mathrm{j} = \mathrm{np}+1 \text{ to } 1; \mathrm{i} = 2 \text{ to } \mathrm{M1}+1 \tag{2.2}$$

$$H_j^{\mathrm{req}} = z_j + P_j^{\mathrm{req}} \text{ for } \mathrm{j} = \mathrm{np}+1 \text{ to } 1 \tag{2.3}$$

where H_{ij} is the piezometric head of downstream node *i* connected directly with node *j*, DH_{ij} is the hydraulic losses of pipe from downstream node *i* to *j*, H_j^{req} is the required pressure head, z_j is the elevation, and P_j^{req} is the required pressure of node *j*. Thus, through this backward hydraulic simulation, i.e. from downstream

ends to upstream end, the minimum upstream piezometric elevation at the head of the network is derived.

The flow regime of the irrigation networks is calculated for on-demand operation of the pressurized irrigation networks using the first model of Clement [31]. The elementary probability p of operation of each outlet is defined as:

$$p = \frac{Q}{r \cdot R \cdot d} = \frac{q \cdot S}{r \cdot R \cdot d} \tag{2.4}$$

where q is the specific continuous discharge (l/s/ha), S is the irrigated area (ha), R is the total number of outlets, d is the nominal discharge of each outlet (l/s), and r is the coefficient of utilization of the network.

The number N of simultaneously operating hydrants is estimated through the probabilistic first formula of Clement:

$$N = R \cdot p + U \cdot \sqrt{Rp(1-p)} \tag{2.5}$$

where U is the standard normal variable corresponding to $P(U)$. The latter is called network operation quality, which is the cumulative probability that among R hydrants, there will be a maximum of N outlets being simultaneously in operation. Thus, considering an equal discharge at all outlets, the total discharge downstream a generic node of the network is given by:

$$Q = N \cdot d = R \cdot p \cdot d + U \cdot \sqrt{Rp(1-p)} \cdot d \tag{2.6}$$

2.3 Fitness evaluation

The least-cost design criterion of an irrigation network is used to evaluate the fitness of the algorithm. In the following applications, the total cost consists of the minimization of the annual recurring cost composed by two main components, the pipeline and the pumping energy cost. Thus, the objective function can be defined as:

$$min\ f = C_{\text{pipe}} + C_{\text{enerpump}} \tag{2.6}$$

where the annual pipeline cost C_{pipe} is given by the following expression:

$$\begin{aligned} C_{\text{pipe}} &= \sum_{i=1}^{Np} L_i \cdot CD_i \cdot C_{rf} \\ C_{rf} &= \frac{r(1+r)^t}{(1+r)^t - 1} \end{aligned} \tag{2.7}$$

and the annual energy cost C_{enerpump} is expressed as follows:

$$C_{\text{enerpump}} = (A \cdot \beta \cdot \gamma \cdot Q \cdot H_{\text{man}})/n \tag{2.8}$$

where γ is the specific weight of water (= 9.81 kN/m^3), A is the irrigation period (h), β is the electricity rate (€/kWh), r is the annual interest rate (%), Q is the pumping flow rate (l/s), H_{man} is the pumping head (m) and n is the combined efficiency of pump and prime mover.

If the variables (pipe diameters) are allowed to take any value of the commercial set, i.e. velocity upper and lower bounds are not inherently incorporated in coding and genetic operations by definition, the respective constraint can be treated with a penalty. Then the objective function of this GA variant takes the following form:

$$\min f = (C_{\text{pipe}} + C_{\text{enerpump}})(1 + a \cdot \text{Pen}V) \tag{2.9}$$

$$\text{Pen}V = \frac{V_{\max}^{sim} - V_{\max}^{def}}{V_{\max}^{def}} \tag{2.10}$$

where *PenV* reflects the relative maximum pipe velocity violation, and $V_{\max}^{sim}$ and $V_{\max}^{def}$ are the maximum simulated and defined velocities, respectively. The parameter a depends on the magnitude of the cost value and is empirically determined by the network designer.

2.4 Genetic operators

The genetic operators used in the following applications are selection, crossover, and mutation. Elitist tournament method is adopted for the selection procedure (Chapter 2, this volume). The selected solutions are combined through crossover to form the offspring according to a predefined rate (e.g. 0.60). In order to enhance the ability to create new information, a blending one-point crossover method is used to combine the variable values from the two parents. A single offspring variable value, p_{new}, comes from a combination of the two corresponding parent variable values at a randomly selected crossover point [32], [33]:

$$p_{\text{new}} = \text{round}\{\lambda * p_{mn} + (1-\lambda) * p_{dn}\} \tag{2.11}$$

where λ is a random number on the interval [0, 1], p_{mn} is nth variable in the mother chromosome, and p_{dn} is the nth variable in the father chromosome.

In case $\lambda = 1$, p_{mn} is reproduced in its entirety, whereas if $\lambda = 0$, the p_{new} is equal to p_{dn}. When $\lambda = 0.5$ [34], the value of p_{new} is an average of the variables of the two parents. The same variable of the second offspring is the complement of the first (i.e. replacing λ by $1-\lambda$). This blending approach is demonstrated to work well on several interesting optimization applications [35]. The rest of the variables of the two offspring are completed similarly to the typical one-point crossover method (Chapter 2, this volume).

One more advantage of this crossover-blending method in water distribution network applications is the fact that it is not allowed to offspring variables (pipe diameters) to get values outside the bounds of the available set of the commercial discrete-valued diameters. Furthermore, the variables will always respect the velocity constraints when the acceptable range of each pipe diameter is defined in the coding scheme within the permissible pipe velocities according to the respective flow rates.

Mutation is applied at each variable of the solution vector by changing its integer value (diameter size) randomly. In order to respect the commercial diameters

bounds and velocity constraints for each pipe, the mutation operator is also designed to alter the diameter size for each parameter within the acceptable predefined values range.

Furthermore, a telescopic pattern control operator is introduced in the algorithm, before the evaluation of generated solutions, that ensures that for any consecutive pair of pipes, the diameter size of the upstream pipe is larger or equal to the one of the downstream pipe. This correction is easily implemented through the use of network layout data stored in the input matrix G, described in the Section 2.2.

3 Applications

In this study, the proposed methodology is tested in two collective pressurized irrigation networks fed by a pumping station. The differences between these irrigation networks mainly rely on the number of parameters (pipes), the geometrical layout, and the irrigation parameters.

The minimum required pressure at each demand node is set to 35 m, which is an adequate value for the application of sprinkler on-farm irrigation method. The commercial available diameters and their respective costs used for the design of the network are given in Table 1. The minimum and maximum velocity for all diameters is set to 0.5 and 2.0 m/s, respectively. Flow rates distribution is calculated through the first Clement formula. In the case of links where the number of downstream outlets is equal or less than 10, the peak discharge is obtained by summating the discharge of each of the outlets. Pipe head losses are calculated using the Hazen–Williams formula (eqn. (2.1)) with a roughness coefficient, $C_{HW} = 150$.

Concerning the economic parameters, the annual interest r is taken as 6% and the life cycle of the infrastructure t is estimated to be 40 years. Pumping energy cost parameters are taken as follows: electricity rate $\beta = 0.085$ €/kWh, combined efficiency of pump and prime mover $n = 0.65$ €/kWh, and pumping irrigation period $A = 1000$ h.

Table 1: Coding and cost of commercial pipe diameters.

Coding	ND (mm)	ID (mm)	Cost (€/m)
0	110	9360	9
1	140	119,15	14.5
2	160	136,19	18
3	200	170,26	26
4	225	191,55	32
5	280	238,39	53
6	315	268,20	65
7	355	302,27	80
8	400	340,60	102
9	450	383,18	130

3.1 Case 1

The first case refers to a typical pumping irrigation network, shown in Fig. 2, used for testing the optimization algorithm. This relatively small-scale (in terms of number of variables) symmetrical network consists of 35 pipes (links), 30 demand nodes with double (60) outlets, and one pumping station at the head of the network (Fig. 2) located at 62 masl. The length of each pipe and the elevation of each node are presented in Tables 2 and 3, respectively. The irrigation parameter values used in the network design are specific continuous discharge $q = 0.005$ l/s/ha, nominal discharge of each outlet $d = 6$ l/s, operation time of the network OT = 16 h, and quality of operation P(U) = 99%. The total irrigated area is 150 ha. The peak Clement discharge at the head (pumping station) of the network is calculated as 192 l/s. Genetic parameters are set as follows: maximum number of generations (MNG) = 500, population size (PS) = 75, crossover rate (CR) = 0.65, and mutation probability = $1/np \approx 0.03$.

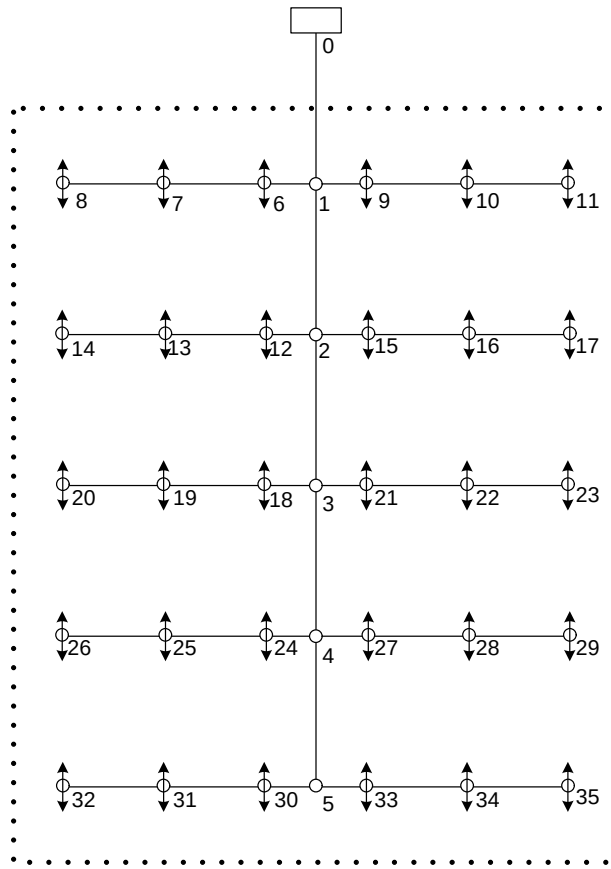


Figure 2: Irrigation network layout (Case 1).

Table 2: Irrigation network nodes elevation (z) data (Case 1).

Node	z (masl)	Node	Z (masl)	Node	z (masl)	Node	z (masl)	Node	z (masl)
1	60.75	8	61.65	15	58.15	22	58.15	29	56.2
2	59.7	9	60.75	16	57.85	23	57.85	30	55.85
3	58.15	10	60.45	17	57.65	24	57.55	31	55.75
4	57.35	11	60.15	18	58.25	25	57.85	32	56.15
5	55.85	12	59.7	19	58.4	26	57.85	33	55.75
6	60.75	13	59.05	20	58.6	27	57.25	34	55.65
7	61	14	60.2	21	58.15	28	56.8	35	55.75

Table 3: Irrigation network pipes length data (Case 1).

Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)
1	325	8	200	15	100	22	200	29	200
2	300	9	100	16	200	23	200	30	100
3	300	10	200	17	200	24	100	31	200
4	300	11	200	18	100	25	200	32	200
5	300	12	100	19	200	26	200	33	100
6	100	13	200	20	200	27	100	34	200
7	200	14	200	21	100	28	200	35	200

3.2 Case 2

In this application, a more extended irrigation network, situated in Epirus, Greece, is optimized. A pumping station at the head of the network with an elevation of 469 masl fulfills the pressure requirements of the irrigation outlets. It has 77 pipes (links) and 63 hydrants with a single outlet. The layout of the irrigation network is presented in Fig. 3, whereas the length of each pipe and the elevation of each node are given in Tables 4 and 5, respectively. The irrigation parameter values used in the network design are specific continuous discharge $q = 0.0057$ l/s/ha, nominal discharge of each outlet $d = 6$ l/s, operation time of the network OT = 18h, and quality of operation P(U) = 99%. The total irrigation area is 163.8 ha. The peak Clement discharge at the upstream end (pumping station) of the network is calculated as 180 l/s. Genetic parameters are set as follows: maximum number of generations (MNG) = 500, population size (PS) = 150, crossover rate (CR) = 0.65, and mutation probability = $1/np \approx 0.015$.

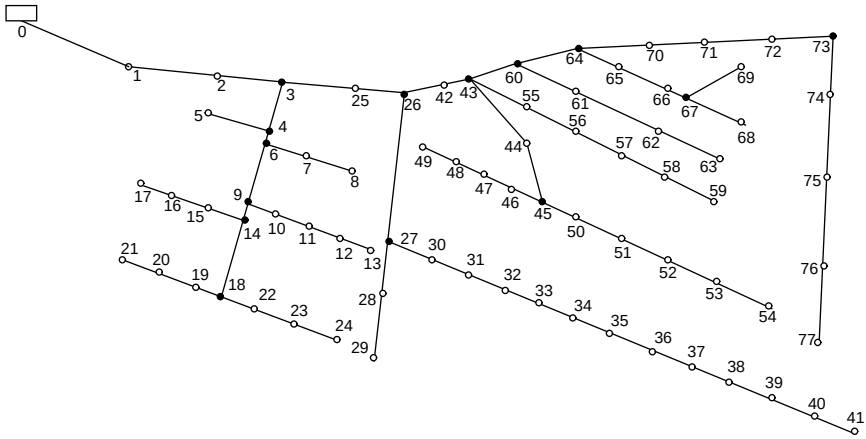


Figure 3: Irrigation network layout (Case 2).

Table 4: Irrigation network nodes elevation (z) data (Case 2).

Node	Elevation (masl)	Node	Elevation (masl)	Node	Elevation (masl)	Node	Elevation (masl)	Node	Elevation (masl)
0	469.0	16	471.9	32	470.1	48	474.5	64	477.2
1	479.5	17	472.2	33	470.1	49	476.2	65	476.5
2	478.0	18	470.0	34	470.0	50	472.0	66	475.1
3	478.0	19	470.0	35	470.1	51	471.9	67	475.1
4	474.3	20	469.5	36	470.2	52	471.5	68	474.1
5	473.8	21	469.6	37	470.3	53	471.4	69	474.8
6	474.0	22	470.0	38	470.5	54	471.2	70	476.8
7	475.0	23	470.1	39	470.4	55	474.1	71	475.9
8	476.0	24	469.8	40	470.3	56	473.5	72	475.8
9	472.8	25	477.8	41	470.2	57	473.5	73	475.9
10	473.5	26	477.7	42	477.5	58	473.3	74	475.2
11	472.8	27	474.0	43	478.2	59	473.2	75	473.8
12	473.0	28	471.9	44	475.0	60	478.0	76	472.1
13	473.0	29	469.2	45	472.1	61	475.8	77	471.0
14	472.1	30	471.0	46	472.6	62	474.2		
15	471.8	31	470.2	47	473.5	63	470.2		

Table 5: Irrigation network pipes length data (Case 2).

Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)
1	710.5	17	140.0	33	130.0	49	130.0	65	35.0
2	165.0	18	180.0	34	110.0	50	30.0	66	230.0
3	130.0	19	25.0	35	85.0	51	130.0	67	25.0

Continued

Table 5: *Continued*.

Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)	Pipe	Length (m)
4	100.0	20	130.0	36	115.0	52	130.0	68	255.0
5	170.0	21	130.0	37	85.0	53	130.0	69	240.0
6	20.0	22	100.0	38	85.0	54	130.0	70	190.0
7	95.0	23	150.0	39	90.0	55	245.0	71	200.0
8	170.0	24	150.0	40	90.0	56	215.0	72	200.0
9	110.0	25	170.0	41	90.0	57	145.0	73	135.0
10	100.0	26	170.0	42	230.0	58	145.0	74	150.0
11	100.0	27	240.0	43	100.0	59	150.0	75	250.0
12	100.0	28	125.0	44	175.0	60	120.0	76	250.0
13	100.0	29	190.0	45	305.0	61	145.0	77	250.0
14	35.0	30	215.0	46	105.0	62	300.0		
15	85.0	31	130.0	47	130.0	63	300.0		
16	125.0	32	130.0	48	130.0	64	210.0		

4 Results and discussion

The best combination of pipe diameters obtained by the proposed GA for the irrigation network of the first application is given in Table 6. This solution corresponds to a total annual cost of 23166.55 € and refers to GA version where velocity constraints are always respected due to their incorporation in the coding and genetic operators' schemes.

For comparison purposes, linear programming (LP) is also used since the optimization problem in our case studies is linearly formulated. The solution found by LP gives an annual cost of 23009.51 €, which is only 0.68% less than the one produced by GA. The LP solution can be considered as a theoretical benchmark. Given that all linear functions are convex, an LP problem has at most one feasible region with 'flat faces' (i.e. no curves) on its outer surface, and the optimal solution will always be found at a 'corner point' on the surface where the constraints intersect. Thus, it is always possible to determine that an LP problem has a globally optimal solution (either a single point or multiple equivalent points along a line). However, the LP solution corresponds to a theoretical lower bound of the cost. By definition, LP allows more than one diameter to be selected for each link, since the lengths of each available diameter of each pipe are the decision variables. So, even if LP can lead to the global minimum cost, it can often produce not practical optimal solutions characterized by inadequately small lengths of a pipe diameter or undesired diameters mixing per pipe. Thus, LP solution often requires a posteriori correction in order to be implemented.

In case of GA's version, where velocity constraint is handled with a penalty in the fitness function of GA, near optimal solutions that slightly violate maximum or minimum velocity may be found. These solutions produce a more economic

Table 6: Best solution – optimal pipe diameters (Case 1).

Pipe	ND	Pipe	ND	Pipe	ND	Pipe	ND	Pipe	ND
1	450	8	140	15	200	22	160	29	140
2	400	9	200	16	160	23	140	30	200
3	355	10	160	17	140	24	200	31	200
4	315	11	140	18	200	25	200	32	140
5	280	12	225	19	160	26	140	33	200
6	200	13	160	20	160	27	225	34	200
7	160	14	140	21	200	28	160	35	140

Total cost = 23166.55 €; pumping head = 42.71 m.

output, even sometimes better than LP. For example, optimal designs with associated annual costs of 23034.16 € and 22992.94 € are found for the first network. The maximum excess of velocity upper bound is 5.4% and 8.7% for the two aforementioned solutions, while 8.6% and 11.4% of pipes are having velocities greater than 2 m/s. The implementation of these network designs depends on the special characteristics and requirements of the case studied, and the final judgment is left to the engineer in charge.

The best solution of the second application is given in Table 7. This solution corresponds to a total cost of 35425.01 €. This solution compared to the one produced by linear programming, which produces a minimum cost equal to 35374.88 €, can

Table 7: Best solution – optimal pipe diameters (Case 2).

Pipe	DN	Pipe	DN	Pipe	DN	Pipe	DN	Pipe	DN
1	400	17	110	33	225	49	110	65	160
2	400	18	200	34	225	50	200	66	140
3	400	19	140	35	200	51	160	67	110
4	280	20	110	36	200	52	140	68	110
5	110	21	110	37	200	53	110	69	110
6	280	22	140	38	160	54	110	70	225
7	110	23	110	39	140	55	200	71	225
8	110	24	110	40	110	56	160	72	225
9	280	25	355	41	110	57	160	73	225
10	160	26	355	42	315	58	110	74	160
11	140	27	280	43	315	59	110	75	160
12	110	28	110	44	280	60	280	76	140
13	110	29	110	45	225	61	140	77	110
14	225	30	280	46	160	62	110		
15	140	31	280	47	160	63	110		
16	110	32	280	48	160	64	280		

Total cost = 35425.01 €; Pumping head = 65.65 m.

Table 8: Best solution – optimal pipe diameters (Case 2).

Pipe	DN	Pipe	DN	Pipe	DN	Pipe	DN	Pipe	DN
1	400	17	110	33	225	49	110	65	160
2	400	18	200	34	225	50	225	66	140
3	400	19	160	35	225	51	160	67	110
4	225	20	110	36	225	52	160	68	110
5	110	21	110	37	200	53	110	69	110
6	225	22	160	38	200	54	110	70	225
7	110	23	140	39	200	55	200	71	225
8	110	24	110	40	160	56	160	72	225
9	225	25	355	41	140	57	160	73	200
10	200	26	355	42	315	58	140	74	200
11	160	27	225	43	315	59	140	75	160
12	110	28	110	44	280	60	280	76	140
13	110	29	110	45	225	61	160	77	110
14	225	30	280	46	225	62	110		
15	140	31	225	47	200	63	110		
16	110	32	225	48	140	64	280		

Total cost: 35244.60 €; pumping head = 64.35 m.
The velocity constraint is handled with a penalty in the fitness function.

be also considered as a global optimum for the case of single diameter per link as it exceeds the LP best solution cost only by 0.14%. The alternative version of GA, with a velocity-related penalty, has reached a best cost of 35244.60 €, which is even better than the one attained by LP. The optimal distribution of pipe diameters for this version of GA is given in Table 8. The respective maximum violation of the velocity upper bound for this solution is 4.16%, whereas 7.8% of pipes are having velocities larger than 2 m/s.

It can be seen that GAs performed very well on both irrigation networks. The first GA variant with the structural incorporation of velocity-conditioned diameters produced in all cases the global minimum with respect to the intrinsic characteristic of a single diameter per pipe. The different characteristics and the extension of the number of parameters among the two networks did not seem to influence the robustness of GA algorithm. As expected, the computational cost is greater in the second application but the time required is not significant.

5 Conclusions

In this study, GAs were designed and applied for the optimization of two pumping irrigation networks. An automatic procedure was proposed in order to minimize the cost design. Namely, the optimal pumping head was automatically calculated through a backward hydraulic simulation based on the optimal pipe diameters combination. The main conclusions can be summarized as follows:

1. In both applications, GAs achieved to attain a minimum cost, very close to the theoretical one obtained by LP. The difference in cost between the two methods is justified due to the fact that GAs optimize a single diameter instead of a diameters mixing per link. However, split-pipe feature is not in many cases the preferred option as the solution may be less realistic.
2. The proposed GA by its coding and genetic operators design respect the hydraulic constraints of the required pumping head, minimum pressure at outlets and nodes, and upper and lower bounds of pipe velocities. There is no need to a posteriori intervention in order to implement the optimal solution in practice.
3. The GAs cannot only handle efficiently nonlinear objective functions of cost compared to LP but also work with discrete commercial diameters as decision variables compared to the continuous NLP programming techniques. Furthermore, the GAs use an automatic way to determine simultaneously the optimal pumping head and pipe diameters without the need of an iterative procedure to find the optimal pumping height that corresponds to the least-cost network design.
4. If a more flexible fulfillment of pipe velocity constraint is considered, the introduction of a properly adjusted penalty in the objective function could reduce the cost with an acceptable and limited violation of the predefined velocity bounds. The feasibility and final adoption of such solution depend on the specific characteristics of the problem under study.
5. The requirement of a telescopic diameter pattern, moving from upwards to downwards of a hydraulic network, can easily be respected in GA with the incorporation of an additional control and correction statement in the algorithm's code, before the fitness evaluation of each solution.

References

[1] Reca, J., Martinez, J., Gil, C. & Banos, R., Application of several metaheuristic techniques to the optimization of real looped water distribution networks. *Water Resources Management*, 22, pp. 1367–1379, 2008.

[2] Holland, J.H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[3] Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A. & Zechman, E., State of the art for genetic algorithms and beyond in water resources planning and management. *Journal of Water Resources Planning and Management*, **136(4)**, pp. 412–429, 2010.

[4] Simpson, A.R., Dandy, G.C. & Murphy, L.J., Genetic algorithms compared to other techniques for pipe optimisation. *Journal of Water Resources Planning and Management Division, ASCE*, **120(4)**, pp. 1589–1603, 1994.

[5] Gessler, J., Pipe network optimization by enumeration. *Proceedings of the Computer Applications for Water Resources*, ASCE, New York, pp. 572–581, 1985.

[6] Savic, D. & Walters, G., Genetic algorithms for least cost design of water distribution networks. *Journal of Water Resources Planning and Management*, **123(2)**, pp. 67–77, 1997.

[7] Alperovits, E. & Shamir, U., Design of optimal water distribution systems. *Water Resources Research*, **13(6)**, pp. 885–900, 1977.
[8] Schaake, J.C. & Lai, D., Linear programming and dynamic programming application to water distribution network design. Report No. 116, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA, 1969.
[9] Fujiwara, O. & Khang, D.B., A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resources Research*, **26(4)**, pp. 539–549, 1990.
[10] Dandy, G.C., Simpson, A.R. & Murphy, L.J., An improved genetic algorithm for pipe network optimisation. *Water Resources Research*, **32(2)**, pp. 449–458, 1996.
[11] Goldberg, D.E., Deb, K., Kargupta, H. & Harik, G.R., Rapid Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms. IllIGAL, Technical Report, 1993.
[12] Wu, Z.Y. & Simpson, A.R., Competent genetic-evolutionary optimization of water distribution systems. *Journal of Computing in Civil Engineering*, **15(2)**, pp. 89–101, 2001.
[13] Vairavamoorthy, K. & Ali, M., Pipe index vector: a method to improve genetic-algorithm-based pipe optimization. *Journal of Hydraulic Engineering*, **131(12)**, pp. 1117–1125, 2005.
[14] Wu, Z.Y. & Simpson, A.R., A self-adaptive boundary search genetic algorithm and its application to water distribution systems. *Journal of Hydraulic Research*, **40(2)**, pp. 191–203, 2002.
[15] Wu, Z.Y. & Walski, T., Self-adaptive penalty approach compared with other constraint-handling techniques for pipeline optimization. *Journal of Water Resources Planning and Management*, **131(3)**, pp. 181–192, 2005.
[16] Walski, T.M., The wrong paradigm – why water distribution optimisation doesn't work. *Journal of Water Resources Planning and Management, ASCE*, **127(7)**, pp. 203–205, 2001.
[17] Halhal, D., Walters, G.A., Savic, D.A. & Ouazar, D., Scheduling of water distribution system rehabilitation using structured messy genetic algorithms. *Evolutionary Computation*, **7(3)**, pp. 311–329, 1999.
[18] Prasad, T.D. & Park, N.-S., Multiobjective genetic algorithms for design of water distribution networks. *Journal of Water Resources Planning and Management*, **130(1)**, pp. 73–82, 2004.
[19] Nicolini, M., Evaluating performance of multi-objective genetic algorithms for water distribution system optimization. *Sixth International Conference on Hydroinformatics*, ed. Phoon, B. Liong, World Scientific Publishing Company, Singapore, vol. 1, pp. 850–857, 2004.
[20] Vamvakeridou-Lyroudia, L.S., Walters, G.A. & Savic, D.A., Fuzzy multiobjective optimization of water distribution networks. *Journal of Water Resources Planning and Management*, **131(6)**, pp. 467–476, 2005.
[21] Di Pierro, F., Khu, S.T., Dragan Savić, D. & Berardi, L., Efficient multiobjective optimal design of water distribution networks on a budget of

simulations using hybrid algorithms. *Environmental Modelling & Software*, **24(2)**, pp. 202–213, 2009.

[22] Norouzi, K. & Rakhshandehroo, G.R., A self organizing map based hybrid multi-objective optimization of water distribution networks. *Iranian Journal of Science and Technology Transaction B: Engineering*, Vol. 35(C1), pp. 105–119, 2011.

[23] Farmani, R., Savic, D.A. & Walters, G.A., Evolutionary multi-objective optimization in water distribution network design. *Engineering Optimization*, **37(2)**, pp. 167–183, 2005.

[24] Mackle, G., Savic, D.A. & Walters, G.A., Application of genetic algorithms to pump scheduling for water supply. *Proceedings of the Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA '95*, IEE, London, pp. 400–405, 1995.

[25] Savic, D. A., Walters, G.A. & Schwab, M., Multiobjective genetic algorithms for pump scheduling in water supply. *Proceedings of the AISB '97, Lecture Notes in Computer Science 1305*, D. Corne & J. L. Shapiro, eds., Springer, Berlin, pp. 227–236, 1997.

[26] Atkinson, R., van Zyl, J.E., Walters, G.A. & Savic, D.A., Genetic algorithm optimisation of level-controlled pumping station operation. *Water Network Modelling for Optimal Design and Management*, Centre for Water Systems, Exeter, UK, pp. 79–90, 2000.

[27] van Zyl, J., Savic, D.A. & Walters, G.A., Operational optimization of water distribution systems using a hybrid genetic algorithm method. *Journal of Water Resources Planning and Management*, **130(2)**, pp. 160–170, 2004.

[28] Rao, Z. & Salomons, E., Development of a real-time, near-optimal control process for water-distribution networks. *Journal of Hydroinformatics*, **9(1)**, pp. 25–37, 2007.

[29] Yates, D.F., Templeman, A.B. & Boffey, T.B., The computational complexity of the problem of determining least capital cost designs for water supply networks. *Engineering Optimization*, **7(2)**, pp. 142–155, 1984.

[30] Reca J. & Martinez, J., Genetic algorithms for the design of looped irrigation water distribution systems. *Water Resources Research*, **42(5)**: W05416, doi:10.1029/2005WR004383, 2006

[31] Lamaddalena, N. & Sagardoy, J.A., Performance analysis of on-demand pressurized irrigation systems, Irrigation and Drainage Paper No. 59, FAO, Rome, 2000.

[32] Radcliff, N.J., Forma analysis and random respectful recombination. *Proceedings of the 4th International Conference on Genetic Algorithms*, Morgan Kauffman, San Mateo, CA, 1991.

[33] Haupt, R.K. & Haupt. S.E., Practiscal Genetic Algorithms, John Wiley & Sons, New York, 2004.

[34] Davis, L., Hybridization and numerical representation. *The Handbook of Genetic Algorithms*. ed. L. Davis, Van Rostrand Reinhold, New York, pp. 61–71, 1991.

[35] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*. 2nd edition Springer-Verlag, New York, 1994.

This page intentionally left blank

CHAPTER 4

Simulated annealing algorithms for water systems optimization

M.C. Cunha & A. Antunes
Department of Civil Engineering, Coimbra University, Portugal.

Abstract

Simulated annealing is a heuristic method based on the analogy with the physical annealing process. In the past 20 years, this method has been used to solve many complex optimization problems, and it has become known for its ability to produce good results in reasonable time. This chapter starts with an explanation of why heuristic methods like simulated annealing are needed to address water systems development problems. The main issues involved in the implementation of this method are examined next, followed by their discussion with respect to application to water systems development problems.
Keywords: optimization, simulated annealing, water systems development.

1 Introduction

Water system development problems usually involve the placement and operation of infrastructure. To find the optimal solution of such problems, when all the costs are considered, is a very difficult task. In fact, non-linear models that include discrete variables have to be solved. This is why such problems are often simplified either by ignoring fixed costs or by linearizing the cost functions.

A variety of optimization models have been developed for the planning and management of water systems, ranging from those relating to the optimal operation of existing infrastructure (where there are no location and construction costs to be considered) to those aiming at planning the exploitation of unused systems (where there are location, construction, and operating costs to be considered). From the point of view of their mathematical characteristics, it can be said that these problems are very different in nature, which have consequences for the degree of complexity of their resolution.

The first type usually involves linear and nonlinear functions containing continuous variables for the flows to be operated, physical management, etc. The literature shows that they can be solved through the use of the classical programming methods, even for large-scale real-world case studies.

The second type usually involves an objective function expressing cost minimization of infrastructure placement and operation. The solution of such problems implies determining the values of decision variables with respect to operating rates and the values of discrete decision variables relating to the number and the location of hydraulic infrastructure, while meeting some constraints. Therefore, the problems are mixed-integer non-linear optimization problems (MINLP), that are among the most difficult optimization problems to solve. There are many engineering fields dealing with this type of problem, which is why the literature reports a variety of attempts to solve them. The corresponding models were often simplified by disregarding the fixed costs incurred by installing the infrastructure and, as such, the discontinuous variables are not considered, resulting in the models becoming standard non-linear programs (NLP). Another way of converting these problems into an NLP is to preselect the type and the number of items of equipment and their location. This can be done in an iterative fashion, updating the type, number, and the location of the equipment items in each iteration, but with this kind of procedure, there will be many doubts about the quality of the solution (even near-optimal solutions may be difficult to obtain). In other cases, cost functions are linearized and they become standard mixed-integer linear problems (MILP).

In the last 25 years, developments in the field of heuristic optimization algorithms have made it possible to solve real-world problems involving planning and management issues without considering the aforementioned simplifications.

Simulated annealing is one of the best known heuristic optimization algorithms that, due to its conception, produce high-quality results. Simulated annealing is a random search based algorithm that allows asymptotic convergence to optimal solutions under mild conditions [1–3]. In the next sections, some theoretical bases and practical aspects concerning its application to water systems development are presented.

2 Theoretical aspects

Simulated annealing is a local search method that theoretically allows the identification of a global optimum of a function. This property is related to the fact that during the search procedure, mainly in the first phase, the algorithm accepts worse solutions. The first example of an annealing algorithm was presented by Metropolis for a thermodynamic problem regarding the evolution of the energetic state of a piece of glass subjected to a high temperature that is progressively decreased. Since then and after the pioneering work by Kirkpatrick *et al.* [4], many applications of annealing algorithms have been developed.

The theoretical bases of annealing algorithms are summed up below.

Consider a system S represented by the set $X = \{1,..., n\}$ of admissible configurations and P a problem representing the minimization of a function $e = e(i) : X \rightarrow R$, that is:

$$P : e^* = \min_{i \in X} e = e(i).$$

According to the thermodynamic terminology, the function $e = e(i)$ is called 'system energy'.

Consider π to be a probability distribution defined through X in terms of a function $f : R^2 \to R$ and a parameter $\tau \in R^+$ with general form

$$\pi(\tau) = [\pi(1,\tau),\ldots,\pi(n,\tau)] = \left[\frac{f[e(1),\tau]}{\sum_i [e(i),\tau]},\ldots,\frac{f[e(n),\tau]}{\sum_i [e(i),\tau]}\right].$$

According to the thermodynamic terminology, the parameter τ is called 'system temperature'.

