IntechOpen

Computational Optimization in Engineering

Paradigms and Applications

Edited by Hossein Peyvandi

COMPUTATIONAL OPTIMIZATION IN ENGINEERING - PARADIGMS AND APPLICATIONS

Edited by **Hossein Peyvandi**

Computational Optimization in Engineering - Paradigms and Applications
http://dx.doi.org/10.5772/62604
Edited by Hossein Peyvandi

Contributors

Yoel Tenne, Florian Martin, Sylvain Gubian, Yang Xiang, Maoxi Tian, Keming Yu, Barry Smith, Augustine Wong, Maral Zafar Allahyari, Ahmed Azab, Makoto Yasuda, Mahsa Amirabdollahian, Bithin Datta

© **The Editor(s) and the Author(s) 2017**
The moral rights of the and the author(s) have been asserted.
All rights to the book as a whole are reserved by INTECH. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECH's written permission.
Enquiries concerning the use of the book should be directed to INTECH rights and permissions department (permissions@intechopen.com).
Violations are liable to prosecution under the governing Copyright Law.

Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 3.0 Unported License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be foundat http://www.intechopen.com/copyright-policy.html.

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in Croatia, 2017 by INTECH d.o.o.
eBook (PDF) Published by IN TECH d.o.o.
Place and year of publication of eBook (PDF): Rijeka, 2019.
IntechOpen is the global imprint of IN TECH d.o.o.
Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Computational Optimization in Engineering - Paradigms and Applications
Edited by Hossein Peyvandi

p. cm.

Print ISBN 978-953-51-3081-9

Online ISBN 978-953-51-3082-6

eBook (PDF) ISBN 978-953-51-4862-3

We are IntechOpen, the world's largest scientific publisher of Open Access books.

3,250+
Open access books available

106,000+
International authors and editors

112M+
Downloads

Our authors are among the

151
Countries delivered to

Top 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Meet the editor

Dr. Hossein Peyvandi received his BSc degree in Electrical Engineering from Sharif University of Technology, Iran, his MSc degree (with distinction) in Electrical Engineering from the University of Tehran, Iran, and his PhD degree in Electrical Engineering from the University of Surrey, UK. He has also been a postdoctoral research scholar at Ulster University, Northern Ireland. Since 2011, he has been with the University of Surrey, UK, where he became an associate lecturer in 2014. He has published more than 20 papers in peer-reviewed journals and conferences, including ISIT, one of the prestigious IEEE conferences on information theory, which was held at MIT, USA, 2012. He also has a patent on cryptography. His current research interests include computational intelligence, neuroinformatics, predictive analytics, self-organizing networks, network security, cryptanalysis, and information theory. Dr. Peyvandi is a fellow of the Higher Education Academy (HEA), UK. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), American Mathematical Society (AMS), and Institute of Engineering and Technology (IET). Since 2012, he has been a TPC member of the International Conference on Data Mining (DMIN), USA. He has served as a reviewer for numerous journals, including *IEEE Signal Processing* and *Wiley Wireless Communication*. He is a recipient of the NEF award accredited by the National Elites Foundation, Iran. He has been involved in a number of R&D projects and three European projects for more than ten years in his research career. He is interested in calligraphy, oil painting, and mountaineering in his spare time.

Contents

Foreword

I feel greatly honored to write a foreword for the book entitled *Computational Optimization in Engineering: Paradigms and Applications* edited by Dr. Hossein Peyvandi.

Optimization has been challenging for centuries. However, computational optimization as a methodology, particularly in the field of engineering, has a history of rapid growth over a couple of recent decades. The advent and development of computational optimization over the last decades have provided very versatile novel methodologies. With the recent developments in science and technology, some new areas of applications for computational optimization have emerged in the field of engineering. There has since been a remarkable period of growth, not only because of the many advantages of novel algorithms but also because in many applications of technology, sophisticated optimization techniques are the only viable choice. Examples of such emerging applications include complex networks that are now evolving to reach unprecedented size and scale and the field of automated medical diagnosis.

Dr. Peyvandi has to be complimented for bringing together an excellent collection of contributions for this important and timely topic. His experience of research positively reflects in the edited book on computational optimization with a comprehensive treatment of the subject. I hope the book will establish itself as a popular text.

Professor Muhammad Ali Imran
Professor of Communication Systems
Vice Dean Glasgow College UESTC
School of Engineering
University of Glasgow
Glasgow, UK

Preface

"Since the fabric of the universe is most perfect, and is the work of the most wise **Creator**, nothing whatsoever takes place in the universe in which some form of **optimality** does not appear."

Leonhard Euler
Swiss Mathematician and Physicist (1707–1783)

The purpose of optimization is to maximize the quality of lives, productivity in time, as well as interests. Therefore, optimization is an ongoing challenge for selecting the best possible among many other inferior designs. For a hundred years in the past, as optimization has been essential to human life, several techniques have been developed and utilized. Such a development has been one of the long-lasting challenges in engineering and science, and it is now clear that the optimization goals in many real-life problems are unlikely to be achieved without resource to computational techniques. The history of such a development in the optimization techniques starts from the early 1950s and is still in progress. Since then, the efforts behind this development dedicated by many distinguished scientists, mathematicians, and engineers have brought us today a level of quality of lives. This book concerns with the computational optimization in engineering and techniques to resolve the underlying problems in real life. The current book contains studies from scientists and researchers around the world from North America to Europe and from Asia to Australia.

In Chapter 1, entitled "Deterministic Annealing: A Variant of Simulated Annealing and Its Application to Fuzzy Clustering," a combination of the deterministic annealing algorithm of optimization with fuzzy c-means has been studied. The method has been further developed using Tsallis entropy maximization, called Tsallis-DAFCM, for the considered optimization problem. In Chapter 2, entitled "Generalized Simulated Annealing," a modified GSA approach has been discussed and implemented using the R package, GenSA. The nonconvex optimization problems in the fields of physics and finance have been analyzed. In addition, an interesting comparison between R packages has been presented in this chapter. In Chapter 3, entitled "A Simulated Annealing-Based Optimization Algorithm," the challenges in metamodels for an optimization process have been discussed. A method of enhancement in the overall prediction accuracy of models has also been proposed. In Chapter 4, entitled "Simulated Annealing of Constrained Statistical Functions," it has been demonstrated how simulated annealing can be used to perform a likelihood-based statistical inference in a constraint optimization problem. In this study, an impressive stress-strength modeling is introduced along with its statistical and numerical properties. In Chapter 5, entitled "Fitting Truncated Mode Regression Model by Simulated Annealing," the truncated mode regression optimization has been considered to explore the conventional income structure in China. In this study, the statistical parameters of an optimization problem have been analyzed.

In Chapter 6, entitled “Facility Layout Problem for Cellular Manufacturing Systems," a heuristic algorithm has been designed to allocate and displace facilities in the radial direction for the underlying optimization problem. In order to improve the search efficiency of the developed algorithm, the different cell size in initialization has been considered. In this chapter, a real-life optimization problem from the industry has been used. In Chapter 7, entitled “Application of Simulated Annealing and Adaptive Simulated Annealing in Search for Efficient Optimal Solutions of a Groundwater Contamination-Related Problem," the optimization problem of source characterization has been considered using a methodology based on ASA. In this study, it has also been shown that ASA provides reliable results for the considered optimization problem.

I would like to add that it was a great time working on the chapters as an editor. I hope the contents of the book attract the interests from all fields of engineering and scientific sectors, as it has been desired to serve so. I welcome any further questions or comments on the book chapters and their contents. Interested readers may send their comments directly to the editor using the following e-mail address.

I would like to thank my family for their support, particularly, Reza for letting me work on the book in times that I should have spent with him. I would like to thank Professor Muhammad Ali Imran, vice dean of UESTC at the University of Glasgow, for his encouragement and kind words. I would also like to thank Ms. Ana Pantar, senior commissioning editor, and Ms. Martina Usljebrka, publishing process manager, for their assistance in preparing the book during the year.

Dr. Hossein Peyvandi
PhD, FHEA, MIEEE, MIET, MSc, BSc
University of Surrey, Guildford, Surrey
United Kingdom

Section 1

Paradigms & Algorithms

Chapter 1

Deterministic Annealing: A Variant of Simulated Annealing and its Application to Fuzzy Clustering

Makoto Yasuda

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66072

Abstract

Deterministic annealing (DA) is a deterministic variant of simulated annealing. In this chapter, after briefly introducing DA, we explain how DA is combined with the fuzzy c-means (FCM) clustering by employing the entropy maximization method, especially the Tsallis entropy maximization. The Tsallis entropy is a q parameter extension of the Shannon entropy. Then, we focus on Tsallis-entropy-maximized FCM (Tsallis-DAFCM), and examine effects of cooling functions for DA on accuracy and convergence. A shape of a membership function of Tsallis-DAFCM depends on both a system temperature and q. Accordingly, a relationship between the temperature and q is quantitatively investigated.

Keywords: deterministic annealing, simulated annealing, free energy, fuzzy c-means clustering, entropy maximization, Shannon entropy, fuzzy entropy, Tsallis entropy

1. Introduction

Statistical mechanics investigates the macroscopic properties of a physical system consisting of many elements. Recently, great research activities of applying statistical mechanical models or tools to information engineering problems have been seen. The entropy maximization method applied to a fuzzy c-means clustering is a good example of such models.

Simulated annealing (SA) [1, 2] is one of the most commonly used optimization techniques and plays an important role in the field of engineering because many of the engineering problems can be formulated as optimization problems. SA is a stochastic relaxation method that treats an objective function as a system energy, and by analogy with the annealing process

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

of solids, searches for its minimum with decreasing the system temperature. SA searches randomly at a high temperature, but more deterministically at a low temperature. As long as a neighborhood can be defined and the temperature is lowered sufficiently slowly, SA is a global optimization technique for solving optimization problems. It requires a very long time to find an optimal global solution because of a stochastic search at each temperature. Thus, SA is an approximation method in practical.

Deterministic annealing (DA) is a deterministic variant of SA, which is first proposed by Rose et al. [3] for a vector quantization algorithm. DA characterizes the minimization problem of the objective function as the minimization of the free energy of the system, which depends on the temperature. DA tracks the minimum of the free energy while decreasing the temperature. Thus, it can deterministically optimize the objective function at each temperature, and is more efficient than SA but does not guarantee the optimal solution. In addition, an effect of a cooling function on the quality of a DA's solution is still unclear.

Membership functions of the fuzzy c-means (FCM) clustering [4] maximized or regularized with entropy [5, 6] have similar forms with the distribution functions that appear in statistical mechanics. For example, a membership function obtained by the Shannon entropy maximization has a similar form with the Boltzmann-Gibbs distribution function [3, 5]. Similarly, a membership function obtained by the fuzzy entropy [7] has a similar form with the Fermi-Dirac distribution function [8]. Annealing methods can be applicable to these membership functions because they contain a parameter that can be corresponded to the system temperature. The advantage of applying the entropy maximization methods to FCM is that fuzzy clustering can be analyzed from both information processing and statistical physical points of view.

Tsallis [9], by extending Boltzmann-Gibbs statistics nonextensively with a generalization parameter q, postulated a generalized formulation of entropy. The entropy is now well known as The Tsallis entropy, which, in the limit of q to 1, approaches the Shannon entropy. The Tsallis entropy is applicable to numerous fields, including physics, bioscience, chemistry, networks, computer science, and so on [10–12]. Menard et al. [13, 14] investigated fuzzy clustering in the framework of nonextensive thermostatistics, and derived the possibilistic membership function by taking the possibilistic constraint into account.

On the other hand, based on the Tsallis entropy, Yasuda [15] defined another form of entropy for FCM, and then derived the membership function by maximizing this entropy within FCM [15]. After that, by combining the membership function with DA, a new fuzzy clustering algorithm (the Tsallis-DAFCM algorithm) has been developed. Tsallis-DAFCM was proved to yield superior results in comparison with the conventional annealing methods.

Similarly to SA, a performance of Tsallis-DAFCM strongly depends on how to decrease the temperature. Among the cooling functions for SA, the very fast annealing (VFA) method decreases the temperature fastest. Thus, VFA is applied to Tsallis-DAFCM to improve its performance, and proved to be effective [16].

In spite of its performance, it remains unknown how appropriate q value and initial annealing temperature T_{high} for Tsallis-DAFCM should be determined according to the data distribution.

One of the important characteristics of the membership function of Tsallis-DAFCM is that centers of clusters are given as a weighted function of the membership function to the power of $q(u_{ik}^q)$. Furthermore, it changes its shape in a similar way by decreasing the temperature or by increasing q. In order to reveal the relationship between the temperature and q, u_{ik}^q is quantitatively analyzed using the Iris Data Set [17]. The result shows that the temperature and q affect u_{ik}^q almost inversely, suggesting that a q-incrementation algorithm is possible. This algorithm might be a solution to the initialvalue problem of Tsallis-DAFCM [18, 19].

This chapter is composed of five sections. Section 1 is this introduction. In Section 2, how DA works is explained generously, and its applications are summarized. In Section 3, each of the components of entropy-maximized FCM clustering methods is explained. In Section 4, VFA is used as the cooling function of Tsallis-DAFCM, and its effects are experimentally investigated. Effects of the temperature and q-values on the membership function of Tsallis-DAFCM are also examined. Section 5 gives a conclusion of this chapter.

2. Deterministic annealing

Vector quantization is a classification method for a big data, which is widely used in the field of image compression. Linde-Buzo-Gray [20] and self-organizing feature map [21] algorithms, for example, are well-known vector quantization algorithms. However, classification results of these algorithms depend on initial settings of the reference vector, and can easily fall into local minima.

In order to overcome the problem, by analogy with statistical mechanics, a DA method has been proposed. DA does not suffer from the initial vector problem. It usually performs better than other methods. However, it does not theoretically guarantee to find a global optimal solution. Furthermore, its performance depends on a way of decreasing the system temperature.

This section gives a brief introduction of the deterministic annealing method. In Section 2.1, a definition and major characteristics of DA are explained. In Section 2.2, DA is compared with SA. In Section 2.3, applications and modifications of DA are summarized.

2.1. Major characteristics of deterministic annealing

In DA, by analogy with statistical mechanics, the free energy F is derived from an objective function J of a problem. At a high temperature, F represents a global structure of J. As the temperature decreases, it gradually reaches J.

Based on these characteristics, at the high temperature, DA is able to find a global minimum of F by the steepest descent method because it should have multiple local minima. When the temperature is lowered a little, F would change its shape only a little. Accordingly, by setting the previous global mimimum as an initial value of the steepest descent mehod, DA searches

the next global minimum. This procedure continues until the temperature is lowered sufficiently, and F reaches J. Consequently, at each temperature, DA searches the local minimum of J deterministically.

2.2. Comparison with simulated annealing

While decreasing the temperature, SA searches the minimum stochastically at each temperature and thus requires a very long time to find an optimal solution. Hence, though theoretically, it is guaranteed to find the optimal solution, SA is practically an approximation method.