The expected value of the system energy at temperature τ will be $\langle e \rangle_\tau = \sum_i e(i)\, \pi(i,\tau)$. It is possible to demonstrate the existence of distributions $\pi(\tau)$ for which $\lim_{\tau\to\infty} \langle e \rangle_\tau = e^*$ [5]. This is the case of distributions where $f[e(i),\tau] = \exp[-g(\tau)\cdot e(i)]$ if $g(\tau)$ meets the following assumptions: $\lim_{\tau\to 0} g(\tau) = +\infty$ and $g'(\tau) < 0$.

Among these distributions, there is the distribution $g(\tau) = a.\tau^{-b}$ and, when $a = 1/k$ and $b = 1$ (where k is the Boltzmann constant), the Boltzmann–Gibbs distribution occur.

To obtain a distribution with these characteristics, it is necessary, in principle, to use a Monte Carlo method of the type proposed by Metropolis because it is usually difficult to establish such distribution by analytical means. This method consists of associating the configuration space and a Markov chain having this distribution as equilibrium distribution in the following manner.

Let $V = \lfloor v_{ij} \rfloor$ be the matrix containing the neighboring configurations of the system, making it probable that a direct move (move with only one step) from configuration i to configuration j could occur, no matter what the configurations are; $\boldsymbol{A}$ be a Markov chain defined over the system's configurations and $A = \lfloor a_{ij} \rfloor$ be a transition matrix of this chain, giving the probability of the direct transition from configuration i to configuration j, defined by

$$a_{ij} = v_{ij} \min\left[1, \frac{\pi(j,\tau)}{\pi(i,\tau)}\right] \Leftarrow i \neq 0$$

$$a_{ii} = 1 - \sum_{i \neq j} a_{ij}.$$

Considering that configurations are organized by the inverse order of the respective energy levels (i.e. $i \langle j \Leftarrow e(i) \rangle e(j)$), it is possible to prove that if the matrix $\boldsymbol{V}$ is symmetric and the chain $\boldsymbol{A}$ is irreducible, then chain $\boldsymbol{A}$ will have unique equilibrium distribution [6]:

$$\pi(\tau) = \pi(1,\tau)\left[1, \frac{\pi(2,\tau)}{\pi(1,\tau)},\ldots,\frac{\pi(n,\tau)}{\pi(1,\tau)}\right].$$

In particular, when $f[e(i),\tau]=\exp[-e(i)/\tau]$, one will have

$$\pi(\tau)=\pi(1,\tau)\left\{1,\exp\left[\frac{-\Delta e(2)}{\tau}\right],\ldots,\exp\left[\frac{-\Delta e(n)}{\tau}\right]\right\},$$

where

$$\pi(1,\tau)=\frac{1}{\sum_i \exp\left[\frac{-\Delta e(i)}{\tau}\right]},$$
$$\Delta e(i)=e(i)-e(1).$$

This distribution is designated hereafter as reference distribution.
It follows that

$$\pi(\tau)=\left(\frac{1}{n^*},\ldots,\frac{1}{n^*},0,\ldots,0\right)\Leftarrow\tau\approx 0,$$

where n^* is the number of global minima, and

$$\pi(\tau)=\left(\frac{1}{n},\frac{1}{n},\ldots,\frac{1}{n}\right)\Leftarrow\tau=+\infty.$$

For the reference distribution at equilibrium, one will have

$$\text{prob}[e(i)-e^*<\partial]\geq 1-(n-1)\exp\left(-\frac{\partial}{\tau}\right),\partial>0.$$

If a random draw sequence of transitions between configurations is performed and these transitions are accepted according to the respective probabilities, it is quite likely that, after a large number of random draws, the system will attain the minimum energy configuration (if this procedure is carried out at very low temperature, the minimum energy configuration will be attained almost certainly).

In formal terms, if the sequence of random draws is represented by $\{S_j\}_{j=1,\infty}$, it is possible to write

$$\lim_{j\to\infty}\text{prob}\{S_j=i\,|\,\tau\}=\pi(i,\tau),\forall i\in\boldsymbol{X},$$
$$\lim_{\tau\to 0}\left(\lim_{j\to\infty}\text{prob}\{S_j=i\,|\,\tau\}\right)=\pi(i,0),\forall i\in\boldsymbol{X}.$$

Regarding the speed of convergence to reach equilibrium, it is known that the convergence rate is geometric and that the convergence factor is equal to the second eigenvalue of the transition matrix [6]. It is also known that for low temperatures and for problems with many local minima, the eigenvalue is close to one (this means a slow rate of convergence). This implies that it is useful to carry out the random draw procedure while decreasing the temperature of the system.

Following the thermodynamic analogy, the sequence $\{\tau_j\}_{j=1,\infty}$ with $\tau_j > \tau_{j+1}, \forall_j$ and $\lim_{j\to\infty} \tau_j = 0$, which represents the temperature levels during the random draw, is called cooling schedule. This schedule should be chosen such that the convergence of the procedure toward the equilibrium is not compromised.

It is possible to prove that if $\boldsymbol{V}$ is symmetric and $\boldsymbol{A}$ is irreducible and if $\pi(\tau)$ is the reference distribution [7], then

$$\lim_{j\to\infty} \text{prob}\{S_j = i|\tau\} = 0, \forall i \notin \boldsymbol{X^m},$$

where

$\boldsymbol{X^m}$: set of local minima of $e = e(i)$;

$$\lim_{j\to\infty} \text{prob}\{S_j = i|\tau\} = 0, \forall i \in \boldsymbol{X'} \Leftrightarrow \sum_{j=1}^{\infty} \exp\left(-\frac{d}{\tau_j}\right) = +\infty,$$

where

$$\boldsymbol{X'}: \boldsymbol{X^m} \backslash \boldsymbol{X}^*;$$

X^* : set of global minima of $e = e(i)$;

d: difference between the local minimum and the global maximum of the energy;

$$\lim_{j\to\infty} \text{prob}\{S_j = i|\tau\} = 0, \forall i \in \boldsymbol{X}^* \Leftrightarrow \sum_{j=1}^{\infty} \exp\left(-\frac{d^*}{\tau_j}\right) = +\infty,$$

where $d^* = \max_{i\in X'} d_i$.

This means that, to achieve the global minimum, it is necessary (and sufficient) that τ_j implies the series

$$\sum_{j=1}^{\infty} \exp\left(-\frac{a}{\tau_j}\right), \forall a \geq d^* \geq \left[\max_{i\in X} e(i) - \min_{i\in X} e(i)\right]$$

to be divergent.

A series such as $\sum_{j=1}^{\infty} 1/(1+j)$, which may be obtained using the cooling schedule $\tau_j = a/\{\ln(1+j)\}$, $j \geq 1$, fulfills this condition. Series obtained through the use of exponential or linear cooling schedule would not fulfill these conditions.

3 Algorithm implementation: general aspects

To implement a simulated annealing algorithm, the solution under evaluation in each iteration (current solution) is improved by generating small random changes in an iterative way. For each new solution, the value of the objective function is evaluated and the solution is accepted or not according to a given probability (Metropolis criterion). The overall procedure for generating solutions, with

each new one being built from the previous one, gives rise to a Markov chain. The broad evaluation of the solution space enables simulated annealing to evolve without being trapped in local optima.

The simulated annealing algorithm can be built using the following steps (summarizing the principles presented in Section 2):

1. choose s_1 {s_1 is the initial solution and current solution for the first iteration}
2. choose τ_1 {τ_1 is the initial temperature}
3. choose τ_f {τ_f is the stopping temperature}
4. $j \leftarrow 0$
5. **repeat**
6. $j \leftarrow j+1$
7. choose at random $s'_j \in \Omega(s_j)$ {$\Omega(s_j)$ is the candidate set of s_j}
8. choose at random $p \in [0,1]$ {p is a probability}
9. **if** $p \leq \min\left\{1, \exp\left(\frac{C(s_j) - C(s'_j)}{\tau_j}\right)\right\}$ {Metropolis Criterion}
   {$C(s_j)$ is the objective function value corresponding to the solution s_j}
   then $s_{j+1} \leftarrow s'_j$ {s_{j+1} future current solution; s'_j candidate solution}
   else $s_{j+1} \leftarrow s_j$
10. choose $\tau_{j+1} \leq \tau_j$
    until $\tau_{j+1} \leq \tau_f$
11. **end**

The general algorithm presented above can be used to solve a multitude of problems, but some choices have to be made. The implementation of the annealing algorithm to solve water system problems can follow a general procedure, as depicted in Figure 1.

Once the cost function is established, two more elements are needed to implement the annealing algorithm: the procedure for generating candidate solutions (known as rearrangements) and the schedule for decreasing temperatures (known as cooling schedule). To implement a simulated annealing algorithm, it is necessary to provide an initial solution. This solution will be randomly changed while the algorithm develops. The development of the algorithm is controlled by the parameter called temperature. The initial value of this parameter is defined, and it will decrease along the iterative procedure until reaching a predefined value. In each iteration, the solution to be analyzed, called the candidate solution, depends on the neighborhood structure employed and the construction of this neighborhood is highly dependent on the specific problem to solve.

To establish the cooling schedule, four parameters have to be considered:

- *ea*: the probability of accepting a transition from the initial solution to a candidate solution whose cost is higher than that of the starting configuration by a given percentage (this parameter is used to define the initial temperature of the annealing process and is called ‘elasticity of acceptance’). In these conditions,

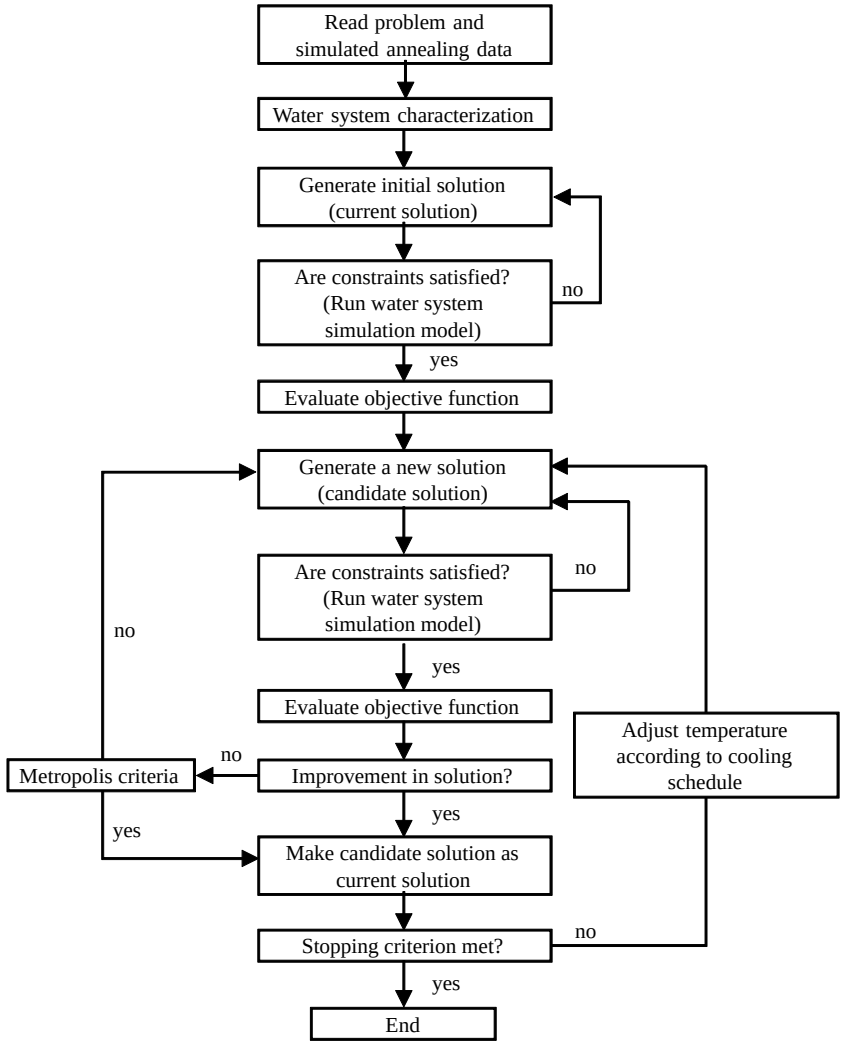


Figure 1: Simulated annealing algorithm flow chart.

one might expect that the final solution would be independent of the initial solution. For practical purposes, the initial temperature τ_1 can be determined as a function of the parameter *ea* through the equation $\tau_1 = 0.1c_0/\ln ea$ (where c_0 is the cost of the initial solution). This expression allows us to identify a temperature for which *ea*% of solutions whose cost is 10% higher than the cost of the starting configuration are accepted.

- n_1: the minimum number of algorithm iterations that will be performed, even without an improvement of the optimum or of the average cost of the solutions analyzed, before decreasing the temperature.

- γ: the cooling factor, rate at which temperature is decreased, whenever a temperature decrease should occur.
- n_2: the number of temperature decreases that will be performed without an improvement of the optimum or the average cost of the solutions analyzed, before stopping the algorithm.

The Metropolis criterion, which is used to accept or reject a candidate solution, is influenced by the temperature parameter (parameter whose name is inspired by the physical process on which simulated annealing is based). A high initial temperature signifies a high rate of solution acceptance and thus a broad covering of the solution space. As the procedure continues, the temperature is lowered and the rate of acceptance decreases.

The definition of the initial and the final temperatures, of a cooling factor for changing the temperature from iteration to iteration and the number of iterations to be performed at each temperature, needs calibration to ensure that the algorithm is robust (i.e. the results of its application are not sensitive to random number generation). A detailed description of the calibration of these parameters can be found in [8] and [9]. Figure 2 shows how the different annealing parameters work together.

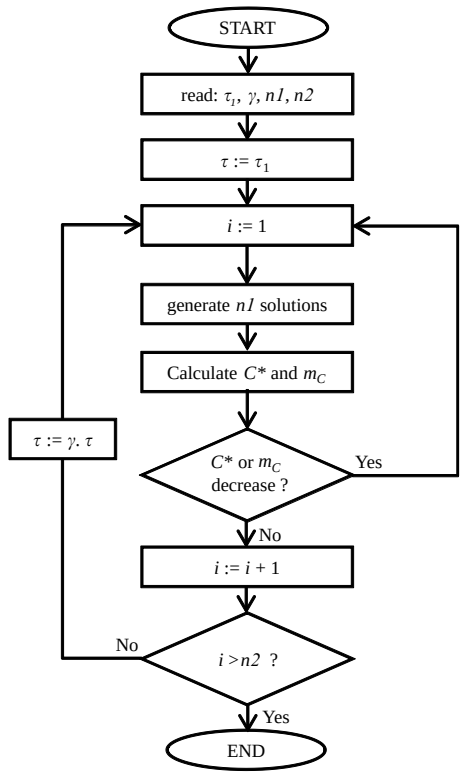


Figure 2: The cooling schedule.

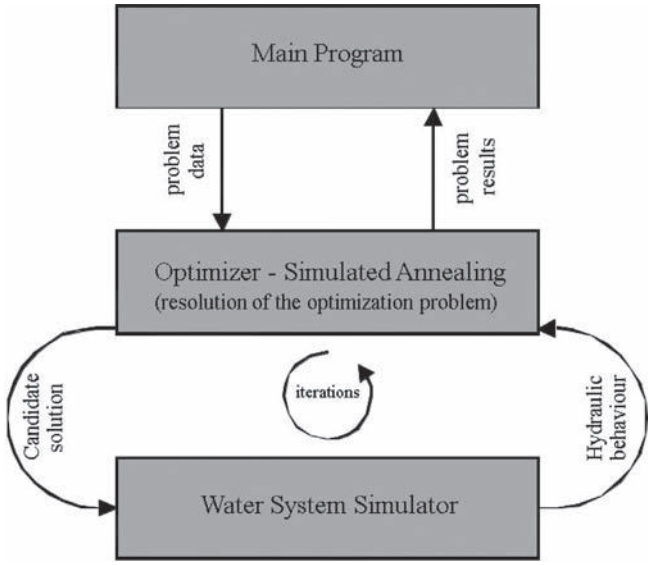


Figure 3: Resolution procedure.

Figure 3 shows how the application of simulated annealing algorithms works to solve water system development problems. It involves a main program including two solvers: the optimization solver and the water system simulator. The optimization solver, including the different steps of the simulated algorithm, calls upon the water system simulator to check the compliance of constraints for each candidate solution.

4 Algorithm implementation: specific aspects

The procedure for generating the initial solution and the candidate solutions (known as rearrangements) and the schedule for decreasing temperatures (known as cooling schedule) is problem dependent. The next sections detail some specific aspects related to the implementation of simulating annealing algorithms for three different water systems.

4.1 Aquifer systems

Let us suppose that an aquifer will be operated to satisfy the demand of a given number of consumption centers at minimum cost. Resolving this decision problem by means of a simulated annealing algorithm implies the following tasks:

- specifying all possible location for well development (previously defined through stakeholder involvement);
- specifying all cost functions;

- generating an initial solution of the problem;
- estimating the cost of the solution;
- checking compliance of the constraints (solving groundwater simulator for drawdown evaluation);
- generating a new solution;
- comparing the new solution with the previous one;
- applying the simulated annealing requirements;
- verifying the convergence criteria.

The initial solution can be set using all the sites previously specified to satisfy the demands. The algorithm can randomly choose the number of sites to satisfy each center and then the demand is equally divided between these sites.

The generation of new solutions (candidate solutions) as the algorithm proceeds should be suggested. As examples of generation, the following options, randomly chosen by the algorithm, can be proposed:

- An open site is chosen at random and the flow pumped there is decreased by a small amount of flow ΔQ. This ΔQ added to another open site is also randomly chosen.
- A randomly chosen open site is closed and the flow pumped there will be pumped at another randomly chosen site (already open or not).
- The flow pumped at a randomly chosen open site will be pumped at a randomly chosen number of open sites.

These ways of generating candidate solutions can be combined in varying proportions during the search procedure.

After the generation of each candidate solution, the effects in terms of costs (objective function) and in terms of drawdowns must be determined. A main program for managing these actions is therefore needed. The optimizer should call upon a groundwater simulator to perform the calculations mentioned.

It is important to have an efficient and robust algorithm, this means that the calibration of the parameters included in the cooling schedule has to be carried out carefully. During this calibration, sets of cooling schedule parameters are evaluated against the quality of the solution obtained and the computer running time. For example, taking the problem described in [10] (for an aquifer management problem, as described below), Figures 4 and 5 show the influence of the parameters corresponding to the minimum number of simulations or algorithm iterations (also called minimum chain length) at each temperature (n_1) and the stopping criteria (n_2) on the quality of the solution and on computer running times. Analyzing these two figures, the values to be chosen can be $n_1 = 150$ and $n_2 = 20$.

The number of simulations or algorithm iterations run at each temperature is a crucial parameter for obtaining an optimal solution. It will have a decisive influence on the computation time and on the rate of convergence.

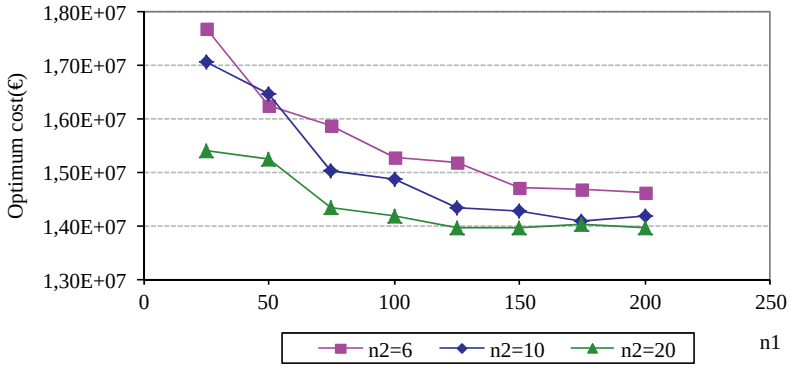


Figure 4: Influence of n_1 and n_2 on the final solution.

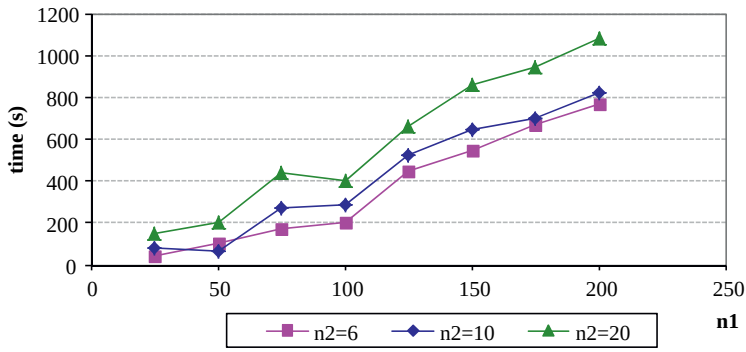


Figure 5: Influence of n_1 and n_2 on the computer running time.

4.2 Water distribution systems

Let us suppose that a looped gravity-fed water distribution system, whose layout is known, is to be designed at minimum cost. Solving this decision problem by means of a simulated annealing algorithm implies the following tasks:

- Specifying all possible commercial diameters that can be used to size the pipes;
- Specifying all cost functions;
- Generating an initial solution of the problem;
- Estimating the cost of the solution;
- Checking compliance of the constraints (resolving hydraulic equilibrium equations simulator for determining flow distribution, node pressures, and velocities);
- Generating a new solution;
- Comparing the new solution with the previous one;
- Applying the simulated annealing requirements;
- Verifying the convergence criteria.

The initial solution can be set by having all the pipes of the maximum diameter.

The generation of new solutions (candidate solutions) as the algorithm proceeds should be suggested. As examples of generation the following options, randomly chosen by the algorithm, may be proposed:

- A pipe is chosen at random and is assigned a new diameter. The new diameter can be larger or smaller than the present diameter. The number of times that the diameter is increased or decreased is also to be calibrated.
- Two pipes are chosen at random and are assigned with new diameters. The new diameters can be larger or smaller than the present diameters. The number of times that the diameters are increased or decreased is also to be calibrated.
- Two pipes are chosen at random and are assigned new diameters. One is assigned a diameter larger than the present one, the other is assigned a diameter smaller than the present one.

After the generation of each candidate solution, the effects in terms of cost (objective function) and hydraulic behavior must be determined. A main program for managing these actions is therefore needed. The optimizer should call upon a hydraulic equilibrium equation simulator to perform the relevant calculations.

For the calibration procedure, Figures 6 and 7 taken from the calibration of the Alperovits and Shamir looped network [11] show the influence of the parameters corresponding to the cooling factor and to the number of simulations (minimum chain length) at each temperature (n_1) on the computational effort. The cooling factor transforms the temperature throughout the cooling procedure in a geometric fashion. This means that higher cooling factors produce slower temperature decrease.

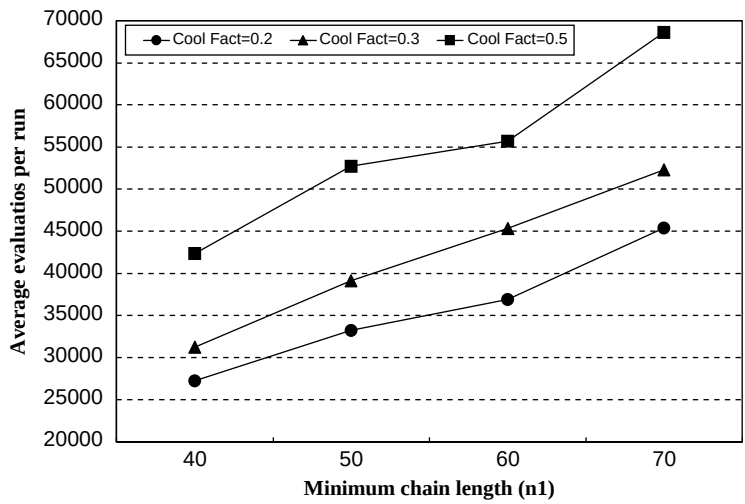


Figure 6: Effect of the cooling factor and the minimum chain length on the computational effort.

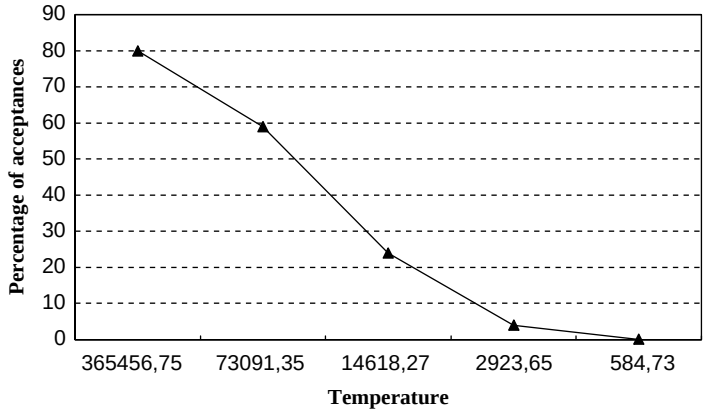


Figure 7: Percentage of acceptances with temperature.

Figure 7 depicts the rate of solution acceptance as the temperature decreases. The shape obtained fits the general pattern described before. In fact, the temperature is used to control the probability of accepting worsening moves. For high temperatures, most uphill moves are accepted. As the temperature level approaches zero, most uphill moves will be rejected.

4.3 Wastewater systems

Let us suppose that the siting and sizing of the different components of a regional wastewater system is to be decided at minimum cost. The essential components of the regional wastewater system are sewer networks and treatment plants. There is a river that will receive the treated effluent.

Solving this decision model by means of a simulated annealing algorithm implies the following tasks:

- specifying all wastewater sources and their location;
- listing all possible locations for siting wastewater treatment plants (previously defined through stakeholder involvement);
- designing a basic network containing all possible links between wastewater sources and wastewater treatment plants' possible locations;
- generating an initial solution of the problem;
- estimating the costs of the solution;
- checking compliance of the constraints;
- generating a new solution;
- comparing the new solution with the previous one;
- applying the simulated annealing requirements;
- verifying the convergence criteria.

The initial solution can be set by randomly assigning the sewers that will be linked to each possible wastewater treatment plant.

The generation of new solutions (candidate solutions) as the algorithm proceeds should be suggested. An example of generation can be proposed:

- A randomly chosen sewer included in the current configuration is replaced by a randomly chosen sewer not included in the current configuration but connected to the same wastewater source or intermediate node (Figure 8).

After the generation of each candidate solution, the effects in terms of cost (objective function) and hydraulic behavior of the sewer system and water quality in the river must be determined. A main program for managing these actions is therefore needed. The optimizer should call upon a hydraulic sewer flow simulator and an advection diffusion simulator to perform the relevant calculations.

Figure 9 depicts information about a case study taken from [13]. It includes 13 wastewater sources and 6 possible locations for wastewater treatment plants.

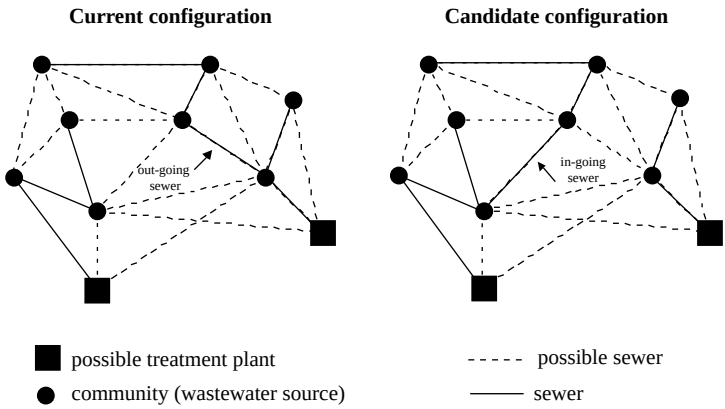


Figure 8: Generation of candidate solutions (adapted from [12]).

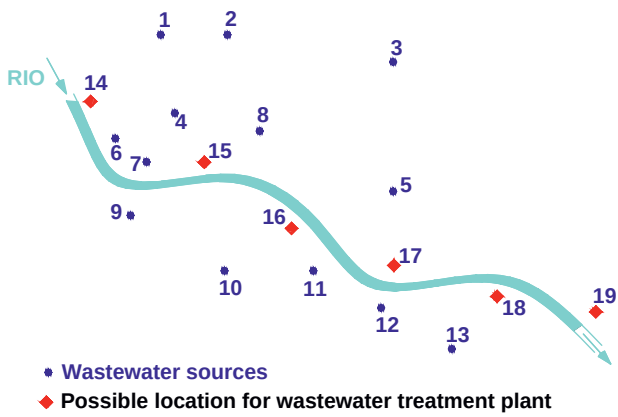


Figure 9: Wastewater system case study.

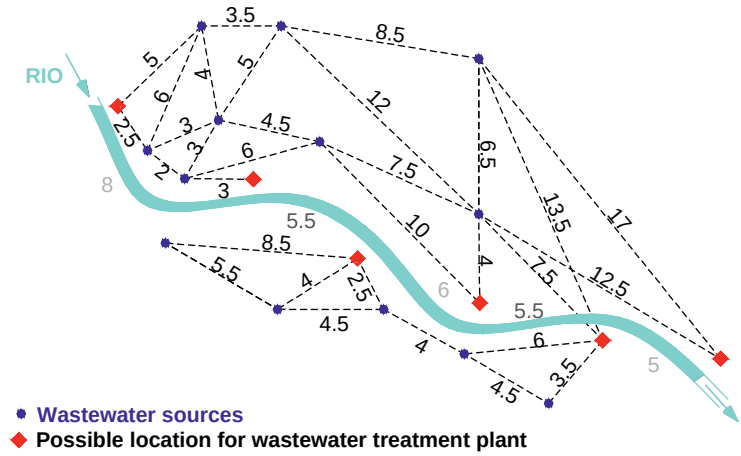


Figure 10: Basic network.

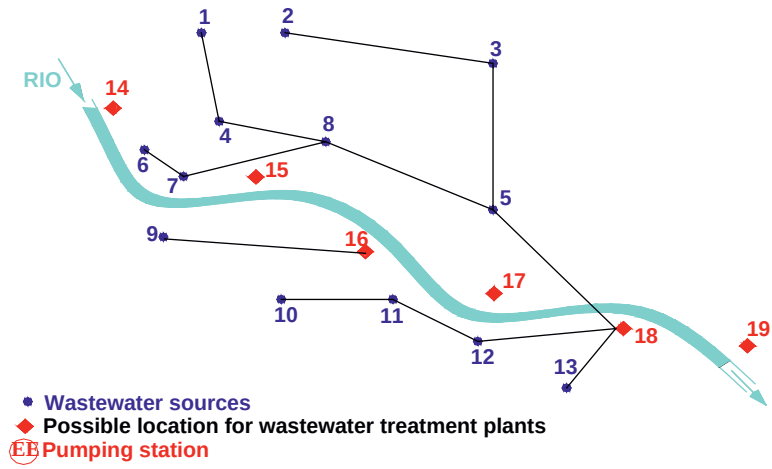


Figure 11: Optimal configuration (DO>7.0 mg/l).

Figure 10 represents the basic network of the problem. The nodes of this network are the wastewater sources, the locations for possible treatment plants, and the possible intermediate nodes. The arcs are the possible sewers linking the different types of nodes. The final solution will be a part of this basic network. Figure 11 presents the optimal configuration for the sewer network and for the location of the wastewater treatment plants (in this case just one) when the only constraint related to the quality parameter to be observed in the river is the minimum value of the dissolved oxygen (*DO*). Figure 12 depicts the optimal configuration when a more demanding environmental solution is needed. This means that various additional quality parameters have to be verified: total nitrogen (N), total phosphorus (P), and Kjeldahl nitrogen (Nkj) values.

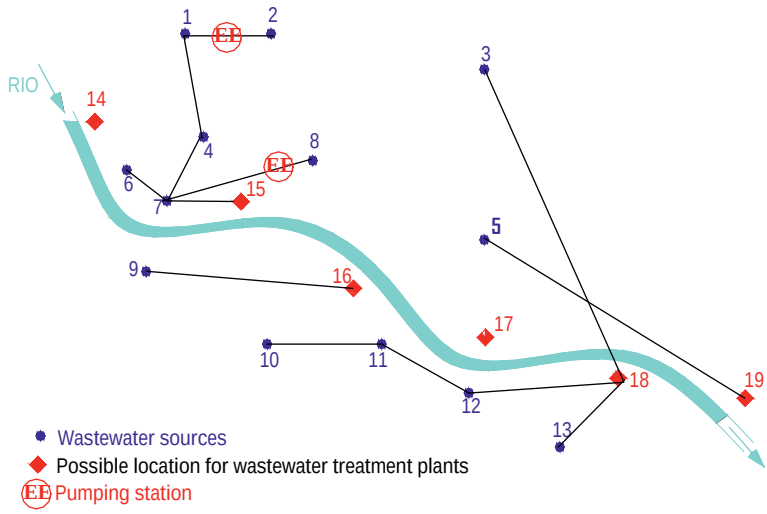


Figure 12: Optimal configuration (DO>7.0 mg/l; N<7.5 mg/l; P<1.0 mg/l, and Nkj<3.5 mg/l).

5 Conclusions

The mathematical nature of water systems planning and design problems is such that heuristic optimization methods like simulated annealing can be very helpful. Simulated annealing is one of the best known heuristic optimization algorithms, which, due to its underlying principles, produces high-quality results. It is a random search based method that allows asymptotic convergence to optimal solutions under mild conditions. Some features of the application of this method to water systems development were discussed and some particular aspects of its implementation were described with respect to different types of water systems. A careful calibration of simulated annealing parameters and a sound generation of candidate solutions are crucial aspects for the success of this method. The use of this type of heuristic method contributes to overcome the difficulties for resolving real-world problems because all their features can properly be taken into account, therefore avoiding excessive problem simplifications.

References

[1] Aarts, E. & Korst, J., *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Chichester, England, 1989.

[2] Reves, C. (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Wiley, Chichester, England, 1990.