On the contrary, DA consumes less computational time because it searches the local minimum deterministically at each temperature. Furthermore, it should be noticed that, in case that multiple local minima exist at some temperature, DA might not be able to find the minimum. For this reason, even theoretically, DA is not guaranteed to find the optimal solution.

Approaches to speed up SA are mainly based on the improvement of a transition method and a cooling function including its parallelization. For example, adaptive SA (ASA) [22], which may belong to the both categories, is an implementation of very fast simulated re-annealing (VFSA) [23]. As compared with the acceleration method called fast annealing (FA) [24] in which the temperature is lowered inversely linear to a number of iterations, ASA is faster than FA. In addition, among many features included in ASA, it can get the benefit of speeding-up by simulated quenching.

In DA, it seems no comprehensive studies on this topic have been conducted.

The summary of comparison is shown in **Table 1**.

	SA	**DA**
Search strategy	Stochastic search based on the Metropolis algorithm.	Deterministic search based on the steepest descent algorithm.
Cooling function	Two categories of cooling functions are well-used. 1. Functions based on statistical analysis. 2. Adaptive functions depending on the problems.	Cooling functions appear in SA are used empirically.
Optimality	The global minimum can be achieved if a temperature is decreased as slow as $T \propto 1/\log(\text{iterations})$.	Not guaranteed.

Table 1. Comparison of SA and DA.

2.3. Applications and modifications of deterministic annealing

The study on DA first addressed avoidance of the poor local minima of data clustering [25]. Then it was extensively applied to various subjects such as combinational optimization problems [26], vector quantization [27, 28], maximum likelihood estimation [29], classifier design [30], and pairwise data clustering [31].

In order to cluster a large number of data, research activities that attempt to parallelize DA have become popular. Kim et al. [32] discussed the parallelization method of DA using GPU, and applied it to color image segmentation. In order to cluster a large number of bioscientific data, Fox et al. [33, 34] parallelized DA using MPI. Qiu et al. [35] compared the DA's performance using C# messaging runtime library CCR with that using MPI.

3. Application of deterministic annealing to fuzzy *c*-means clustering maximized with entropy

One of the important applications of DA is fuzzy clustering. In this section, we focus on fuzzy *c*-means (FCM) clustering. By maximizing the objective function of FCM with various entropies, membership functions similar to the statistical mechanical distribution functions are obtained. These membership functions can be easily combined with DA, because they contain a parameter corresponding to a system temperature.

In this section, first we outline the formulation of FCM. Then, we describe how to apply the entropy maximization methods to FCM. In Section 3.1, the classical FCM clustering method is introduced. In Section 3.2, various entropy maximization methods are explained. The free energies for each entropy maximization method are derived in Section 3.3. In Section 3.4, representative annealing functions are presented.

3.1. Fuzzy *c*-means clustering

Let $X = \{x_1, \cdots, x_n\}\left(x_k = \left(x_k^1, \cdots, x_k^p\right) \in R^p\right)$ be a data set in p-dimensional real space, which is to be divided into c clusters. Let $V = \{v_1, \cdots, v_c\}\left(v_i = \left(v_i^1, \cdots, v_i^p\right) \in R^p\right)$ be the centers of the clusters, and let $u_{ik} \in [0, \quad 1] \quad (i = 1, \cdots, c; k = 1, \cdots, n)$ be the membership functions. Then, let

$$J = \sum_{k=1}^{n}\sum_{i=1}^{c} {u_{ik}}^m d_{ik} \quad \left(d_{ik} = \left\|\mathbf{x}_k - \mathbf{v}_i\right\|^2;\ m \in R;\ 1 < m\right) \tag{1}$$

be the objective function of FCM to be minimized. Under the normalization constraint of

$$\sum_{i=1}^{c} u_{ik} = 1 \ \ (^{\forall}k), \tag{2}$$

the Lagrange function L is given by

$$L = J - \sum_{k=1}^{n} \eta_k \left(\sum_{i=1}^{c} u_{ik} - 1 \right), \tag{3}$$

where η_k denotes the Lagrange multiplier. $\partial L / \partial u_{ik} = 0$ gives the membership function of the form:

$$u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{1}{m-1}} \right]^{-1}. \tag{4}$$

Similarly, v_i can be determined by $\partial L / \partial v_i = 0$ as follows:

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik} \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}}. \tag{5}$$

Desirable cluster centers can be obtained by calculating Eq. (4) and Eq. (5) repeatedly.

3.2. Entropy maximization method for fuzzy *c*-means

3.2.1. Shannon entropy maximization

Shannon entropy for FCM takes the form:

$$S_{SE} = -\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik} \log u_{ik}. \tag{6}$$

By setting m to 1 in Eq. (1), under the normalization constraint of Eq. (2), the Shannon entropy functional is given by

$$\delta S_{SE} - \sum_{k=1}^{n} \alpha_k \delta \left(\sum_{i=1}^{c} u_{ik} - 1 \right) - \beta \sum_{k=1}^{n} \sum_{i-1}^{c} \delta \left(u_{ik} d_{ik} \right), \tag{7}$$

where α_k and β denote the Lagrange multipliers. The stationary condition for Eq. (7) leads to the following Gaussian membership function:

http://dx.doi.org/10.5772/66072

$$u_{ik} = \frac{e^{-\beta d_{ik}}}{\sum_{j=1}^{c} e^{-\beta d_{jk}}}, \tag{8}$$

and the same formula for v_i in Eq. (5).

3.2.2. Fuzzy entropy maximization

Fuzzy entropy for FCM is defined as [36]

$$S_{FE} = -\sum_{k=1}^{n}\sum_{i=1}^{c}\{u_{ik}\log u_{ik} + (1-u_{ik})\log(1-u_{ik})\}. \tag{9}$$

The fuzzy entropy functional is given by

$$\delta S_{FE} - \sum_{k=1}^{n}\alpha_k\delta\left(\sum_{i=1}^{c}u_{ik} - 1\right) - \beta\sum_{k=1}^{n}\sum_{i-1}^{c}\delta\left(u_{ik}d_{ik}\right). \tag{10}$$

The stationary condition for Eq. (10) leads to the following membership function:

$$u_{ik} = \frac{1}{e^{\alpha_k + \beta d_{ik}} + 1}, \tag{11}$$

and the same formula for v_i in Eq. (5). Eq. (11) is similar to the Fermi-Dirac distribution function.

3.2.3. Tsallis entropy maximization

The Tsallis entropy for FCM is defined as [15]

$$S_{Tsallis} = -\frac{1}{q-1}\left(\sum_{k=1}^{n}\sum_{i-1}^{c}{u_{ik}}^{q} - 1\right), \tag{12}$$

where $q \in R$ is a real number. In case of the Tsallis entropy maximization, the objective function should be rewritten as

$$J_{Tsallis} = \sum_{k=1}^{n}\sum_{i=1}^{c} {u_{ik}}^{q} d_{ik}. \tag{13}$$

Accordingly, the Tsallis entropy functional is given by

$$\delta S_{Tsallis} - \sum_{k=1}^{n} \alpha_k \delta\left(\sum_{i=1}^{c} u_{ik} - 1\right) - \beta \sum_{k=1}^{n}\sum_{i-1}^{c} \delta\left({u_{ik}}^{q} d_{ik}\right). \tag{14}$$

The stationary condition for Eq. (14) leads to the membership function of the form:

$$u_{ik} = \frac{\left\{1 - \beta(1-q)d_{ik}\right\}^{\frac{1}{1-q}}}{Z}, \tag{15}$$

where

$$Z = \sum_{j=1}^{c} \left\{1 - \beta(1-q)d_{jk}\right\}^{\frac{1}{1-q}}.$$

v_i is defined as

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} {u_{ik}}^{q} \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}}. \tag{16}$$

3.3. Free energy for entropy maximized fuzzy *c*-means

In each entropy, maximization methods introduced in Section 3.2, β can be regarded as the inverse of the system temperature T^{-1}. This feature makes it possible to apply DA, and Shannon-entropy-maximized FCM with DA (Shannon-DAFCM, hereafter), fuzzy-entropy-maximized FCM with DA (fuzzy-DAFCM, hereafter), and Tsallis-entropy-maximized FCM with DA (Tsallis-DAFCM, hereafter) have been developed.

3.3.1. Free energy for Shannon-DAFCM

In Shannon-DAFCM, by analogy with statistical mechanics [37], the sum of the states (the partition function) for the grand canonical ensemble of FCM can be expressed as

$$\xi = \prod_{k=1}^{n} \sum_{i=1}^{c} \mathrm{e}^{-\beta d_{ik}} \tag{17}$$

The free energy is derived as

$$F = -\frac{1}{\beta} \log \xi = -\frac{1}{\beta} \sum_{k=1}^{n} \log \sum_{i=1}^{c} \mathrm{e}^{-\beta d_{ik}}. \tag{18}$$

Stable thermal equilibrium requires a minimization of the free energy. By formulating deterministic annealing as a minimization of the free energy, $\partial F / \partial v_i = 0$ yields the same expression for v_i as that in Eq. (5).

3.3.2. Free energy for fuzzy-DAFCM

Similarly, in fuzzy-DAFCM, the grand partition function for the grand canonical ensemble for FCM can be written as

$$\Xi = \prod_{k-1}^{n} \prod_{i=1}^{c} \left(1 + \mathrm{e}^{-\alpha_k - \beta d_{ik}} \right). \tag{19}$$

The free energy is calculated as

$$F = -\frac{1}{\beta} \left(\log \Xi - \alpha_k \frac{\partial \log \Xi}{\partial \alpha_k} \right) = -\frac{1}{\beta} \sum_{k=1}^{n} \left\{ \sum_{i=1}^{c} \log \left(1 + \mathrm{e}^{-\alpha_k - \beta d_{ik}} \right) + \alpha_k \right\}. \tag{20}$$

$\partial F / \partial v_i = 0$ yields the same expression for v_i as that in Eq. (5).

3.3.3. Free energy for Tsallis-DAFCM

In Tsallis-DAFCM, the free energy can be derived as

$$F = J_{Tsallis} - TS_{Tsallis} = -\frac{1}{\beta} \sum_{k=1}^{n} \frac{Z^{1-q} - 1}{1 - q}, \tag{21}$$

where T is a system temperature. $\partial F / \partial v_i = 0$ yields the same expression for v_i as that in Eq. (16).

3.4. Effect of annealing temperature on clustering

3.4.1. Dependency of shape of membership function on temperature

While reducing the system temperature T, DA achieves thermal equilibrium at each temperature by minimizing the free energy. Thus, DA searches a cluster distribution that minimizes the free energy at each temperature. When the temperature is high, the membership functions distribute widely. This makes clusters to which a data belong fuzzy. In case of Tsallis-DAFCM, when q is nearly equal to 2, the width of the membership function is almost proportional to $\sqrt{T}$. On the contrary, at the low temperature, fuzzy clustering approaches hard clustering. The relationship $F = J_{\text{Tsallis}} - TS_{\text{Tsallis}}$ in Eq. (21) suggests that, at the higher temperature, the larger entropy state or chaotic state is caused by a widening of the extent of the membership function.

3.4.2. Cooling function

In SA, the temperature decreases according to a cooling function or an annealing schedule. The representative cooling functions for SA [38] are:

(I) Exponential function

$$T = T_{high} r^t, \tag{22}$$

where T_{high} is the highest initial temperature, r is a parameter which defines a temperature reduction rate, and t is a number of iterations of temperature reduction.

(II) Inversely linear function

$$T = \frac{T_{high}}{t}. \tag{23}$$

(III) Inversely logarithmic function

$$T = \frac{T_{high}}{\ln t}. \tag{24}$$

(IV) Inversely exponential function

$$T = \frac{T_{high}}{e^{rt}}. \tag{25}$$

(V) Very fast annealing

Rosen [39] proposed another inversely exponential function known as VFA. VFA decreases the temperature exponentially in a similar way to ASA:

$$T = \frac{T_{high}}{e^{rt^{(1/D)}}}, \tag{26}$$

where D is a dimension of a state space.[1] **Figure 1** compares plots of Eq. (25) and Eq. (26).

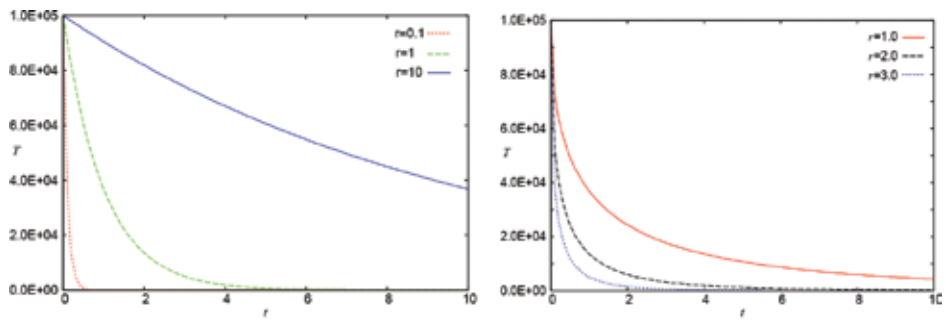

Figure 1. Plots of (a) Eq. (25) and (b) Eq. (26) ($T_{high} = 1.0 \times 10^5$, $D = 2$).

4. Tsallis-entropy-maximized fuzzy *c*-means clustering with deterministic annealing

In this section, we focus on Tsallis-DAFCM, and its important experimental results are explained. In Section 4.1, we present the Tsallis-DAFCM clustering algorithm. In Section 4.2, how VFA affects Tsallis-DAFCM is experimentally investigated. In Section 4.3, effects of the temperature and q-values on the membership function are examined.

4.1. Tsallis-DAFCM clustering algorithm

The Tsallis-entropy-maximization method, fuzzy c-means clustering, and the deterministic annealing method can be combined as the following Tsallis-DAFCM clustering algorithm [15]:

1. Set the number of clusters c, the highest temperature T_{high}, the temperature reduction rate m, and the threshold of convergence test δ_1 and δ_2;

[1] In clustering, a state space refers to an input space of a data set. For example, if a data set consists of 3 attributes, D is set to 3.

2. Generate c clusters at random positions. Set current temperature T to T_{high};

3. Calculate the membership function u_{ik} by Eq. (15);

4. Calculate the centers of clusters v_i by Eq. (17);

5. Compare the difference between the current cluster centers v_i and the cluster centers obtained in the previous iteration $\overline{v}_i$. If the convergence condition $max_{1 \le i \le c} \left|\left| \mathbf{v}_i - \overline{\mathbf{v}}_i \right|\right| < \delta_1$ is satisfied, then go to **6**, otherwise go back to **3**;

6. Compare the difference between the current cluster centers and the cluster centers obtained at the previous temperature $\overline{\overline{v}}_i$. If the convergence condition $max_{1 \le i \le c} \left|\left| \mathbf{v}_i - \overline{\overline{\mathbf{v}}}_i \right|\right| < \delta_2$ is satisfied, then stop, otherwise decrease the temperature using the cooling function, and go back to **3**.

4.2. Effect of cooling functions

In general, the temperature should be reduced gradually in DA. However, this takes a long time to converge. If VFA is applicable to Tsallis-DAFCM, it is of great advantage to this method. Accordingly, VFA is tested as a cooling function of Tsallis-DAFCM.

In this subsection, both Shannon- and Tsallis-DAFCM are examined.

4.2.1. Experiment 1

In experiment 1, the numerical data composed of five clusters and 2000 data points are used, as shown in **Figure 2**. The parameters are set as follows: c = 10, δ_1 = 50, δ_2 = 2, q = 1.5, T_{high} = 1.0×10^6 or 1.0×10^5.

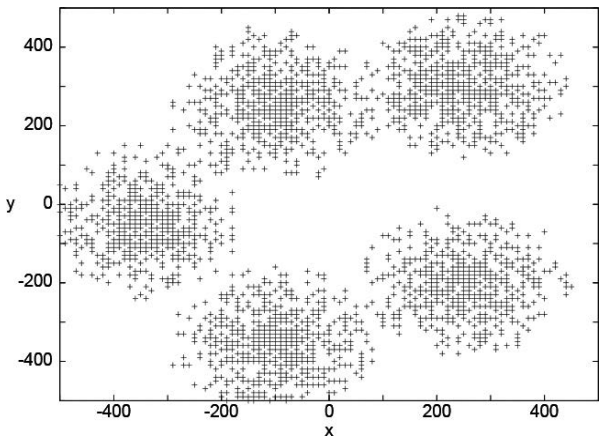

Figure 2. Numerical data.

http://dx.doi.org/10.5772/66072

First, the inversely exponential cooling function is applied to Tsallis-DAFCM. The changes of β parameterized by r are plotted in **Figure 3**. In case of r = 1000.0, when $T_{\text{high}} = 1.0\times10^5$, as T is lowered from **Figures 3 (A)** to **(D)**, data are clustered gradually and desirably. In case of r = 10.0 and r = 1.0 (**Figures 3 (E)** and **(F)**, respectively), it is observed that u_{ik} and v_i converge more rapidly.

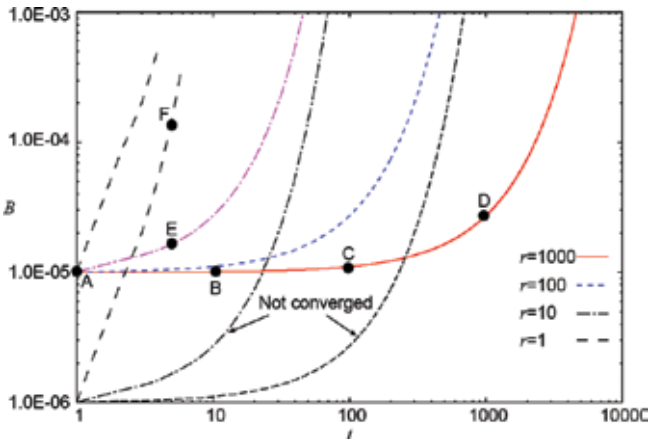

Figure 3. Increasing of β by inversely exponential cooling function.

However, when $T_{\text{high}} = 1.0\times10^6$, the algorithm fails to converge with r = 100.0 and 10.0 (expressed by "Not converged" in **Figure 3**) because the initial distribution of u_{ik} is too wide. This result indicates that it is important to set both T_{high} and r properly.

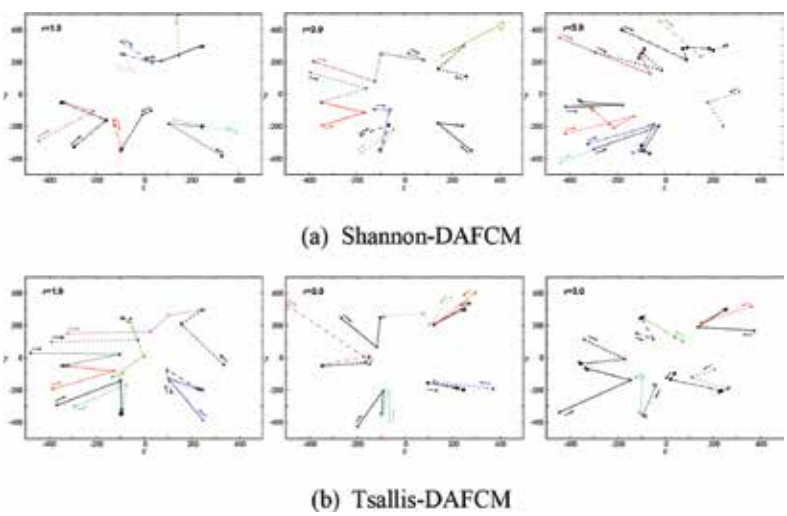

Figure 4. Shifts of cluster centers during clustering obtained by (a) Shannon-DAFCM and (b) Tsallis-DAFCM.

In order to clarify the adaptability of VFA as a cooling function of DA, numerical experiments on Shannon- and Tsallis-DAFCM are performed. The shifts of cluster centers with decreasing temperature are illustrated in **Figures 4 (a)** and **(b)**. Initially, clusters are located randomly. Then, at the higher temperature, β is comparatively small and clusters move to near the center of gravity of data because the membership functions are extend over the data area and become extremely uniform. As T is lowered, contrarily, the membership functions become narrower and the associations of data to the clusters become less fuzzy. In this process, in Shannon-DAFCM, the clusters move to their nearest local data distribution centers. However, in Tsallis-DAFCM, the clusters can move a long distance to optimal positions because the membership functions have gentle base slopes.

Figures 5 (a) and **(b)** illustrates the three-dimensional plots of u_{ik} in the progress of Shannon- and Tsallis-DAFCM clustering combined with VFA. When the temperature is as high as 3.7×10^4, roughness of u_{ik} of Tsallis-DAFCM is smaller than that of Shannon-DAFCM. After that, the shapes of both membership functions do not change greatly, because VFA reduces the temperature extremely only at the early annealing stage. When the temperature is lowered to 1.3×10^4, both methods cluster data desirably.

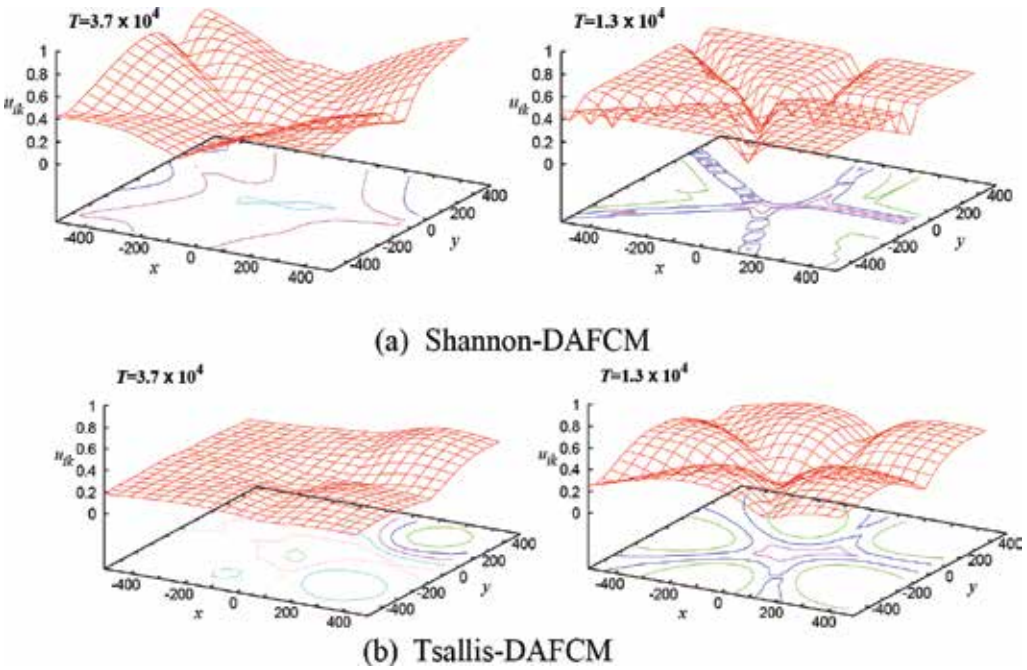

Figure 5. Initial and final landscapes of u_{ik} of (a) Shannon-DAFCM and (b) Tsallis-DAFCM.

Consequently, because Tsallis-DAFCM has gentle slope in the region far from the origin, clusters can move long distance to optimal positions stably. This feature makes it possible to reduce the temperature rapidly. Thus, VFA is suitable as a cooling function of Tsallis-DAFCM.

On the other hand, final cluster positions obtained by Shannon-DAFCM tend to depend on their initial positions.

4.2.2. Experiment 2

In experiment 2, the Iris Data Set [17] consisting of 150 four-dimensional vectors of iris flowers is used. Three clusters of flowers detected are Versicolor, Virginia, and Setosa. Each cluster consists of 50 vectors. VFA is used as a cooling function of DA. The parameters are set as follows: $c = 3$, $\delta_1 = 0.1$, $\delta_2 = 0.01$, $q = 1.5$, $T_{\text{high}} = 2.0$.

The minimum, maximum, and average values of misclassified data points of 100 trials are summarized in **Table 2**. It can be seen that Shannon-DAFCM gives slightly better results than Tsallis-DAFCM. However, it is also confirmed that Tsallis-DAFCM gives its best results when the temperature reduction rate r is set to 1 or 2, though the best result for Shannon-DAFCM is obtained only when $r = 2$. Furthermore, variances of Tsallis-DAFCM are smaller than those of Shannon-DAFCM. These features indicate that a wide range of r values are applicable to Tsallis-DAFCM. On the other hand, with larger r values, Shannon-DAFCM becomes unstable.

Shannon-DAFCM				**Tsallis-DAFCM**			
r	**Min.**	**Max.**	**Ave.**	**r**	**Min.**	**Max.**	**Ave.**
1	15	16	15.01	1	14	14	14.00
2	11	13	12.97	2	14	15	14.01
3	11	14	13.59	3	14	15	14.68

Table 2. Misclassified data points of Iris Data Set (100 trials).

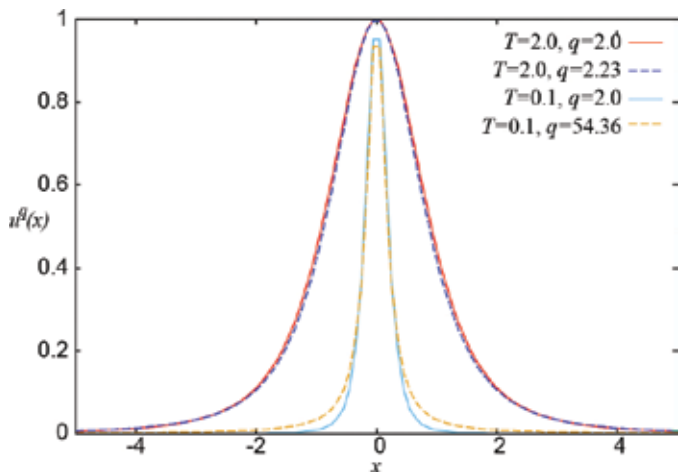

Figure 6. Plots of $u^q(x)$.

4.3. Dependencies of the membership function on temperature and q

In Eq. (16), it can be seen that u_{ik}^{q} plays an important role as a weight value to each x_k, and it determines v_i. For this reason, dependencies of u_{ik}^{q} on T and q are to be investigated.

In this subsection, for simplicity, v_i is set to be 0, because this makes the denominator of Eq. (15) become the sum of the same formulas of its numerator. **Figure 6** illustrates the numerator of u_{ik}^{q} (expressed by $u^q(x)$) as a function of x_k, parameterized by T and q, respectively. In order to plot the shape of u_{ik}^{q} as a function of the distance between the cluster center and various data points, in this figure, x_k is replaced to a continuous variable $\boldsymbol{x}$. **Figure 6** confirms that the extent of u_{ik}^{q} becomes narrower with increasing q. On the contrary, as the temperature decreases, the distribution becomes narrower.

4.3.1. Quantitative relationship between temperature and q

As stated in the previous subsection, T and q inversely affect the extent of u_{ik}^{q}, which changes in a similar way with decreasing T or increasing q. In order to examine the quantitative relationship between T and q in more detail, they are changed independently as follows:

First, we define

$$u^q(x,T,q) = \left\{1 - \frac{1-q}{T}x^2\right\}^{\frac{q}{1-q}}. \tag{27}$$

Then, Eq. (27) is calculated by fixing T and q to some constants T_0 and q_0. Next, by decreasing T, we search the q values that minimize the sum of squares of the residuals of the following two functions:

$$\sum_{S=0}^{Smax} \left| u^q(\Delta xS, T_0, q_0) - u^q(\Delta xS, T, q) \right|^2. \tag{28}$$

In the following calculations, the parameters are set as follows: $T_{\text{high}}(= T_0) = 2.0$; the domain of x is $0 \le x \le 100$; the number of sampling points of the sum of residuals $S_{\max} = 10{,}000$; $\Delta x = 0.01$.

For q (= q_0) values of 1.01, 2.0, 6.0, 10.0, and for T decreasing from T_{high}, the q values that minimize Eq. (28) (expressed by $q_{\min}$) are shown in **Figure 7 (a)**. **Figure 7 (b)**, on the other hand, shows the results of cases in which q is set to 2.0 and T is lowered from $T_{\text{high}} = 2.0$, 20.0, 100.0, 200.0. Approximate curves plotted in **Figure 7** are obtained by fitting the data to the following exponential function:

http://dx.doi.org/10.5772/66072

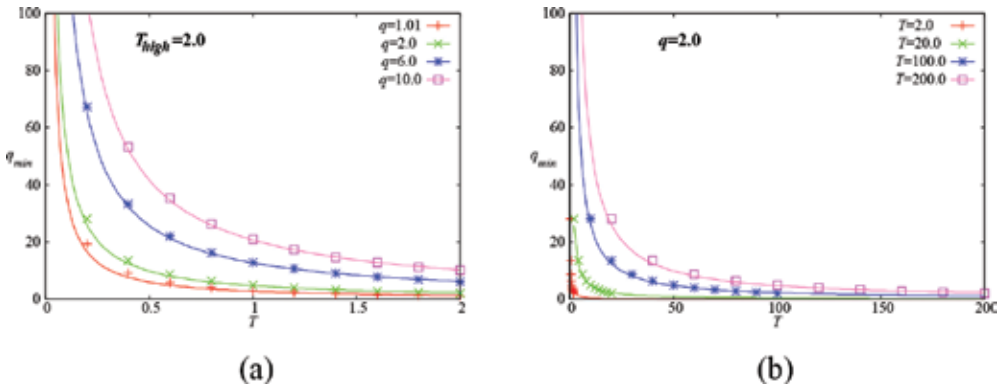

Figure 7. Plots of q_{min} as a function of T parameterized by (a) q and (b) T_{high}.

$$q_{\min} = aT^{-b}, \tag{29}$$

where a and b denote the fitting parameters. Optimal values for these parameters obtained by the least squares method are summarized in **Table 3** and **Table 4**. From these tables, it is concluded that b is nearly equal to 1.0 indicating that q is inversely proportional to T. In addition, it can be seen that, though b does not change its value much, a increases with increasing T.

q	a	b
1.01	2.71	1.126
2	4.67	1.066
6	12.65	1.023
10	20.64	1.014

Table 3. Parameters of approximate curves ($T_{\text{high}} = 2.0$).