[3] Aarts, E.H., Korst, J.H. & Van Laarhoven, P.J., *Simulated Annealing, in Local Search in Combinatorial Optimization*, ed. E. Aarts & J.K. Lenstra, John Wiley & Sons, New York, pp. 91–120, 1997.
[4] Kirkpatrick, S., Gelatt C. & Vecchi, M., Optimization by simulated annealing. *Science*, **220(4598)**, pp. 671–680, 1983.
[5] Rosier, Y.M., Troyon, M. & Liebling, Th., Probabilistic exchange algorithms and Euclidean traveling salesman problems. *OR Spektrum*, **8**, pp. 151–154, 1986.
[6] Lundy, M. & Mees, A., Convergence of an annealing algorithm. *Mathematical Programming*, **34**, pp. 111–124, 1986.
[7] Hajek, B., Cooling schedules for optimal annealing. *Mathematics of Operations Research*, **13**, pp. 311–329, 1988.
[8] Cunha, M.C. & Sousa, J.J.O., Water distribution network design optimization: Simulated Annealing approach. *Journal of Water Resources Planning and Management, ASCE*, **125(4)**, pp. 215–221, 1999.
[9] Cunha, M.C., On solving aquifer management problems with simulated annealing algorithms. *Water Resources Management*, **13(3)**, pp. 153–169, 1999.
[10] Fragoso, T., Cunha, M.C. & Lobo-Ferreira, J.P., Aquifer development optimization: the case of Palmela aquifer. *Water Resources Management II, International Series on Progress in Water Resources*, Vol 8, ed. C.A. Brebbia, WIT Press, Southampton, UK, pp. 3–12, 2003.
[11] Alperovits, E. & Shamir, U., Design of optimal water distribution systems. *Water Resources Research*, **13(6)**, pp. 885–900, 1977.
[12] Sousa, J., Ribeiro, A., Cunha, M.C. & Antunes, A., An optimization approach to wastewater systems planning at regional level. *Journal of Hydroinformatics*, **4(2)**, pp. 115–123, 2002.
[13] Cunha, M.C., Pinheiro, L., Afonso, P. & Sousa, J., Decision-aid models for the implementation of the water framework directive, *Proceedings of the 4th International Conference in Decision Making in Urban and Civil Engineering*, DMUCE, CD Edition, 10 p., Porto, 2004.

This page intentionally left blank

CHAPTER 5

Particle swarm optimisation

J. Izquierdo[1], I. Montalvo[2], R. Pérez-García[1] & M. Herrera[1]
[1]*FluIng-IMM, Universitat Politècnica de València, Valencia, Spain.*
[2]*Ingeniousware, Karlsruhe, Germany.*

Abstract

Due to the various limitations shown by classical techniques of optimisation when dealing with many real-world engineering problems, a number of paradigms have been produced over the last 20 years that claim to be better suited to providing useful engineering solutions for these types of problems. One of the evolutionary algorithms that has shown great potential for solving various optimisation problems is (PSO), which stands for *particle swarm optimisation*. PSO is a multi-agent optimisation system inspired by the social behaviour of a group of migrating birds trying to reach an unknown destination. In this chapter, the basics of PSO are provided together with some modifications that considerably improve the performance of the standard algorithm. This variant features: a mixed continuous-discrete variant of PSO; a mechanism to enrich population diversity; and a self-adapting feature that spares engineers the task of parameter selection and fine-tuning. This variant can find solutions efficiently for various optimisation problems in the water field. In addition, we provide the necessary details for this algorithm to work with multi-objective optimisation problems. Multiobjective optimisation is essential in many decision-making processes. From a practical standpoint, the development of a multi-objective optimisation process enables the combination of economic, engineering, and policy viewpoints when searching for a solution to a problem. Finally, we reference a number of applications of these approaches in various areas of hydrology, hydraulics, and water resource management, and present the details of the complex problem of designing water distribution networks, together with the solutions to two real-world case studies.

Keywords: design, multi-objective optimisation, particle swarm optimisation, reliability, water distribution network.

1 Introduction

Optimisation in engineering, in general, and in water systems, in particular, is crucial for many reasons. Design is necessary to implement new configurations, improve existing systems, continue satisfying various needs, and expand systems to meet new conditions. Taking into account the uncertainty of data (especially in existing configurations), it is often necessary to solve difficult inverse problems where optimisation techniques are also of key importance. There are many examples of important industrial problems in the water field because there is a great deal of interest in mechanisms for managing sustainable water resources at a reasonable cost.

Optimisation is a typically constrained nonlinear search problem involving both continuous and discrete variables. The problem in hand is frequently a mixed continuous and discrete constrained nonlinear optimisation problem that is often highly dimensional and multimodal. Highly dimensional means that many decision variables influence the solution. Multimodal expresses the idea that there are many local optima in the search space. There is no single search algorithm for solving many real-world optimisation problems without compromising solution accuracy, computational efficiency, and problem completeness.

Classical methods of optimisation involve the use of gradients or higher order derivatives of the fitness function. But these methods are not well suited for many real-world problems since they cannot process inaccurate, noisy, discrete, and complex data. Robust methods of optimisation are often required to generate suitable results.

Over the last 20 years, many researchers, including those in the water field, have embarked on the implementation of a range of *evolutionary algorithms*: genetic algorithms, ant colony optimisation, PSO, simulated annealing, shuffled complex evolution, harmony search, and memetic algorithms, among many others. Some of these techniques are the objective of other chapters in this book.

This chapter presents the principles of PSO, which is one of the evolutionary algorithms that has shown great potential for the solution of various optimisation problems. The PSO algorithm was first developed by Kennedy and Eberhart [1] and is a multi-agent optimisation system inspired by the social behaviour of a group of migrating birds trying to reach an unknown destination. In addition to one of the versions of standard PSO, we describe several modifications (developed by the authors of this chapter) that considerably improve the performance of the standard algorithm when used to find solutions to various optimisation problems in the water field [2–6].

The remainder of this chapter is presented as follows. Firstly, PSO is concisely presented. Secondly, the proposed adaptations, namely, a mixed continuous-discrete variant of PSO, a mechanism to enrich diversity that greatly improves the performance of PSO, and a self-adapting characteristic that avoids the task of parameter selection, are then described. By developing the necessary elements, we then provide an adaptation of this algorithm for multi-objective optimisation problems. Finally, we show the results of specific applications to selected case-studies

regarding a very well-known urban water problem, namely, the design of water distribution systems (WDS). This is a crucial problem of industrial interest in the water field since increasing urban development represents a permanent challenge for the management of many resources – and especially water.

2 Description of particle swarm optimisation

PSO is an evolutionary computation technique that was first developed by Kennedy and Eberhart [1]. The particle swarm idea originated as a simulation of a simplified social system, the graceful but unpredictable choreography of a flock of birds. The word 'swarm' is used after a paper by Millonas [7], who developed several models for artificial life and examined certain principles in swarm intelligence. The selection of the term 'particle' comes from classical mechanics and is justified by the fact that positions and velocities are applied to the population elements, despite the fact that are considered to have zero mass and volume. Kennedy and Eberhart's first idea was to simulate the social behaviour of a flock of birds in their attempt to reach, when flying through the field (search space), their unknown destination (fitness function), e.g. the location of food resources.

2.1 Description of standard PSO

Each problem solution in PSO is a bird of the flock and is referred to as a particle. In this algorithm, birds evolve in terms of their individual and social behaviour and mutually coordinate their movement towards their destination [8].

Each bird keeps track of its coordinates in the problem space, remains aware of its recent trajectory, and remembers a dynamic specific position: the best solution (best local position) it has achieved so far. Birds also communicate among themselves and are able to identify the bird in the best position (best global position). In a coordinated way each bird evolves by changing its velocity so that it accelerates towards both its best position and the best position obtained so far by any bird in the flock without forgetting its recent trajectory. This enables each bird to explore the search space from its new location. The process is repeated until the best bird reaches a certain desired location. It is worth noting that, according to the description, the process involves not only individual intelligent behaviour (including memory) but also social interaction. In this way, birds somehow follow their recent history, learn both from their own experience (local search), and from group experience (global search).

PSO shows common evolutionary computation features including: (1) initialisation with a population of random solutions; (2) search for optima by updating generations; and (3) particle evolution through the problem space by following specific strategies.

The process initially starts with a group of M particles, which have been randomly generated, representing different solutions of the problem. The ith particle, X_i, is represented by its location in a d-dimensional subset, $S \subset R^d$, where

d corresponds to the number of variables of the problem. Any set of values of the d variables, determining the particle location, represents a candidate solution for the optimisation problem:

Find $\min_{X \in S} F(X)$, subject to appropriate constraints,

where F is the fitness function associated with the problem, a minimisation problem without loss of generality. The optimal solution is then searched for by iteration. The performance of each particle is measured using this fitness function, according to the problem in hand.

During the process, as already explained, each particle i is associated with three vectors:

- current position, $X_i = (x_{i1,}\ldots, x_{id})$;
- best position, $Y_i = (y_{i1,}\ldots, y_{id}) = \text{argmin}(F(X_i(t)), F(X_i(t-1)))$, reached in previous cycles; and
- flight velocity $V_i = (v_{i1,}\ldots, v_{id})$, which makes it evolve.

The bird which is in the best position, $Y^* = \text{argmin}\{F(X_i(t), i = 1,\ldots, M\}$, is identified for every iteration, t.

During each generation, the velocity of each particle is updated in a process based on its recent trajectory, its best encountered position, the best position encountered by any particle, and a number of parameters:

$$V_i \leftarrow \omega V_i + c_1\, rand\,()\,(Y_i - X_i) + c_2\, rand\,()\,(Y^* - X_i). \tag{1}$$

The parameters in (1) are as follows: ω is a factor of inertia suggested in [8] that controls the impact of the velocity history on the new velocity; c_1 and c_2 are two positive acceleration constants, called the cognitive and social parameters, respectively; *rand*() represents a function that creates random numbers between 0 and 1 (two independent random numbers enter eqn (1)).

Expression (1) is used to calculate the ith particle's new velocity, a determination that takes into consideration three main terms: the particle's previous velocity, the distance of the particle's current position from its own best position, and the distance of the particle's current position from the swarm's best experience (position of the best particle).

In each dimension, particle velocities are clamped to minimum and maximum velocities, which are user-defined parameters,

$$V_{\min} \le V_{ij} \le V_{\max}, \tag{2}$$

in order to control excessive roaming by particles outside the search space. These important parameters are problem dependent. They determine the resolution with which regions between the present position and the target (best so far) positions are searched. If velocities are too great, particles might fly through good solutions. If they are too slow, on the other hand, particles may not explore sufficiently beyond locally good regions – becoming easily trapped in local optima and unable to move far enough to reach a better position in the problem space. Usually, $V_{\min}$ is taken as $-V_{\max}$.

Finally, the position of each particle is updated every generation. This is performed by adding the velocity vector to the position vector,

$$X_i \leftarrow X_i + V_i. \tag{3}$$

Each particle or potential solution moves to a new position according to expression (3).

2.2 Combining continuous and discrete variables

The previously described algorithm can be considered as the standard PSO algorithm, which is applicable to continuous systems and cannot be used for discrete problems. Several approaches have been put forward to tackle discrete problems with PSO [9–12]. The approach we propose for discrete variables involves the use of the integer part of the discrete velocity components. In this way, the new velocity of discrete components will be an integer and, as a consequence, the new updated positions will share this characteristic since the initial population, in its turn, must also have been generated using only integer numbers. According to this simple idea, expression (1) will be replaced by

$$V_i \leftarrow fix(\omega V_i + c_1\, rand\,()\,(Y_i - X_i) + c_2\, rand\,()\,(Y^* - X_i)), \tag{4}$$

for discrete variables, where $fix(\cdot)$ is a function that takes the integer part of its argument. However, it should be taken into account that the new velocity discrete values must be controlled by suitable bounds as in (2). However, there is a singular aspect regarding velocity bounds that must be taken into consideration so that the algorithm can treat both continuous and discrete variables in a balanced way. In [3], it was found that using different velocity limits for discrete and continuous variables produce improved results.

2.3 Enriched diversity

The main drawback of PSO is the difficulty in maintaining acceptable levels of population diversity while balancing local and global searches; and as a result, suboptimal solutions are prematurely obtained [13]. Some evolutionary techniques maintain population diversity by using some more or less sophisticated operators or parameters. Several other mechanisms for forcing diversity in PSO can be found in the literature [14–16]. In general, the random character that is typical of evolutionary algorithms adds a degree of diversity to the manipulated populations. Nevertheless, in PSO these random components are unable to add sufficient diversity.

Frequent collisions of birds in the search space, especially with the leader, can be detected – as shown in [4]. This caused the effective size of the population to fall and the algorithm's effectiveness is consequently impaired. The study in [17] introduces a PSO derivative in which a few of the best birds are selected to check collisions, and colliding birds are randomly re-generated if collision occurs. This random re-generation of the many birds that collide with the best birds has been

shown to avoid premature convergence as it prevents clone populations from dominating the search. The inclusion of this procedure into PSO greatly increases diversity as well as improves convergence characteristics and the quality of the final solutions.

2.4 Self-adapting parameters

The role of the inertia, ω, in (1) and (4) is considered critical for the convergence behaviour of the PSO algorithm. Although inertia was constant in the early stages of the algorithm, it is currently allowed to vary from one cycle to the next. As it facilitates the balancing of global and local searches, it has been suggested that ω could be allowed to adaptively decrease linearly with time – usually in a way that initially emphasises global search and then, with each cycle of the iteration, increasingly prioritises local searches [18]. A significant improvement in the performance of PSO, with decreasing inertia weight across generations, is achieved by using the proposal [19]:

$$\omega = 0.5 + \frac{1}{2\left(\ln\left(t\right)+1\right)}. \tag{5}$$

In the variant we propose, the acceleration coefficients and the clamping velocities are neither set to a constant value, as in standard PSO, nor set as a time-varying function, as in adaptive PSO variants [20]. Instead, they are incorporated into the optimisation problem [5]. Each particle is allowed to self-adaptively set its own parameters by using the same process used by PSO – and given by expressions (1) or (4), and (3). To this end, these three parameters are considered as three new variables that are incorporated into the position vector X_i. In general, if d is the dimension of the problem, and p is the number of self-adapting parameters, the new position vector for particle i will be:

$$X_i = (x_{i1}, \ldots, x_{id}, x_{id+1}, \ldots, x_{id+p}). \tag{6}$$

These new variables do not enter the fitness function, but rather they are manipulated by using the same mixed individual-social learning paradigm used in PSO. Also, V_i and Y_i that give the velocity and thus far best position for particle i increase their dimension, correspondingly.

By using expressions (1) or (4), and (3), each particle is additionally endowed with the ability to self-adjust its parameters by taking into account the parameters it had at its best position in the past, as well as the parameters of the leader, which facilitated this best particle's move to its privileged position. As a result, particles use their cognition of individual thinking and social cooperation to improve their positions, as well as improving the way they improve their position by accommodating themselves to the best-known conditions, namely, their conditions and their leader's conditions when they achieved the thus-far best position.

3 Multi-objective PSO

In multi-objective optimisation, scores of objectives are not scalars but vectors that is to say that the objective space is multidimensional with as many dimensions as objectives considered. A change in the decision space may produce a positive increment in some components (objectives) of these vectors while, at the same time, cause lower values in other objectives. In a departure from the normal behaviour of particles in PSO – derived from the ordered nature of real numbers – particles now use the dominant solution concept when deciding on a *better* position: solution A is said to dominate another solution B when A is better than B in at least one objective, and not worse in the others. Two solutions are called indifferent or incomparable if neither dominates the other. Dominant solutions will always be considered as *better* solutions. Figure 1 illustrates the solution space (decision space), with three decision variables, x_1, x_2, and x_3, and their corresponding points in the objective space, with two objectives, namely, minimisation of z_1 and maximisation of z_2.

The empty particles are incomparable because none is better than the others in both objectives (check in the objective space), but the second-from-top empty particle dominates the solid particles since it is better in both objectives: lower z_1-value and higher z_2-value. Moreover, the particle corresponding to the second-from-bottom empty dot in the objective space seems to dominate the lowest three solid particles (lower z_1-value and equal z_2-value). In contrast, the other three empty particles do not dominate any of the solid particles. Finally, the upper solid particle dominates the three lower solid particles.

The objective of multi-objective optimisation is to obtain at least an approximation of the set of nondominated particles (in the decision space) and its image on the objective space, which is the so-called Pareto front [21] (in red, in Fig. 1).

The leadership in a swarm must be determined in a different manner than in the classical PSO algorithm. The most natural option is to select as leader the closest particle to the so-called *utopia point* in the objective space. The utopia point is

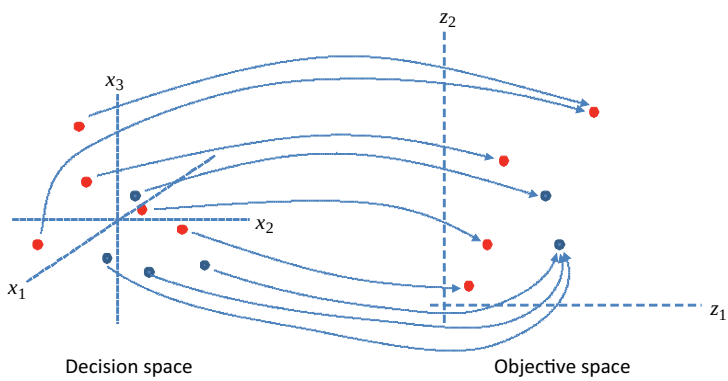


Figure 1: Relation between decision and objective spaces.

defined as the point in the objective space whose components give the best values for every objective. The utopia point is an unknown point since the best value for every objective is not known at the beginning (and perhaps during the whole process). Accordingly, we use a dynamic approximation of this utopia point, termed *singular point*, which is updated with the best values found so far during the evolution of the algorithm [22].

As each objective may be expressed in different units, it is necessary to make a regularisation for evaluating distances in the objective space. Coordinates may be regularised in terms of percentage, considering that for every component, the worst and best values of the corresponding objective are 0% and 100%, respectively. The percentage corresponding to any other value may be calculated using linear interpolation. To establish the distance between any two objective vectors, the components are first regularised in terms of percentage and then the Euclidean distance between them is calculated. The worst and best objective values are not usually known *a priori* and are updated while the solution space is explored.

Figure 2 shows a two-dimensional representation of the concept of a singular point. The most interesting solutions are located near the singular point and not too far from the ends of the Pareto front. For this reason, instead of seeking a complete and detailed Pareto front, we may be more interested in precise details around the singular point. However, situations may occur as shown in Fig. 2 (right) when non-symmetric Pareto fronts with respect to the singular point develop. As a result, poorly detailed sections on the Pareto front may appear. It seems plausible that problem complexity is the cause of this asymmetry in many real-world multi-objective optimisation problems.

It is difficult to find a general heuristic rule for deciding which parts of the Pareto front should be more closely represented and how much detail the representation of the Pareto front should contain. Various methods of inducing the Pareto

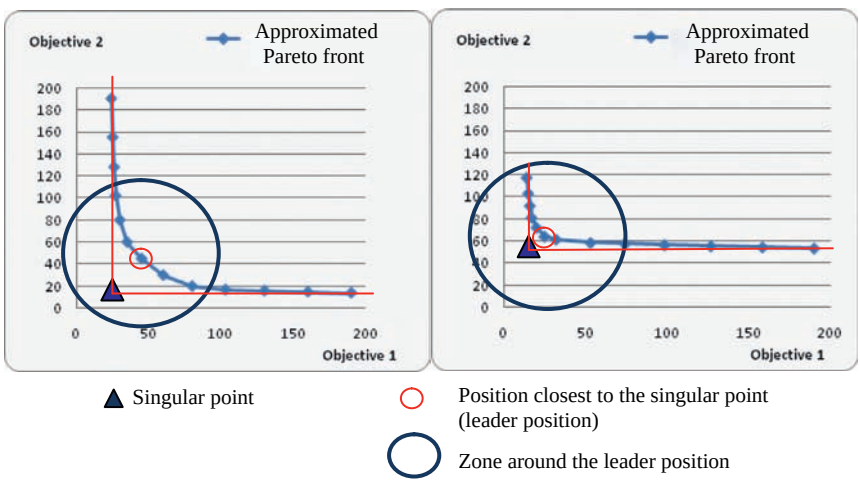


Figure 2: Singular point in an approximated Pareto front.

front completeness may be devised. We describe below a possible approach based on dynamic population increase to enrich the Pareto front density, and another alternative approach based on human computer interaction to complete poorly represented areas of the Pareto front.

3.1 Dynamic population increase

During the search process swarms are able to increase their population when needed in order to better define the Pareto front: a particle whose solution already belongs to the Pareto front may, on its evolution, find another solution belonging to the front. In this situation, a new clone of the particle is placed where the new solution is found, thus increasing the density of particles on the Pareto front. Greater densities on the Pareto front must be restricted to cases where the new clone has at least one of its neighbours located further away than some minimal permissible distance in the objective space. For example, in Fig. 3 (left), a particle J, whose objective vector is located at position P_J, finds a new position $NewP_J$, also belonging to the Pareto front.

The consequence is shown in Fig. 3 (right): a new particle k is added to the swarm by cloning the particle at position $NewP_J$, while particle J will continue to be active and considering the point P_J as its best objective position. This occurs because the new point P_k has at least one neighbour located further away than the minimum permissible distance for at least one of the objectives. In Fig. 3 (right), the point to the left of P_k is located at a distance, regarding Objective 1, that is greater than the minimal distance considered for the increase of density on the Pareto front. It should be noted that two particles are considered as neighbours when no other particle is located between them for at least one of the objectives considered in the problem.

3.2 Human–computer interaction

During evolution there may be areas poorly represented on the Pareto front. Users are allowed to add new swarms for searching in the desired region of the objective space. The concept of singular point is now extended to any desired area in the objective space for particles to search around.

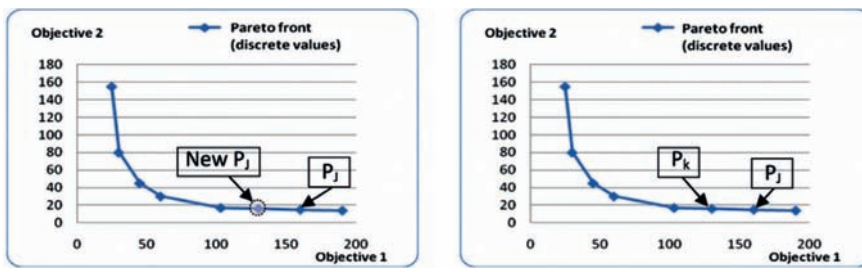


Figure 3: Particle cloning.

Decisions are strongly dependent on the individuals solving the problem and on the problem itself. The user can specify additional points where the algorithm should focus the search, and specify how much detail a region should contain. This must be achieved in real time during the execution of the algorithm. Once a new singular point is added, a new swarm is created with the same characteristics as the swarm created first. Swarms will run in parallel, but they share (and can modify) the information related to the Pareto set. Particles from any swarm can be added to the Pareto set. If the user changes the fixed values for a singular point, then the corresponding swarm selects a new leader considering the new location of the singular point.

Human interaction with the algorithm in real time also enables the incorporation of human behaviour so that the human becomes a member of the swarm by proposing new candidate solutions. Eventually, such a solution can be incorporated into the Pareto front or lead the behaviour of a group of particles. User solutions will always be evaluated in the first swarm created. If a particle is being evaluated then the user request waits until the evaluation of the particle is finished. If a solution proposed by the user is being evaluated then any particle belonging to the first swarm should wait for evaluation. Once any solution is evaluated, the algorithm checks whether it could be incorporated into the Pareto front. Synchronisation is achieved among all the swarms so that they have open access for managing the Pareto front. Proposed solutions could even become leaders of the swarm(s) if they are good enough. At this point, human behaviour begins to have a proactive role during the evolution of the algorithm.

Participation by several human agents with different perspectives on a problem is close to what happens in the practice of engineering decision making, where politicians, economists, engineers, and environment specialists are all involved in final decisions. The idea of incorporating user experience into the search process is a step forward in the development of computer-aided design.

3.3 The algorithm

New particles are used that are based on the behaviour of particles in PSO. Swarms running in parallel may be distributed in different computers, and it must be ensured that the swarms can communicate among themselves; a peer-to-peer scenario could be a good choice for this task. The steps of the algorithm (for a swarm) are summarised below.

1. Set parameters and initialize the number of iterations to zero.
2. Generate random population of M particles: $\{X_i(k)\}_{i=1}^{M}$
3. Evaluate the fitness of particles and set the local best location for each particle equal to its current location.
4. Form the Pareto front and list the particles belonging to the front.
5. Build the singular point.

6. Find the closest particle to the singular point and establish it as swarm leader.
7. While not in termination condition, do the following:
    a. Execute from $i = 1$ to number of particles.
    Start
        i. Change the position of the particle:
        Determine the inertia parameter $\omega(k)$, according to (5).
        Calculate the new velocity, $V_i(k+1)$, for particle i according to (1) or (4).
        Set a new position, $X_i(k+1)$, for particle i according to (3).
        ii. Calculate the new fitness function vector for particle i in its new position.
        iii. If the new fitness function vector for particle i dominates the fitness function vector that the particle had before moving to the new position, then set the new position as the best position currently found by particle i.
        iv. If particle i is in the list of particles belonging to the Pareto front then:
        If the new fitness function vector may also be a point on the Pareto front and this new position has at least one of its neighbours located further than the minimal permissible distance from any of the objectives, then add a new particle j (a clone of i) with P_k and P_{kbest} located at the current position of i;
        else
        try to add (if possible) the particle i (at its new position) to the Pareto front; if the particle is added, remove from the list any dominated solution; dominated clones are eliminated from the swarm.
        v. If particle i is closer to the singular point than any other particle in the swarm, then set particle i as the leader of the swarm with regard to the singular point.
        vi. If particle i is not currently the leader of the swarm, but coincides in position with the leader, then re-generate particle i randomly.
        End
    b. Increase the iteration number.
8. Show the Pareto front and related results.

Some steps in this algorithm (specifically, 3 and 7-a-ii) involve particle fitness calculations. In the next section, we consider various functions characterizing the fitness of a particle –a candidate network. One of them involves the precise knowledge of all the network node pressures, which are obtained by solving the continuity and the energy equations. Various tools to analyze water networks have been developed in the past. Among them, EPANET2 [23], based on gradient-like techniques, is a software program for analysing water networks in the steady-state flow, and is used to evaluate the hydraulic performance of the solutions. Additional analyses in extended period simulation, or any other transient analysis to assess the goodness of solutions, may be performed without any change in the core of the given algorithm. The algorithm and its connection

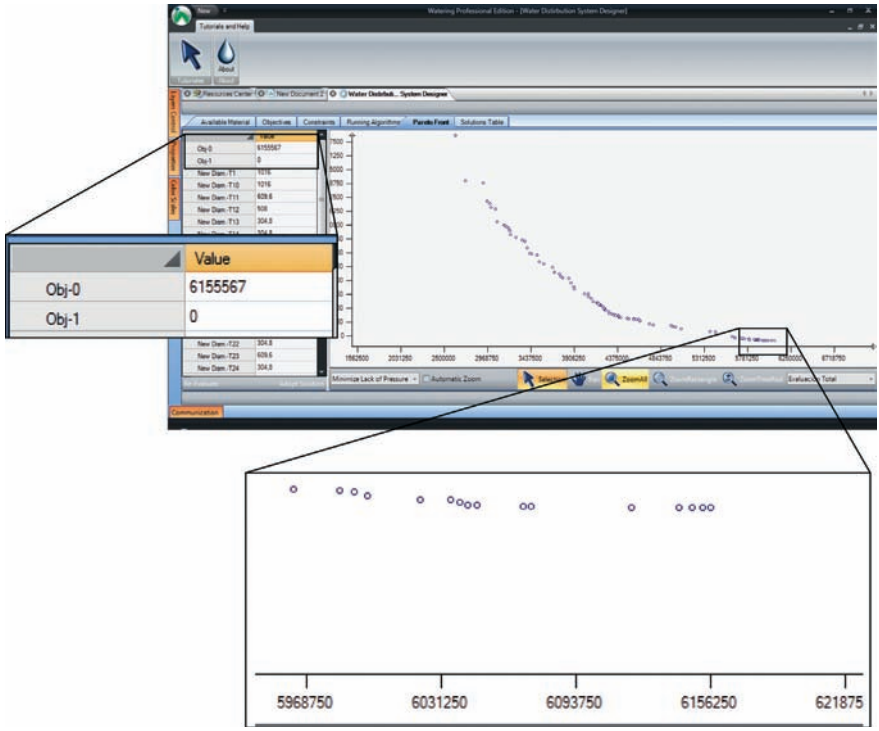


Figure 4: Pareto front for the Hanoi network as seen on the software interface.

with EPANET2 are implemented in a software program called WaterIng[1] [24], developed for water distribution system design and analysis. WaterIng is in constant evolution and may be downloaded from its website – the installation includes a file with network data as an example. A user guide is also available to learn the main concepts of how to design a water distribution system using the software.

A two-dimensional representation of the Pareto front showing the interface of the software implementing the described algorithm is shown in Fig. 4. In this particular case, Objective 0 represents an investment cost, and Objective 1 represents the lack of compliance with certain problem constraints. We have used WaterIng for the application using multi-objective optimisation in the next section.

4 Applications

PSO has shown great potential for the solution of various optimisation problems [3, 11, 13, 19, 25–28]. The authors have used the algorithmic engine of the

[1] www.ingeniousware.net

software mentioned in the previous section to solve problems in various areas [25, 27, 29, 30]. These fields have included: hydrology [31–36], hydraulics [37–46], and WR management [47–51], in addition to the references quoted in this chapter.

We devote the rest of the section to present an important application in water supply management – together with the solutions to two real-world case studies.

4.1 The design of water distribution systems

WDS are undoubtedly alive. They are born, grow, age, and deteriorate, need care (preventive care but also sometimes surgery), are expected to work properly, must meet basic requirements even under adverse circumstances, and so on. Our aim is to ensure a long and quality life for WDS. As a consequence, the design of WDS cannot be thought of as a single, material, and static design. This is one reason why WDS design optimisation is one of the most heavily researched areas in hydraulics. [52–55] and [24] are but a sample of significant references selected with intervals of around five years since 1985 to date. Of special universal relevance is the *annual WDS analysis* (WDSA) conference endorsed by the EWRI-ASCE; and in which WDS design is one of its more relevant recurrent topics.

The optimal design of a WDS aims to determine the values of all involved variables in such a way that all the demands are satisfied, even under certain failure conditions, while the investment and maintenance costs of the system are minimal (see, for instance, [56]). A general strategy for solving the optimal design problem of a WDS involves the balancing of several factors: finding the lowest costs for layout and sizing using new components; reuse or substitution of existing components; creating a working system configuration that fulfils all water demands (including water quality); adherence to the design constraints; and guaranteeing a certain degree of system reliability [57, 58].

The diameters of the new pipes are the basic variables of the problem. Nevertheless, additional variables that depend on the design characteristics of the system may be required: storage volume, pump head, the type of rehabilitation to be carried out for various parts of the network, etc. The estimation of individual costs will always depend on these variables. The correct approach to assess the costs for each element is important when defining the corresponding objective function, which has to be fully adapted to the problem under consideration: design, enlargement, rehabilitation, operation design, etc. It is also important that this objective function reflects with utmost reliability the total cost of the system during its entire lifetime. Various authors have used, in their optimisation, an objective function that only considers the costs of the pipelines (new and/or additional and duplicated pipelines) while others have taken into account other various costs involved (some examples are [59, 60]).

Satisfying the demands both in quantity and quality represents another objective. Minimum pressure values are frequently specified to guarantee a minimum level of service quality. This condition is sometimes enforced into the problem as constraints for all the consumption nodes. In contrast, in multi-objective approaches, this condition is issued as another objective.

Reliability mainly refers to the ability of the network to provide consumers with adequate and high-quality supply, under normal and *abnormal* conditions. The reliability of water systems can be classified as mechanical reliability and hydraulic reliability. The former usually refers to failures of system components, such as pipe breakage or pumps being out of service. The latter refers to uncertainty coming mainly from nodal demand and pipe roughness. There is no universal agreement about what is the best measure of reliability and what is an acceptable level of reliability (see, for example [61, 62]).

Various approaches exist for assessing the reliability of a water distribution system [63–65]. We consider here the proposal raised in [66]. It ensures that the system offers a certain level of reliability by considering costs incurred by the lack of supply satisfaction. The authors have found that the improvement obtained for various systems by considering these costs in the fitness function implies only moderate increases regarding the initial investment costs.

A more detailed representation of these three objectives follows. The first objective takes into account the pipeline costs (other costs may be easily included); the second objective considers the lack of compliance with minimal values for the pressure at each node of the network; and the third objective evaluates some reliability measure by considering incurred costs for service disruptions.

The mathematical formulation for the first objective, the investment cost of the pipes required for the design, is as follows:

$$C = \sum_{i=1}^{L} c_i l_i, \tag{7}$$

where all (L) individual pipes are summed. $D = (D_1, \ldots, D_L)^t$ is the vector of the pipe diameters. The costs per meter, depending on the diameter of pipe i, D_i, is given by c_i and its corresponding length by l_i. Note that D_i is chosen from a discrete set of available diameters and c_i is a nonlinear function of diameter.

The second objective, P, measures the lack of pressure in the nodes with respect to a prefixed minimal value. This objective is also a function of the selected pipe diameters (through the hydraulic model). For nodes with pressure larger than this minimal value, the associated individual terms vanish, and one uses the usual Heaviside step function H in the explicit expression for P:

$$P = \sum_{j=1}^{N} H(p_{\min} - p_i) \cdot (p_{\min} - p_i). \tag{8}$$

EPANET2 is used to evaluate the actual pressure at consumption nodes for a specific solution. The integration of such software to run various analyses or simulations for potential solutions of the problem is performed during the optimisation process that is developed within the evolutionary algorithms [2, 4, 5] – such as the algorithm presented in this paper.