T_{high}	a	b
2	4.67	1.066
20	54.33	1.066
100	302.06	1.066
200	632.35	1.066

Table 4. Parameters of approximate curves ($q = 2.0$).

As a result, by using the approximate relationship of T and $q_{\min}$, instead of annealing or T-reduction, q-incrementation clustering might be possible [18].

5. Conclusion

In this chapter, we first explained the major characteristics of DA and compared it with SA. DA is a variant of SA and searches a minimum deterministically. Thus, generally it is more efficient than SA. We then explained how DA could be applied to the fuzzy c-means clustering by employing the entropy maximization method.

After that, by focusing on Tsallis-entropy-maximized FCM combined with DA (Tsallis-DAFCM), an effect of VFA on DA was examined. VFA reduces the temperature extremely only at the early annealing stage, and the experimental result showed that this feature improved the performance of Tsallis-DAFCM because it has gentle slope in the region far from the origin and clusters can move long distance to optimal positions from the beginning.

The Tsallis entropy is an extension of the Shannon entropy with a generalization parameter q. A shape of a membership function of Tsallis-DAFCM strongly depends on both the temperature and q. Accordingly, a relationship between the temperature and q was quantitatively investigated, and it was experimentally confirmed that they affected the area covered by the membership function almost inversely. Based on the result, a development of a q-incrementation algorithm is our future subject.

Author details

Makoto Yasuda

Address all correspondence to: yasuda@gifu-nct.ac.jp

National Institute of Technology, Gifu College, Gifu, Japan

References

[1] E. Aarts, J. Korst. Simulated Annealing and Boltzmann Machines. Chichester: John Wiley & Sons; 1989.

[2] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. Optimization by simulated annealing. Science. 1983; 220: 671–680.

[3] K. Rose, E. Gurewitz, B. C. Fox. A deterministic annealing approach to clustering. Pattern Recognition Letters. 1990; 11(9): 589–594.

[4] J.C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Prenum Press; 1981.

[5] R.-P. Li, M. Mukaidono. A maximum entropy approach to fuzzy clustering. In: Proceedings of the 4th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE/ IFES '95); 1995 March 20–24; Yokohama, Japan; 1995. p. 2227–2232.

[6] S. Miyamoto, M. Mukaidono. Fuzzy c-means as a regularization and maximum entropy approach. In: Proceedings of the 7th International Fuzzy Systems Association World Congress; 1997 June 25–29; Prague, Czech Republic; 1997. p. 86–92.

[7] A. DeLuca, S. Termini. A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. Information and Control. 1972; 20: 301–312.

[8] L. D. Landau, E. M. Lifshitz. Statistical Physics (Part 1). Oxford: Butterworth Heinemann; 1980.

[9] C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. Journal of Statistical Physics. 1988; 52(1–2): 479–487.

[10] S. Abe, Y. Okamoto, editors. Nonextensive Statistical Mechanics and Its Applications. New York: Springer; 2001.

[11] M. Gell-Mann, C. Tsallis, editors. Nonextensive Entropy—Interdisciplinary Applications. New York: Oxford University Press; 2004.

[12] C. Tsallis, editor. Introduction to Nonextensive Statistical Mechanics. New York: Springer; 2009.

[13] M. Menard, V. Courboulay, P.-A. Dardignac. Possibilistic and probabilistic fuzzy clustering: unification within the framework of the non-extensive thermostatistics. Pattern Recognition. 2003; 36(6): 1325–1342.

[14] M. Menard, P. Dardignac, C. C. Chibelushi. Non-extensive thermostatistics and extreme physical information for fuzzy clustering. International Journal of Computational Cognition. 2004; 2(4): 1–63.

[15] M. Yasuda. Entropy maximization and very fast deterministic annealing approach to fuzzy c-means clustering. In: Proceedings of the 5th Joint International Conference on So Computing and 11th International Symposium on Intelligent Systems; 8–12 December 2010; Okayama, Japan; 2010. p. 1515–1520.

[16] M. Yasuda. Deterministic annealing approach to fuzzy c-means clustering based on entropy maximization. Advances in Fuzzy Systems. 2011; 960635: 9.

[17] UCI Machine Learning Repository: Iris Data Set [Internet]. 1988. Available from: http://archive.ics.uci.edu/ml/datasets/Iris

[18] M. Yasuda. Q-increment deterministic annealing fuzzy c-means clustering using Tsallis entropy. In: Proceedings of the 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKS); 19–21 August 2014; Xiamen, China; 2014. p. 31–35.

[19] M. Yasuda. Quantitative analyses and development of a q-incrementation algorithm for FCM with Tsallis entropy maximization. Advances in Fuzzy Systems. 2015; 404510: 7.

[20] Y. Linde, A. Buzo, R. M. Gray. An algorithm for vector quantizer design. IEEE Transaction on Communication. 1980; Com-28(1): 84–95.

[21] T. Kohonen. Self-Organizing Maps. 3rd ed. New York: Springer; 2000.

[22] L. Ingber. Adaptive Simulated Annealing. In: H. A. Oliveira, A. Petraglia Jr., L. Ingber, M. A. S. Machado, M. R. Petraglia (Eds.), Stochastic Global Optimization and Its Applications with Fuzzy Adaptive Simulated Annealing. New York: Springer; 2012. p. 33–61.

[23] L. Ingber. Very fast simulated re-annealing. Mathematical Computer Modelling. 1989; 12(8): 967–973.

[24] H. Szu, R. Hartley. Fast simulated annealing. Physics Letters A. 1987; 122(3–4): 157–162.

[25] K. Rose. Deterministic annealing for clustering, compression, classification regression, and related optimization problems. Proceedings of the IEEE. 1998; 86(11): 2210–2239.

[26] K. Rose, E. Gurewitz, G. C. Fox. Constrained clustering as an optimization method. IEEE Transaction on Pattern Analysis and Machine Intelligence. 1993; 15(8): 785–794.

[27] J. Buhmann, H. Kuhnel. Vector quantization with complexity costs. IEEE Transaction on Information Theory. 1993; 39(4): 1133–1143.

[28] K. Rose, E. Gurewitz, G. C. Fox. Vector quantization by deterministic annealing. IEEE Transaction on Information Theory. 2002; 38(4): 1249–1257.

[29] N. Ueda, R. Nakano. Mixture density estimation via EM algorithm with deterministic annealing. In: Proceedings of 1994 IEEE Neural Networks for Signal Processing; 6–8 September 1994; Ermioni, Greek. IEEE; 1994. p. 69–77.

[30] D. Miller, A. V. Rao, K. Rose, A. Gersho. A global optimization technique for statistical classifier design. IEEE Transaction on Signal Processing. 1996; 44: 3108–3122.

[31] T. Hofmann, J. Buhmann. Pairwise data clustering by deterministic annealing. IEEE Transaction on Pattern Analysis and Machine Intelligence. 1997; 19: 1–14.

[32] E. Kim, W. Wang, H. Li, X. Huang. A parallel annealing method for automatic color cervigram image segmentation. In: Medical Image Computing and Computer Assisted Intervention, MICCAI-GRID 2009 HPC Workshop; 2009 September 20–24; London, UK; 2009.

[33] G. C. Fox, D. R. Mani, S. Pyne. Detailed results of evaluation of DAVS(c) deterministic annealing clustering and its application to LC-MS data analysis. 2013. Available from: http://grids.ucs.indiana.edu/ptliupages/publications/DAVS2.pdf

[34] G. C. Fox, D. R. Mani. Parallel deterministic annealing clustering and its application to LC-MS data analysis. In: 2013 IEEE International Conference on Big Data; 6–9 October 2013; Silicon Valley, USA. IEEE; 2013. p. 665–673.

[35] X. Qiu, G. C. Fox, H. Yuan, S. Bae, G. Chrysanthakopoulos, F. Nielsen. Performance of multicore systems on parallel data clustering with deterministic annealing. In: International Conference on Computer Science 2008; 23–25 June 2008; Krakow, Poland. Springer-Verlag; 2008. p. 407–416.

[36] M. Yasuda, T. Furuhashi, S. Okuma. Statistical mechanical analysis of fuzzy clustering based on fuzzy entropy. IEICE Transaction on Information and Systems. 2007; E90-D(6): 883–888.

[37] L. E. Reichl. A Modern Course in Statistical Physics. New York: John Wiley & Sons; 1998.

[38] S. M. Sait, H. Youssef. Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems. U.S.A.: Wiley-IEEE Computer Society Press; 2000.

[39] B. E. Rosen. Function optimization based on advanced simulated annealing. In: Proceedings of the IEEE Workshop on Physics and Computation (PhysComp '92); 2–4 October 1992; Dallas, U.S.A. IEEE; 1992. p. 289–293.

Chapter 2

Generalized Simulated Annealing

Yang Xiang, Sylvain Gubian and Florian Martin

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66071

Abstract

Many problems in mathematics, statistics, finance, biology, pharmacology, physics, applied mathematics, economics, and chemistry involve the determination of the global minimum of multidimensional real-valued functions. Simulated annealing methods have been widely used for different global optimization problems. Multiple versions of simulated annealing have been developed, including classical simulated annealing (CSA), fast simulated annealing (FSA), and generalized simulated annealing (GSA). After revisiting the basic idea of GSA using Tsallis statistics, we implemented a modified GSA approach using the R package GenSA. This package was designed to solve complicated nonlinear objective functions with a large number of local minima. In this chapter, we provide a brief introduction to this R package and demonstrate its utility by solving non-convexoptimization problems in different fields: physics, environmental science, and finance. We performed a comprehensive comparison between GenSA and other widely used R packages, including rgenoud and DEoptim. GenSA is useful and can provide a solution that is comparable with or even better than that provided by other widely used R packages for optimization.

Keywords: classical simulated annealing (CSA), fast simulated annealing (FSA), generalized simulated annealing (GSA), GenSA

1. Introduction

Determining the global minimum of a multidimensional function is the focus of many problems in statistics, biology, physics, applied mathematics, economics, and chemistry [1–6]. Although there is a wide spectrum of problems, computing the global minimum remains a challenging task, because, for example, modern problem dimensionality is increasing.

The optimization of convex functions is usually reasonably conducted using standard optimization approaches, such as simplex optimization, the steepest descent method, and

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

the quasi-Newton method. These methods can also provide reasonable results for the study of simple non-convex functions with only a few dimensions and well-separated local minima.

Deterministic methods are usually faster than stochastic methods although they tend to be trapped into a local minimum. To overcome this particular issue, stochastic methods have been widely developed and can determine a good approximation of the global minimum with a modest computational cost. Among stochastic methods, genetic algorithms [7], evolution algorithms [8], simulated annealing (SA) [9], and taboo search [10–12] have been successfully applied.

Among popular approaches, genetic algorithms [7] mimic the process of natural DNA evolution. In this approach, a population of randomly generated solutions is generated. The solutions are encoded as strings and evolve over many iterations toward better solutions. In each generation, the fitness of each individual in the population is evaluated, and in the next generation, strings are generated by crossover, mutation, and selection, based on their fitness. Differential evolution belongs to such genetic algorithms.

Ant colony optimization (ACO) [13] is another set of stochastic optimization methods, which is inspired by ants wandering to find food for the colony. An ant starts wandering randomly while laying down pheromone trails that will influence other ants because they will be attracted (increase in probability) by the trail, and if they eventually locate food, will return and reinforce the trail. To avoid the algorithm converging to local minima, the pheromone trail is set to evaporate proportionally to the time it takes to traverse the trail to decrease its attractiveness. As a consequence, the pheromone density of short paths becomes higher than that of longer paths. The design of ACO perfectly matches graph-based optimization (e.g., traveling salesman problem), but it can be adapted to determine the global minimum of real-valued functions [14] by allowing local random moves in the neighborhood of the current states of the ant.

The SA algorithm was inspired by the annealing process that takes place in metallurgy, whereby annealing a molten metal causes it to achieve its global minimum in terms of thermodynamic energy (crystalline state) [9]. In the SA algorithm, the objective function is treated as the energy function of a molten metal, and one or more artificial temperatures are introduced and gradually cooled, which is analogous to the annealing technique, to attempt to achieve the global minimum. To escape from local minima, this artificial temperature (or set of temperatures) acts as a source of stochasticity. Following the metallurgy analogy, at the end of the process, the system is foreseen to reside inside the attractive basin of the global minimum (or in one of the global minima if more than one global minimum exists). In classical simulated annealing (CSA), the visiting distribution is a Gaussian function (a local search distribution) for each temperature. It has been observed that this distribution is not optimal for moving across the entire search space [5]. Generalized simulated annealing (GSA) was developed to overcome this issue by using a distorted Cauchy-Lorentz distribution.

For a more extensive review of stochastic optimization algorithms, see the review provided by Fouskakis and Draper [15].

http://dx.doi.org/10.5772/66071

The R language and environment for statistical computing will be the language of choice in this chapter because it enables a fast implementation of algorithms, access to a variety of statistical modeling packages, and easy-to-use plotting functionalities. These advantages make the use of R preferable in many situations to other programming languages, such as Java, C++, Fortran, and Pascal [16].

In this chapter, we elaborate further on the background and improvements of GSA and the use of the R package GenSA [17], which is an implementation of a modified GSA. We will also discuss the performance of GenSA and show that it outperforms the genetic algorithm (R package rgenoud) and differential evolution (R package DEoptim) in an extensive testbed comprising 134 testing functions based on the success rate and number of function calls. The core function of GenSA is written in C++ to ensure that the package runs as efficiently as possible. The utility of this R package and its use will be presented by way of several applications, such as the famous Thomson problem in physics, non-convex portfolio optimization in finance, and kinetic modeling of pesticide degradation in environmental science.

2. Method

As mentioned above, SA methods attempt to determine the global minimum of any objective function by simulating the annealing process of a molten metal. Given an objective function $f(x)$ with $x = (x_1, x_2, \ldots, x_n)^T$, we attempt to determine its global minimum using SA. The general procedure for SA is as follows:

1. Generate an initial state $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)^T$ randomly and obtain its function value $E^0 = f(x^0)$. An initial temperature T^0 is set. *imax* is set to be any big integer.
2. For step $i = 1$ to *imax*,
   - The temperature T^i is decreased according to some cooling function.
   - Generate a new state $x^i = x^{i-1} + \Delta x$, where Δx follows a predefined visiting distribution (e.g., Gaussian distribution). $E^i = f(x^i)$ and $\Delta E = E^i - E^{i-1}$.
   - Calculate the probability p of $x^{i-1} \rightarrow x^i$.If $p < random(0,1)$, x^i is set back to its previous state x^{i-1} and E^i is also set back to E^{i-1}.
3. Output the final state x^{imax} and its function value E^{imax}.

We provide more details of SA methods as follows.

2.1. Classical simulated annealing (CSA)

According to the process of cooling and the visiting distribution, SA methods can be classified into several categories, amongwhich CSA [9], fast simulated annealing (FSA) [18], and the GSA [19] are the most common.

In CSA, proposed by Kirkpatrick et al., the visiting distribution is a Gaussian function, which is a local search distribution [5, 19]:

$$g(\Delta x) \propto \exp\left(-\frac{(\Delta x)^2}{T}\right), \tag{1}$$

where Δx is the trial jump distance of variable x and T is an artificial temperature in the reduced unit. In a local search distribution, for example, a Gaussian distribution Δx is always localized around zero. The jump is accepted if it is downhill of the energy/fitness/objective function. If the jump is uphill, it might be accepted according to an acceptance probability, which is computed using the Metropolis algorithm [20]:

$$p = \min\left[1,\ \exp\left(-\frac{\Delta E}{T}\right)\right]. \tag{2}$$

Geman and Geman [21] showed that for the classical case, a necessary and sufficient condition for having probability 1 of ending at the global minimum is that the temperature decreases logarithmically with the simulation time, which is impossible in practice because this would dramatically increase the computational time.

2.2. Fast simulated annealing (FSA)

In 1987, Szu and Hartley proposed a method called FSA [18], in which the Cauchy-Lorentz visiting distribution, that is, a semi-local search distribution, is introduced:

$$g(\Delta x) \propto \frac{T}{\left(T^2 + (\Delta x)^2\right)^{\frac{D+1}{2}}}, \tag{3}$$

where D is the dimension of the variable space. In a semi-local search distribution, for example, the Cauchy-Lorentz distribution, the jumps Δx are frequently local, but can occasionally be quite long. The temperature T in FSA decreases with the inverse of the simulation time, and the acceptance algorithm is the Metropolis algorithm shown in Eq. (2).

2.3. Generalized simulated annealing (GSA)

2.3.1. Introduction to GSA

A generalization of classical statistical mechanics was proposed by Tsallis and Stariolo [19]. In the Tsallis formalism, a generalized statistic is built from generalized entropy:

$$s_q = k\frac{1-\sum p_i^q}{q-1}, \tag{4}$$

where q is a real number, i is the index of the energy spectrum, and s_q tends to information entropy:

http://dx.doi.org/10.5772/66071

$$s = -k\sum p_i \ln p_i \tag{5}$$

when $q \to 1$. Maximizing the Tsallis entropy with the constraints

$$\begin{array}{l} \sum p_i = 1, \text{ and} \\ \sum p_i^q \varepsilon_i = const, \end{array} \tag{6}$$

where ε_i is the energy spectrum, and the generalized probability distribution is determined to be

$$p_i = \frac{[1-(1-q)\beta\varepsilon_i]^{\frac{1}{1-q}}}{z_q}, \tag{7}$$

where z_q is the normalization constant that ensures that p_i integrates to 1. This distribution pointwise converges to the Gibbs-Boltzmann distribution, where q tends to 1.

The GSA algorithm [19] refers to the generalization of both CSA and FSA according to Tsallis statistics. It uses the Tsallis-Stariolo form of the Cauchy-Lorentz visiting distribution, whose shape is controlled by the parameter q_v:

$$g_{q_v}\left(\Delta x(t)\right) \propto \frac{[T_{q_v}(t)]^{-\frac{D}{3-q_v}}}{\left[1+(q_v-1)\frac{\left(\Delta x(t)\right)^2}{[T_{q_v}(t)]^{\frac{2}{3-q_v}}}\right]^{\frac{1}{q_v-1}+\frac{D-1}{2}}}. \tag{8}$$

Please note that q_v also controls the rate of cooling:

$$T_{q_v}(t) = \frac{2^{q_v-1}-1}{(1+t)^{q_v-1}-1} T_{q_v}(1), \tag{9}$$

where T_{q_v} is the visiting temperature. In turn, a generalized Metropolis algorithm is used for the acceptance probability:

$$p_{q_a} = \min\left\{1, \ [1-(1-q_a)\beta\Delta E]^{\frac{1}{1-q_a}}\right\}, \tag{10}$$

where $\beta = 1/(KT_{q_a})$. The acceptance probability is controlled by the artificial temperature, T_{q_a}. When $q_v = 1$ and $q_a = 1$, then GSA is exactly CSA. Another special instance is given by $q_v = 2$ and $q_a = 1$ for which GSA is exactly FSA. Asymptotically, GSA has a similar behavior thanthe stochastic steepest descentas $T \to 0$. A faster cooling than CSA and FSA is achieved when $q_v > 2$.

It has been shown in a few instances that GSA is superior to FSA and CSA. Xiang et al. established that a pronounced reduction in the fluctuation of energy and a faster convergence

to the global minimum are achieved in the optimization of the Thomson problem and Nickel cluster structure [4–6]. Lemes et al. [22] observed a similar trend when optimizing the structure of a silicon cluster.

2.3.2. Improvement of GSA

A simple technique to accelerate convergence is as follows:

$$T_{q_a} = \frac{T_{q_v}}{t} \tag{11}$$

where T_{q_a} is the acceptance temperature, T_{q_v} is the visiting temperature, and t is the number of time steps. Our testing shows that this simple technique accelerates convergence [6].

The performance of GSA can be further improved by combining GSA with a local search method, large-scale bound-constrained Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [23] for a smooth function, or Nelder-Mead [24] for a non-smooth function. A local search is performed at the end of each Markov chain for GSA.

3. Results

The GenSA R package has been developed and added to the toolkit for solving optimization problems in the Comprehensive R Archive Network (CRAN) R packages repository. The package GenSA has proven to be a useful tool for solving global optimization problems [17].

3.1. Usage

The GenSA R package provides a unique function called GenSA whose arguments were described in the associated help [25]:

par: Numeric vector. Initial values for the parameters to be optimized over. Default value is NULL, in which case, the initial values will be generated automatically.

lower: Numeric vector with a length of par. Lower bounds for the parameters to be optimized over.

upper: Numeric vector with a length of par. Upper bounds for the parameters to be optimized over.

fn: A function to be minimized, where the first argument is the vector of parameters over which minimization is to take place. It should return a scalar result. The function has to always return a numerical value; however, not applicable (NA) and infinity are supported.

...: Allows the user to pass additional arguments into fn.

http://dx.doi.org/10.5772/66071

control: A list of control parameters, discussed below. The control argument is a list that can be used to control the behavior of the algorithm. Some components of control are the following:

- **maxit**: Integer. Maximum number of iterations of the algorithm. Default value is 5000, which could be increased by the user for the optimization of a very complicated objective function.
- **threshold.stop**: Numeric. The program will stop when the objective function value is≤ threshold.stop. Default value is NULL.
- **smooth**: Logical. TRUE when the objective function is smooth, or differentiable almost everywhere, and FALSE otherwise. Default value is TRUE.
- **max.call**: Integer. Maximum number of calls of the objective function. Default value is 10,000,000.
- **max.time**: Numeric.Maximum running time in seconds.
- **temperature**: Numeric. Initial value for the temperature.
- **visiting.param**: Numeric.Parameter for the visiting distribution.
- **acceptance.param**: Numeric.Parameter for the acceptance distribution.
- **verbose**: Logical. TRUE means that messages from the algorithm are shown. Default value is FALSE.
- **trace.mat**: Logical. Default value is TRUE, which means that the trace matrix will be available in the returned value of the GenSA call.

The default values of the control components are set for a complicated optimization problem. For typical optimization problems with medium complexity, GenSA can determine a reasonable solution quickly; thus,using the variable **threshold.stop** to the expected function value is recommended to make GenSA stop at an earlier stage. A maximum run time can be also set by **max.time** argument or **max.call** argument for setting the maximum run time or number of calls, respectively.

3.2. Performance

An extensive performance comparison of currently available R packages for continuous global optimization problems has been published [26]. In this comparison, 48 benchmark functions were used to compare 18 R packages for continuous global optimization. Performance was measured in terms of the quality of the solutions determined and speed. The author concluded that GenSA and rgenoud are recommended in general for continuous global optimization [26]. Based on this conclusion, we set out to perform a more extensive performance test by including more benchmark functions and the additional algorithm DEoptim. The SciPy Python scientific toolkit provides an extensive set of 196 benchmark functions. Because these 196

benchmark functions are coded in Python, we had to convert the Python code to R code. As a result, a subset containing 134 functions was available for this test. One hundred runs using a random initial starting point were performed for every combination of the 134 benchmark functions and the aforementioned three methods. We used a local search method to further refine the best solution provided by Deoptim, because this technique provides a more accurate final result [17]. The default values of the parameters of every package were used in this comparison. A tolerance of 1e-8 was used to establish whether the algorithm determines the global minimum value.

The reliability of a method can be measured by the success rate%, which is defined as the number of successful runs over 100 runs. For each testing function, the number of function calls required to achieve the global minimum was recorded for every method, and we refer to this as the number of function calls. Please note that rgenoud required a longer time to complete 100 runs compared with GenSA and DEoptim. **Table 1** shows the success rate% and the average number of function calls (in parentheses).

The mean of the success rate% over all the benchmark functions is 92, 85, and 86% for GenSA, DEoptim, and rgenoud, respectively. Because the number of function calls changes dramatically, the median rather than the mean of the number of function calls is provided: 244.3 for GenSA, 1625.9 for Deoptim, and 1772.1 for rgenoud.

A heatmap of the success rate% for GenSA, DEoptim, and rgenoud is displayed in **Figure 1**. The color scaling from red to green represents the success rate% from 0 to 100. Clearly, GenSA has a larger green region (high success rate%) than DEoptim and rgenoud.

Both the success rate% and number of function calls show that GenSA performed better than DEoptim and rgenoud in the testbed composed of 134 benchmark functions.

3.3. Applications

3.3.1. The Thomson problem in physics

The physicist J.J. Thomson posed the famous Thomson problem after proposing his plum pudding atomic model, based on his knowledge of the existence of negatively charged electrons within neutrally charged atoms [27]. The objective of the Thomson problem is to determine the minimum electrostatic potential energy configuration of N equal point charges constrained at the surface of a unit sphere that repel each other with a force given by Coulomb's law. The Thomson model has been widely studied in physics [28–31]. In the Thomson model, the energy function is

$$E = \frac{1}{2}\sum_{j\neq i}\frac{1}{|\vec{r}_i - \vec{r}_j|}. \tag{12}$$

The number of metastable structures (local minima) of the Thomson problem grows exponentially with N [28]. The region containing the global minimum is often small compared with those of other minima [30]. The Thomson problem has been selected as a benchmark for global

http://dx.doi.org/10.5772/66071

Function name	GenSA	DEoptim-LBFGS	rgenoud
AMGM	100% (93.0)	100% (62.6)	100.0% (88.8)
Ackley01	100% (746.2)	100% (1710.0)	100.0% (1840.1)
Ackley02	100% (182.9)	100% (1703.6)	100.0% (2341.6)
Ackley03	100% (352.5)	100% (1420.2)	100.0% (1860.9)
Adjiman	100% (33.3)	100% (1133.9)	100.0% (1677.5)
Alpine01	100% (756.0)	70% (2756.8)	95.0% (46852.2)
Alpine02	100% (56.4)	100% (913.6)	100.0% (1688.0)
BartelsConn	100% (263.1)	100% (1539.8)	100.0% (2343.5)
Beale	100% (145.6)	100% (1311.8)	100.0% (1711.0)
BiggsExp02	100% (85.6)	100% (763.3)	100.0% (1710.5)
BiggsExp03	100% (190.7)	100% (2614.4)	100.0% (1975.9)
BiggsExp04	100% (3498.8)	88% (8182.5)	100.0% (4234.5)
BiggsExp05	98% (40117.8)	14% (10864.2)	17.0% (13871.5)
Bird	100% (112.3)	100% (1777.3)	100.0% (1702.9)
Bohachevsky1	100% (875.1)	100% (1107.5)	100.0% (2673.1)
Bohachevsky2	100% (1372.9)	100% (1155.2)	76.0% (2554.5)
Bohachevsky3	100% (623.4)	100% (1342.4)	96.0% (2596.9)
BoxBetts	100% (129.2)	100% (1866.3)	100.0% (2018.8)
Branin01	100% (42.2)	100% (1747.6)	100.0% (1694.9)
Branin02	100% (1495.7)	28% (2830.9)	96.0% (1752.3)
Brent	100% (11.0)	100% (987.5)	100.0% (1687.9)
Bukin02	100% (39.9)	100% (1477.4)	100.0% (1679.7)
Bukin04	100% (217.9)	100% (1029.4)	100.0% (1744.2)
Bukin06	0% (NA)	0% (NA)	0.0% (NA)
CarromTable	100% (119.5)	100% (2040.9)	100.0% (1729.6)
Chichinadze	100% (517.4)	100% (1063.9)	92.0% (2219.8)
Colville	100% (4515.8)	100% (8230.9)	100.0% (2996.1)
CosineMixture	100% (22.0)	100% (3875.3)	100.0% (1670.6)
CrossInTray	100% (70.8)	100% (1512.8)	100.0% (1772.1)
CrossLegTable	0% (NA)	0% (NA)	2.0% (51829.0)
CrownedCross	0% (NA)	0% (NA)	2.0% (16045.0)
Cube	100% (1717.2)	100% (2030.3)	52.0% (21649.8)
DeVilliersGlasser01	100% (2343.8)	0% (NA)	1.0% (43919.0)
DeVilliersGlasser02	2% (173501.0)	0% (NA)	0.0% (NA)
Deb01	100% (57.1)	100% (4000.0)	100.0% (1700.8)
Deb03	100% (78.8)	100% (4028.9)	100.0% (1708.1)