The presence of loops in water distribution networks adds complexity to the design problem since the optimisation algorithms, due to their nature, attempt to avoid redundancies, in particular, unnecessary loops. This action does not favor

the reliability of the system. Considering explicitly some kind of reliability within the fitness function is one of the most difficult tasks faced by researchers in the area. Many researchers have asked (see [61, 62], for example) "What is the best measure of reliability and what level of reliability is acceptable?"

In this chapter, we consider the proposal in [66] that indirectly assesses reliability from an economic point of view by considering the costs of the water not delivered due to problems in the system. Precisely, the third objective is defined by

$$R = \sum_{k=1}^{L} w_k \cdot L_k \cdot d_k^{-u}, \tag{9}$$

where:

- w_k is a coefficient associated with each pipe, of the form $a \cdot t_f \cdot (c_f + c_a \cdot V_f)$.
- $a \cdot L \cdot d^{-u}$ gives the number of expected failures per year of one pipe, as a function of diameter, d, and length, L, (a and u are known constants)
- t_f is the average number of days required to repair the pipe
- c_f is the daily repair average cost
- c_a is the average cost of the water supplied to affected consumers, in monetary units per unit volume
- $V_f = 86400 \cdot Q_{\text{break}}$ is the daily volume of water that should be supplied to the affected consumer (86400 = number of second per hour) due to the loss of water of Q_{break} cubic meters per second.

All the constant values entering this equation have been taken from [66]. Consideration of other values corresponding to other specific cases is straightforward.

The optimisation problem may be addressed as a single objective problem or as a multi-objective problem. If a single objective approach is implemented, any of the defined objectives can be considered and as many other objectives introduced as desired by way of 'suitable' penalty costs. For example, the following fitness function

$$F(D, V_f) = C + \alpha P + \beta R, \tag{10}$$

considers the minimisation of the piping costs, C, while considering a penalty term αP that considers the lack of compliance with some required minimum pressure level, and another penalty cost, βR, regarding the lack of reliability as the cost incurred for not satisfying the supply due to problems in the system (α and β are suitable penalty factors).

4.2 A single objective problem

The following problem has been considered in [67]. It analyses a sector of the WDS of a Latin American capital (see the layout in Fig. 5) using the PSO variant presented above.

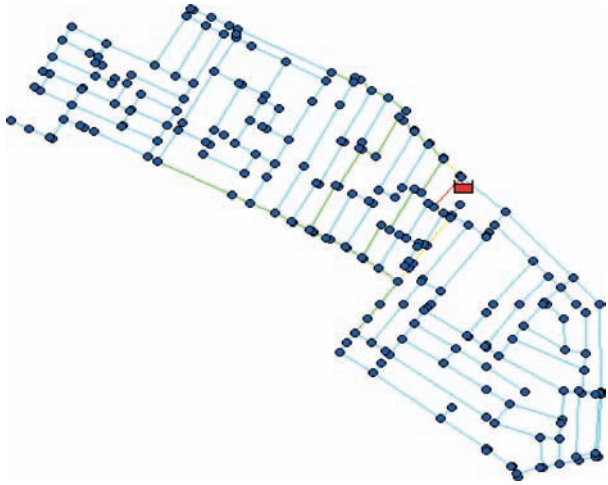


Figure 5: Sector of a WDS in a Latin American capital and the solution when considering reliability.

Table 1: Comparison between costs for both solutions.

Diameter [mm]	Without reliability		With reliability	
	Length [m]	Cost [soles]	Length [m]	Cost [soles]
100	17731.10	2077021.41	15822.31	1853425.63
150	606.39	88023.28	2077.69	301597.04
200	0.00	0.00	328.79	62937.56
250	0.00	0.00	108.70	26206.24
300	0.00	0.00	0.00	0.00
Total cost (soles)	2165044.69		2244166.47	

This sector is fed by a tank, and has 294 lines amounting to 18.337 km of pipe and 240 nodes consuming 81.53l/s in total.

Figure 5 also presents the solution obtained by using eqn. 10, which includes the minimum pressure requirement and reliability as penalty terms. The first column of Table 1 specifies the various diameters. This solution is only a mere 3.65% more expensive than the solution obtained by using the fitness function $G = C + \alpha P$ (no reliability consideration). Table 1 presents a comparison between the initial investment costs for both solutions.

Figures 6 and 7 present the solution when not considering reliability. These figures are used to show the performance differences between both solutions following pipe breaks.

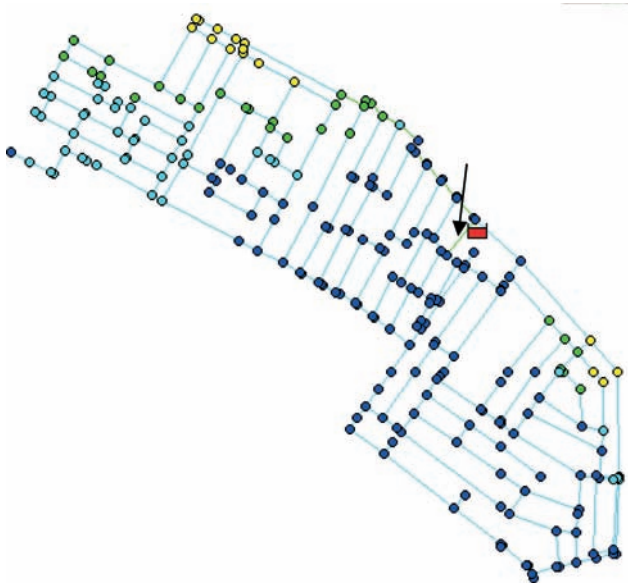


Figure 6: Lower performance after ignoring reliability due to closure of marked pipe.

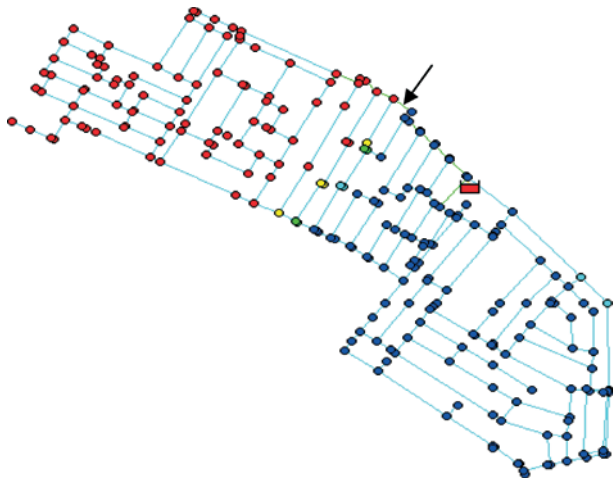


Figure 7: Higher impact on performance after ignoring reliability due to closure of marked pipe.

The effects of closing the pipe indicated by the arrow can be observed in Figure 6. Even though no pressures less than 10 m can be observed, almost half of the nodes (red points) do not meet the required minimum pressure of 15 m (nodes in dark blue). Figure 7 shows the higher impact produced by another closed pipe. Now slightly clear points (on the upper left) are consumption nodes with a pressure lower than 10 m. Again, this will not happen for the more reliable design (Figure 5) obtained from *F*, no matter which pipe is out of service.

The formulation we consider here aims at minimising the cost of a new network with the diameters of the pipes as design variables, while satisfying a minimum pressure in all the nodes and, at the same time, providing a certain amount of enforced reliability by guaranteeing the service under determined failure scenarios. The scenarios considered here follow the approach of 'breaking' by turn all the pipes of a specific design to check if all the constraints are fulfilled by the design when subjected to these circumstances. If the test is negative the design is suitably penalised. In this way, designs will develop increasing reliability. To undergo those tests, the system must be analysed for any of those specific 'breakages'. Only solutions assessed as being feasible by EPANET2 are considered.

4.3 A multi-objective analysis

Multi-objective approaches are clearly preferred for a number of reasons – even though they are more expensive from a computational point of view. It is clear that, sooner or later, a decision must be made to balance the various objectives. Using penalty factors is a decision, somehow arbitrary, taken *a priori*. On the contrary, a multi-objective approach provides a set of solutions that will help *a posteriori* decision making and provide decision makers with a richer range of solutions and alternatives.

We present another case-study corresponding to a different sector of the WDS of the same Latin American capital (Fig. 8). The design involves the three objectives described above in a multi-objective solution: minimising the investment

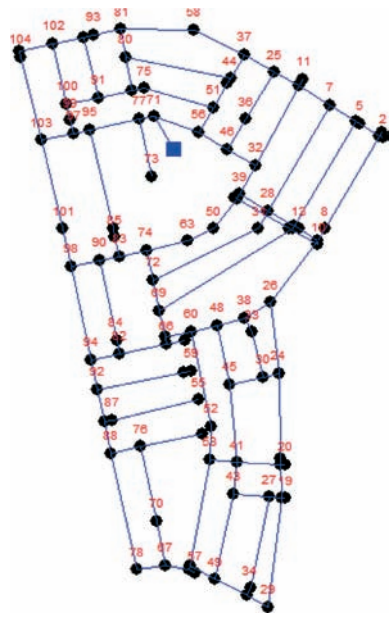


Figure 8: A real-world network.

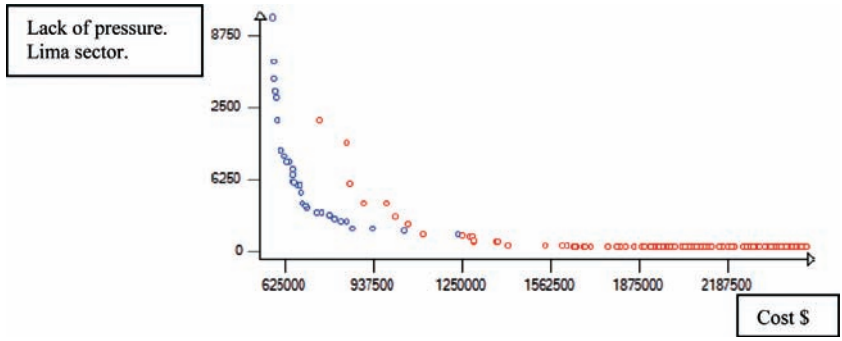


Figure 9: Approximated Pareto front for a real world case.

cost; minimising the lack of pressure at demand nodes; and minimising additional costs because of reliability problems.

This network is fed by a tank and made of 132 lines and 104 consumption nodes, its total length being 9.055 km, and the total consumed flowrate amounting to 47.09l /s.

In Figure 9 a bi-dimensional (cost against lack of pressure) representation of the approximated Pareto front that considers the three mentioned objectives is presented. Two swarms (one covering the left zone of the Pareto front and the other the right more horizontal area) can be identified that build the three-dimensional Pareto front of this problem with three objectives. Observe that the representation is a projection in two dimensions of points in the real Pareto front, which is a surface in a three-dimensional space.

Plenty of enriched information that helps the decision-making process is provided by this type of representation of the Pareto front. For example, it becomes evident, as expected, that after some point, the rate at which the minimum pressure can be increased in the network is much lower than the rate at which initial investment costs must be increased to achieve the desired pressure level.

The development of a multi-objective optimisation process enables the combination of economic, engineering, and policy viewpoints when searching for a solution to a problem. For example, the relationship between the initial investment cost and the minimum pressure in the network may help decide, among other factors, which pressure would be better to use for the final solution. In this case (in which there is a limited budget to implement the design), the decision maker has at his or her disposal a clear guideline to assess how much the quality may be improved if the budget is increased by a certain amount. This is an added value of the multi-objective approach when solving the problem of optimal design of WDS.

5 Conclusions

Classical methods of optimisation are poorly suited for many real world problems since they are unable to process inaccurate, noisy, discrete, and complex data. In this chapter, we have presented the principles of PSO, an evolutionary algorithm that has shown great efficiency for the solution of various optimisation problems.

We have also described three modifications that considerably improve the performance of the standard algorithm for finding solutions to various optimisation problems. Firstly, a proposal for enabling continuous and discrete variables to coexist in one PSO formulation. Secondly, a mechanism for enriching diversity and so improving the performance of PSO. And thirdly, a self-adapting feature that avoids the cumbersome task of parameter selection and fine tuning. The elements to adapt this algorithm to multi-objective optimisation problems have then been provided. Finally, after mentioning various applications in the water field, we have shown the results of specific applications to selected case-studies regarding the design of WDS – using a very well-known urban water problem.

Further improvements should be considered for the multi-objective algorithm presented in this paper. For example, the inclusion of problem-dependent rules – thus taking advantage of expert knowledge – would further facilitate the process of finding solutions by enhancing human–computer interaction while introducing more reality. Also, the consideration of a wider environment where swarms (or even other algorithms – evolutionary or otherwise) with various specialisation tasks could coexist and cooperate in the optimisation process. In addition, all the algorithms used must be further developed to take advantage of emerging technologies in the field of parallel and distributed computing. Some of these research lines have been addressed recently in [67], which is the source for most of the contents of this chapter.

Acknowledgements

This work has been developed with the support of the project IDAWAS, DPI2009-11591, of the Dirección General de Investigación of the Ministerio de Educación y Ciencia, and ACOMP/2010/146 of the Conselleria d'Educació of the Generalitat Valenciana. The first author is also indebted to the Universitat Politècnica de València for the sabbatical leave granted during the first semester of 2011. The use of English in this paper was revised by John Rawlins.

References

[1] Kennedy, J. & Eberhart, R.C., Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks, Piscataway*, NJ, pp. 1942–1948, 1995.

[2] Montalvo, I., Izquierdo, J., Pérez, R. & Tung, M.M., Particle swarm optimization applied to the design of water supply systems. *Computer & Mathematics with Applications*, **56(3)**, pp. 769–776, 2007.

[3] Izquierdo, J., Montalvo, I., Pérez, R. & Fuertes, V.S., Design optimization of wastewater collection networks by PSO. *Computer & Mathematics with Applications*, **56(3)**, pp. 777–784, 2007.

[4] Montalvo, I., Izquierdo, J., Pérez, R. & Iglesias, P.L., A diversity-enriched variant of discrete PSO applied to the design of water distribution networks. *Engineering Optimization*, **40(7)**, pp. 655–668, 2008.

[5] Montalvo, I., Izquierdo, J., Pérez, R. & Herrera, M., Improved performance of PSO with self-adaptive parameters for computing the optimal design of water supply systems. *Engineering Applications of Artificial Intelligence*, **23(5)**, pp. 727–735, 2010.
[6] Izquierdo, J., Montalvo, I., Pérez, R. & Tavera, M., Optimization in water systems: A PSO approach. *Proceedings of the Business and Industry Symposium, BIS*, Ottawa, Canada, 2008.
[7] Millonas, M.M., Swarms, phase transition, and collective intelligence. *Artificial Life III*, ed. C.G. Langton, Addison Wesley: Massachusetts, pp. 417–445, 1994.
[8] Shi, Y.F. & Eberhart, R.C., A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998.
[9] Al-Kazemi, B. & Mohan, C.K., Multi-phase discrete particle swarm optimization. *JCIS*, pp. 622–625, 2002.
[10] Rastegar, R., Meybodi, M.R. & Badie, K., A new discrete binary particle swarm optimization based on learning automata. *Proceedings of the 2004 International Conference on Machine Learning and Applications*, Louisville, KY, USA, 2004.
[11] Liao, C.J., Tseng, C.T. & Luarn, P., A discrete version of particle swarm optimization for flowshop scheduling problems. *Comput Oper Res*, **34(10)**, pp. 3099–3111, 2007.
[12] Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C. & Wang, Q.X., Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, **103(5)**, pp. 169–176, 2007.
[13] Dong, Y., Tang, B.X.J. & Wang, D., An application of swarm optimization to nonlinear programming. *Computers & Mathematics with Applications*, **49(11–12)**, pp. 1655–1668, 2005.
[14] Angeline, P.J., Using selection to improve particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 84–89, Anchorage, AL, USA, 1998.
[15] Løvbjerg, M., Rasmussen, T.K. & Krink, T., Hybrid particle swarm optimiser with breeding and subpopulations. *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, CA, USA, 2001.
[16] Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D. & Vrahatis, M.N., Stretching technique for obtaining global minimizers through particle swarm optimization. *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN, USA, 2001.
[17] Herrera, M., Izquierdo, J., Montalvo, I. & Pérez-García, R., Injecting diversity into particle swarm optimization. Application to water distribution system design. *Advances in Computer Science and Engineering*, **6(2)**, pp. 159–179, 2011.
[18] Shi, Y. & Eberhart, R.C., Empirical study of particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation*. Washington, DC, USA, 1999.
[19] Jin, Y.X., Cheng, H.Z., Yan, J.I. & Zhang, L., New discrete method for particle swarm optimization and its application in transmission network expansion planning. *Electric Power Systems Research*, **77(3–4)**, pp. 227–233, 2007.

[20] Arumugam, M.S. & Rao, M.V.C., On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. *Appl Soft Comput*, **8(1)**, pp. 324–336, 2008.
[21] V. Pareto, *Cours d'economie politique*, Université de Lausanne, Lausanne, Switzerland, 1896.
[22] Montalvo, I., Izquierdo, J., Schwarze, S. & Pérez-García, R., Multiobjective Particle Swarm Optimization applied to water distribution systems design: An approach with human interaction. *Mathematical and Computer Modelling*, **52**, pp. 1219–1227, 2010.
[23] Rossman, L.A., *EPANET2, User's Manual. Cincinnati* (IN), USA, Environmental Protection Agency, 2000.
[24] Montalvo, I., Martínez-Rodríguez, J.B., Izquierdo, J. & Pérez-García, R., Water distribution system design using agent swarm optimization. *Water Distribution System Analysis 2010 – WDSA2010*. Tucson, AZ, USA, 2010.
[25] Herrera, M., Izquierdo, J., Montalvo, I., García-Armengol, J. & Roig, J.V., Identification of surgical practice patterns using evolutionary cluster analysis. *Mathematical and Computer Modelling*, **50**, pp. 705–712, 2009.
[26] Janson, S., Merkle, D. & Middendorf, M., Molecular docking with multiobjective particle swarm optimization. *Applied Soft Computing*, **8**, pp. 666–675, 2008.
[27] Izquierdo, J., Minciardi, R., Montalvo, I., Robba, M. & Tavera, M., Particle Swarm Optimization for the biomass supply chain strategic planning. *Proceedings of 4th Biennal Meeting, iEMSs 2008: International Congress on Environmental Modelling and Software*, pp. 1272–1280, Barcelona, Spain, 2008.
[28] Pan, Q.K., Tasgetiren, F. & Liang, Y.C., A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers and Operations Research*, **35(9)**, pp. 2807–2839, 2008.
[29] Díaz, J.L., Herrera, M., Izquierdo, J., Montalvo, I. & Pérez-García, R., A Particle Swarm Optimization derivative applied to cluster analysis. *Proceedings of the iEMSs 2008 (International Congress on Environmental Modelling and Software – 4th Biennial Meeting)*, Barcelona, 2008.
[30] Izquierdo, J., Montalvo, I., Pérez, R. & Fuertes V.S., Forecasting pedestrian evacuation times by using swarm intelligence. *Physica A: Statistical Mechanics and its Applications*, **388(7)**, pp. 1213–1220, 2009.
[31] Gill, M.K., Kaheil, Y.H., Khalil, A., McKee, M. & Bastidas L., Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research*, **42**, W07417, p. 14, 2006.
[32] Chau, K.W., Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *Journal of Hydrology*, **329(3–4)**, pp. 363–367, 2006.
[33] Jiang, Y., Hu, T., Huang C.C. & Wu, X., An improved particle swarm optimization algorithm. *Applied Mathematics and Computation*, **193(1)**, pp. 231–239, 2007.

[34] Zhang, X., Srinivasan, R., Zhao, K. & van Liew M., Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrological Processes*, **23(3)**, pp. 430–441, 2009.

[35] Kuok, K.K., Harun, S. & Shamsuddin, S.M., Particle swarm optimization feedforward neural network for hourly rainfall runoff modeling in bedup basin. *Malaysia. International Journal of Civil & Environmental Engineering IJCEE*, **9(10)**, 2010.

[36] Lü, H., Yu, Z., Horton, R., Zhu, Y., Wang, Z., Hao, Z. & Xiang L., Multiscale assimilation of root zone soil water predictions. Article first published online: 15 March 2011. DOI: 10.1002/hyp.8034.

[37] Jung, B.S. & Karney, B., Fluid transients and pipeline optimization using GA and PSO: the diameter connection. *Urban Water Journal*, **1(2)**, pp. 167–176, 2004.

[38] Muleta, M.K., Boulos, P.F., Orr, C.H. & Ro, J.J., Using Genetic Algorithms and Particle Swarm Optimization for Optimal Design and Calibration of Large and Complex Urban Stormwater Management Models. *ASCE Conference Proceeding* **200**, 113, DOI:10.1061/40856(200)113, 2006.

[39] Kumar, D.N. & Reddy M.J., Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resources Planning and Management, ASCE*, **133(3)**, pp. 192–201, 2007.

[40] Reddy, M.J. & Kumar, D.N., Optimal reservoir operation for irrigation of multiple crops using elitist-mutated particle swarm optimization. *Hydrological Sciences Journal*, **52(4)**, pp. 686–701, 2007.

[41] Reddy, M.J. & Kumar, D.N., Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. *Hydrological Processes*, **21(21)**, 2007.

[42] Afshar, M.H., Rebirthing particle swarm optimization algorithm: Application to storm water network design. *Canadian Journal of Civil Engineering*, **35(10)**, 2008.

[43] Scheerlinck, K., Pauwels, V.R.N., Vernieuwe, H. & de Baets, B., Calibration of a water and energy balance model: Recursive parameterization versus particle swarm optimization. *Water Resources Research*, **45(10)**, doi:10.1029/2009WR008051, 2009.

[44] Cheng, C., Liao, S., Wu, X. & Chau, K.W., Two stage particle swarm optimisation for long-term operation of a hydroelectric power system. *Transactions Hong Kong Institution of Engineers*, **16(3)**, pp. 8–14, 2009.

[45] Geem, Z.W., Particle-swarm harmony search for water network design. *Engineering Optimization*, **41(4)**, pp. 297–311, 2009.

[46] Zhang, J., Wu, Z., Cheng, C.-T., & Zhang S.-Q., Improved particle swarm optimization algorithm for multi-reservoir system operation, *Water Science and Engineering*, **4(1)**, pp. 61–73, 2011.

[47] Baltar, A.M. & Fontane, D.G., Use of multiobjective particle swarm optimization in water resources management. *Journal of Water Resources Planning and Management*, **134(3)**, pp. 257–268, 2008.

[48] Reddy, M.J. & Kumar, D.N., Performance evaluation of elitist-mutated multi-objective particle swarm optimization for integrated water resources management. *Journal of Hydroinformatics*, **11(1)**, pp. 78–88, 2009.
[49] Yu, L. & Wang, J.-q., Analysis of water resources planning based on particle swarm optimization algorithm. *2nd International Workshop on Intelligent Systems and Applications (ISA)*, Wuhan, China, pp. 1–3, 2010.
[50] Chau, K., River stage forecasting with particle swarm optimization. *Lecture Notes in Artificial Intelligence*, **3029**, pp. 1166–1173, 2004.
[51] Chang, W., A novel particle swarm optimization for optimal scheduling of hydrothermal system. *Energy and Power Engineering*, **2(4)**, 2010.
[52] Walski, T.M., State of the art: Pipe network optimization. *Computer Applications in Water Resources, ASCE*, 1985.
[53] Goulter, I. C., Systems analysis in water-distribution network design: From theory to practice. *Journal of Water Resources Planning and Management, ASCE*, **118(3)**, pp. 238–248, 1992.
[54] Lansey, K.E., Optimal design of water distribution systems. *Water Distribution System Handbook*, ed. L.W. Mays, McGraw-Hill: New York, 2000.
[55] Wu, Z.Y. & Walski, T., Self-adaptive penalty cost for optimal design of water distribution systems. *Journal of Water Resources Planning and Management*, **131(3)**, pp. 181–192, 2005.
[56] Izquierdo, J., Pérez, R. & Iglesias, P.L., Mathematical models and methods in the water industry. *Mathematical and Computer Modelling*, **39(11–12)**, pp. 1353–1374, 2004).
[57] Goulter, I.C. & Coals, A.V., Quantitative approaches to reliability assessment in pipe networks. *Journal of Transportation Engineering*, **112(3)**, pp. 287–301, 1986.
[58] Goulter, I.C. & Bouchart, F., Reliability-constrained pipe network model. *Journal of Hydraulic Engineering*, **116(2)**, pp. 211–229, 1990.
[59] Matías, A.S., *Diseño de redes de distribución de agua contemplando la fiabilidad, mediante algoritmos genéticos. PhD thesis*, Universidad Politécnica de Valencia, Valencia, Spain, 2003.
[60] Dandy, G.C. & Engelhardt, M.O., Multi-objective trade-offs between cost and reliability in the replacement of water mains. *Journal of Water Resources Planning and Management*, **132(2)**, pp. 79–88, 2006.
[61] Savic, D.A., (2005) Coping with risk and uncertainty in urban water infrastructure rehabilitation planning. Acqua e città – i convegno nazionale di idraulica urbana, S'Agnello (NA), 28–30.
[62] Martínez-Rodríguez, J.B., Montalvo, I., Izquierdo, J. & Pérez-García, R., Reliability and tolerance comparison in water supply networks. *Water Resources Management*, **25**, pp. 1437–1448, 2011.
[63] Mays, L.W., Review of reliability analysis of water distribution systems. *Stochastic Hydraulic*, Rotterdam, The Netherlands, 1996.
[64] Ostfeld, A. & Shamir, U., Design of optimal reliable multiquality water-supply systems. *Journal of Water Resources Planning and Management*, **122(5)**, pp. 322–333, 1996.

[65] Tanyimboh, T.T. & Setiadi, Y., Joint layout, pipe size and hydraulic reliability optimization of water distribution systems. *Engineering Optimization*, **40(8)**, pp. 729–747, 2008.
[66] Martínez, J.B., Quantifying the economy of water supply looped networks. *Journal of Hydraulic Engineering – ASCE*, **133(1)**, pp. 88–97, 2007.
[67] Montalvo, I., *Diseño óptimo de sistemas de distribución de agua mediante Agent Swarm Optimization*. Doctoral dissertation, Universitat Politècnica de València, Valencia, Spain, 2011.

This page intentionally left blank

CHAPTER 6

Tabu search for managing groundwater

R.C. Peralta[1] & I.M. Kalwij[2]
[1]*Department of Civil and Environmental Engineering, Utah State University, United States of America.*
[2]*Kalwij Water Dynamics Inc, British Columbia, Canada.*

Abstract

The family of tabu search (TS) heuristic global search optimizer types can solve problem types ranging from linear programming (LP) through mixed integer nonlinear programming (MINLP). TS is most commonly used for problems that are not readily solved by classical linear programming and gradient search nonlinear programming algorithms (e.g., problems having nondifferentiable objective functions).

TS uses its memory of tested solutions to guide development of new solutions. For example, a TS algorithm maintains a tabu list of inferior solutions, and will avoid expending effort on solutions similar to the poor ones. User input designates the minimum acceptable difference between a new strategy and a previous poor strategy. On the other hand, the TS will try to develop new solutions that are in the neighborhood of known good solutions. User input determines acceptable and desirable proximity. Other rules for discarding and favoring candidate solutions help make TS very adaptable to many situations.

As it moves discrete distances thru the solution space, TS can avoid being trapped in local optima, making it a global optimizer. TS is especially useful for discrete problems (having a finite number of solutions). Especially when it directs other optimization algorithms, TS is referred to as a metaheuristic optimizer.

Because of the ease with which TS can couple with other optimizers, and incorporate problem-specific rules, TS applications in hydrologic science are increasing. This chapter discusses TS concepts, features, and coupling with other heuristic optimizers (HO) such as genetic algorithm. The resulting hybrid HO effectively and efficiently solves complex nonlinear groundwater optimization problems.

Keywords: GA–TS hybrid, groundwater management, heuristic optimization, pump and treat, remediation, Tabu search.

1 Introduction

Tabu search (TS) is a heuristic and thus is a nonclassical optimizer that can solve linear thru nonlinear-mixed integer optimization problems. It is considered a global optimizer because it does not require a continuously differentiable objective function, and it can avoid being trapped in local optima. A TS maintains a record of solutions that it develops, and the results of its evaluations. It uses rules to avoid spending unnecessary effort in evaluating inferior solutions. Other rules make it more likely to develop superior solutions.

By the late 1970s, TS was applied for solving nonlinear combinatorial optimization problems. In 1986 Fred Glover, developer of this approach, published the concept [1]. Via publications in 1989 [2] and 1990 [3], he increased research community awareness of TS. Other valuable publications followed [4]. Glover and Laguna [5] summarize many TS applications for different industries. These include TS use for optimization problem types ranging from integer programming (IP) through mixed integer nonlinear programming (MINLP) problems. Applications cover most arenas that require optimal planning, resource allocation and design.

This chapter emphasizes TS application for managing groundwater resources. TS applications in hydrologic science and engineering are increasing. One of the earliest that we are aware of is by Zheng and Wang [6]. They apply TS with simulated annealing (SA) to develop optimal groundwater model parameters for one-dimensional hypothetical problems. SA is a heuristic optimizer, developed by Kirkpatrick *et al*. in 1983 [7].

In later work, Zheng and Wang [8] integrated TS with LP to develop an optimal pumping strategy for hypothetical groundwater remediation problems. They used TS to select well locations, and used linear programming (LP) to determine optimal pumping rates. Peralta *et al*. [9–11] applied TS concepts within their groundwater simulation–optimization model to optimize groundwater pump and treat system designs and strategies.

Peralta *et al*. [12] applied hybrid optimization techniques to develop 30-year transient pump and treat designs and strategies for Blaine Naval Ammunition Depot in Nebraska. The goal was to minimize total cost of containment and cleanup of dissolved phase plumes of explosives and several hydrocarbons. Employed hybrids included a TS coupled with a genetic algorithm (GA–TS) and with simulated annealing (SA–TS). GA is a heuristic optimizer, developed by John Holland and his colleagues in 1975 [13, 14]. In these GA–TS and SA–TS hybrids, TS guides the GA or SA to more efficiently search the solution space. Kalwij and Peralta [15] further enhanced the GA–TS concept by incorporating an adaptive component to the algorithm. Other GA–TS transport optimization applications include [16–18].

TS applications in water distribution network optimization are Cunha and Ribeiro [19] and Lippai *et al*. [20]. Examples of other applications of TS in hydrologic science are Yeh *et al*. [21] (groundwater contamination source identification), Tung and Chou [22] (groundwater parameter zonation), and Martinez *et al*. [23] (water level monitoring network design).

TS can be implemented by itself, however, it is more commonly used to guide other algorithms (either of nonclassical or classical origin). Glover [3] originally describes TS as '*as an approach for guiding other heuristics to overcome the limitations of local optimality, both in a deterministic and a probabilistic framework.*' There are many different ways to implement TS, and design is often specific to the optimization problem type and application. This chapter discusses TS in the context of solving groundwater contaminant transport-related MINLP problems.

This chapter briefly describes TS fundamentals, and TS coupling with a GA (producing a GA–TS hybrid). Section 3 applies this hybrid to solving transport optimization problem and contrasts computational efficiency of a GA–TS and its adaptive counterpart with a straightforward GA. The summary section mentions the outlook for further TS development in hydrological science.

2 Tabu search concepts

The intent of this section is to provide a brief overview of the governing elements of TS. For more in-depth theory, the reader is referred to the well-recognized manuscripts by Glover [2, 3, 24], and many other publications and manuscripts [4]. Glover and Laguna [24] is the first published comprehensive monograph on the subject matter.

TS employs adaptive memory and responsive exploration [2, 3, 24, 25], leading to a more intelligent search of a solution space. It mimics memory to avoid undesirable regions of the solution space and focus on acceptable regions. "We learn from our mistakes" is an adage that characterizes an important TS feature (TS assumes that if a developed strategy has an inferior objective function value (OFV), then any strategy that is near this inferior strategy will also produce an inferior OFV). Therefore, the optimizer will stay away from this forbidden ('taboo' or 'tabu') region of the solution space and will search more in that part of the solution space that will potentially yield superior result (hopefully it will yield the globally optimal solution). In effect, choosing a poor strategy can yield more information than choosing a good strategy. Figure 1 shows a flow chart of a simplified stand-alone TS. Here TS is the only optimizer and newly developed strategies are evaluated with respect to the tabu list. If a strategy satisfies the tabu conditions, meaning that it is not a desirable strategy and is rejected, one proceeds with evaluating the second best strategy, and so on. Enforcing tabu conditions guides optimization toward optimality.

TS creates a balance between search intensification and diversification [24, 25]. Adaptive memory implies that the algorithm retains knowledge about strategies previously developed during optimization and their visited solution space regions.

This means that a newly developed strategy is evaluated in relation to previously developed strategies. TS distinguishes between short-term and long-term memory, and controls the size of the subset of strategies that a newly developed strategy is evaluated against. In essence, new strategies are preferably developed within the good neighborhood area. The more intense search is in the region having the best strategies. TS maintains at least a specified distance (or search coarseness) between

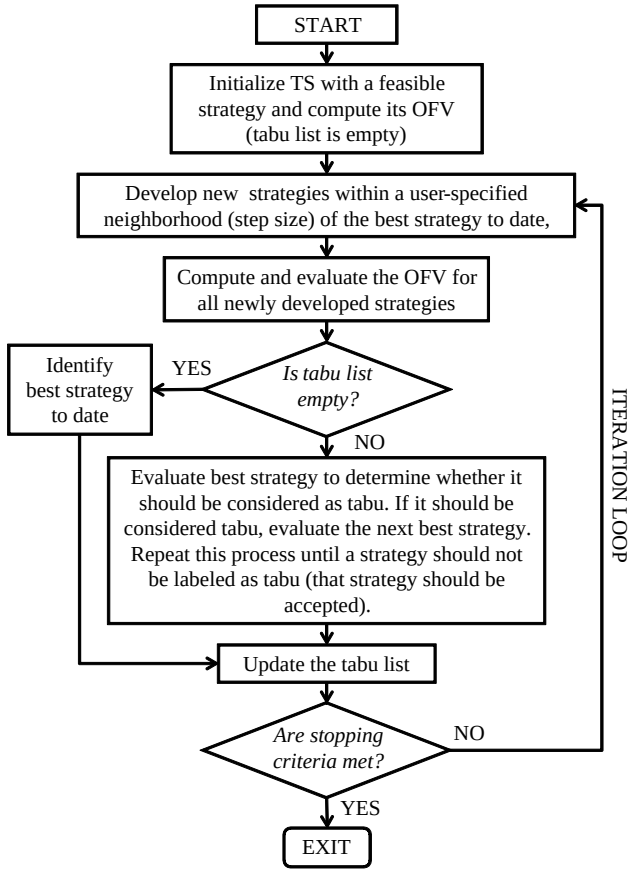


Figure 1: Example flow chart for TS basic concept.

developed strategies. This is especially useful for continuous optimization problems that essentially have an infinite number of solutions.