Function name	GenSA	DEoptim-LBFGS	rgenoud
Decanomial	100% (1519.3)	100% (741.4)	100.0% (2050.8)
DeckkersAarts	100% (74.6)	100% (1525.0)	100.0% (1988.7)
Dolan	100% (2504.7)	1% (10293.0)	82.0% (25067.4)
DropWave	100% (3768.7)	85% (4009.8)	83.0% (2973.3)
Easom	97% (5077.5)	100% (1343.0)	100.0% (1875.7)
EggCrate	100% (122.9)	100% (912.2)	100.0% (1697.2)
ElAttarVidyasagarDutta	100% (675.7)	93% (1625.9)	100.0% (2115.7)
Exp2	100% (85.3)	100% (781.2)	100.0% (1707.7)
Exponential	100% (20.6)	100% (580.3)	100.0% (1682.5)
FreudensteinRoth	100% (395.4)	83% (2620.7)	100.0% (1700.9)
Gear	100% (2225.8)	100% (1118.0)	93.0% (1609.6)
Giunta	100% (39.6)	100% (592.3)	100.0% (1686.6)
GoldsteinPrice	100% (158.7)	100% (1023.0)	100.0% (1703.5)
Gulf	100% (1739.0)	100% (4076.4)	1.0% (1810.0)
Hansen	100% (149.7)	100% (104.0)	100.0% (132.0)
HimmelBlau	100% (53.7)	100% (2384.8)	100.0% (1698.7)
HolderTable	100% (138.0)	100% (2010.3)	100.0% (1701.7)
Hosaki	100% (49.8)	100% (583.2)	100.0% (1699.5)
Infinity	100% (225.8)	100% (113.4)	100.0% (492.0)
JennrichSampson	0% (NA)	0% (NA)	0.0% (NA)
Keane	100% (21.4)	100% (679.1)	100.0% (629.5)
Leon	100% (128.0)	100% (1338.2)	35.0% (2156.5)
Levy05	100% (152.8)	100% (144.7)	100.0% (224.6)
Levy13	100% (867.3)	100% (1138.5)	100.0% (1771.7)
Matyas	100% (33.8)	100% (967.7)	100.0% (1702.4)
McCormick	100% (42.2)	100% (747.8)	100.0% (1705.4)
Michalewicz	100% (97.3)	100% (69.0)	100.0% (157.3)
MieleCantrell	100% (2935.1)	100% (1942.7)	100.0% (3438.9)
Mishra03	81% (138334.7)	0% (NA)	24.0% (5745.0)
Mishra04	0% (NA)	0% (NA)	13.0% (1833.2)
Mishra05	100% (1289.1)	77% (1396.9)	100.0% (1680.0)
Mishra06	0% (NA)	0% (NA)	0.0% (NA)
Mishra07	100% (38.9)	100% (3055.9)	100.0% (1679.7)
Mishra08	100% (1519.3)	100% (741.4)	100.0% (1900.8)
Mishra09	100% (80.0)	85% (4832.0)	99.0% (12356.0)

http://dx.doi.org/10.5772/66071

Function name	GenSA	DEoptim-LBFGS	rgenoud
Mishra11	100% (30.9)	100% (3152.2)	100.0% (1613.1)
MultiModal	100% (134.5)	100% (481.8)	100.0% (1691.6)
NewFunction01	1% (243864.0)	0% (NA)	27.0% (9823.3)
NewFunction02	0% (NA)	0% (NA)	33.0% (12260.0)
Parsopoulos	100% (29.9)	100% (3267.7)	100.0% (1683.9)
Paviani	100% (423.3)	100% (18764.2)	100.0% (2168.7)
PenHolder	100% (94.3)	100% (1391.3)	100.0% (1701.6)
Plateau	100% (35.5)	100% (22.4)	100.0% (27.6)
Powell	100% (692.4)	100% (6501.1)	100.0% (2052.4)
Price01	100% (27.4)	100% (2107.0)	100.0% (1686.6)
Price02	100% (1242.4)	92% (4031.3)	85.0% (2087.7)
Price03	100% (3840.5)	64% (2574.6)	56.0% (1855.1)
Price04	100% (440.1)	100% (1075.4)	94.0% (6413.9)
Qing	0% (NA)	0% (NA)	0.0% (NA)
Quadratic	100% (26.0)	100% (872.2)	100.0% (1695.8)
Quintic	100% (928.2)	76% (2694.1)	36.0% (42496.5)
Rastrigin	100% (482.4)	98% (3753.6)	100.0% (1840.2)
Ripple01	100% (5742.4)	71% (4053.3)	99.0% (4758.6)
Ripple25	100% (414.7)	99% (3165.8)	100.0% (1771.5)
RosenbrockModified	100% (1343.5)	24% (3625.6)	95.0% (2329.0)
RotatedEllipse01	100% (26.0)	100% (1345.8)	100.0% (1699.9)
RotatedEllipse02	100% (28.0)	100% (1218.1)	100.0% (1700.8)
Salomon	95% (9790.3)	86% (4050.4)	21.0% (3143.1)
Schaffer01	99% (7105.6)	92% (3092.8)	59.0% (4520.3)
Schaffer02	100% (2078.9)	100% (2125.2)	100.0% (2677.8)
Schaffer03	100% (2786.3)	91% (4096.2)	99.0% (3236.3)
Schaffer04	100% (2920.9)	98% (4063.5)	86.0% (6505.1)
Schwefel01	100% (131.2)	100% (651.8)	100.0% (1734.5)
Schwefel04	100% (63.9)	100% (842.8)	100.0% (1698.8)
Schwefel06	100% (3536.0)	100% (2302.3)	100.0% (1790.4)
Schwefel20	100% (2079.8)	100% (1670.2)	100.0% (1779.1)
Schwefel21	100% (2158.6)	100% (1853.2)	100.0% (1774.8)
Schwefel22	100% (2772.0)	100% (1678.6)	100.0% (1775.0)
Schwefel26	100% (131.3)	100% (1648.2)	100.0% (1687.5)
Schwefel36	100% (361.3)	100% (1298.7)	100.0% (1930.6)

Function name	GenSA	DEoptim-LBFGS	rgenoud
SixHumpCamel	100% (83.3)	100% (909.2)	100.0% (1695.7)
Sodp	100% (64.3)	100% (477.7)	100.0% (1697.5)
Sphere	100% (17.4)	100% (730.2)	100.0% (1683.2)
Step	100% (471.0)	100% (219.3)	100.0% (1131.5)
Step2	100% (433.4)	100% (222.0)	100.0% (1177.1)
StyblinskiTang	100% (94.7)	100% (861.4)	100.0% (1824.2)
TestTubeHolder	100% (839.0)	98% (3957.5)	100.0% (2088.5)
ThreeHumpCamel	100% (86.3)	100% (817.5)	100.0% (1699.0)
Treccani	100% (49.1)	100% (1015.2)	100.0% (1696.4)
Trefethen	64% (20972.8)	0% (NA)	46.0% (29439.5)
Trigonometric02	100% (1766.2)	100% (1319.6)	99.0% (4809.6)
Ursem01	100% (47.7)	100% (682.1)	100.0% (1753.8)
Ursem03	100% (584.0)	100% (1645.7)	100.0% (1777.7)
Ursem04	100% (210.2)	100% (1372.3)	100.0% (1847.3)
UrsemWaves	100% (2565.0)	26% (4025.8)	50.0% (1674.8)
VenterSobiezcczanskiSobieski	100% (698.3)	100% (1748.0)	100.0% (2004.8)
Vincent	100% (41.4)	100% (3013.4)	100.0% (1703.5)
Wavy	100% (535.1)	100% (3110.8)	100.0% (1745.0)
WayburnSeader01	100% (156.1)	96% (2258.3)	93.0% (16665.5)
WayburnSeader02	100% (262.8)	100% (1772.3)	100.0% (1826.0)
Wolfe	100% (50.3)	100% (3252.9)	100.0% (1720.7)
XinSheYang02	100% (1048.0)	87% (2284.5)	100.0% (1768.5)
XinSheYang04	100% (457.0)	88% (3329.7)	100.0% (1777.5)
Xor	100% (26204.2)	48% (18245.1)	100.0% (1852.9)
YaoLiu09	100% (482.3)	98% (3753.6)	100.0% (1824.3)
Zacharov	100% (59.7)	100% (944.3)	100.0% (1696.6)
Zettl	100% (123.4)	100% (846.8)	100.0% (1712.5)
Zirilli	100% (131.0)	99% (702.3)	100.0% (1695.1)

Table 1. The success rate% and average number of function calls (in parentheses) for GenSA, DEoptim, and rgenoud.

optimization algorithms in a number of previous studies [4, 5, 28, 32]. The Thomson problem has been solved by both deterministic methods, including steepest descent [28], and stochastic methods, including (but not limited to) constrained global optimization (CGO) [29], the GSA algorithm [4, 5], genetic algorithms [30], and the Monte Carlo method [31, 33]. Typically, deterministic methods with multiple starts can provide a good solution when there are fewer point charges, whereas stochastic methods have to be used when N is large.

http://dx.doi.org/10.5772/66071

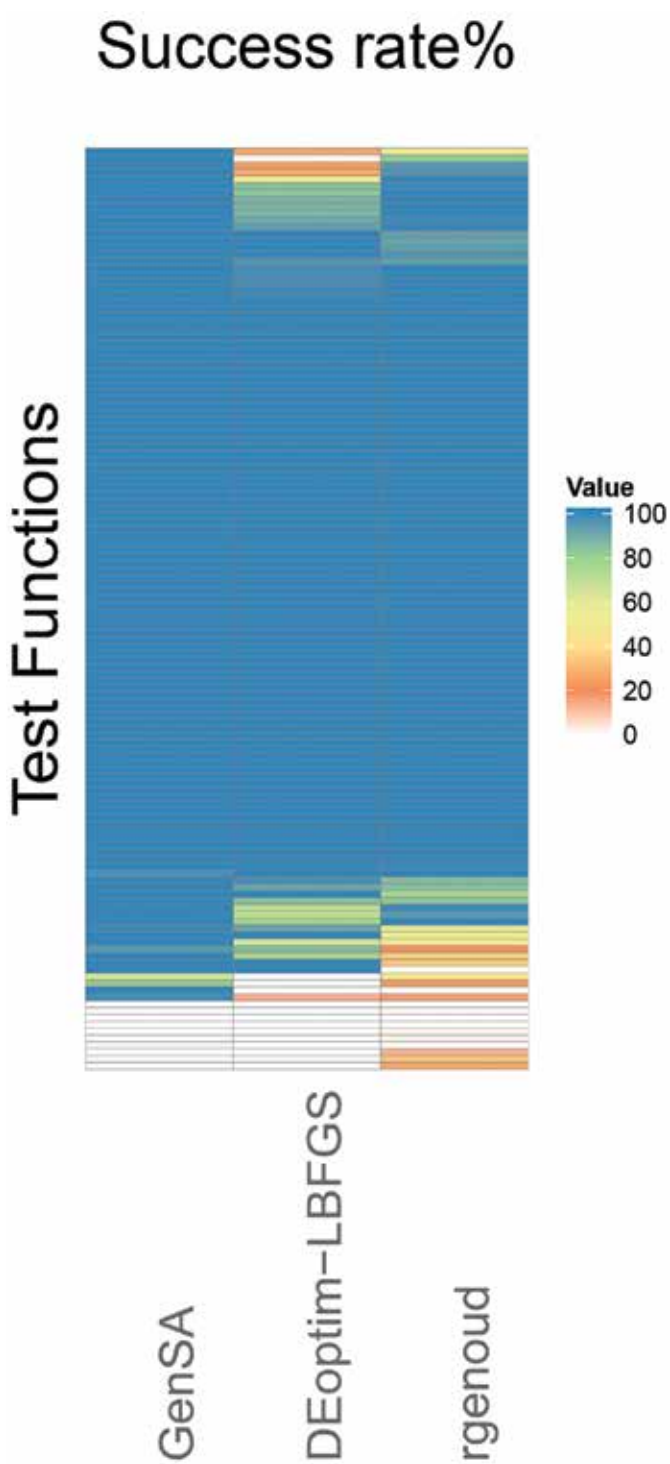

Figure 1. Heat map of the success rate% for GenSA, DEoptim, and rgenoud.

We can define an R function for the Thomson problem as follows:

```
>Thomson.fn<- function(x){
fn.call<<- fn.call + 1
x <- matrix(x, ncol = 2)
y <- t(apply(x, 1, function(z){
c(sin(z[1]) * cos(z[2]),
sin(z[1]) * sin(z[2]), cos(z[1]))}))
n <- nrow(x)
tmp<- matrix(NA, nrow = n, ncol = n)
index<- cbind(as.vector(row(tmp)), as.vector(col(tmp)))
index<- index[index[, 1] < index[, 2],, drop=F]
rdist<- apply(index, 1, function(z){
tmp<- 1/sqrt(sum((y[z[1],] - y[z[2],])^2))
})
res<- sum(rdist)
return(res)
}
```

In this example, we chose six point charges because our purpose is only to show how to use our package GenSA. The global energy minimum of six equal point charges on a unit sphere is 9.98528137 [28].

We applied GenSA with default settings to the Thomson problem. As the number of point charges is small, GenSA can determine the global minimum easily. We set the maximum number of function calls allowed, max.call, to 600:

```
>n.particles<- 6 # regular octahedron with global minimum
9.98528137
>lower.T<- rep(0, 2 * n.particles)
>upper.T<- c(rep(pi, n.particles), rep(2 * pi, n.particles))
>require(GenSA)
>options(digits = 9)
>set.seed(1234)
>fn.call<<- 0
>out.GenSA<- GenSA(par = NULL, lower = lower.T, upper = upper.T,
fn = Thomson.fn, control = list(max.call=600))
>print(out.GenSA[c("value", "counts")])
$value
[1] 9.98528137
$counts
[1] 600
>cat("GenSA call function", fn.call, "times.\n")
GenSA call function 600 times.
```

The global minimum 9.98528137 for six point charges is determined within 600 function calls.

http://dx.doi.org/10.5772/66071

3.3.2. Kinetic modeling of pesticide degradation

Various types of pesticides have been widely used in modern agriculture. It is important to calculate the concentration of a pesticide in groundwater and surface water. We will show how GenSA can be used to fit a degradation model for a parent compound with one transformation product. All the data and models are from the R packages mkin [34] and FOCUS (2006) [35].

After loading the library, we obtain the data (FOCUS Example Dataset D). The observed concentrations are in the column named "value" at the time specified in column "time" for the two observed variables named "parent" and "m1."

```
>require(mkin)
>require(GenSA)
>require(deSolve)
>options(digits = 9)
>set.seed(1234)
>str(FOCUS_2006_D)
'data.frame':44 obs. of3 variables:
$ name : Factor w/2 levels "m1","parent": 2 2 2 2 2 2 2 2 2 2 ...
$ time :num0 0 1 1 3 3 7 7 14 14 ...
$ value: num99.5 102 93.5 92.5 63.2 ...
```

The measured concentration of parent and m1 change with time is displayed in the lower panel of **Figure 2**.

According to the kinetic model displayed in the upper panel of **Figure 2**, we define the derivative function as follows:

```
>df<- function(t, y, parameters){
+d_parent<- -parameters["k_parent_sink"] * y["parent"] -
+parameters["k_parent_m1"] * y["parent"]
+d_m1 <- parameters["k_parent_m1"] * y["parent"] -
+parameters["k_m1_sink"] * y["m1"]
+return(list(c(d_parent, d_m1)))
+}
```

There is one initial concentration, parent_0, and three kinetic parameters, k_parent_m1, k_parent_sink, and k_m1_sink, whose values need to be fitted out by fitting the concentration curves. The initial concentration of m1, m1_0, is always zero. We define the objective function, fn, as the sum of the squares of residuals (deviations predicted from observed values for both the parent and m1):

```
>fn<- function(x,
+names_x = c("parent_0", "k_parent_sink", "k_parent_m1",
 "k_m1_sink"),
+names_y = c("parent", "m1"),
+names_parameters = c("k_parent_sink", "k_parent_m1",
 "k_m1_sink"),
```

```
+tspan = if (!is.null(dat.fitting))
+sort(unique(dat.fitting[["time"]])) else NULL,
+df, dat.fitting = NULL){
+m1_0 = 0
+names(x) <- names_x
+y0 <- c(x["parent_0"], m1_0)
+names(y0) <- names_y
+parameters<- x[c("k_parent_sink", "k_parent_m1", "k_m1_sink")]
+stopifnot(!is.null(tspan))
+out<- ode(y0, tspan, df, parameters)
+if (is.null(dat.fitting)){
+rss<- out
+} else{
+rss<- sum(sapply(c("parent", "m1"), function(nm){
+o.time<- as.character(dat.fitting[dat.fitting$name == nm,
"time"])
+sum((out[, nm][match(o.time, out[, "time"])] -
+dat.fitting[dat.fitting$name == nm, "value"])^2,
na.rm = TRUE)
+}))
+}
+return(rss)
+}
```

Then, we use GenSA to estimate the four parameters. As the model is not complicated, we limit the running time of GenSA to 5seconds by setting max.time=5:

```
>res<- GenSA(fn = fn, lower = c(90, rep(0.001, 3)),
+upper = c(110, rep(0.1, 3)),
+control = list(max.time = 5), df = df,
+dat.fitting = FOCUS_2006_D)
>names(res$par) <- c("parent_0", "k_parent_sink", "k_parent_m1",
"k_m1_sink")
>print(round(res$par, digits = 6))
parent_0 k_parent_sinkk_parent_m1k_m1_sink
99.5984910.0479200.0507780.005261
```

GenSA successfully determines the correct value of the initial concentration of the parent and the three kinetic parameters. The fitting curves for the parent and m1 are displayed in the lower panel of **Figure 2**.

3.3.3. Finance: risk allocation portfolios

Portfolio selection problems were addressed by mean-risk models in the 1950s. The most popular measures of downside risk are the value-at-risk (VaR) and conditional VaR(CVaR). Portfolio weights for which the portfolio has the lowest CVaR and each investment can

http://dx.doi.org/10.5772/66071

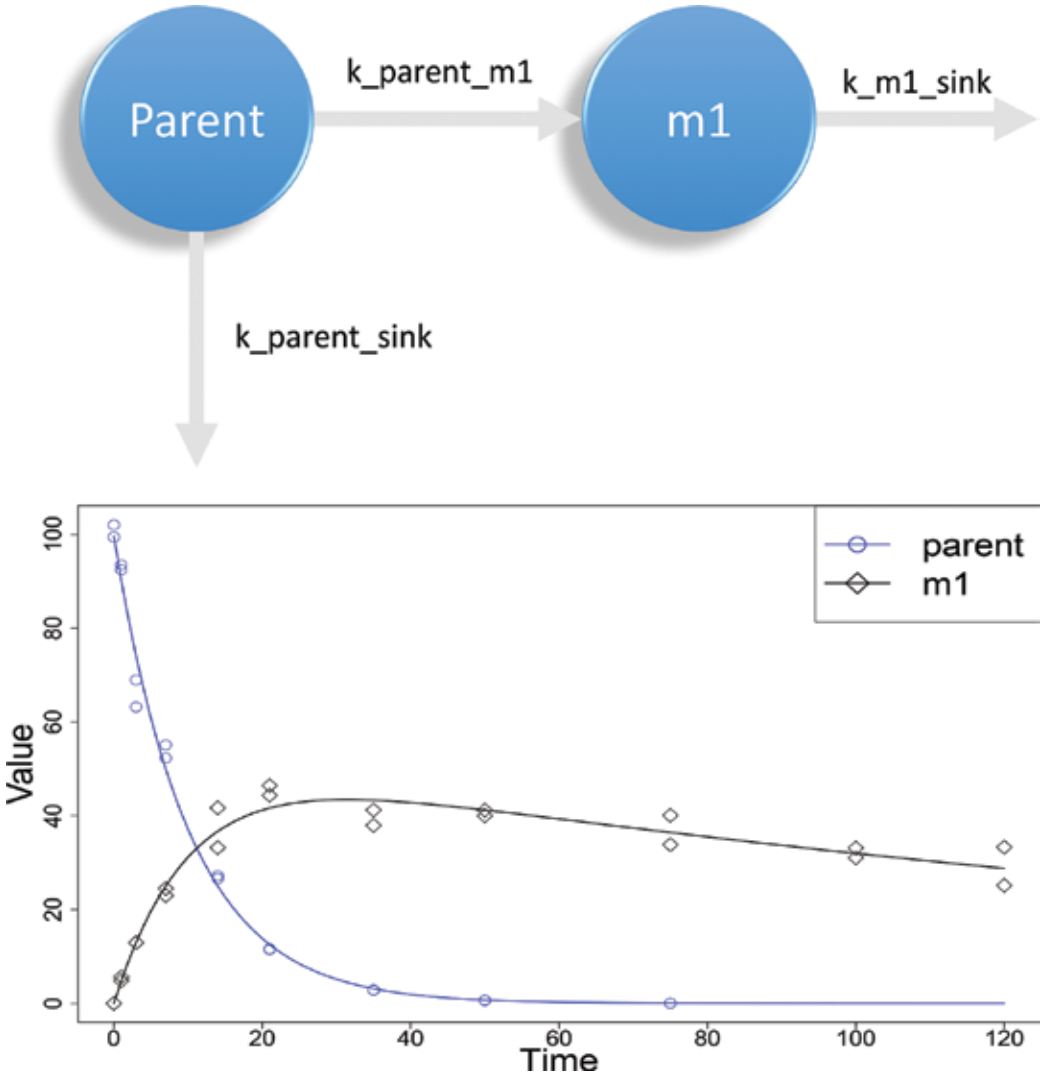

Figure 2. Kinetic modeling of pesticide degradation. Upper panel: illustration of pesticide degradation model. Lower panel: experimental concentration data and fitting curves for parent and m1.

contribute at most 22.5% to the total portfolio CVaR risk were estimated using differential evolution algorithms in Mullen et al. [16] and Ardia et al. [36]. The code for the objective function in portfolio optimization is rewritten below from Ardia et al. [36]:

```
>library("quantmod")
> tickers <- c("GE", "IBM", "JPM", "MSFT", "WMT")
>getSymbols(tickers, from = "2000-12-01", to = "2010-12-31")
[1] "GE" "IBM" "JPM" "MSFT" "WMT"
> P <- NULL
>for(ticker in tickers){
```

```
+tmp<- Cl(to.monthly(eval(parse(text = ticker))))
+P <- cbind(P, tmp)
+}
>colnames(P) <- tickers
> R <- diff(log(P))
> R <- R[-1,]
> mu <- colMeans(R)
> sigma <- cov(R)
>library("PerformanceAnalytics")
>pContribCVaR<- ES(weights = rep(0.2, 5),
+method = "gaussian", portfolio_method = "component",
+mu = mu, sigma = sigma)$pct_contrib_ES
>obj<- function(w) {
+if (sum(w) == 0) { w <- w + 1e-2}
+w <- w/sum(w)
+CVaR<- ES(weights = w,
+method = "gaussian", portfolio_method = "component",
+mu = mu, sigma = sigma)
+tmp1 <- CVaR$ES
+tmp2 <- max(CVaR$pct_contrib_ES - 0.225, 0)
+out <- tmp1 + 1e3 * tmp2
+return(out)
+}
```

GenSA can be used to determine the global optimum of this function using a bounded search domain from 0 to 1 values for the five parameters to be optimized:

```
>library(GenSA)
>lb<- rep(0, 5) # lower bounds, minimum values for all 5 parameters
are 0
>ub<- rep(1, 5) # upper bounds, maximum values for all 5 parameters
are 1
> out1.GenSA <- GenSA(fn = obj, lower = lb, upper = ub)
```

For non-differentiable objective functions, the smooth parameter in the control argument can be set to FALSE, which means that the Nelder-Mead method is used in the local search:

```
> out2.GenSA <- GenSA(fn=obj, lower=rep(0, 5), upper=rep(1, 5),
+control=list(smooth=FALSE, max.call=3000))
The max.call parameter is set to 3000 to make the algorithm stop
earlier:
> out2.GenSA$value
[1] 0.1141484884
> out2.GenSA$counts
[1] 3000
>cat("GenSA call functions", fn.call.GenSA, "times.\n")
GenSA call functions 3000 times.
>wstar.GenSA<- out2.GenSA$par
>wstar.GenSA<- wstar.GenSA/sum(wstar.GenSA)
```

http://dx.doi.org/10.5772/66071

```
>rbind(tickers, round(100 * wstar.GenSA, 2))
[,1] [,2] [,3] [,4] [,5]
tickers "GE" "IBM" "JPM" "MSFT" "WMT"
"18.92" "21.23" "8.33" "15.92" "35.6"
>100 * (sum(wstar.GenSA * mu) - mean(mu))
[1] 0.03790568876
```

GenSA determined a minimum of 0.1141484884 within 3000 function calls.

4. Discussion and conclusions

The discrete optimization problem, in particular, the feature selection problem, exists extensively. GSA can also be used for the discrete problem. Please refer to [37] for details.

GSA is a powerful method for the non-convex global optimization problem. We developed an R package GenSA based on Tsallis statistics and GSA. In an extensive performance testbed composed of 134 benchmark functions, GenSA provided a higher average success rate% and a smaller median of the number of function calls compared with two widely recognized R packages: DEoptim and rgenoud. GenSA is useful and can provide a solution that is comparable with or even better than that provided by other widely used R packages for optimization.

R is very good for program prototype. When there is a need for heavy computation, other computational languages, such as C/C++, Fortran, Java, and Python,are recommended. Considering both speed and usability, aPython version of GSA, PyGenSA, is being developed and will be released within the SciPy scientific toolkitat the end of 2016.

Acknowledgements

This work would not have been possible without the R open-source software project. We would like to thank our colleague Julia Hoeng for inspirational discussions and support. This study was funded by Philip Morris International.

Author details

Yang Xiang†*, Sylvain Gubian† and Florian Martin

*Address all correspondence to: yang.xiang@pmi.com

Philip Morris International R&D, Philip Morris Products S.A., Neuchâtel, Switzerland

† These authors contributed equally to this work

References

[1] Agostini FP, Soares-Pinto DD, Moret MA, Osthoff C, Pascutti PG. Generalized simulated annealing applied to protein folding studies. Journal of Computational Chemistry. 2006 Aug 1;27(11):1142–55.

[2] Serra P, Stanton AF, Kais S, Bleil RE. Comparison study of pivot methods for global optimization. The Journal of Chemical Physics. 1997 May 1;106(17):7170–7.

[3] De Guire V, Caron M, Scott N, Ménard C, Gaumont-Leclerc MF, Chartrand P, Major F, Ferbeyre G. Designing small multiple-target artificial RNAs. Nucleic Acids Research. 2010 Jul 1;38(13):e140.

[4] Xiang Y, Sun DY, Fan W, Gong XG. Generalized simulated annealing algorithm and its application to the Thomson model. Physics Letters A. 1997 Aug 25;233(3):216–20.

[5] Xiang Y, Gong XG. Efficiency of generalized simulated annealing. Physical Review E. 2000 Sep 1;62(3):4473.

[6] Xiang Y, Sun DY, Gong XG. Generalized simulated annealing studies on structures and properties of Ni n (n =2-55) clusters. The Journal of Physical Chemistry A. 2000 Mar 30;104(12):2746–51.

[7] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press, 839 Greene Street, Ann Arbor, MI 48104-3209; 1975.

[8] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 1997 Dec 1;11(4):341–59.

[9] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983;220(4598):671–80.

[10] Cvijovic D, Klinowski J. Taboo search: an approach to the multiple minima problem. Science. 1995 Feb 3;267(5198):664.

[11] Cvijović D, Klinowski J. Taboo search: An approach to the multiple-minima problem for continuous functions. In Handbook of Global Optimization 2002 (pp. 387-406). Springer Publishing Company, 11 West 42nd Street, 15th Floor, New York, NY 10036, US.

[12] Glover F, Taillard E. A user's guide to tabu search. Annals of Operations Research. 1993 Mar 1;41(1):1–28.

[13] Dorigo M. Optimization, learning and natural algorithms. Ph. D. Thesis, Politecnico di Milano, Italy. 1992.

[14] Toksari MD. Ant colony optimization for finding the global minimum. Applied Mathematics and Computation. 2006 May 1;176(1):308–16.

[15] Fouskakis D, Draper D. Stochastic optimization: a review. International Statistical Review. 2002 Dec 1;70(3):315–49.

[16] Mullen KM, Ardia D, Gil DL, Windover D, Cline J. DEoptim: An R package for global optimization by differential evolution. Journal of Statistical Software. 2011;40(6):1–26.

[17] Xiang Y, Gubian S, Suomela B, Hoeng J. Generalized simulated annealing for global optimization: the GenSA Package. R Journal. 2013 Jun 1;5(1):13–28.

[18] Szu H, Hartley R. Fast simulated annealing. Physics Letters A. 1987 Jun 8;122(3–4):157–62.

[19] Tsallis C, Stariolo DA. Generalized simulated annealing. Physica A: Statistical Mechanics and its Applications. 1996 Nov 15;233(1):395–406.

[20] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. The Journal of Chemical Physics. 1953 Jun 1;21 (6):1087–92.

[21] Geman S, Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence. 1984 Jun 6(6):721–41.

[22] Lemes MR, Zacharias CR, Dal Pino A. Generalized simulated annealing: Application to silicon clusters. Physical Review B. 1997 Oct 15;56(15):9279.

[23] Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing. 1995 Sep;16(5):1190–208.

[24] Nash JC. Compact numerical methods for computers: linear algebra and function minimisation. Boca Raton, FL: CRC Press; 1990.

[25] Gubian S, Xiang Y, Suomela B, Hoeng J. Package GenSA. Version 1.1.6. 2016. Available from: https://cran.r-project.org/web/packages/GenSA/GenSA.pdf.

[26] Mullen KM. Continuous global optimization in r. Journal of Statistical Software. 2014 Sep 28;60(6):1–45.

[27] Thomson JJ. XXIV. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1904 Mar 1;7(39):237–65.

[28] Erber T, Hockney GM. Equilibrium configurations of N equal charges on a sphere. Journal of Physics A: Mathematical and General. 1991 Dec 7;24(23):L1369.