Responsive exploration encourages the TS optimizer to explore promising regions in addition to the already-established good neighborhood [24]. This diversification helps the optimizer avoid entrapment within local optima. Responsive exploration includes maintaining a tabu list (a list that records inferior strategies, henceforward termed inferior neighborhoods). A tabu list can also be designed to record previous moves rather than strategies. Either way, a newly developed strategy is evaluated with respect to this list, to ensure that the optimizer does not revisit or get too close to inferior neighborhoods.

3 Hybrid optimization types

Hybrids are combinations of two or more optimization approaches, and we consider them to be a subgroup of nonclassical optimization types. Here, we discuss

how particular TS concepts and elements can guide a GA in optimization (GA–TS). The same TS principles and algorithm structure can be applied to a hybrid SA–TS. However, some TS features can be specific to the optimizer being guided. For example, the choice of intensifying TS search around elite strategies is GA-specific. The discussed hybrid optimization method handles nonlinear continuous domain, time-varying solutions for multiple stress periods. The GA–TS simultaneously optimizes for multiple stress periods (as opposed to sequentially optimizing one stress period after another). Simultaneous optimization of multiple stress periods increases optimization complexity [17]. This hybrid optimization method was designed to specifically solve computationally intensive groundwater optimization problems, but is adaptable to other fields as well.

Figure 2 presents a GA–TS flow chart. It shows a GA outer loop (generation iterations) and an inner loop (iterations within a generation). The GA proceeds

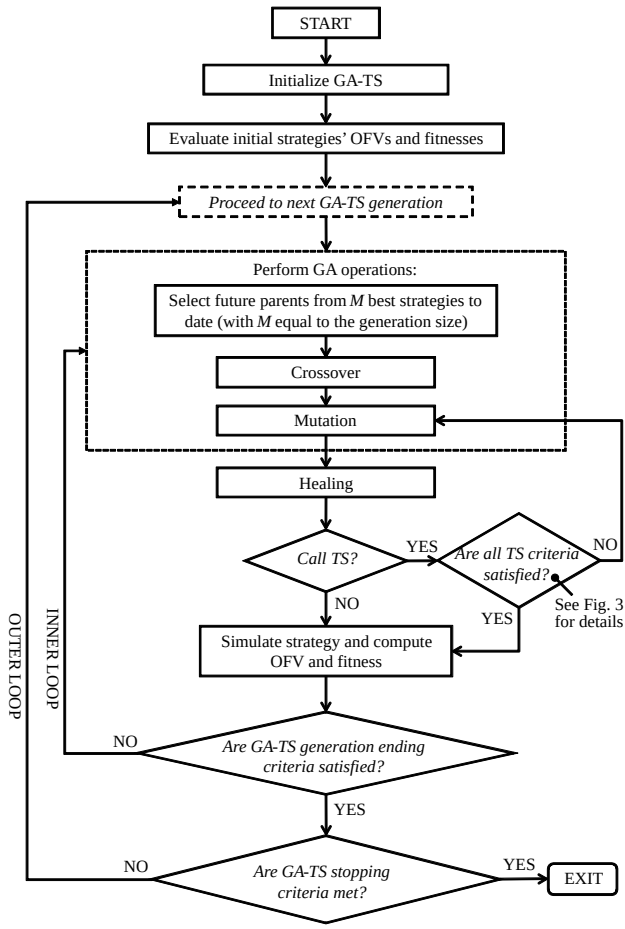


Figure 2: GA–TS flow chart (Modified from Peralta *et al*. [17]).

to the next generation after developing a number of new strategies (the number equals the generation size or population size). The GA terminates after the total number of generations has been completed (outer loop). Alternatively, a GA can be directed to terminate when the optimizer has converged to an assumedly globally optimal solution. However, there is often uncertainty concerning whether a globally optimal solution has been obtained for a large-scale highly nonlinear problem, and one often settles for a nearly globally optimal strategy.

One can direct the GA to terminate if OFV improvements do not satisfy a minimum acceptable threshold value after a prespecified number of generations. In that case, the optimizer has possibly become trapped in a local optimum (for a nonconvex problem).

One initializes a GA–TS using preexisting strategies or semirandomly or randomly generated strategies. These strategies are evaluated with respect to OFV and fitness. The inner loop includes standard GA operations such as parent selection, crossover, and mutation. This algorithm also includes advanced features such as elitism and healing. Elitism controls how many of the best strategies to date are used in the parent selection procedure. Healing ensures that a newly developed strategy satisfies decision-variable-based constraints [15].

After healing, a strategy is evaluated by TS. If all applicable TS criteria are satisfied, the strategy (set of stimuli) is simulated to compute system responses (state variable values). Simulation models (S-models) are used to compute system responses. (A model that couples simulation modeling and optimization is commonly referred to as a simulation–optimization or S–O model.) The hybrid GA–TS method allows simulation only of solutions that satisfy TS criteria. This prevents waste of much computation time (S-models can require significant computer time).

Figure 3 details steps performed when invoking TS. TS is implemented in such a way that it guides the optimization method to [15]:

1. Avoid reproducing strategies similar to any already developed strategies;
2. Intensify search in the solution space region (neighborhood) that potentially yields superior strategies;
3. Avoid reproducing inferior strategies by avoiding inferior neighborhoods; and
4. (If applicable) Avoid developing strategies having a nonpenalized OFV inferior to the OFV of the best strategy to date.

Although Fig. 3 mentions one criterion in each of Steps 1–3, within any of those steps, multiple criteria of the same type can be used. For example, a different source search coarseness can be used to evaluate how close new strategies are to elite strategies, versus how close they are to tabu strategies.

The act of invoking TS can be probability based, allowing one to either always use the TS criteria, or allowing its use to be stochastic. Determination of invocation occurs at the beginning of each GA generation. When the TS is not invoked, the GA–TS has more freedom to explore the solution space. This might result in more diverse strategies.

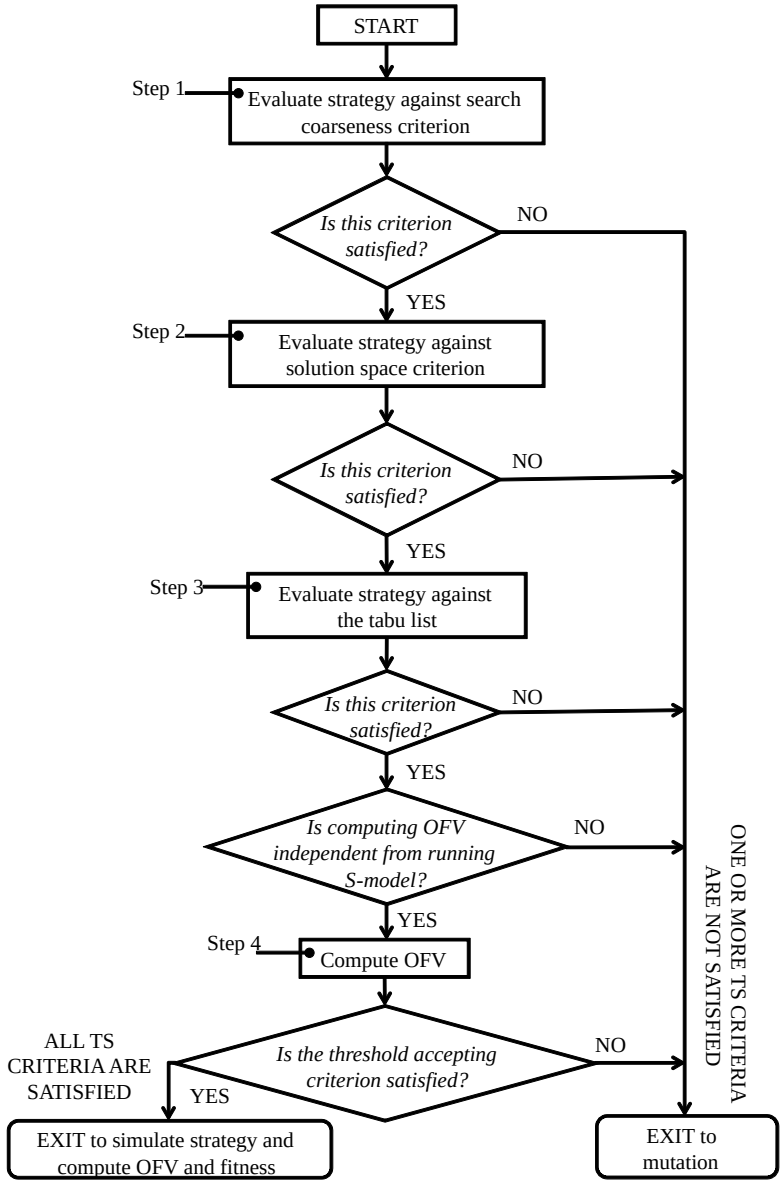


Figure 3: Representative flow chart of a TS evaluation algorithm.

Step 1 in Fig. 3 evaluates with respect to the search coarseness criterion. Satisfying this criterion ensures that a minimum acceptable distance is maintained between a newly developed strategy and previously generated elite or general strategies. This avoids reproducing duplicate or similar strategies already developed.

Step 2 in Fig. 3 evaluates with respect to the solution space step size criterion. Satisfying this ensures that the maximum allowable change of each decision variable of a newly developed strategy with respect to its parents is within the desired threshold. This criterion forces the GA–TS to search in the good neighborhood (of elite strategies).

Step 3 in Fig. 3 ensures that the newly developed strategy is not on the tabu list or in the neighborhood of a tabu strategy. If this strategy is on the tabu list or close to a strategy on the tabu list, the TS algorithm does not accept that strategy.

Some TS steps are only beneficial sometimes. For example, one can force the algorithm to only accept strategies having an OFV that satisfies a specified threshold accepting value (Step 4 in Fig. 3). This criterion can only be evaluated *a priori* if the OFV does not require running simulation models (e.g, it is solely decision–variable based) [15].

Table 1 summarizes suggested ranges for dimensionless TS control parameter values. These values were developed by evaluating a range of control parameters for a hypothetical transport optimization problem [15, 26]. This involved trial-and-error testing of 50 control parameter sets (TS probability, search coarseness, solution space size, tabu list size, and threshold accepting value), each repeated 10 times to account for the randomness of heuristic optimization. For clarity, for each parameter in the set, the second column in the table refers to the corresponding TS step(s) illustrated in Fig. 3.

The third column of Table 1 shows the combinations of parameter value ranges that achieve convergence to the correct subset of wells for 60% to 90% of the 10 optimization runs. Also, combinations within these parameter value ranges, cause 60% to 80% of the runs to converge within 1% of the best optimal solution [15].

The authors noticed that different values work better for different optimization stages. Some parameter values that are ideal for the beginning part of an optimization run should be modified when more closely approaching optimality [15]. An auto-adaptive feedback-driven GA–TS approach developed to test this observation dynamically changes TS parameters during run-time [15]. This includes parameters such as TS probability, search coarseness, solution space size, and tabu list size. When and how much these control parameters are modified during

Table 1: Empirically developed and suggested TS control parameters values (adapted from Kalwij and Peralta [15]).

TS control parameter	Corresponding TS Step (Fig. 3)	Value ranges that yield convergence[(a)]
Probability of invoking the TS(-)	---	0.20–0.50
Search coarseness (%)	1	0.1–0.2
Solution space size (%)	2	15–50
Tabu list size (%)	3	10–40
Threshold accepting value (%)	4	0.001–1

(a) this implies convergence to the correct subset of wells for 60–90% of the 10 optimization runs; and 60–80% of the runs converge to within 1% of the best optimal solution.

optimization is a function of optimization performance. For example, if for a prespecified number of consecutive GA–TS generations, the OFV does not improve, the search coarseness and solution space control parameters are reduced by ΔX (eqns (1) and (2)) [15].

$$\Delta X = \beta \Delta X_{\text{old}} + (1-\beta)\Delta X_{\text{new}} \tag{1}$$

$$\Delta X_{\text{new}} = X \frac{\left|OFV_0 - OFV_{\text{new}}\right|}{\left|OFV_0\right|}, \tag{2}$$

where ΔX is the change in search coarseness or solution space size parameter; ΔX_{old} and ΔX_{new} are the old and new changes in parameter value, respectively; β has a value [0, 1] and controls how much ΔX is based on ΔX_{old} and ΔX_{new}; ΔX_{new} is a function of the present parameter value (X), and the first feasible OFV (OFV_0) and the best OFV to date (OFV_{new}).

Equation (2) prevents too large parameter value reductions during early optimization. Adaptive control parameter modifications at the end of a generation generally decrease search coarseness and refine solution space size parameter values. However, if the OFV does not improve after a prespecified number of modifications, search coarseness and solution space size parameter values are reset to predefined values, thereby preventing the optimizer from becoming stuck in local optima.

The auto-adaptive approach includes a gradually increasing TS probability and tabu list size. How much these are increased depends on the optimization stage in which the alterations occur [15].

Section 4 discusses the performance of this auto-adaptive approach, contrasted to a GA and (nonadaptive) GA–TS, applied to a real-world transport optimization problem.

There are many ways to implement auto-adaptive concepts. Here we demonstrate just a few, to illustrate the potential of this optimization approach.

4 Application to Blaine Naval Ammunition Depot

4.1 Situation and optimization approach

Blaine Naval Ammunition Depot (NAD), in Hastings, Nebraska, has significant contamination of groundwater by volatile hydrocarbons, and some from dissolved phase 2,4,6-trinitrotoluene (TNT) explosive. Figure 4 shows the finite difference grid of flow and transport simulation models of the contaminated aquifer. Figures 5 and 6, respectively, show trichloroethylene (TCE) and TNT plumes simulated to exist in January 2003. The project report contains study area and simulation model details [11]. The study area consists of two mutually exclusive zones. Contamination is within the irregularly shaped cleanup zone. Other cells lie within the exclusion zone. The greatest concentration of a species predicted to result from a pumping strategy in a zone at a particular time is $\text{cmax}_{\text{species, zone, time}}$.

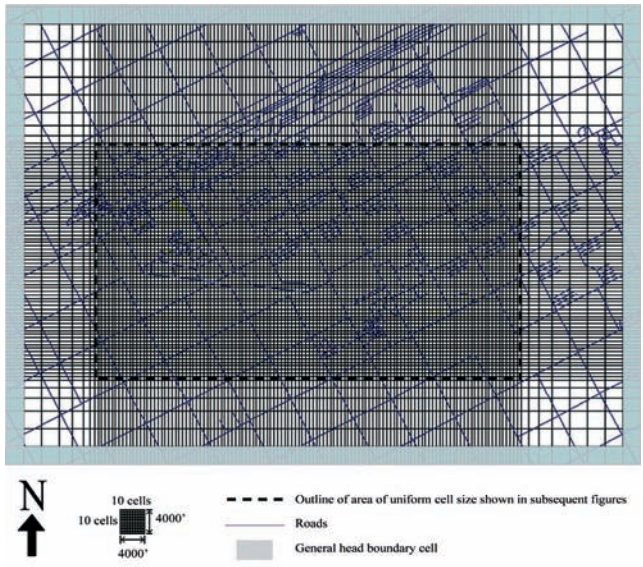


Figure 4: Finite difference grid superimposed on Blaine naval ammunition depot road map.

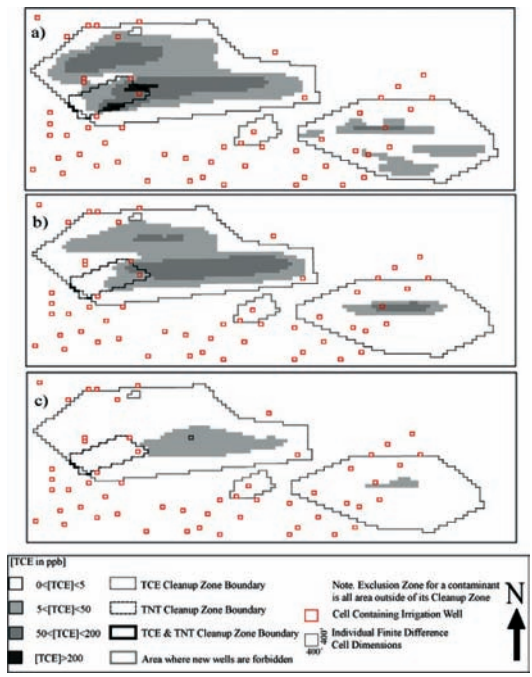


Figure 5: Initial (simulated for 1 Jan 2003) TCE concentrations exceeding 5.0 ppb, and part of finite difference grid with rows and columns numbered in: (a) Layer 3, (b) Layer 4, and (c) Layer 5.

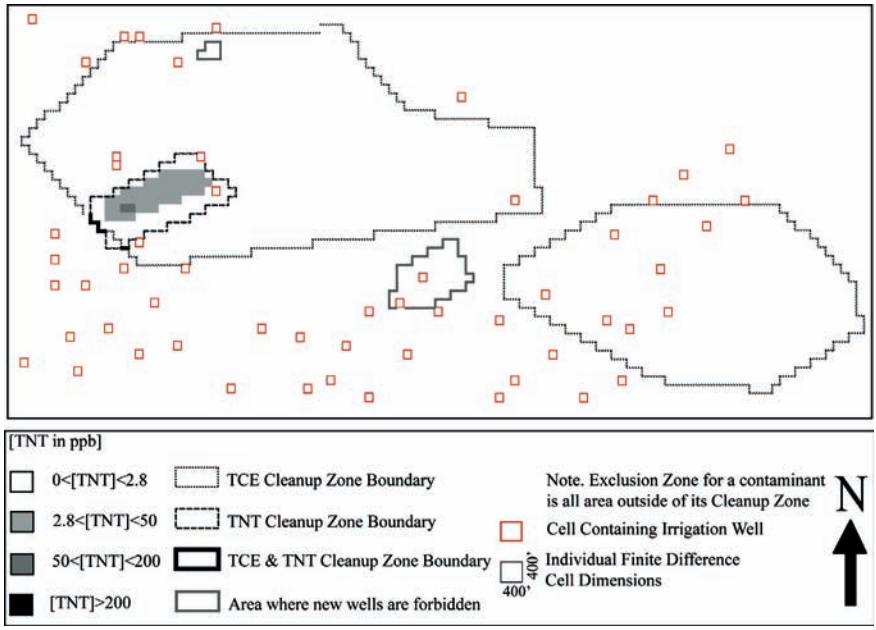


Figure 6: Initial (simulated 1 January 2003) TNT concentrations exceeding 2.8 ppb in layer 3, and part of finite difference grid.

Here we discuss one of three optimization problem formulations posed by NAD (eqns (3) and (4)). The formulation minimizes the present value of the cost of containing TCE and TNT plumes for 30 years (six five-year pumping periods), and remediating them to not exceed $cmax^U$ by no later than the end of 30 years.

$$\text{Minimize } Z = \sum_{k=1}^{6} \left\{ \begin{array}{l} DFC_k (CE_k + CT_k + CD_k) \\ +DFF_k (FM_k + FS_k) \\ +DFV_k (VE_k + VT_k + VD_k) \end{array} \right\}, \tag{3}$$

where

Z = present value of 30 years of strategy implementation; k = a 5-year period of uniform pumping (k = 1 at time 0). Capital costs can be incurred, and wells can be installed, and begin, change, or cease pumping only at beginnings of years 1, 6, 11, 16, 21, and 26; DFC_k = a factor used to convert capital expense occurring at the beginning of period k, into a present value; DFF_k = a factor used to convert annual expenses occurring during period k into a present value; DFV_k = a factor used to convert costs of pumped and treated flows occurring during period k, into a present value; CE = capital costs of new extraction wells ($400K); CT = capital cost of treatment ($1.0K per gpm); CCD = capital cost of discharge piping ($1.5K per gpm); FM = fixed cost of management ($115K/yr); FS = fixed cost of sampling and analysis ($300K/yr); VE = variable costs of electricity for operating wells ($0.046K/gpm); VT = variable cost of treatment ($0.283K/gpm); and VD = variable cost of discharge ($0.066K/gpm).

Subject to

$q_{\hat{e}}^{L} \le q_{\hat{e},k} \le q_{\hat{e}}^{U}$ for each candidate extraction well (maximum rates are 350, 700, & 1050 gpm if screened in one, two or three layers, respectively)

$\text{cmax}_{\text{species,zone,time}} \le \text{cmax}^{U}_{\text{species,zone,time}}$

where

$q_{\hat{e},k}$ = extraction pumping rate at well $\hat{e}$ during period k

$$\text{cmax}^{U}_{\text{TCE, cleanup zone, 30 yr}} = \text{cmax}^{U}_{\text{TCE, exclusion zone, every 5 yrs}} = 5 \text{ ppb}$$
$$\text{cmax}^{U}_{\text{TNT, cleanup zone, 30 yr}} = \text{cmax}^{U}_{\text{TNT, exclusion zone, every 5 yrs}} = 2.8 \text{ ppb} \quad (4)$$

Employed was an S–O model consisting of embedded flow and transport simulation models (MODFLOW groundwater flow model [27] and MT3DMS transport model [28], and linked GA and TS search optimization algorithms (GA–TS).

4.2 Application and results

At the time this optimization effort was performed, one simulation of both flow and TCE transport for just one pumping strategy required almost one hour of real time. To reduce total computational effort, before optimization, a set of candidate pumping wells and an initial feasible pumping strategy were developed. These were provided as input to the S–O model. The S–O model subsequently computed an optimal pumping strategy for those candidate wells, that employed each well at some time during the 30 years. We repeated this (in effect an NLP optimization), until we believed we had the best transient pumping strategy for that set of candidate wells. We term it the base optimal strategy.

Then we compared how well three S–O implementations could approach the OFV of the base optimal strategy. The implementations were:

Model (A) nonadaptive genetic algorithm alone (GA)
Model (B) nonadaptive genetic algorithm with coupled tabu search (GA–TS)
Model (C) adaptive genetic algorithm with coupled tabu search (Adaptive GA–TS)

Employed TS parameters were search coarseness (1%), solution space step size (30%), tabu list size (10%), and TS probability (0.20). These values fall within the ranges shown in Table 1. All GA input parameter were the same for all three models. Model A does not use TS control parameters; Models B and C use identical TS control parameters, with the only difference that for Model B these parameter values are fixed throughout optimization, whereas for Model C TS control parameter values adjust during optimization subject to optimization performance.

We made multiple optimization runs using each of models A–C. Figure 7 shows dimensionless average convergence for each model toward the base optimal strategy (the base optimal strategy is one of the runs). The denominator of the *y*-axis is the improvement that the base optimal strategy was over the initial feasible solution. By viewing where the S–O curve reached the right *y*-axis, we see that S–O Model C achieved an average 98.38% of the improvement obtained by the base

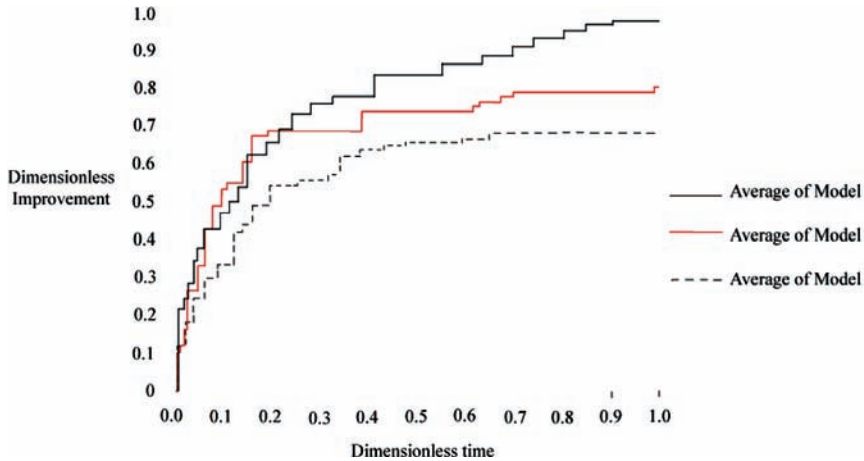


Figure 7: Dimensionless convergence of GA, GA–TS, and auto-adaptive GA–TS optimization algorithms.

optimal strategy. S–O models B and A achieved average values of 79.59% and 66.47%, respectively (the higher the value the closer the optimizer converges to the best optimal result achieved during this exercise).

Giving the GA–TS hybrid (Model B) the ability to intelligently change optimization parameters during optimization (the result is Model C), caused a 0.1879 (or 18.79%) dimensionless OFV improvement (0.7959–0.9838). Augmenting a GA (Model A) with TS ability to produce Model B caused a 0.1313 improvement (13.13%).

For Fig. 7, the denominator of the *x*-axis is the total time required by the respective S–O Model A, B, or C. This shows that Model B was converging faster than Model A; Model C was converging faster than Model B (also, Model C improves overall strategies more quickly than Model B). Model C achieves 85.3% of total possible OF improvement at 50% of the total computational run time. Model B achieves 73.5% OF improvement for the same time frame. When time is limited it is beneficial if an optimizer can converge faster. Stochastic processes can display such results.

5 Closure and outlook

Although tabu search (TS) can be used as a stand-alone heuristic optimizer, it is often used as a metaheuristic optimizer. In that capacity, it guides other optimizers, such as genetic algorithm (GA) and simulated annealing. We illustrate that a hybrid GA–TS optimizer converges to an optimal solution more rapidly than a GA optimizing alone. Yet faster is a hybrid GA–TS that auto-adaptively changes TS algorithm operational parameter values during optimization. Here we demonstrate GA–TS use in a simulation–optimization model that performs contaminant transport optimization for large-scale groundwater contaminant plume remediation. We have also used TS with simulated annealing for hydraulic optimization dewatering.

Tabu search features can be added as appropriate for particular problems. That flexibility makes the continued development of new tabu search applications and algorithms very likely in many fields, including water science and engineering.

References

[1] Glover, F., Future paths for integer programming and links to artificial intelligence, *Comput*ers and *Operations* Research, **13(5)**, pp. 533–549, 1986.
[2] Glover, F., Tabu search – Part I, *ORSA Journal on Computing*, **1(3)**, pp. 190–206, 1989.
[3] Glover, F., Tabu search – Part II, *ORSA Journal on Computing*, **2(1)**, pp. 4–32, 1990.
[4] Home page of Fred Glover, http://leeds-faculty.colorado.edu/glover/.
[5] Glover, F., Tabu search and adaptive memory programming – advances, applications and challenges. In: *Interfaces in Computer Science and Operations Research. Barr, Helgason and Kennington, eds.*, Kluwer Academic Pub., 1996.
[6] Zheng, C. & Wang, P., Parameter structure identification using tabu search and simulated annealing. Advances in Water Resources, **19(4)**, pp. 215–224, 1996.
[7] Kirkpatrick, S., Gelatt, Jr., C.D. & Vecchi, M.P., Optimization by simulated annealing. *Science*, **220(4598)**, pp. 671–680, 1983.
[8] Zheng, C. & Wang, P.P., An integrated global and local optimization approach for remediation system design. *Water Resources Research*, **35(1)**, pp. 137–148, 1999.
[9] Peralta, R.C. & Wu, S., (Hydro Geo Systems Group), Optimal pumping-treatment-reinjection strategies for CS-10 TCE plume, Massachusetts Military Reservation CS-10 TCE plume. Report submitted to EnviroTech Center, 2000.
[10] Peralta, R.C., Wu, S., & Kalwij, I.M., Optimal P&T designs for Umatilla Army Ammunition Depot. *Project Completion Report for U.S. Navy*, 2002.
[11] Peralta, R.C., Kalwij, I.M. & Wu, S., Optimal P&T designs for Tooele Army Depot. *Project Completion Report for U.S. Navy*, 2002.
[12] Peralta, R.C., Kalwij, I.M. & Wu, S., Optimal pumping strategies for TCE and TNT plumes at Blaine Naval Ammunition Depot, Hastings, NE. *Project Completion Report for U.S. Navy*, 2002.
[13] Holland, J.H., *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press, 1975.
[14] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston: Addison-Wesley Longman, 1989.
[15] Kalwij, I.M. & Peralta, R.C., Non-adaptive and adaptive hybrid approaches for enhancing water quality management, *Journal of Hydrology*, **358(3–4)**, pp.182–192, 2008.
[16] Kalwij, I.M. & Peralta, R.C., Simulation/optimization modeling for robust pumping strategy design, *Ground Waters*, **44(4)**, pp. 574–582, 2006.

[17] Peralta, R.C., Kalwij, I.M. & Wu, S., Practical remedial design optimization for large complex plumes. *Journal of Water Resources Planning and Management*, **134(5)**, pp. 422–431, 2008.
[18] Kalwij, I.M. & Peralta, R.C., Intelligent space tube optimization for speeding ground water remedial design. *Ground Water*, **46(6)**, pp. 829–840, 2008.
[19] Cunha, M. & Ribeiro, L., Tabu search algorithms for water network optimization. *European Journal of Operational Research*, **157(3)**, pp. 746–758, 2004.
[20] Lippai, I, Heaney, J.P. & Laguna, M., Robust water system design with commercial intelligent search optimizers. *Journal of Computing in Civil Engineering*, **13(3)**, pp. 135–143, 1999.
[21] Yeh, H.-D. Chang, T.-H. & Lin, Y.-C., Groundwater contaminant source identification by a hybrid heuristic approach. *Water Resources Research*, **43**, W09420, doi:10.1029/2005WR004731, 2007.
[22] Tung, C.-P. & Chou, C.-A., Application of tabu search to ground water parameter zonation, *Journal of the American Water Resources Association*, **38(4)**, pp. 1115–1125, 2002.
[23] Martínez, S.O., Merwade, V. & Maidment, D., Linking GIS, hydraulic modeling, and tabu search for optimizing a water level-monitoring network in South Florida, *Journal of Water Resources Planning and Management*, **136(2)**, pp. 167-176, 2010.
[24] Glover, F. & Laguna, M., *Tabu Search*. Norwell, MA: Kluwer Academic Publishers, 1997.
[25] Glover, F., Laguna, M. & Martí., M., Principles of Tabu Search (Chapter 23). *Handbook of Approximation Algorithms and Metaheuristics*. London: Taylor & Francis Group, LLC, 2007.
[26] SSOL, *SOMOS Software User's Manual*. Systems Simulation/Optimization Laboratory, Department of Biological and Irrigation Engineering, Utah State University, 2004.
[27] McDonald, M.G. & Harbaugh., A.W., *MODFLOW: A Modular Three Dimensional Finite-Difference Ground-Water Flow Model*. USGS Techniques of Water-Resources Investigations, Washington, DC (Book 6, Chapter A1), 1988.
[28] Zheng, C. & Wang, P.P., *MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion and Chemical Reactions of Contaminants in Groundwater Systems*. Documentation and user's guide. Contract Report SERDP-99-1, U.S. Army Engineer Research and Development Center, Vicksburg, MS, 1999.

This page intentionally left blank

CHAPTER 7

Harmony search algorithm

N.P. Theodossiou & I.P. Kougias
Division of Hydraulics and Environmental Engineering, Department of Civil Engineering, Aristotle University of Thessaloniki, Greece.

Abstract

During the past few years, a new method has been added to those optimization techniques inspired by natural phenomena like genetic algorithms. The music-inspired harmony based optimization algorithm, commonly referred to as Harmony Search Algorithm (HSA), is a novel, powerful optimization technique based on the observation of music creation and its quest of the perfect state of harmony. In this chapter, the concept and the basic characteristics of the harmony search algorithm are comprehensively presented. Since the method is still in full development, a number of variations aiming to improve its efficiency have appeared. Their main differences are also outlined in this chapter along with a benchmark evaluation of their effectiveness. Finally, an application on a classic multireservoir system management problem is presented, in order to demonstrate the efficiency of the Harmony Search Algorithm.
Keywords: optimization techniques, Harmony Search Algorithm, music-inspired harmony based optimization algorithm, Rosenbrock function, Griewangk function, multi-reservoir management.

1 Introduction

1.1 Relationship between music and mathematics

Very early in time, people realized the strong relationship between music and mathematics. Since the ancient times, mathematicians tried to interpret the governing rules of mathematics using the art of music. Amongst them the work of Pythagoras, the Greek philosopher and mathematician, stands out. Pythagoras tried to investigate the laws that dominate sounds and thus he, informally, founded the science of acoustics.

In the following centuries, this relationship was retained. Very often musicians tried to use mathematics in order to deeply understand music. On the other hand, mathematicians, like Fibonacci, through their work, assisted artists and composers to seek harmony.

During recent times, since the Baroque period, this bond has been strengthened. Sometimes as a conscious effort by musicians-composers and sometimes as part of a rumored and almost mystical relationship, mathematics and music came closer.

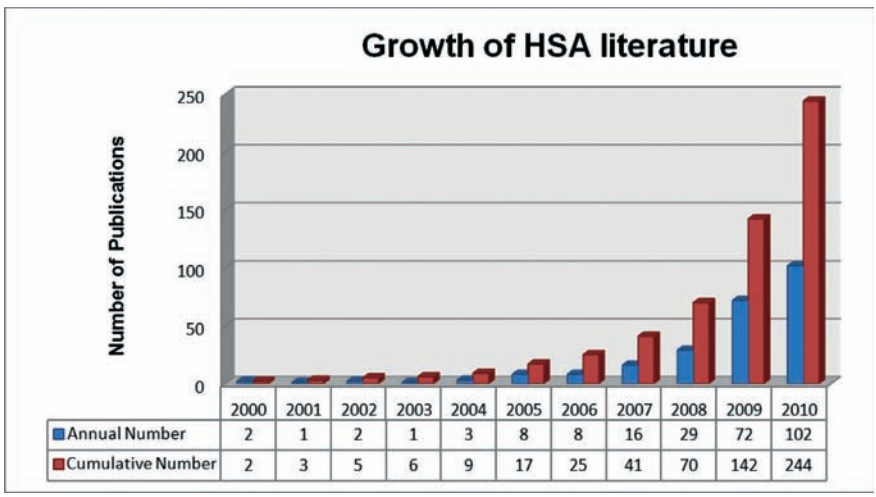
Iannis Xenakis represents a very special example from the 20th century. Xenakis combined very skillfully his identities as a mathematician and a composer. His work, through which he was recognized as one the most important composers of the century, is based on the use of mathematical functions to produce music and sounds.

This relationship inspired Zong Woo Geem in the development of a new optimization algorithm, a meta-heuristic algorithm, based on music. Being a Civil Engineer and a musician himself, he presented in 2000 the Harmony Search Algorithm through an application in the design of water supply networks [1, 2].

1.2 Spreading of the method

Even since the first applications of the method, its significant perspectives were evident. The Harmony Search Algorithm (HSA) does not require the determination of initial values and it has less mathematical demands resulting to much simpler computer programming. Moreover, since it does not require the use of the binary system, the computer code is shorter and faster during execution. All these reasons have led to a wider spreading of successful applications, from the solution of Sudoku puzzles to the composition of music and even to the design of space technology.

Respectively, the number of scientific publications presents an increasing trend. While until 2004 the relevant publications were limited, since 2005 the



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Annual Number	2	1	2	1	3	8	8	16	29	72	102
Cumulative Number	2	3	5	6	9	17	25	41	70	142	244

Figure 1: Increase of annual scientific publications.

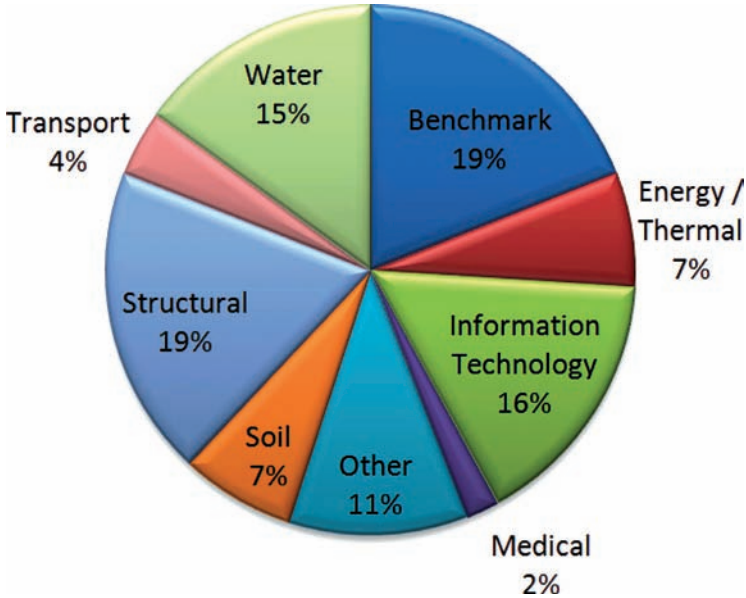


Figure 2: Analysis of scientific fields implementing applications in 2010.

annual number of published papers increased significantly. More specifically the number of published papers in 2005 was more than double those published in 2004, while during 2009 nine times more papers were published compared to the respective number of 2005 and 2006 [3, 4]. In 2010, the number of publications in journals exceeds more than three times the number published in 2009.

From all the above, it becomes very clear that the Harmony Search Algorithm, despite its very short presence amongst the optimization methods, has attracted the attention of the scientific community demonstrating significant results. As a matter of fact since 2009, as a result of the escalation of the use of the algorithm in IT technology, the number of publications has exponentially increased.

Even though I was originally designed for the optimization of the design of water supply networks, its applications soon expanded in a wide range of scientific fields. Applications in structural, geotechnical, and mechanical engineering, in transports, in medicine even in the design of space stations were developed. Thus, while during the first years almost 50% of the applications were targeting water resources management and other hydraulic works, the wide spreading of the method has led to a reduction of this percentage to less than 15% in 2010 [4]. However, research regarding hydraulic works is still expanding to other fields as groundwater modeling and management [5, 6].

2 Presentation of the Harmony Search Algorithm

2.1 The concept of harmony

The term harmony, in music, refers to the sound effect caused by two or more instruments playing at once. By terms of acoustics, harmony refers to the relationship between different sound waves, coming from either musical instruments or human voices, and to their interaction that defines whether the final result is characterized as pleasant or not. Although in music, as in every artistic expression, beautiful and acceptable are subjective, through the years some objective rules regarding harmony have been developed. These rules follow the trend of each artistic period and each musical current. It is not a coincidence that Pythagoras's calculations during the 6^{th} century BC resulted to harmonic sounds that are considered as such even in the classical harmony of the 18^{th} century.

2.2 Method analysis

2.2.1 The search of harmony in music

The new algorithm was inspired by the way a musician plays within a music group. During rehearsals or a concert, a musician has three choices:

i. To play the known – basic melody of the musical piece. This melody is known as the 'theme' and it characterizes every piece. It is obviously known and already in the memory of the musician.
ii. To play something similar to the basic melody. To slightly change the 'theme' enriching the piece with notes never played before.
iii. To start an improvisation by selecting new note sequences that will form a completely new music material.

2.2.2 The basic elements of the algorithm

The Harmony Search Algorithm is a stochastic meta-heuristic method based on the sequential production of possible solutions. It belongs to the category of 'neighborhood meta-heuristics' that produce one possible solution per iteration. This procedure is completely different from that of the population methods that produce a number of possible solutions in every iteration (e.g., genetic algorithms).

Every possible solution consists of a set of values of the decision variables of the function that needs to be optimized. Each one of these sets of values is called a 'Harmony.' During the optimization process, a number of 'harmonies' equal to the 'Harmony Memory size' are stored in the 'Harmony Memory' (HM), a database that includes the produced set of solutions. The algorithm ends when the predefined total number of iterations has been achieved [7].

2.2.3 Development of the Harmony Search Algorithm

The first stage for the application of the Harmony Search Algorithm is the description of the problem and the definition of its parameters. This is a very important

stage since the formulation of the decision variables, their boundaries, and their participation in the objective function are decisive for a realistic approach of the optimization problem.

After the definition of the variables, the matrix containing the Harmony Memory is formulated. The dimensions of the matrix are $m \times n$, where m is the Harmony Memory size and n the number of variables involved in the objective function. The procedure demands that before the first run, the matrix must be complete with initial values. This is usually achieved with sets of random values of the decision variables.

At this point, the algorithm is ready to start to produce and evaluate new 'Harmonies' by simulating the search for harmony in music through the application of the following mechanisms in respect to the procedures described in Section 2.2.1:

Use of one of the set of variables stored in the Harmony Memory (HM). The Harmony Memory consideration is very important since it ensures that good harmonies, solutions that provide good results, will be considered during the optimization process and will form the basis for the production of even better solutions. For the effective application of this procedure, the Harmony Memory Consideration rate (HMCR) is used. If this index is assigned small values then only a few of the good solutions stored in the HM will be taken into consideration in the evaluation of new solutions resulting to a very slow convergence. On the other hand, large values of HMCR will impose restrictions in the ability of the procedure to investigate a large field of possible solutions. Technically, HMCR takes values larger than 70%, sometimes even exceeding 95%.

(ii) A usually smaller percentage of the solutions selected with the application of the previous mechanism are allowed to be slightly altered. This is the second mechanism of the algorithm. In this case the pitch adjusting rate (PAR) is used. The concept is that if the algorithm chooses to slightly alter the term xi selected from the harmonic memory, this will be performed by selecting a neighboring value of x_i, such as

$$x_i^{\text{new}} = x_i \pm \text{Random (bw)} \tag{1}$$

where Random (bw) is a random number between 0 and 1 expressing the bandwidth of the adjustment. This procedure is similar to the mutation in genetic algorithms. It is worth noting that although PAR can only take small values it is considered to play an important role in the convergence of the algorithm.

(iii) The third procedure is improvisation, which means to introduce completely random values in new solutions. The probability of introducing random values is (100-HMCR)%. In this way, the variability of solutions is enriched. Contrary to the PAR parameter through which the algorithm investigates the area around the values of the Harmony Memory, randomization aims in widening the field of solutions, thus ensuring the determination of global optimal solutions of the objective function. After the creation of a new 'Harmony,' its performance is evaluated according to the corresponding value of the objective function. If this performance is better than that of the worst 'Harmony' stored in the Harmony Memory,

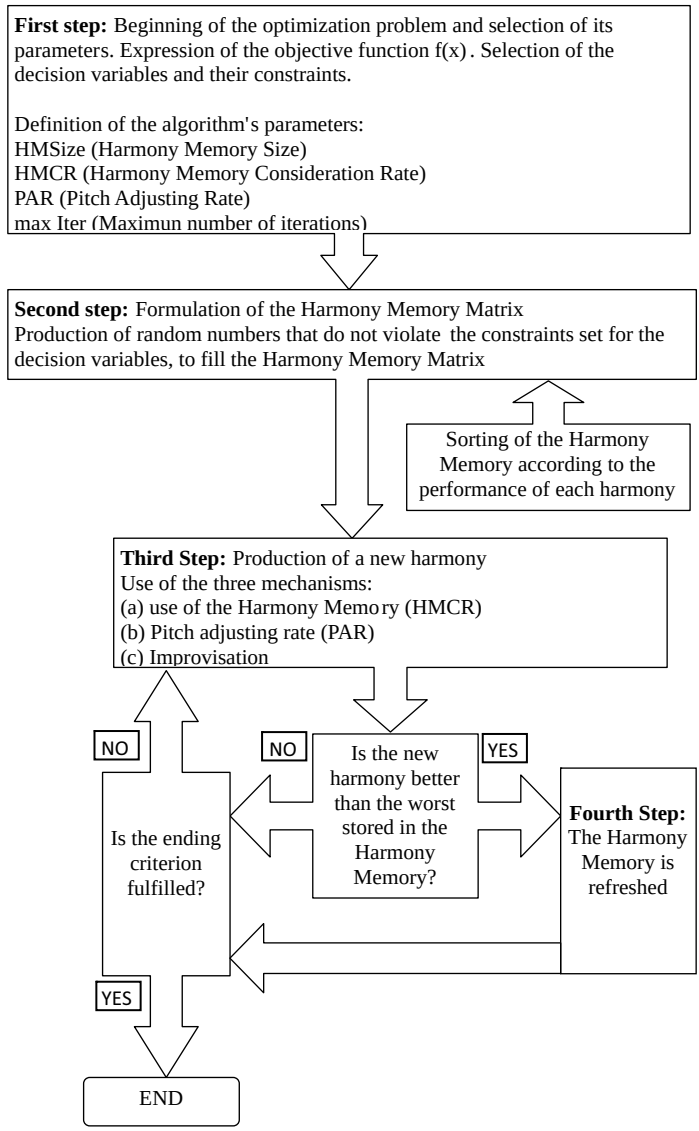


Figure 3: Flow chart of Harmony of Harmony Search Algorithm.

it replaces it. This procedure is repeated until the ending criterion, usually a maximum number of iterations, is reached. Choosing values for the parameters mentioned above has always been an issue. Moreover, the size of Harmony Memory is one more parameter that needs calibration. Different scientists have proposed various formations for HMCR, PAR, and HM size, often quite different from each other. Xin She Yang has proposed HMCR = 0.7–0.9 and PAR = 0.1–0.5 [8], while Zong Woo Geem generally proposes lower values for PAR index [9].

In the present chapter, a calibration of HSA parameters is implemented, using two test functions. This calibration aims to add some knowledge regarding the fine tuning of Harmony Search Algorithm.

3 Variations of Harmony Search Algorithm

After the successful implementations of HSA and the impressive results, various scientists have proposed variations of HSA which might perform better in certain problems. These algorithms share the same basic structure, but some changes in their setup cause different performance, which in some cases has been proven to be superior.

3.1 Improved Harmony Search Algorithm from ensemble of music players

In 2006, Zong Woo Geem proposed a variation of the classic algorithm called 'Improved Harmony Search from Ensemble of Music Players' which introduced a new mechanism [10].

This new mechanism, called 'ensemble of music players' is applied after the creation of a new 'Harmony,' between steps (iii) and (iv) (see figure 3) presented earlier. The basic idea of this variation is that very often in a music ensemble, some of the musical instruments appear to have a stronger bond. The similarity of the melody played by two instruments, the continuous questions and answers, the musical dialogs are a common practice in music. A characteristic example is that of two groups of violins within a symphonic orchestra, where the first violins are in a continuous dialog with the group of second violins. It is obvious that their connection is much stronger than that between the violins and the percussions, for example.

Similarly, in many optimization problems some of the decision variables are more connected than others. For example the pumping rates between neighboring wells are interrelated. An increase in the pumping rate of one well results in a decrease in the pumping rate of the other, in order to maintain the constraints set on the water level drop of a groundwater aquifer.

The value of a decision variable in a new 'Harmony' x_j' created by the algorithm can depend on the value of another decision variable x_i', if the two variables x_i and x_j already in the Harmony Memory are correlated. The above observation is applied through the following equation:

$$x_i' \leftarrow \text{fn}(x_j'), \text{ where } \max\{[\text{corr}\,(x_i, x_j)]^2\} \text{ and } i \neq \text{j} \qquad (2)$$

Furthermore:

– The function corr() indicates the correlation between variables $x_i = (x_i^1, x_i^2, \ldots, x_i^{HMS})$ and $x_j = (x_j^1, x_j^2, \ldots, x_j^{HMS})$ using the coefficient of linear correlation r.

– max{} is a function that determines the stronger correlation between the variables x_j and x_i based on the term r^2.

– fn() is a function that defines the term x_i' based on the value of x_j'. Function fn() determines the value of x_i that appears most often between the pairs x_i and x_j, when x_j is equal to x_j'.

In order to implement this variation in a percentage of the new harmonies created, a new parameter called ECR (ensemble consideration rate) is introduced ($0 \leq ECR \leq 1$). After the creation of the new 'Harmony' using the four, instead of three, mechanisms, the algorithm is applied in a similar way and the new 'Harmony' is either included in the Harmony Memory or it is rejected.

3.2 Improved Harmony Search Algorithm

In 2007, scientists from the University of Tehran under M. Mahdavi, suggested another variation, which they called 'Improved Harmony Search' [11].

This variation is based on the observation that the parameter HMCR and generally the mechanism that uses the Harmony Memory, aims at a wider search of optimal solutions (global search). Respectively, the pitch adjusting rate mechanism aims in optimizing search on a local level (local search). The PAR parameter and the bandwidth bw are very important for the optimal function of the algorithm. Thus Mahdavi et al. considered the use of constant values for these two parameters, as a weakness of the algorithm.

He suggested that small values of the parameter PAR along with large values of the bandwidth bw can result to low efficiency of the algorithm and significant increase in the number of necessary iterations needed for the convergence to the global optimum. At the same time he observed that while small values of the bandwidth bw at later iterations improve the efficiency of the algorithm, at early iterations they cause problems. So he arrived at the conclusion that during the early stages of the algorithm the value of bw should be allowed to take larger values in order to assist the algorithm to produce a variety of available solutions in the Harmony Memory. During the later stages of the algorithm though, a combination of large values for the parameter PAR with small values of the bandwidth, can improve significantly the efficiency of the algorithm.

The above remarks led Mahdavi to introduce non-constant varying with time values for PAR and bw. So the improved harmony search (IHS) algorithm allows PAR to increase linearly and the bandwidth bw to decrease exponentially. These variations are introduced, as in the classical algorithm, during the third step and vary dynamically as follows:

$$PAR(gn) = PAR_{min} + [(PAR_{max} - PAR_{min}) \times gn]/MaxIter \quad (3)$$

where PAR is the pitch adjusting rate

PAR_{max} and PAR_{min} are the maximum and minimum values of PAR, respectively

MaxIter is the maximum number of iterations

and gn is the current iteration

$$bw(gn) = bw_{max} \exp(c \times gn), \quad (4)$$

where bw is the bandwidth

bw_{max} and bw_{min} are the maximum and minimum values of bw, respectively

MaxIter is the maximum number of iterations

and $c = \ln(bw_{min}/bw_{max})/MaxIter$

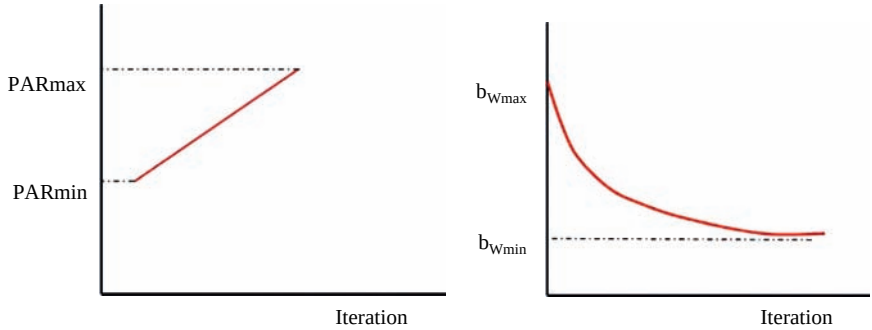


Figure 4: Variation of PAR and bw in improved Harmony Search Algorithm [11].

3.3 Global-best Harmony Search Algorithm

In 2008, M. Omran with M. Mahdavi, who presented the previous variation, created a new algorithm which they called Global-Best Harmony Search [12]. The specific variation includes elements of another meta-heuristic method, the particle swarm optimization. This algorithm is based on the fact that the position of a member of a swarm depends on its previous positions and those of the best member of the swarm. In the same way Omran and Mahdavi propose a variation on the function of the pitch adjusting rate mechanism. Thus they introduce a varying value for PAR and they replace the bandwidth bw.

Every time the algorithm chooses to perform a pitch adjustment, it will replace the chosen value from the Harmony Memory with one of the values of the parameters of the best 'Harmony' stored in the Harmony Memory. This adjustment led to remarkable results during the benchmark tests and is applicable in both discrete and continuous problems.

The computer code that introduces the variation proposed by Omran and Mahdavi, is presented through the following pseudocode:

for each i e; [1,*N*] **do**
if *U*(0,1) ≤ HMCR **then**/*use of Harmony Memory*/
begin
$x_i' = x_i^j$, where $j \sim$ U(1,. . . , HMS)
if *U*(0,1) ≤ PAR **then**/* Pitch Adjusting Rate */
begin
$x_i' = x_i'$, where *best* is the index of the best 'Harmony' stored in the harmony memory and $k \sim U(1,N)$.
endif
else/*improvisation*/
x_i'= random
endif
done

3.4 Self-adaptive Harmony Search Algorithm

This specific variation of the Harmony Search Algorithm was presented in 2010 by Chia-Ming Wang and Yin-Fu Huang [13]. Like the two previous variations it interferes with the structure of the pitch adjusting rate mechanism of the classical algorithm.

Once again it was recognized that the determination of the dominating parameters of the algorithm was one of the main problems of its application, since for some of them, even the order of magnitude remained unknown.

After testing the proposed variation they expressed their reservations concerning the continuous increase of the PAR coefficient according to the Improved Harmony Search Algorithm. They considered that the search performed through the pitch adjustment mechanism should be intense during the first stages and should then gradually decrease. In this way, they estimated that early convergence and oscillation around local optima could be avoided.

This claim was enforced by the observation that the application of a constant value for PAR in the Global-best Harmony Search Algorithm improved its efficiency compared to that of its linear increase.

The suggested variation included the linear reduction of the PAR coefficient during the application, from 1 to 0. At the same time they related the adjustment width with the maximum and minimum values of the variable observed, after every iterations in the Harmony Memory. Thus the new variable x_i selected to be pitch adjusted from the Harmony Memory, can be altered in two ways:

$$x_i + [\max(HM_i) - x_i] \times \text{ran}[0,1) \tag{5}$$

$$x_i - [x_i - \min(HM_i)] \times \text{ran}[0,1) \tag{6}$$

where: x_i is the value selected to be pitch adjusted from the harmony memory
where $\max(HM_i)$ and $\min(HM_i)$ are the maximum and minimum values of the variable in the Harmony Memory, respectively.
and ran [0,1) a random number between 0 and 1

The specific structure was tested using examples solved with the other variations and the results indicated an efficient and successful algorithm.

3.5 Parameter-setting-free Harmony Search Algorithm

Acknowledging the problems raised by the several variations concerning the determination of the parameters, Dr Z.W. Geem, the creator of the original Harmony Search Algorithm, designed a new variation, published in 2011, that does not need the adjustment of parameters. This includes a dynamic variation of the Harmony Memory Consideration Rate (HMCR) [14].

This specific variation is described as follows. The Harmony Search Algorithm starts and is executed for a small number of iterations (for example 3 times the Harmony Memory size). At that point another matrix is created with the same dimensions as the Harmony Memory matrix. This matrix is called Functional

Memory and includes the origin of the elements of the Harmony Memory. In this way, if an element of the Harmony Memory comes from the improvisation mechanism, the functional memory stores this information.

This specific variation regulates automatically the parameters in respect to the number of elements produced by each mechanism. In this way:

HMCR = (Number of elements originating from the use of memory or PAR)/(HM size)

PAR = (Number of elements originating from PAR)/(HM size)

Every time a new solution, better than those stored in the Harmony Memory, is produced, the functional memory is renewed. At the same time the values of the parameters, HMCR and PAR, are altered.

It should be noted that another particularity of this method is that, apart from the continuous alteration of HMCR and PAR parameters during the solution process, to each one of the decision variables corresponds a different value of these parameters. This is a result of the fact that these parameters depend on the number of elements in the Harmony Memory corresponding to a certain origin.

In order to avoid early convergence of this variation, Geem introduced another adjustment. It is obvious that if the parameter HMCR reaches the value of 1, it will remain constant during the solution process. The same result will occur if the parameter PAR reaches the value of 0. For this reason, the algorithm must be able to avoid early convergence when these parameters have values close to 1 and 0, respectively. This is accomplished with the application of the following control equations:

$$\text{HMCR} = [\text{HMCR} + \Phi_{\text{HMCR}} \times \text{ran}(-1,1)] \tag{7}$$

If the above term is not between [0, 1], then HMCR retains its value

$$\text{PAR} = [\text{PAR} + \Phi_{\text{PAR}} \times \text{ran}(-1,1)] \tag{8}$$

If the above term is not between [0, 1], then PAR retains its value where the terms Φ_{HMCR} and Φ_{PAR} are noise coefficients with suggested values 0.05 and 0.1, respectively.

This variation of the Harmony Search Algorithm was applied in a problem of optimized design of a water supply network. The results as far as fast and efficient convergence to the globally optimal solution were very encouraging. A significant advantage of this technique according to Geem is that it does not require assignment of values to the algorithms parameters. The only value required is the maximum number of iterations. This characteristic can help to the wider spreading of the method, since its application does not require knowledge of the theoretical background of the Harmony Search.

It must be noted though that even if it is not required to assign values to the problem's parameters, one still needs to adjust the noise coefficients Φ. It is expected that the future wider application of this variation of the Harmony Search Algorithm will contribute to the clarification of whether it is actually more efficient and more user-friendly.

4 Evaluation of the variations of Harmony Search Algorithm

In order to demonstrate these variations, their field of application and their effectiveness, two of the most successful ones are analyzed: 'Improved Harmony Search' and 'Global-Best Harmony Search.' As extensively presented in the previous paragraphs the Improved HSA proposes dynamic change of PAR parameter and its bandwidth throughout the solution. Thus in Improved HSA [11], PAR increases linearly from a PAR_{min} to a PAR_{max} value, while the bandwidth decreases exponentially in order to narrow the local search toward the global optimum. Respectively, the Global-Best HSA [12] includes the linear change of PAR similarly with Improved HSA. However, every time Pitch Adjustment is about to happen, Global-Best HSA considers only variables that constitute the best-so far solution.

4.1 Test functions

In order to calibrate HSA and its variations, two test – functions were used, Rosenbrock and Griewangk [15]. These two functions have already been used by many scientists including those involved in HSA's research. Geem and Wang & Huang [13] have already used them in order to calculate and improve HSA's performance.

4.1.1 Rosenbrock function

Rosenbrock function has been often used in mathematical optimization as a performance test for optimization techniques. It is also known as Rosenbrock's valley, or Rosenbrock's banana function. In the two-dimensional problem, the minimum is inside a long, narrow, parabolic shaped valley (Fig. 5). It is difficult to locate

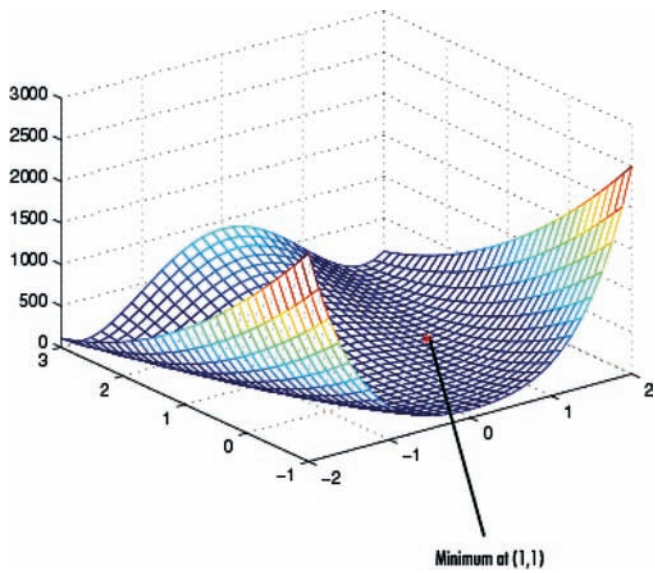


Figure 5: Two-variable Rosenbrock function.

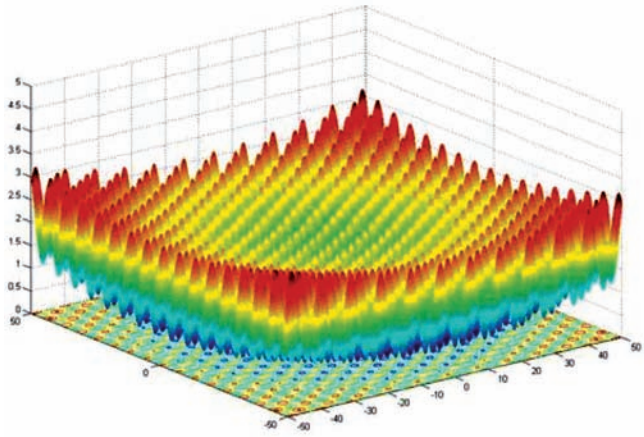


Figure 6: Two-variable Griewangk function.

this minimum and obviously it is far more difficult when the function expands in 30 dimensions and consists of 30 variables.

$$f(x) = \sum_{n=1}^{N-1} [100(x_{n+1} - x_n^2)^2 + (1 - x_n)^2] \tag{9}$$

Minimum $f(1, 1) = 0$,
Search domain: $-2.048 \le x_i \le 2.048$,
$n = 1, 2, \ldots, 30$
Number of local minima: several local minima

4.1.2 Griewangk function

Griewangk function (also known as Griewank function) has also been widely used to test the convergence of optimization functions (Fig. 6).

$$f(x) = 1 + \sum_{n=1}^{N} \frac{x_n^2}{4000} - \prod_{n=1}^{N} \cos(x_n) \tag{10}$$

Minimum $f(0, \ldots, 0) = 0$
Search domain: $-512 \le x_i \le 512$
$n = 1, 2, \ldots, 30$
Number of local minima: several local minima

4.2 Calibration procedure

4.2.1 'Classic' HSA

In order to start an optimization process using HSA, the user needs to input values for the HMCR and PAR parameters and define the size of HM. Defining the exact

number for these parameters has always been difficult, although scientists have a general idea of the limits between PAR and HMCR should be kept.

The authors examined the performance of HSA for the whole range of these parameters [16]. We have performed several runs for HMCR ranging from 60% to 97% and PAR ranging between 3% and 15%. HMCR took seven different values (0.6, 0.7, 0.77, 0.85, 0.9, 0.95, and 0.97) and PAR took five different values for each HMCR value (0.03, 0.05, 0.07, 0.09, and 0.15). Each run was performed twice and the average was calculated. This aimed to improve the quality of the results and reduce the influence of randomly generated good or bad solutions. Thirty-five runs in total were executed, each run performed twice.

In order to examine the impact of HM size, these 35 runs were performed for different HM size, equal to 10, 30, and 50 (Tables 1 and 2).

After the first runs, we realized that HSA, as we expected, is able to locate the global optimum. Besides, in the recent past, HSA implementations succeeded in finding local optimum in similar test functions. However, we decided to limit each run at 25,000 iterations. The reason for this was that we did not want to offer the Algorithm countless iterations in order to locate optima, but we wanted to mark its

Table 1: HSA: HMCR, PAR & HM size values.

HMCR	PAR	HM Size
0.60		
0.70	0.03	
0.77	0.05	10
0.85	0.07	30
0.90	0.09	50
0.95	0.15	
0.97		

Table 2: IHSA & G-B HSA:HMCR and HM size values.

HMCR	HM Size
50	
60	
65	
70	10
75	30
80	50
90	
95	
97	
99	

performance for various ways of calibration. Thus, watching what the Algorithm could succeed in only 25,000 calculations provided us the information regarding the calibration which leads to a better performance.

4.2.2 Improved HSA and Global-Best HSA

As we have already mentioned, Improved HSA (IHSA) and Global-Best HSA (G-B HSA) use a dynamic PAR value. This value ranges between two constant values PAR_{min} and PAR_{max} ($0 < PAR < 1$) and increases linearly during the optimization. For this reason, there was no need to calibrate PAR; only HM size and HMCR were calibrated.

4.3 Results

4.3.1 'Classic' Harmony Search Algorithm

HSA in its original form showed its best performance for the calibration shown in Table 3. It is interesting to note that results deriving from Rosenbrock function agree with Geem's proposals (HM Size = 30, HMCR ≈ 95%, PAR ≈ 5%–7%) [9]. However, results deriving from Griewangk suggest a higher Memory size and a slightly higher PAR with a lower HMCR index.

Regarding the average performance of HSA if we exclude PAR index and we only adjust HMCR and HM size it resulted that for Rosenbrock function the best performance occurs for HM size = 10 units and HMCR = 85%, whereas for Griewangk the best performance comes for HM size = 50 units and HMCR = 77%. It is important to note that for this formation, the Algorithm derives results with the smallest standard deviation (σ). This means that for several runs (independently from PAR) HSA gives results of the same order of magnitude.

4.3.2 Improved HSA

Improved HSA showed a very good performance throughout the experiments showing in some occasions a better performance than 'classic' HSA. The calibration procedure showed that this variation works better for higher values of HM size, maximizing its performance for HM size = 50 for both tests.

Experiments have shown that IHSA performs better when HMCR is ranging between 90% and 97%.

The only exception occurred in Griewangk test and when the HM size became relatively high.

Table 3: Optimum calibration for 'classic' HAS.

	Rosenbrock	Griewangk
min value	2.05	12.39
HM Size	30	50
HMCR	90%	77%
PAR	9%	15%

4.3.3 Global-best Harmony Search Algorithm

Global-Best HSA has shown the best performance compared to the other two algorithms. And although the number of iterations was limited to 25,000, G-B HSA managed to locate the global optimum of the Griewangk test-function for every HM size formation. Its performance in Rosenbrock's function was not as good, but still G-B HSA provided equal results compared to the other methods.

G-B HSA seemed to work better when HMCR was given relatively high values. However, for lower values of HM size, the HSA was performing better when HMCR was ranging between 50% and 65%. Figures 7 and 8 illustrate the performance of the three Harmony-based Algorithms in Griewangk and Rosenbrock tests.

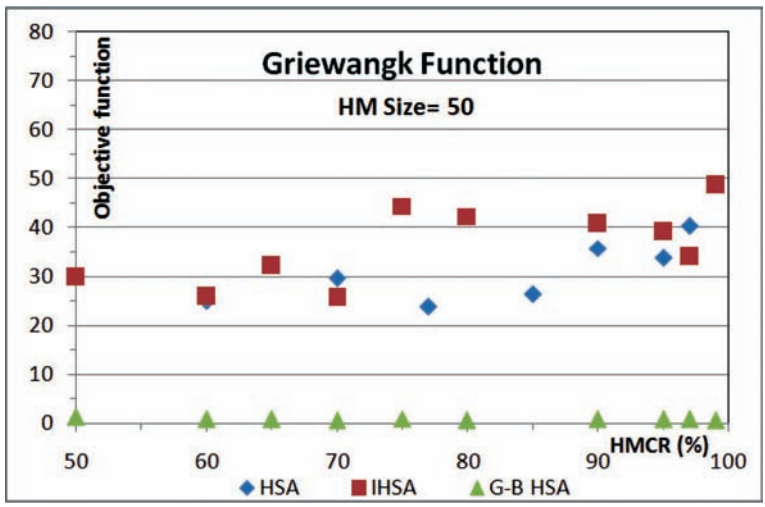


Figure 7. Performance of the 3 methods in Griewangk test.

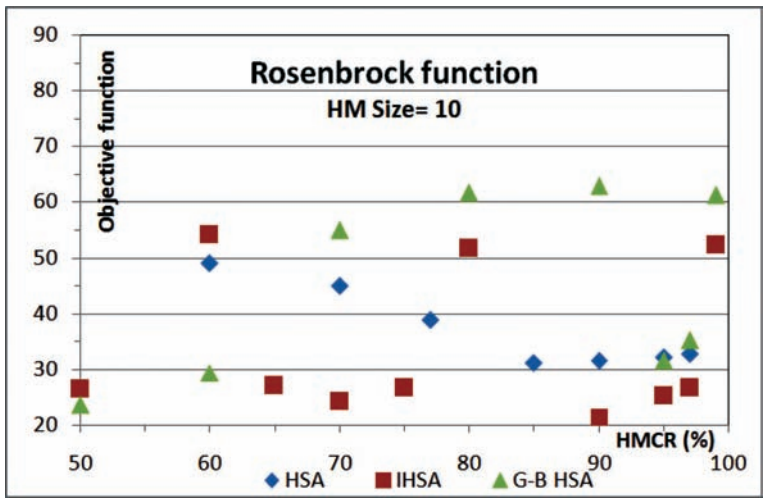


Figure 8. Performance of the 3 methods in Rosenbrock test.

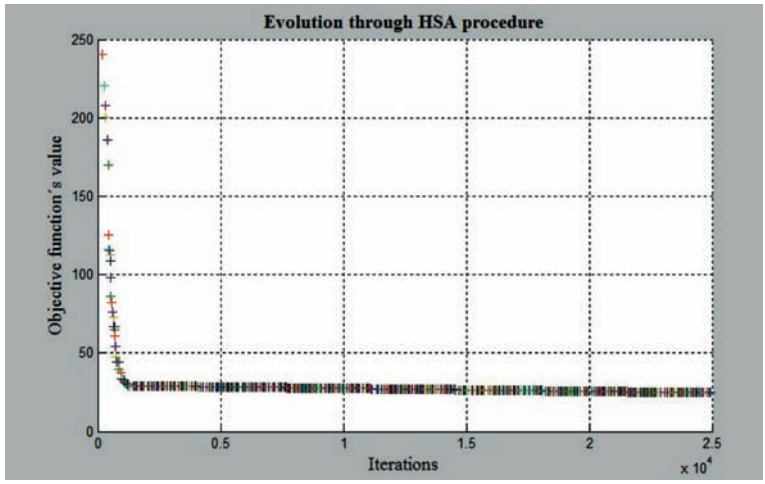


Figure 9. Typical Run of HSA in Griewangk test – function (HMSize = 50, HMCR = 95%, PAR= 7%) [16].

Generally, it is very important to note that for all three Algorithms, the performance improved as the size of Harmony Memory was increasing. Thus, in both test-functions the classic HSA and its variations approached closer to the global optimum for HM size = 50. This can be explained by the fact that a larger Harmony Memory entails richer 'music material.' In other words, as HM size increases, more possible solutions are included in it. These solutions not only may be an optimum option, but they may lead to the production of even better solutions through Memory Consideration, Pitch Adjustment, and Improvisation. However, further increase of HM size showed no further improvement on the performance. On the contrary HSA showed a difficulty to converge to optimum solutions, oscillating between local optima.

5 Application to a multireservoir system

5.1 Description of the problem

In this section, the optimization of the dam system of Fig. 10 is presented. This problem was first introduced in 1974 by Ven Te Chow and Gonzalo Cortes-Rivera [17]. It was designed as a demanding application to be optimized with Discrete Differential Dynamic Programming (DDDP). Its main difference, compared to conventional problems, is that inflows and constraints are not integer. As a result, the solution space is considerably wider and convergence to optimum management becomes a challenging task.

The optimization of this particular problem with the use of a stochastic algorithm, and more specifically the Harmony Search Algorithm, was first presented by Kougias and Theodossiou [18].

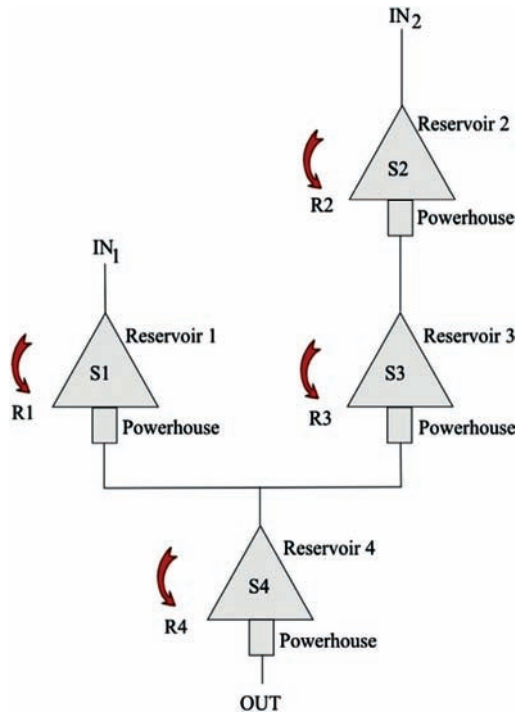


Figure 10. The multi-reservoir system.

In this system, benefits derive from the use of water toward irrigation and hydropower generation. The aim is to determine the optimum operation of the four-reservoir system over a period of 12 time steps.

5.2 Problem's constraints

The Inflow parameters IN_1 and IN_2 that flow in Reservoirs 1 and 2 in each of the 12 time steps are given in Table 4. The initial and final states of the reservoirs are identical and are required to be:

$$S_1(0) = S_2(0) = S_3(0) = S_1(12) = S_2(12) = S_3(12) = 6 \quad (11)$$

$$S_4(0) = S_4(12) = 8_{\text{water units}} \quad (12)$$

The releases from the four dams through turbines are as follows (values in water units):

$$0.005 \le R_1 \le 4.0 \quad (3)$$

$$0.005 \le R_2 \le 4.5 \quad (4)$$

$$0.005 \le R_3 \le 4.5 \quad (5)$$

$$0.005 \le R_4 \le 8.0 \quad (6)$$

Table 4: Inflows in Reservoirs 1 and 2.

Time step	IN_1	IN_2
1	0.5	0.4
2	1.0	0.7
3	2.0	2.0
4	3.0	2.0
5	3.5	4.0
6	2.5	3.5
7	2.0	3.0
8	1.25	2.5
9	1.25	1.3
10	0.75	1.2
11	1.75	1.0
12	1.0	0.7

It can be easily understood that this minor change in the constraints is causing a huge impact in the size of the search space. In the 'classic problem' R1 would range between $0 \leq R1 \leq 4$, taking one of the five possible integer values. However, with this change, R1 can take thousands different possible values (decimal precision of third digit). If we consider that this adjustment applies also to R2, R3, R4, it can be safely said that the search space is becoming incomparably wider.

This adjustment has an even stronger effect in stochastic methods. These techniques detect good solutions scanning the search space. As a result any significant increase of the space's size will raise difficulties. Moreover, a vast search space usually includes various local optima. Thus, misleading is a strong possibility and convergence to global optimum solutions might be infeasible.

Table 5 shows the maximum permissible storage in the four reservoirs during the time period of the 12-time steps. The minimum permissible storage is equal to 1. This minimum storage is common for all four reservoirs and it is constant during the function of the system

As a result, the equations that express the water stored in each reservoir throughout the 12 time-steps are:

$$S_1(t+1) = S_1(t) - R_1(t) + IN_1(t) \tag{13}$$

$$S_2(t+1) = S_2(t) - R_2(t) + IN_2(t) \tag{14}$$

$$S_3(t+1) = S_3(t) - R_3(t) + R_2(t) \tag{15}$$

$$S_4(t+1) = S_4(t) - R_4(t) + R_1(t) + R_3(t) \tag{16}$$

The Benefit criterion considers the use of water for power generation and irrigation. Reservoirs 1, 2, and 3 generate electricity through turbines, while water from Reservoir 4 is used for hydropower generation and irrigation. The objective function to be maximized is

$$\text{maximize: } \sum_{i=1}^{4} \sum_{t=0}^{11} b_i(t) \times R_i(t) \tag{17}$$

Table 6 summarizes the benefit coefficients (b_i, $i = 1$–4) included in eqn (17).

Table 5: Maximum permissible storage.

Time step	Dam 1	Dam 2	Dam 3	Dam 4
1	12.0	17.0	8.0	15.0
2	12.0	15.0	8.0	15.0
3	10.0	15.0	8.0	15.0
4	9.0	15.0	8.0	15.0
5	8.0	12.0	8.0	15.0
6	8.0	12.0	8.0	15.0
7	9.0	15.0	8.0	15.0
8	10.0	17.0	8.0	15.0
9	10.0	18.0	8.0	15.0
10	12.0	18.0	8.0	15.0
11	12.0	18.0	8.0	15.0

Table 6: Benefit coefficients.

Time steps	b_1	b_2	b_3	b_4
1	1.1	1.4	1.0	2.6
2	1.0	1.1	1.0	2.9
3	1.0	1.0	1.2	3.6
4	1.2	1.0	1.8	4.4
5	1.8	1.2	2.5	4.2
6	2.5	1.8	2.2	4.0
7	2.2	2.5	2.0	3.8
8	2.0	2.2	1.8	4.1
9	1.8	2.0	2.2	3.6
10	2.2	1.8	1.8	3.1
11	1.8	2.2	1.4	2.7
12	1.4	1.8	1.1	2.5

5.3 Results

The multiple reservoir system was optimized by Kougias and Theodossiou [18] with HSA using Matlab. A program was designed and created, providing some interested results. As it is already mentioned this was the first successful optimization of this particular problem with a stochastic technique. This is due to the continuous nature of the decision variables. This distinctiveness makes this problem really demanding for stochastic techniques, because the solution search space is vast. Figure 11 shows a typical run of the program.

Chow and Cortes Rivera reported that the linear programming solution converged to a maximum benefit equal to 308.2665 [17]. Soon, Murray and Yakowitz presented a solution using Differential Dynamic Programming (DDP) which converged to 308.234 [19]. The same authors also used Discrete Differential Dynamic Programming which resulted to a convergence to 307.98 [19].

HSA converged to a maximum of 307.218. This solution represents the 99.66% of the best known solution. Thus, it is safe to say that HSA concluded to very satisfying results.

This convergence was the best result of few runs. An extra advantage was that there was no need for calibration of HSA's parameters throughout the solution. The values of the parameters proposed in the HSA literature, produced the best results.

However, the number of iterations needed was considerably higher than the iterations in simpler problems. This resulted to a slow convergence to the optimum solution. Convergence to the global-optimal solution in the integer problem

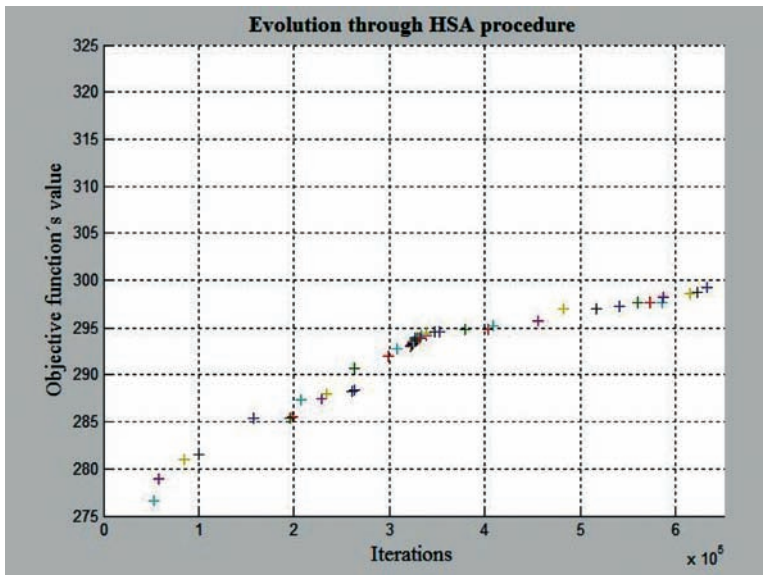


Figure 11: Gradual evolution of the objective function.

Table 7: Optimum water release.

Time step	R_1	R_2	R_3	R_4
1	1	1.295	0.005	0.005
2	0.005	0.005	0.005	0.005
3	0.005	0.22	0.005	0.855
4	2.99	0.05	0.8	8
5	4	1.62	4.5	8
6	4	3.56	4.25	8
7	4	4.5	4.5	8
8	2.485	4.5	2.77	8
9	0.01	1.3	4.5	1.92
10	1.995	0.63	0.93	0.005
11	0.005	4.41	0.03	0.005
12	0.005	0.21	0.005	0.005

Table 8: Water storage for the optimum solution.

Time step	S_1	S_2	S_3	S_4
	6	6	6	8
1	5.5	5.105	7.29	9
2	6.495	5.8	7.29	9.005
3	8.49	7.58	7.505	8.16
4	8.5	9.53	6.755	3.95
5	8	11.91	3.875	4.45
6	6.5	11.85	3.185	4.7
7	4.5	10.35	3.185	5.2
8	3.265	8.35	4.915	2.455
9	4.505	8.35	1.715	5.045
10	3.26	8.92	1.415	7.965
11	5.005	5.51	5.795	7.995
12	6	6	6	8

appeared only after ≈30.000 iterations. On the contrary, in the present application, the program converged after several hundred thousand iterations. Considering that the total search space comprises ≈1012 possible solutions, Harmony Search Algorithm converged to an optimum management only after searching 2% of the total space. These results show that HSA is a robust optimization tool, even for complicated continuous problems.

Table 7 shows the optimum water release for each dam, as it resulted from HSA. In Table 8, the Water storage of the reservoirs is presented. These storages correspond to the optimum solution and along with the releases of Table 7, show that all constraints are satisfied by the optimum solution.

6 Conclusions

Harmony Search Algorithm has already attracted the scientific interest. HSA has shown such a performance and an ability to locate optimal solutions, which led to its quick spread and evolution. Moreover, its simple structure not only makes the creation of HSA programs easier, but concludes to a faster performance and less computing time. Calibrating HSA's parameters may be of crucial importance toward the solution of complex problems. For this reason further research and experiments need to be made.

References

[1] Geem, Z.W., Optimal design of water distribution networks using harmony search. PhD Thesis, Korea University, Seoul, Korea, 2000.

[2] Geem, Z.W. & Kim, J.H., Loganathan G.V., Harmony search optimization: Application to pipe network design. *International Journal of Modeling and Simulation*, **22(2)**, pp. 125–133, 2002.

[3] Ingram, G. & Zhang, T., Overview of applications and developments in the harmony search algorithm, In: Geem, Z.W. *Music-Inspired Harmony Search Algorithm*, Springer, Berlin, SCI191, pp. 15–37, 2009.

[4] Kougias, I. & Theodossiou, N., A new music-inspired harmony based optimization algorithm. Theory and applications, In: *International Conference on Protection and Restoration of the Environment X*, Corfu, Greece, 2010.

[5] Ayvaz, M.T., Application of Harmony Search algorithm to the solution of groundwater management models. *Advances in Water Resources*, **32**, pp. 916–924, 2009.

[6] Kougias, I. & Theodossiou, N., A new music-inspired harmony based optimization algorithm. Application in water resources management problems, In: *International Conference: Protection and Restoration of the Environment X*, Corfu, Greece, 2010.

[7] Geem, Z.W., Kim, J.H. & Loganathan, G.V., A new heuristic optimization algorithm: Harmony search. *Simulation*, **76(2)**, pp. 60–68, 2001.

[8] Yang, X.S., Harmony search as a metaheuristic algorithm. In: Geem Z.W. *Music-Inspired Harmony Search Algorithm*, Springer, Berlin, SCI191, pp. 1–14, 2008.

[9] Geem, Z.W., *Optimal Scheduling of Multiple Dam System Using Harmony Search Algorithm*, eds. F. Sandoval et al, IWANN 2007, LNCS 4507, pp. 316–323, 2007.

[10] Geem, Z.W., *Improved Harmony Search from Ensemble of Music Players*, Springer-Verlag, Berlin, KES 2006, Part I, LNAI 4251, pp. 86–93, 2006.

[11] Mahdavi, M., Fesanghary, M. & Damangir, E., An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, **188**, pp. 1567–1579, 2007.

[12] Omran, M.G.H., Mahdavi, M., Global-best harmony search. *Applied Mathematics and Computation*, **198**, pp. 643–656, 2008.

[13] Wang, C.M. & Huang, Y.F., Self-adaptive harmony search algorithm for optimization, *Expert Systems with Applications*, **37**, pp. 2826–2837, 2010.
[14] Geem, Z.W. & Cho, Y.H., Optimal design of water distribution networks using parameter-setting-free harmony search for two major parameters. *Journal of Water Resources Planning and Management*, **137(4)**, 2011, pp. 377–380, 2011.
[15] Luke, S., Essentials of Metaheuristics. Available at: http://cs.gmu.edu/~sean/book/metaheuristics/, 2009.
[16] Theodossiou, N., Kougias, I., Kakoudakis, K. & Doikos, K. Harmony Search Algorithm, a novel optimization technique. Parameter calibration and applications on water resources management problems. In: *Proceedings of 3rd International Conference on Environmental Management, Engineering, Planning and Economics (CEMEPE 2011) & SECOTOX Conference*, Skiathos, Greece, 2011.
[17] Chow, V.T & Cortes, R., *Applications of DDDP in Water Resources Planning*, Research Report 78, University of Illinois, Water Resources Center, Urbana, 1974.
[18] Kougias, I. & Theodossiou, N., Optimization of multi-reservoir management using Harmony Search Algorithm (HSA). In: *Proceedings of 3rd International Conference on Environmental Management, Engineering, Planning and Economics (CEMEPE 2011) & SECOTOX Conference*, Skiathos, Greece, 2011.
[19] Murray, D.M. & Yakowitz, S., Constrained differential dynamic programming and its application to multireservoir control, *Water Resources Research*, **15(5)**, pp. 1017–1027, 1979.

CHAPTER 8

Groundwater optimal management using an outer approximation approach

George P. Karatzas
Department of Environmental Engineering, Technical University of Crete, Polytechneioupolis, 73100 Chania, Greece.

Abstract

In this chapter, a nonclassical global optimization method, the outer approximation method, is presented which is combined with groundwater simulators for the solution of groundwater management problems. In the first part, the mathematics of the outer approximation is presented. The main characteristic of the outer approximation method is the minimization of a concave objective function over a convex or concave feasible domain. The second part of the present chapter includes the application of the outer approximation method to groundwater optimal design problems.
Keywords: groundwater management, optimization, outer approximation.

1 Introduction

In the early 1950s, numerical simulation models were presented in an attempt to obtain solutions for large problems of oil reservoirs. This effort was later expanded to address subsurface management problems related to groundwater 'quality' and 'quantity.' These models couple and solve simultaneously the governing equations of groundwater flow, mass transport, and chemical reaction.

The main objective of such models is to 'predict' the groundwater movement and the contaminant transport through a subsurface system. In many cases, these problems are very complicated and difficult to describe mathematically. The accurate description of the physical system, necessary for an optimal prediction to be obtained, requires a large volume of data. In addition, analytical solutions cannot be used due to the inhomogeneities, irregularities, and uncertainties of the physical system. Therefore, numerical simulation must be employed. In the past, several

numerical simulator models of groundwater flow and transport have been presented based on the theories of finite elements and finite differences (Sutra, Modflow and MD3d, Princeton Transport Code (PTC), FEMWATER, etc).

In the early 1970s, groundwater numerical simulation models (Deninger [1]) were combined with optimization techniques and became a powerful tool for solving groundwater management problems. The motivation of the attempt to combine simulation with optimization was the desire to determine the 'best' solution (among several feasible solutions) that could be applied to a groundwater management problem.

As indicated by Gorelick *et al*. [2], simulation models are often inadequate because the problems of aquifer management do not involve prediction alone. Rather, they involve both simulation, for prediction, and optimization. The role of optimization is to determine the best operating policy for a particular objective, taking into account the restrictions that exist on a site-specific basis.

The combination of groundwater simulation and optimization techniques can be thought of as organized and methodical trial-and-error methods. However, in contrast to most trial-and-error approaches, the objective, constraints, and solution search strategies are clearly specified. An optimization problem is formulated mathematically as a problem that minimizes or maximizes an objective function subject to a set of constraints that are based on physical, economic, technical, or social restrictions.

2 Example

An optimal design is desired for the following groundwater management problem: an aquifer is the main source of water supply for the city of Waterton. A minimum amount of water supply is required to cover the daily needs of the city. There exists a limit to the maximum amount of water that the aquifer can yield on a daily basis. This restriction is imposed by the fact that the water level in the aquifer has to remain above a certain threshold to ensure substantial groundwater flow. In addition, the pumping stations have a maximum potential for water extraction depending on the available pumps. The engineering design is to determine the number of wells that should be installed, the location of the wells, and the amount of water that should be pumped from each well so that all the above requirements are satisfied and the entire management process is accomplished with the minimum possible cost.

This problem can be formulated as follows:

MINIMIZE the total pumping cost

Subject to

- The amount of water (total pumping) being greater than or equal to the minimum amount required for the city's needs.
- The pumping action not lowering the water level below a certain threshold.
- The pumping rates at each station not exceeding the maximum specified value for each pump.

Let us consider now the mathematical formulation of the problem. The variable parameters involved in the formulation are identified from the problem description presented above. Such are the pumping rates and the water level in the aquifer (hydraulic head). These parameters are distinguished in two categories: (i) the parameters of the decision making process, called *decision variables* and (ii) the parameters related to the physical system, called *state variables*. The pumping rates are decision variables and the hydraulic heads are state variables.

How do we approach the problem as a groundwater management problem where simulation is combined with optimization?

A groundwater simulator is developed first. This requires defining the area of interest and collecting all the data related to the parameters of the physical system. Topographic, meteorological, geological, and surface condition information must be collected for the accurate description of the system. More specifically, this information includes:

- data related to the geological parameters such as porosity, hydraulic conductivity, aquifer's depth, storativity, etc. (system parameters),
- initial and boundary conditions
  - initial water level, topographic conditions, meteorological conditions (e.g. rainfall depth),
  - conditions related to the interaction of surface water and groundwater (existing lakes, rivers, ponds, and reservoirs).

The development of the numerical simulator is followed by the calibration of the model to ensure that the model provides an accurate representation of the existing field conditions.

Next, the groundwater numerical simulator is combined with the optimization model. The objective here is to establish forward and backward feedback between the optimization model and the numerical simulator. The optimization selects the decision variables, passes the decision variables to the numerical simulator, and performs the evaluation of the constraints. The optimal solution determines the values of the decision variables (pumping rates) for which the objective function is minimized and all the constraints are satisfied.

The decision and state variables are defined as follows. The simulator is used to represent the physical system by a *finite difference* or *finite element* mesh. The numerical simulator can be either two-dimensional (Fig. 1) or three-dimensional (Fig. 2).

The location of the decision variables can be assigned at any mesh node by the decision maker. Several nodes can be considered as potential pumping well locations. Let $I = \{1, \ldots, n\}$ be the set of selected potential pumping well locations. Let q_i denote the pumping rate assigned to node i of the mesh, $i \in I$. Then, the vector of decision variables $\mathbf{q}$ is defined as $\mathbf{q} = [q_1, \ldots, q_n]$. By proposing several potential pumping well locations, the optimization model will select the 'best' locations to be activated, that is, those locations where the value of the pumping rate is greater than zero.

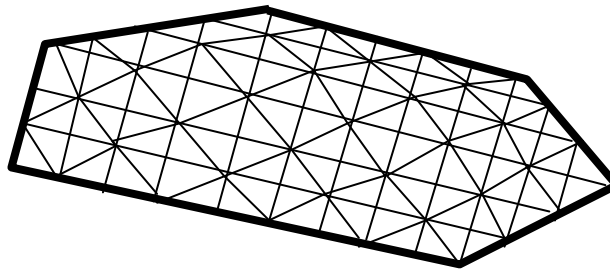


Figure 1: A two-dimensional (2D) mesh grid.

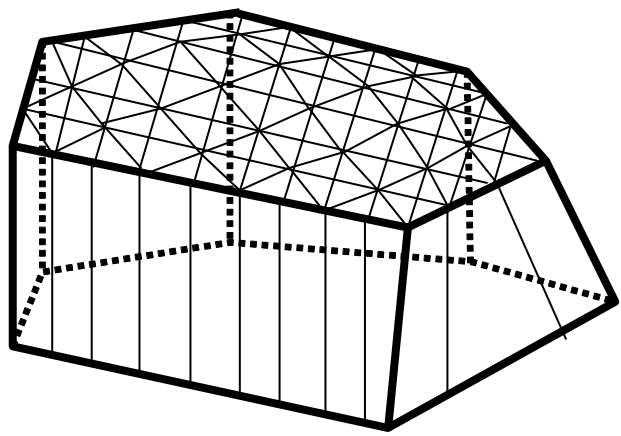


Figure 2: A three-dimensional (3D) mesh grid.

Based on the problem constraints, the sum of all q_i's should be greater than or equal to the town minimum daily water supply needs, Q^*. This is the first constraint of the optimization problem.

Finally, there is the restriction of the upper bounds at each pumping station. According to this constraint, the pumping rate q_i at any pumping location cannot exceed a prespecified upper limit q_i^*.

The objective function is the *total pumping cost* per unit time and is expressed as the sum of the products of the unit cost coefficient α_i (\$/L^3) times the pumping rate q_i (L^3/T) over all pumping well locations.

A selected solution is declared *optimal* if it achieves the minimum objective function value and satisfies all the aforementioned constraints. This is mathematically expressed as follows:

$$\min \quad f(\mathbf{q}) = \sum_{i=1}^{N} \alpha_i q_i$$

subject to:

$$\sum_{i=1}^{N} q_i \geq Q^* \quad i \in I$$

$$h_j \geq h_j^* \quad \forall j \in J$$
$$q_i \leq q_i^* \quad \forall i \in I$$

where all the symbols are as previously defined.

The optimization and numerical simulation are combined as follows. The optimization model selects the pumping combination that minimizes the objective function (total pumping cost). The information is passed to the numerical simulator and the model is executed. The output of this simulation consists of the values of the hydraulic heads (one for each node). The values for the specific observation locations are forwarded to the optimization algorithm. The optimization algorithm, in turn, evaluates the constraints (hydraulic heads and upper pumping bounds) and an optimal solution is declared if all constraints are satisfied. If any of the constraints is violated, then the algorithm selects a new combination of pumping rates and the process is repeated. The Waterton example is a typical case of groundwater 'quantity' management.

The same procedure can be applied in the case of groundwater 'quality' management problems. In this case, the numerical simulator couples the groundwater flow equation and the mass transport equation of the contaminant. The starting point in 'quality' management problems is the determination of the location and the perimeter of the contaminant plume. This will determine the initial conditions of the contaminant concentration field at the beginning of the remediation period required by the simulator. A concentration value is assigned at each node of the mesh. Similar to the quantity management problem, certain nodes are considered as observation points for concentration. The optimization algorithm evaluates the concentration at the observation points c_j, $j \in J$, and ensures that their values remain below the pre-specified (cutoff) level.

The mathematical formulation of the problem is:

$$\min \quad f(\mathbf{q}) = \sum_{i=1}^{N} a_i q_i$$

subject to:

$$\sum_{i=1}^{N} q_i \geq Q^* \quad i \in I$$
$$c_j \leq c_j^* \ \forall \ j \in J$$
$$q_i \leq q_i^* \ \forall \ i \in I$$

In some cases, it is also possible to have a mixed management problem where both hydraulic head and concentration constraints are implied. Such a formulation is:

$$\min \quad f(\mathbf{q}) = \sum_{i=1}^{N} a_i q_i$$

subject to:

$$\sum_{i=1}^{N} q_i \geq Q^* \quad i \in I$$

$$c_j \le c_j^* \ \forall\ j \in J$$
$$h_k \ge h_k^* \ \forall\ k \in K$$
$$q_i \le q_i^* \quad \forall\ i \in I$$

In some cases, a groundwater quality management problem is approached as a 'contaminant plume containment problem.' In this case, a hydraulic gradient control method is applied to prevent the contaminant plume of moving toward a certain direction. The concept of this is to create a hydraulic gradient around the contaminant plume (*capture zone*) and ensure that the groundwater flow is toward this zone (Fig. 3).

Groundwater should flow inward the line that defines the capture zone. The constraints are imposed on certain points along this line as shown in Fig. 3. It is required that at any point *j* of the line the hydraulic head at location 2 be greater than that at location 1 so that no flow crosses the line in the downstream direction.

The remediation design includes several potential well locations that can be either extraction wells inside the capture zone or injection wells outside the capture zone. The role of an extraction well is to reduce the hydraulic head by pumping. The opposite applies to an injection well. The mathematical formulation of this optimization problem is:

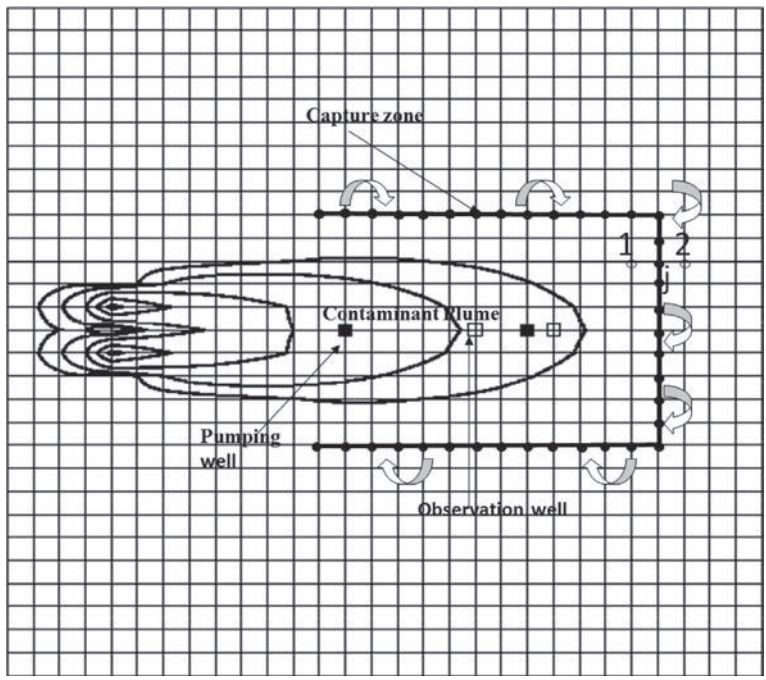


Figure 3: The hydraulic gradient control approach.

$$\min \quad f(\mathbf{q}) = \sum_{i=1}^{N} a_i q_i$$

subject to:

$$\sum_{i=1}^{N} q_i \geq Q^* \quad i \in I$$

$$h_j^2 - h_j^1 \geq b_j^* \quad \forall\, j \in J$$

$$q_i \leq q_i^* \qquad \forall\, i \in I$$

where b_j^* denotes the prespecified hydraulic head difference at location j.

It should be noticed at this point that the first constraint in many cases is not included in the formulation of the optimization problem. This constraint is necessary in cases where the amount of the extracted groundwater has to satisfy the community's water demands.

3 The mathematics of groundwater management problems

Groundwater management problems can be very simple or very complicated. This depends on the formulation of the problem. A groundwater management problem with a linear objective function and a linear set of constraints is characterized as a simple problem and is relatively easy to solve. Problems where the decision variables do not appear in any power and/or in a product form have linear behavior. The geometric representation of a linear objective function or constraint is a straight line for problems with one decision variable (one-dimensional problems, 1D), a plane for 2D problems, or a hyperplane for multidimensional problems (Fig. 4).

The most complicated form of groundwater management problems appears when either the objective function or any of the constraints are nonlinear. Typical examples for 1D and 2D problems are shown in Fig. 5.

In most practical cases, hydraulic head and hydraulic gradient constraints have linear behavior with respect to the pumping rate. If the objective function is also linear, the management problem is characterized as linear. Linear problems involve only flow equations in the numerical simulation. It should be noted that in order for the above constraints to exhibit linear behavior the aquifer has to be in the steady state (no changes with time). These problems can be solved using classical linear programming techniques (simplex method). Several software packages exist to solve problems in this category (Lindo, Minos, Modman).

Groundwater management problems that involve concentration constraints are nonlinear problems (since the mass transport equation has a nonlinear behavior) and are known as *groundwater quality management problems.* In this case, the objective function can be either linear or nonlinear. This kind of problems is more difficult to be solved due to the nonlinear behavior.

The degree of difficulty in solving nonlinear groundwater management problems using optimization techniques is mostly dependent on the behavior of the objective and constraint functions.

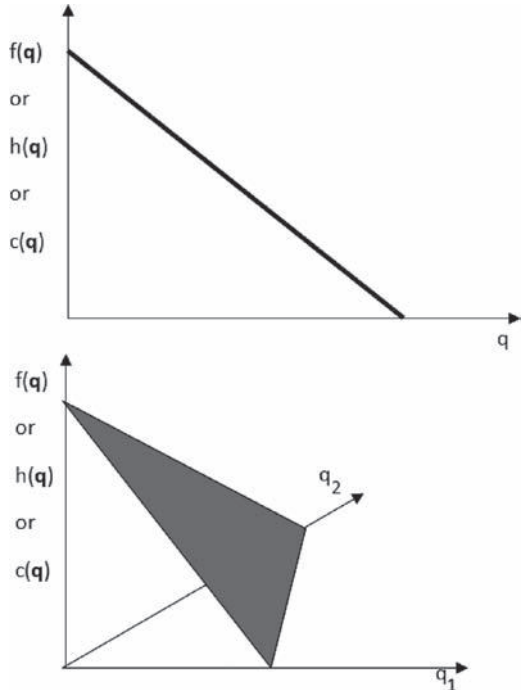


Figure 4: Linear behavior of a function for 1D and 2D problem.

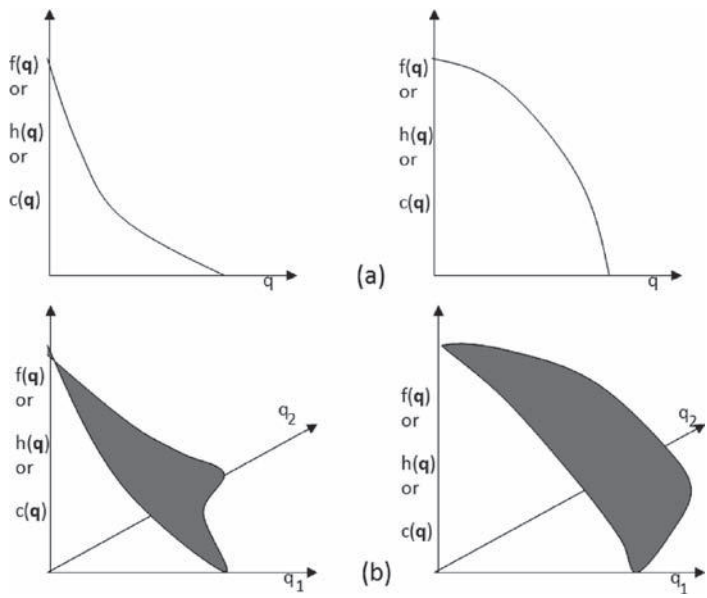


Figure 5: Nonlinear behavior of a function for 1D (a) and 2D (b) problems.

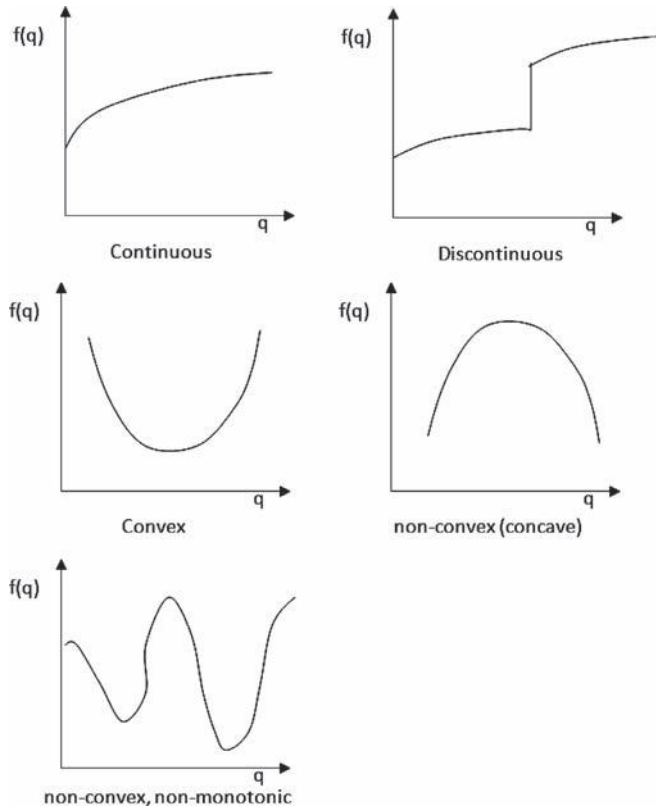


Figure 6: Typical behaviors of an objective function.

Regarding the objective function, the mathematical formulation can be either minimization (e.g. minimization of the total pumping cost) or maximization (e.g. maximization of the total pumping) of the function. The objective function can be linear or nonlinear. A nonlinear function can be continuous or discontinuous, convex or nonconvex (concave), monotonic or nonmonotonic. Some typical examples are presented in Fig. 6.

The most complicated case is the nonconvex, nonmonotonic function, where most of the optimization techniques have difficulties to determine the 'global optimal' and instead terminate the process at a local optimal. Figure 7 illustrates the concept of local and global optimal.

The constraint functions can also be linear or nonlinear, convex or nonconvex. The geometric representation of the constraint functions is illustrated in Fig. 8.

In case that the problem has several constraints the *feasible region* is defined as the intersection of all the constraints.

The objective function is imposed over the feasible region defined by the set of constraints. The optimal solution must be either inside the feasible region or along the perimeter of this region (Fig. 9).

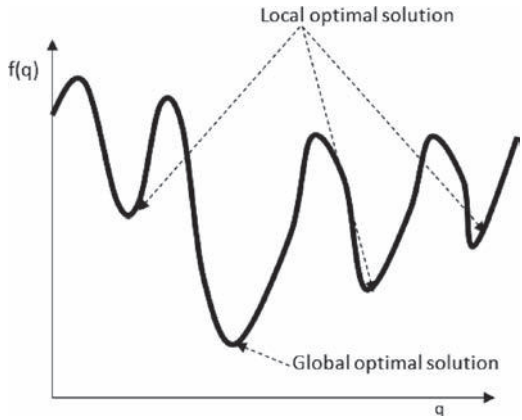


Figure 7: The concept of local and global optimal in a minimization problem.

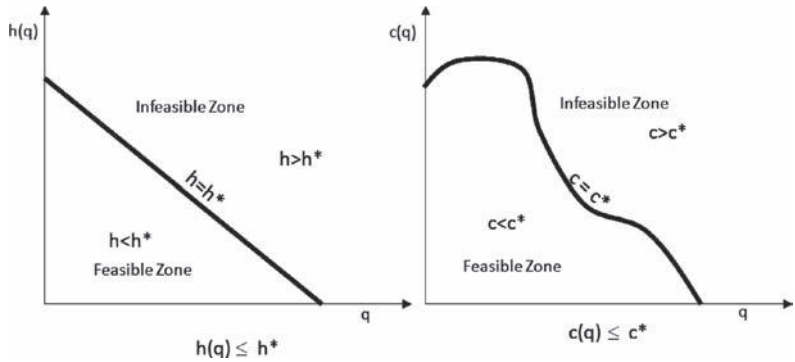


Figure 8: Geometrical representation of the constraints.

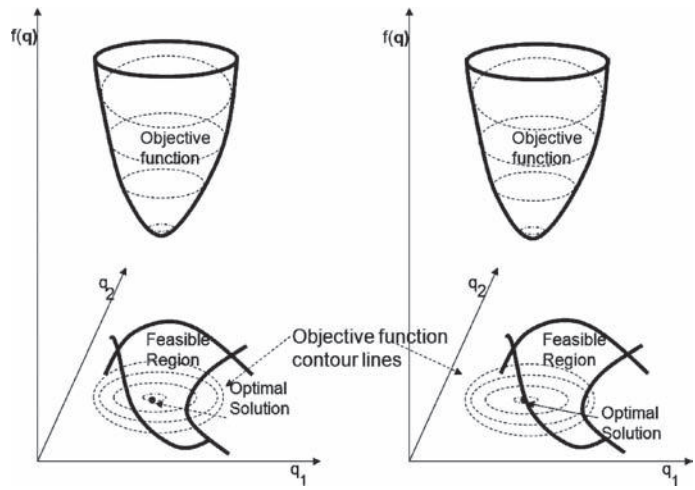


Figure 9: Geometric representation of the objective function, feasible region and optimal solution for a 2D minimization problem.

4 The outer approximation method

Based on the theory presented in the previous sections, we introduce the outer approximation method for the solution of water resources management problems.

The outer approximation method is applicable to problems formulated to minimize a concave objective function over a compact set of constraints. Methods for concave minimization include successive approximation methods, successive partition methods, and cutting plane methods. The category of successive approximations methods includes the outer approximation method or relaxation method, the inner approximation method, and the successive underestimation method (Horst and Tuy [3]). The concept of the outer approximation method, in the area of concave programming, was introduced by Tuy [4]. The method takes advantage of the basic property of a concave function *f* that the minimum of the function defined over a compact set of constraints always occurs at one of the most extreme points of the set (Fig. 10).

In the area of water resources, and specifically, for the solution of groundwater management problems, the outer approximation method was first introduced by Karatzas and Pinder [5] for convex feasible regions, followed by Karatzas and Pinder [6] for concave domains. New versions of the above work, with additional capabilities of the original methodology, were presented by Papadopoulou *et al*. [7] and Spiliotopoulos *et al*. [8].

The method utilizes a cutting hyperplane technique. The feasible region is initially approximated by an enclosing polytope defined by a set of points, the vertices. Among the vertices of that polytope, the one that minimizes the objective function is determined. If this vertex is feasible, then the optimal solution is found and, based on the properties of the objective function, the obtained optimal solution is a global optimal solution. Otherwise, an iterative procedure is applied. At each step of this procedure, a hyperplane is introduced, that 'cuts' through the polytope dividing it into two parts, such that only one contains the entire feasible

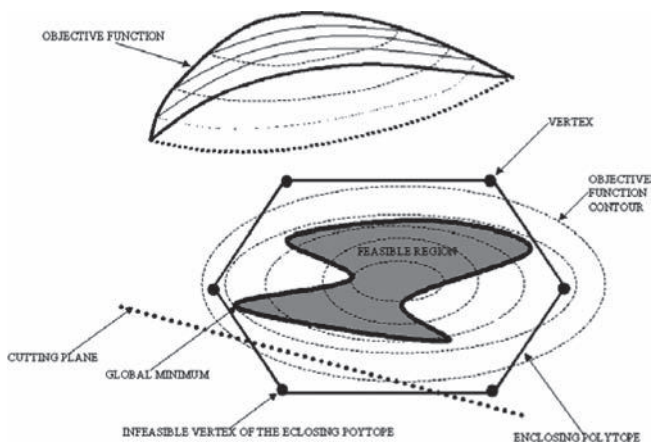


Figure 10: The concept of the outer approximation method.

region. The vertices of the polytope that contains the feasible region are examined as to which one minimizes the objective function and whether, that vertex, satisfies all the constraints. If all constrains are not satisfied, the process is repeated (Karatzas and Pinder [5] and [6], Spiliotopoulos [9]).

The problem is defined as follows:

$$\text{minimize } f(\mathbf{x})$$

such that:

$$g_i(\mathbf{x}) \leq \mathbf{x}^* \quad i=1,2,...,m$$
$$\mathbf{x} \geq 0$$

where:

$f : R^n \rightarrow R$ is a real-valued concave function defined in R^n

$g_i(\mathbf{x})$ are real-valued functions with convex or nonconvex behavior.

If all the functions $g_i(\mathbf{x})$ are convex, then the generated feasible region, D, is a closed convex set. Otherwise, the set is nonconvex. The outer approximation method that determines the global optimal solution can be outlined as follows (Karatzas and Pinder [5] and [6]):

1. Define an initial enclosing polytope, D_1. The polytope is determined by a set of vertices such that the feasible region of the problem is enclosed. (Figure 11) shows an example of an enclosing polytope defined by its vertices (v^1;v^2;v^3;v^4).
2. Determine the vertex $\mathbf{x}^k$ that minimizes the objective function.
3. If at the vertex $\mathbf{x}^k$ all the constraints are satisfied, the optimal solution is determined.

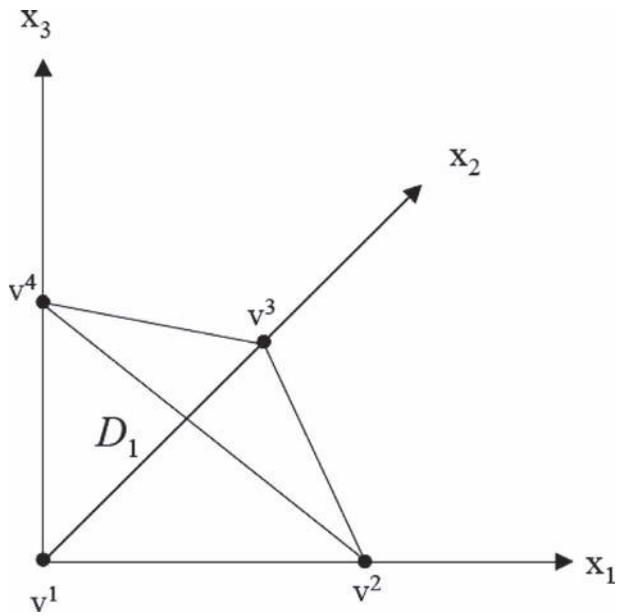


Figure 11: Enclosing polytope – outer approximation method.

Otherwise, find the most violated constraint $g^k(\mathbf{x})$, where

$$g^k(\mathbf{x}) = \max\{g_i^k(\mathbf{x})\} > 0, i = 1,2,\ldots,m$$

4. Determine the behavior of the most violated constraint;
   a. if the constraint is convex, introduce the cutting hyperplane as follows: for problems with linear constraints, the cutting hyperplane is the most violated constraint itself:

$$h^k(\mathbf{x}) = g^k(\mathbf{x})$$

For problems with nonlinear convex constraints a 'linearized' form of the most violated constraint is introduced to play the role of the cutting hyperplane, expressed as follows:

$$h^k(\mathbf{x}) = \frac{\partial g^k(\mathbf{x})}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}^k) + g^k(\mathbf{x})$$

This hyperplane is tangent to the constraint at the minimizing vertex. For the simple case of a one-dimensional problem, the tangent plane is shown in Fig. 12.

   b. if the constraint is concave, a tangent hyperplane would cut part of the feasible region. A modified procedure to determine the cutting hyperplane is introduced.

The edges of the enclosing polytope that connect the minimizing vertex to its adjacent ones are identified. Along each edge, the points $\mathbf{z}^i$ are defined such that:

$$\mathbf{z}^i = \mathbf{x}^k + \theta_i(\mathbf{y}^i - \mathbf{x}^k),\ \mathbf{z}^i \neq \mathbf{x}^k$$

and

$$g^k(\mathbf{z}^i) \geq 0, \quad i = 1,2,\ldots,n, \quad \mathbf{z}^i \neq \mathbf{x}^k$$

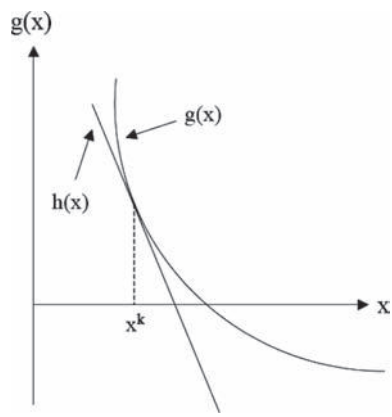


Figure 12: Cutting hyperplane for convex constraint.

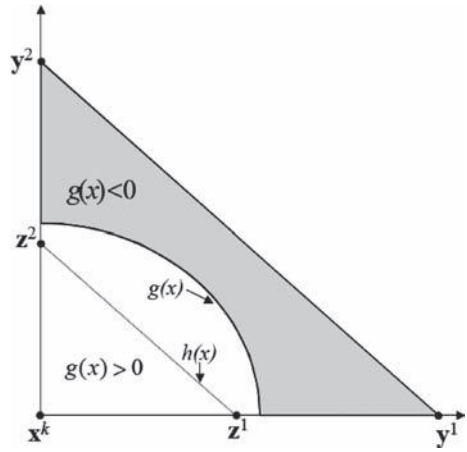


Figure 13: Cutting hyperplane for concave constraint.

where θ_i is a constant greater than zero and $\mathbf{y}^i$ is an adjacent vertex. In other words, the points $\mathbf{z}^i$ are defined as points along the edges of the enclosing polytope so that the constraint at those points is violated. The cutting hyperplane passes through these points (Fig. 13).

It is introduced using the following equation:

$$h^k(\mathbf{x}) = \mathbf{e}\mathbf{D}^{-1}(\mathbf{x} - \mathbf{x}^k) = 1 \qquad \text{(I)}$$

where **D** is the $n \times n$ matrix:

$$\mathbf{D} = [\mathbf{z}^1 - \mathbf{x}^k, \mathbf{z}^2 - \mathbf{x}^k, \ldots, \mathbf{z}^n - \mathbf{x}^k]$$

and **e** is a row vector of ones, i.e. **e** = (1, 1, …, 1) (Hillestand and Jacobsen [10]).

Example Consider the points:

$\mathbf{x}^k = (0,0)$, $\mathbf{z}^1 = (2,0)$, and $\mathbf{z}^2 = (0,3)$ (Fig. 14) (Spiliotopoulos [9]).

In this simple 2D problem, the cutting plane is the line segment that connects the points $\mathbf{z}^1$ and $\mathbf{z}^2$. Then,

$$\mathbf{D} = [\mathbf{z}^1 - \mathbf{x}^k, \mathbf{z}^2 - \mathbf{x}^k] = \begin{bmatrix} 2-0 & 0-0 \\ 0-0 & 3-0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

and

$$\mathbf{D}^{-1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.33 \end{bmatrix}$$

Any point **x** that belongs to the cutting hyperplane should satisfy eqn (I).

To verify that consider the point x = (1.33,1) which lies on the segment ($\mathbf{z}^1$,$\mathbf{z}^2$).

Equation I becomes

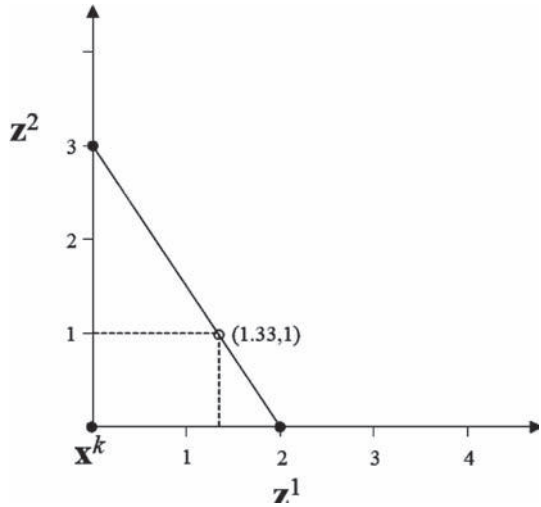


Figure 14: Example for cutting hyperplane for concave constraint.

$$\mathbf{e}\mathbf{D}^{-1}\left(\mathbf{x}-\mathbf{x}^{k}\right)=\begin{pmatrix}1 & 1\end{pmatrix}\begin{bmatrix}0.5 & 0\\ 0 & 0.33\end{bmatrix}\begin{pmatrix}1.33-0\\ 1-0\end{pmatrix}=1$$

5. Determine the sets of vertices for which $h^k(\mathbf{x})$ becomes positive and negative and reject the set for which it becomes positive. Determine the new vertices generated by introducing the cutting hyperplane. Repeat steps 2–5 (Spiliotopoulos [9]).

5 Example

A groundwater management problem for a hypothetical contaminated aquifer.

Consider a homogeneous, isotropic aquifer with dimensions 870 m × 870 m, hydraulic conductivity 0.125 m/h, longitudinal dispersivity 18 m, transverse dispersivity 1.8 m, diffusion coefficient 0.00001 m^2/h, and porosity 0.2. Using a finite-element scheme, the aquifer is represented by 900 nodes and 841 elements and a time step (Δt) equal to 2,928 h (4 months).

A 2D finite element model for areal flow and transport of a no decaying, adsorbing-desorbing contaminant in a groundwater aquifer was used for the representation of the above system (Bredehoeft and Pinder [11]; Ahlfeld [12]).

Two contaminant sources (indicated by the black triangles in Fig. 15) were polluting the aquifer for 15 years prior to remediation. Figure 15 shows the finite-element mesh, initial and boundary conditions, locations of the contaminant sources, normalized concentration contours at the end of the 15-year period, and the proposed remediation scenario.

The proposed remediation design was as follows: a clean water injection well is pumping at location W1 with the pumping rate upper bound equal to 9 m^3/h and

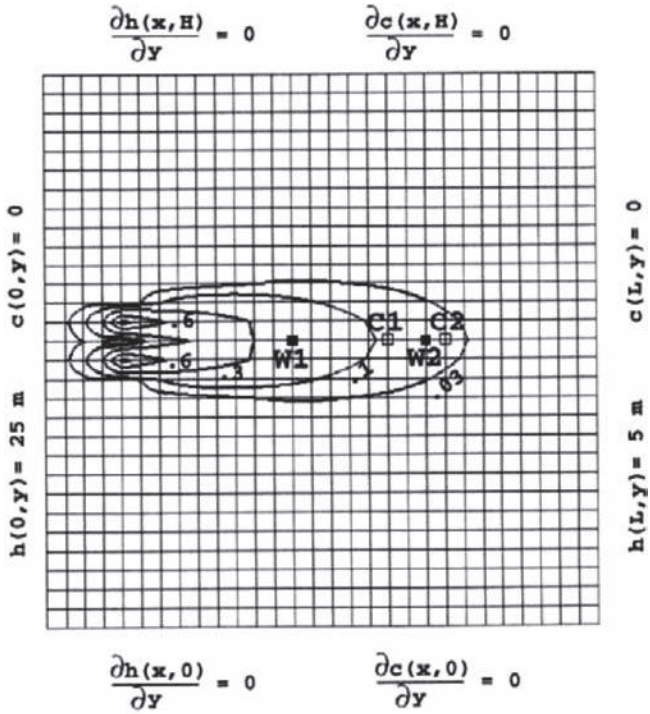


Figure 15: The mesh, the initial conditions and the remediation scheme for the example problem (Karatzas and Pinder [6]).

an extraction well is pumping at location W2 with the pumping rate upper bound equal to 20 m^3/h (pumping wells are indicated by black squares). It is requested that the normalized concentration at the end of a five-year remediation period be less than or equal to 0.09 at location C1 and less than or equal to 0.03 at location C2 (observation points are indicated by white squares). The objective is to determine the optimal solution with the minimum cost.

Figure 16 shows the behavior of the constraints for the above remediation scheme. The *x*-axis represents the injection-pumping rate at well W1 and the *y*-axis the extraction rate at well W2.

The two curves, indicated as constraints #1 and #2, represent the response of the concentration functions $c_i - c_i^* = 0$, $i = 1, 2,$ with respect to the pumping rates at observation locations C1 and C2, respectively. c_i^* is the target concentration at location *i* at the end of the remediation period.

As illustrated in Fig. 16, constraint #1 is a nonconvex function while constraint #2 is convex. The feasible region determined by the two constraints, and the two upper bounds of the pumping rates is indicated by the shaded area in Fig. 16.

The number of decision variables (well locations) determines the dimensionality of the optimization problem. In the present case, the feasible region (Fig. 16) is defined by the two concentration constraints and upper bounds of the decision variables (maximum pumping rates of the wells). The optimal solution was

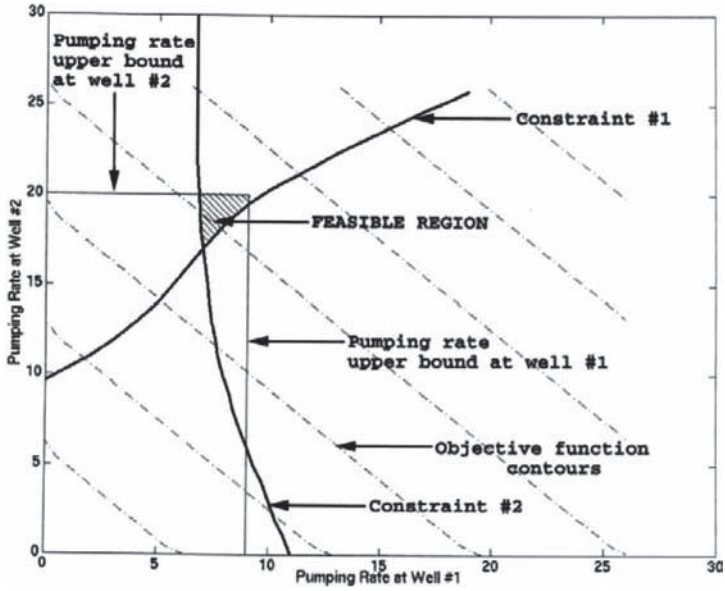


Figure 16: Representation of the constraints and the feasible region for the example problem (Karatzas and Pinder [6]).

obtained after twelve (12) optimization steps, which means 12 cutting planes were introduced to determine the optimal point (Karatzas and Pinder [6]). The optimal solution was obtained as follows:

Step = 1
X (1) = 0.000
X (2) = 0.000
No optimal solution

Step = 2
X (1) = 9.280
X (2) = 0.000
No optimal solution

Step = 3
X (1) = 9.000
X (2) = 1.312
No optimal solution

Step = 4
X (1) = 6.466
X (2) = 13.190
No optimal solution

Step = 5
X (1) = 9.000
X (2) = 11.162
No optimal solution

Step = 6
X (1) = 5.708
X (2) = 16.739
No optimal solution

Step = 7
X (1) = 6.762
X (2) = 16.362
No optimal solution

Step = 8
X (1) = 6.997
X (2) = 16.277
No optimal solution

Step = 9
X (1) = 6.897
X (2) = 16.602
No optimal solution

Step = 10
X (1) = 6.955
X (2) = 16.681
No optimal solution

Step = 11
X (1) = 6.945
X (2) = 16.713
No optimal solution

Step = 12
X (1) = 6.959
X (2) = 16.724
Optimal solution was found
objective value = 367250.0

At each step the **PTC** (Babu and Pinder [13], Babu *et al*. [14], Pinder [15]) was employed as the numerical simulation tool to evaluate the concentration levels based on the selected pumping rates from the optimization part. This information is returned to the optimization algorithm that determined if feasibility occurred.

Herein, the obtained optimal solution, for $X(1) = 6.959$ m^3/h and $X(2) = 16.724$ m^3/h, corresponds to the coordinators of the lower point of the feasible region (Fig. 16). Considering the graphical representation of the objective function behavior, as shown in Fig. 16, one can verify that the algorithm of the outer approximation accurately determines the global optimal solution of the problem.

6 Conclusions

The outer approximation method is a technique that determines a global optimal solution of a concave function over convex or nonconvex domains. In this chapter,

applications of the method were presented to groundwater management problems. Two significant advantages of this method are that the derivative of the objective function is not required, and at each optimization step (every time that a new hyperplane is introduced) only the derivatives of one constraint with respect to the decision variables are needed. In addition, the method easily handles linear constraints or combinations of linear and nonlinear constraints.

References

[1] Deininger, R.A., Systems analysis of water supply systems, *Water Resources Bulletin*, **6(4)**, pp. 573–579, 1970.

[2] Gorelick, S.M., Freeze, R.A., Donohue, D. & Keely, J.F., *Groundwater Contamination Optimal Capture and Containment*, Lewis Publishers, Boca Raton, FL, 1993.

[3] Horst, R. & Tuy, H., *Global Optimization*, Springer-Verlag, New York, 1990.

[4] Tuy, H., Concave programming under linear constraints. *Dokl. Akad. Nauk SSSR*, **159**, 32–35, 1964. (*Engl. Transl., Sov. Math.*, (Providence), 5, 1437–1440, 1964).

[5] Karatzas, G.P. & Pinder, G.F., Groundwater management using numerical simulation and the outer approximation method for global optimization. *Water Resources Research*, **29(10)**, pp. 3371–3378, 1993.

[6] Karatzas, G.P. & Pinder, G.F., The solution of groundwater quality management problems with a nonconvex feasible region using a cutting plane optimization technique. *Water Resources Research*, **32(4)**, pp. 1091–1100, 1996.

[7] Papadopoulou, M.P., Pinder, G.F. & Karatzas, G.P., Enhancement of the outer approximation method for solution of concentration-constrained optimal-design groundwater-remediation problem. *Water Resources Research*, **39(7)**, 1185–1194, 2003.

[8] Spiliotopoulos, A.A., Karatzas, G.P. & Pinder, G.F., A multiperiod approach to the solution of groundwater management problems using the outer approximation method. *European Journal of Operational Research*, **157**, pp. 514–525, 2004.

[9] Spiliotopoulos, A.A., *Solution of Groundwater Management Problems Using Concave and Biconcave Optimization Techniques*, PhD Thesis, Department of Civil and Environmental Engineering, University of Vermont, 1999.

[10] Hillestad, R.J. & Jacoben, S.E., Reverse convex programming, *Applied Mathematics* and *Optimization*, **6**, pp. 63–78, 1980.

[11] Bredehoeft, J.D. & Pinder, G.F., Mass transport in flowing groundwater. *Water Resources Research*, **9(1)**, 1973.

[12] Ahlfeld, D.P., Designing contaminated groundwater remediation systems using numerical simulation and nonlinear optimization, *Ph.D. Dissertation*, Department of Civil Engineering, Princeton University, NJ, 1987.

[13] Babu, D.K. & Pinder G.F., 1984. A finite element–finite difference alternating direction algorithm for 3-dimensional groundwater transport, *Advances in Water Resources*, **7(3)**, pp. 116–119.

[14] Babu, D.K., Pinder, G.F., Niemi, A., Ahlfeld, D.P. & Stothoff, S.A., 1997. Chemical transport by three-dimensional groundwater flows, Princeton University, 84-WR-3.
[15] Pinder, G.F., *Groundwater Modeling using Geographical Information Systems*, John Wiley & Sons, New York, p. 233, 2002.

Urban Water

Edited by: ***S. MAMBRETTI****, Politecnico di Milano, Italy and* ***C.A. BREBBIA****, Wessex Institute of Technology, UK*

Containing the proceedings of the first international conference organised by the Wessex Institute of Technology on the Design, Construction, Maintenance, Monitoring and Control of Urban Water Systems, this book covers an area of increasing worldwide concern. As our cities continue to expand, their urban infrastructure needs to be re-evaluated and adapted to new requirements related to the increase in population and the growing areas under urbanisation.

New water systems are also required to reduce the risk associated with floods, network failures, and problems associated with inadequate networks. New systems should reduce economic losses and environmental impacts as well as promote a higher degree of reliability to users. Improved management, measurement and control mechanisms are needed to ensure the efficiency and safety of urban water systems.

Topics include the following in the area of Water Supply: Surface Water and Ground Water Sources; Water Supply Networks; Coping with Water Scarcity; Safety and Security of Water Systems; Water Quality; Water and Sustainability; Water Savings; Water Re-use. The Following Topics are related to Urban Drainage: Waste Water Treatment and Disposal; Structural Works and Infrastructure; Networks Design; Real Time Control; Water Quality Issues; Combined Sewer Overflows; Storage Tanks; Flood Control; Environmental Impact; Industrial Waste Water.

WIT Transactions on the Built Environment, Vol 122
ISBN: 978-1-84564-576-2 eISBN: 978-1-84564-577-9
Published 2012 / 306pp / £132.00

Water and Society

Edited by: ***D. PEPPER****, University of Nevada, Las Vegas, USA;* ***M. LAITURI****, Colorado State University, USA and* ***C.A. BREBBIA****, Wessex Institute of Technology, UK*

This book contains the papers presented at a conference co-organised by the University of Nevada-Las Vegas and the Wessex Institute of Technology to facilitate trans-disciplinary communication on issues related to the nature of water and its use and exploitation by society. With adequate water supply becoming a critical issue in more and more areas, there is a great and urgent need to bridge the gap between the broad spectrum of social sciences and humanistic disciplines and the specialists in physical and natural sciences, biology, environmental sciences, and health. Many issues are also trans-national in nature and relate to rights of states and hence it is essential to discuss these at an international level to arrive at equitable and binding solutions that will ensure the rights of society to quality water supplies.

This book discusses: The Nature of Water; Water as a Human Right; Water as the Source of Life; Water in a Changing Climate; Future Water Demands and Adaptation Strategies; Water Resources Contamination; Surface and Sub-Surface Water Resources; Irrigation and Desertification; Water, Sanitation and Health; Transnational Water Rights; Legislation and Controls; Water through the Ages; Lessons to be Learnt; and Water and Disaster Management.

WIT Transactions on Ecology and the Environment, Vol 153
ISBN: 978-1-84564-556-4 eISBN: 978-1-84564-557-1
Published 2012 / 576pp / £248.00

WIT*Press*
Ashurst Lodge, Ashurst, Southampton,
SO40 7AA, UK.
Tel: 44 (0) 238 029 3223
Fax: 44 (0) 238 029 2853
E-Mail: witpress@witpress.com

Flood Prevention and Remediation

Edited by: ***F.C.B. MASCARENHAS****, COPPE-UFRJ, Brazil*

The trend is for the world's population to concentrate in cities, and for flooding to occur more and more frequently in urban areas. The economic losses and social risks are amplified when inundation reaches the built environment. Yet it has been difficult to adequately manage urban flood problems, because flood risk and flood costs are not easy to quantify due to the subjective nature of these evaluations and to the lack of standardised methodologies. The editor of this book has organised contributions to address that problem.

Human settlements have grown near watercourses since ancient times. Water supply, irrigation, navigation, wastewater conveyance and city defence are some of the uses that have been responsible for this choice. Even floods played an important role, facilitating soil fertilisation. Man-made actions, however, especially in urban watersheds, significantly modify the natural water cycle, increasing the magnitude of floods and their potential damages. Consequently, flood damage is one of the most important issues to be dealt with in the present day. Several different studies show that floods are one of the most significant natural hazards, with severe losses, in terms of both lives and money.

Flood Prevention and Remediation presents several techniques and approaches to help in dealing with flood problems. Chapters 1 and 2 show simplified mathematical modelling of floods and the results of multifunctional landscape flood control measures in the city of Rio de Janeiro. The remaining book chapters present remedial work against debris after floods in Venezuela; measures for mitigation of flood areas in Japan; studies on flood risk assessment and management in Mediterranean basins; the attitudes of residents in the Tokyo Bay area toward flood hazards.

ISBN: 978-1-84564-560-1 eISBN: 978-1-84564-561-8
Published 2011 / 176pp / £76.00

Water Resources Management VI

Edited by: ***C.A. BREBBIA*** *and* ***V. POPOV****, Wessex Institute of Technology, UK*

Water Resources Management VI contains the proceedings of the sixth conference in a biennial series, begun in 2001, where scientists, experts and practitioners exchange their knowledge and experiences regarding sustainable management of water resources.

The growth of human population combined with the irregularity in precipitation and water availability may restrict even further the access to water in certain regions of the world. This problem is made more severe by anthropogenic activities that affect the quality of the water available.

Only by comparing the performance of the current technologies and practices can the best platforms be selected as the basis for future improved strategies. It is apparent that the goal of sustainable water resources management is not static; it changes as the conditions in the world alter due to, for instance, to growing population and climate change. Therefore the developments in the field must respond with the same agility. New ways of thinking are required in order to successfully predict future trends and prepare adequate sustainable solutions.

The presentations published in this book are grouped into the following sections: Water Management and Planning; River Basin Management; Urban Water Management; Hydrological Modelling; Hydraulic Engineering; Water Quality; Pollution Control; Irrigation Issues; Special Session on Sharing our Water Resources; Flood Risk; Waste Water Treatment and Management; Waste Water Treatment and Re-use; Water Resources and Economics; The Right to Water.

WIT Transactions on Ecology and the Environment, Vol 145
ISBN: 978-1-84564-514-4 eISBN: 978-1-84564-515-1
Published 2011 / 864pp / £372.00