[29] Altschuler EL, Williams TJ, Ratner ER, Dowla F, Wooten F. Method of constrained global optimization. Physical Review Letters. 1994 Apr 25;72(17):2671.

[30] Morris JR, Deaven DM, Ho KM. Genetic-algorithm energy minimization for point charges on a sphere. Physical Review B. 1996 Jan 15;53(4):R1740.

[31] Wales DJ, Ulker S. Structure and dynamics of spherical crystals characterized for the Thomson problem. Physical Review B. 2006 Dec 27;74(21):212101.

[32] Altschuler EL, Pérez–Garrido A. Defect-free global minima in Thomson's problem of charges on a sphere. Physical Review E. 2006 Mar 6;73(3):036108.

[33] Wales DJ, McKay H, Altschuler EL. Defect motifs for spherical topologies. Physical Review B. 2009 Jun 22;79(22):224115.

[34] Ranke J. mkin: Routines for fitting kinetic models with one or more state variables to chemical degradation data, 2013b. URL http://CRAN.R-project.org/package=mkin.

[35] Boesten JJ, Aden K, Beigel CY, Beulke S, Dust M, Dyson JS, Fomsgaard IS, Jones RL, Karlsson S, Van der Linden AM, Richter O. Guidance document on estimating persistence and degradation kinetics from environmental fate studies on pesticides in EU registration. Report of the FOCUS Work Group on Degradation Kinetics, EC Doc. Ref. Sanco/10058/2005, version. 2005;1.

[36] Ardia D, Boudt K, Carl P, Mullen KM, Peterson BG. Differential evolution with DEoptim. R Journal. 2011 Jun 1;3(1):27–34.

[37] Xiang Y, Hoeng J, Martin F, inventors; Philip Morris Products SA, assignee. Systems and methods for generating biomarker signatures with integrated dual ensemble and generalized simulated annealing techniques. United States patent application US 14/409,679.2013 Jun 21.

Chapter 3

A Simulated Annealing Based Optimization Algorithm

Yoel Tenne

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66455

Abstract

In modern engineering finding an optimal design is formulated as an optimization problem which involves evaluating a computationally expensive black-box function. To alleviate these difficulties, such problems are often solved by using a metamodel, which approximates the computer simulation and provides predicted values at a much lower computational cost. While metamodels can significantly improve the efficiency of the design process, they also introduce several challenges, such as a high evaluation cost, the need to effectively search the metamodel landscape and to locate good solutions, and the selection of which metamodel is most suitable to the problem being solved. To address these challenges, this chapter proposes an algorithm that uses a hybrid simulated annealing and SQP search to effectively search the metamodel. It also uses ensembles that combine prediction of several metamodels to improve the overall prediction accuracy. To further improve the ensemble accuracy, it adapts the ensemble topology during the search. Finally, to ensure convergence to a valid optimum in the presence of metamodel inaccuracies, the proposed algorithm operates within a trust-region framework. An extensive performance analysis based on both mathematical test functions and an engineering application shows the effectiveness of the proposed algorithm.

Keywords: simulated annealing, metamodelling, ensembles, trust-region, expensive optimization problems

1. Introduction

With the advent of high performance computing, intricate computer simulations can now model real-world physics with high accuracy. This, in turn, transformed the engineering design process into a simulation-driven process in which candidate designs are evaluated by

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

a computer simulation instead of a laboratory experiment. In this set-up, the design process is formulated as an optimization problem with several unique features [1]:

- The computer simulation acts as the objective function, since it assigns objective values to candidate designs. However, the simulation is often a legacy code or commercial software whose inner workings are inaccessible to the user, and so there is no analytic expression that defines how candidate designs are mapped to objective values. Such a *black-box function* precludes the use of optimization algorithms that require an analytic function.
- Each simulation run is *computationally expensive*, that is, it requires a lengthy run time, and this severely restricts the number of candidate designs that can be evaluated.
- Both the real-world physics being modelled and the numerical simulation process may result in an objective function, which has a complicated landscape, and this further complicates the optimization process.

An optimization strategy that has proven effective in such problems is that of *metamodel-assisted* optimization, namely, where a metamodel approximates the true expensive function and provides predicted objective values at a lower computational cost [1]. However, the integration of metamodels and ensembles into the optimization search introduces several challenges:

- Locating a good solution requires effectively searching the metamodel, which can have a complicated landscape with multiple local solutions, and hence can be a difficult task.
- Since function evaluations are expensive, only a small number of evaluated vectors will be available and hence the metamodel will be inaccurate. In severe cases, the optimization search can converge to a false optimum, namely, which was artificially created by the metamodel's inaccurate approximation of the true expensive function.
- A variety of metamodels have been proposed, for example, artificial neural networks (ANNs), Kriging and radial basis functions (RBFs) [2, 3], but the optimal variant is problem-dependant, and is typically not known *a priori*. In an attempt to address this issue, *ensembles* employ several metamodels concurrently and aggregate their individual predictions into a single one [4, 5]. However, the effectiveness of ensembles depends on their topology, namely, which metamodels they incorporate, but the optimal topology is again typically unknown *a priori*, and may be impractical to identify by numerical experiments due to the high cost of each simulation run.

To address this issue, this chapter proposes an optimization algorithm that uses a hybrid-simulated annealing (SA) search followed by a local refinement of solutions based on an SQP search. In this manner, this set-up achieves both an effective global and local search, which assists in locating good solutions. To address the issue of inaccurate metamodel predictions, the proposed algorithm operates within a trust region (TR) approach that manages the metamodel and ensures convergence to a valid optimum. Finally, to further improve the prediction accuracy the proposed algorithm uses ensembles and selects the most suitable topology during the search.

http://dx.doi.org/10.5772/66455

Accordingly, this chapter presents several contributions: (a) a hybrid SA-SQP metamodel-assisted search, (b) integration within a TR framework, and (c) continuous selection of the ensemble topology during the search. An extensive performance analysis based on both mathematical test functions and an engineering problem of airfoil shape optimization shows the effectiveness of the proposed algorithm.

The remainder of this chapter is organized as follows: Section 2 provides the background information, Section 3 describes the proposed algorithm, then Section 4 provides an extensive performance evaluation and discussion, and finally Section 5 concludes this chapter.

2. Background

2.1. Optimization techniques

As mentioned in Section 1, simulation-driven problems often include a challenging objective function, such as having multiple local optima or lacking an analytic expression. In such settings, classical gradient-based optimization algorithms can perform poorly, and therefore researchers have explored various alternative approaches [1]. One class of such algorithms is the *nature-inspired metaheuristics*, which include evolutionary algorithms (EAs), particle swarm optimization (PSO), and SA. The latter was inspired by the physics of the annealing process in metals: initially a metal has a high temperature and so particles have a high probability of moving to a higher energy state. As the metal cools in the annealing process, particles are more likely to move to a lower energy level than to a higher one. The process is completed once the system has reached the lowest possible energy level, typically its temperature of equilibrium with the environment. In the realm of global optimization, these mechanics have been translated into a heuristic search, which starts with an initial vector, namely, a candidate solution. During the search, the current vector is perturbed so that new vectors in its vicinity are obtained. These vectors are evaluated and replace the original vector if: (a) they are better, or (b) they are worse and with probability p, which is analogous to the energy state changes. As p decreases, the search is transformed from being explorative to being more localized. Two main parameters of the SA algorithm are the *annealing schedule*, namely, the duration of the search process, which is determined by the manner that the temperature is decreased, and the selection probability function, which defines the dynamic threshold for accepting a worse solution. Algorithm 1 gives a pseudocode of a baseline SA algorithm, while Section 3 gives the specific parameter settings of the SA implementation used in this study.

The underlying mechanism of the SA algorithm was originally proposed by Metropolis et al. [6], while the more common annealing-inspired formulation was later proposed by Černy [7] and Kirkpatrick et al. [8]. Since then the algorithm has been widely used in the literature, and some recent examples include [9] in finance, [10] in machine learning, [11] in chemical engineering and [12] in production line machine scheduling.

2.2. Expensive black-box problems

Computationally expensive optimization problems are common across engineering and science. Typically in such problems, candidate designs are parameterized as vectors of design variables and sent to the simulation for evaluation, as shown in **Figure 1**.

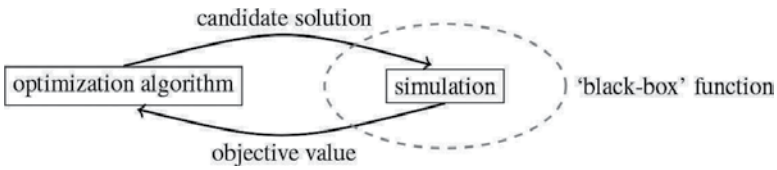

Figure 1. The layout of an expensive black-box optimization problem. The optimization algorithm generates candidate solutions, and these are evaluated by the simulation, which acts as a 'black-box' function, to obtain their corresponding objective values.

As mentioned in Section 1, metamodels are often used in such settings to alleviate the high computational cost of the simulation runs [2, 3]. However, the integration of metamodels into the search introduces two main challenges:

- Prediction uncertainty: Due to the high cost of the simulation runs only a small number of designs can be evaluated, which in turn degrades the prediction accuracy of the metamodel and leads to optimization with uncertainty regarding the validity of the predicted objective values [13]. To address this, the metamodel needs to be updated during the search to ensure that its accuracy is sufficient to drive the search to a correct final result. To accomplish this, the proposed algorithm is structured based on the TR approach [14]. In this way, the algorithm performs a sequence of *trial steps* that are constrained to the TR, namely, the region where the metamodel is assumed to be accurate. Based on the success of the trial step, namely, if a new optimum has been found, the TR and the set of vectors are updated. Section 3 presents a detailed description of the TR approach implemented in this study.
- Metamodel suitability: Various metamodel variants have been proposed, but the optimal variant is problem dependant and is typically unknown *a priori* [15]. To address this, ensembles employ multiple metamodels concurrently and combine their predictions into a single one to obtain a more accurate prediction [4, 5, 16]. **Figure 2** shows an example based on the Rosenbrock function.

  The ensemble *topology*, namely, which metamodel variants are incorporated, is typically determined *a priori* and is unchanged during the search. However, the topology directly affects the prediction accuracy, and hence an unsuitable topology can degrade the search effectiveness. As an example, RBFs, RBFN and Kriging metamodels (described in Appendix 1) were used to generate several ensemble topologies. The same testing and training samples were used with all the topologies (sized 30 and 20 vectors, respectively), such that each ensemble was trained with training sample and its prediction accuracy was tested on

http://dx.doi.org/10.5772/66455

the testing sample. The prediction accuracy was measured both with the root mean squared error (RMSE),

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{f}(x_i)-f(x_i)\right)^2} \tag{1}$$

and the R^2 indicator,

$$R^2 = \frac{S_t - S_r}{S_t}, \tag{2a}$$

where

$$S_t = \sum(f(x_i)-\overline{f(x_i)})^2, \quad S_r = \sum(f(x_i)-\hat{f}(x_i))^2, \tag{2b}$$

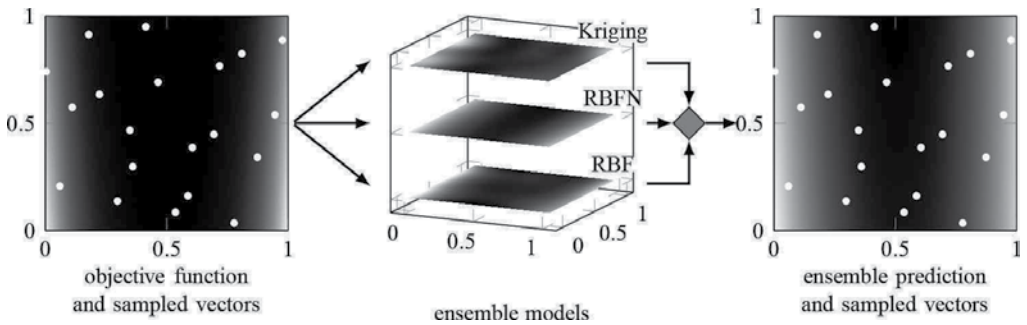

Figure 2. An ensemble topology consisting of RBF, RBFN and Kriging metamodels. The overall prediction is the aggregation of the individual predictions.

Ensemble topology						
	R+RN		**R+K**		**RN+K**	
Function	**RMSE**	R^2	**RMSE**	R^2	**RMSE**	R^2
Ackley-5D	**1.172e+00**	-1.830e-02	1.630e+00	-2.613e-02	1.176e+00	-9.521e-01
Rastrigin-10D	3.149e-02	-2.407e-01	**2.471e-02**	-1.812e-01	2.907e-02	-3.381e-01
Rosenbrock-20D	6.101e-04	-3.566e-01	4.968e-04	-4.541e-01	**3.859e-04**	-1.410e-01
Schwefel 2.13-30D	1.003e-03	-9.573e-01	**2.234e-04**	-7.182e-02	3.212e-04	-1.321e-01

Note: R: RBF, RN: RBF network, K: Kriging.

Table 1. Statistics for prediction accuracies of different topologies.

where $f(x)$, and $\hat{f}(x)$ are the true and the ensemble predicted values, respectively, and x_i, $i = 1\ldots n$ are the testing vectors. **Table 1** presents the test results, from which it follows that the prediction accuracy varied with the topology and that no single topology was the overall best.

Addressing this issue, the algorithm proposed in this study selects the most suitable ensemble topology during the search, as explained in the following section.

3. Proposed algorithm

This section describes the algorithm proposed in this study, which is designed to address the issues described in Sections 1 and 2. The proposed algorithm operates in five main steps, as follows:

Step 1. *Initialization*: The algorithm begins by generating an initial sample of vectors based on the optimal Latin hypercube design (OLHD) method [17]. The method ensures that the resultant sample is space-filling, namely, adequately covers the search space, which in turn improves the prediction accuracy of the metamodels. The OLHD method exploits patterns of point locations for optimal Latin hypercube designs based on a variation of the maximum distance criterion to produce near-optimal designs efficiently. After generating the sample, the vectors are evaluated with the true expensive function.

Step 2. *Ensemble selection*:

Step 2.1. Following the cross-validation (CV) procedure [18], the vectors that have been evaluated so far are split into a training set and a testing set. In turn, each candidate metamodel variant is trained with the training set and is tested on the testing set. The prediction accuracy is measured with the root mean squared error (RMSE), which is calculated as

$$\varepsilon_j = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\left(m_j(x_i) - f(x_i)\right)^2}, \tag{3}$$

where x_i, $i = 1\ldots l$, are the vectors in the testing set, $f(x_i)$ is the exact and already known function value at x_i, and $m_j(x)$ is the prediction of the jth metamodel variant that has been trained with the training set.

Step 2.2. The evaluated vectors are again split into a training and a testing set. For each candidate ensemble topology, the following steps are performed:

- Each metamodel variant that is active in the ensemble is assigned an *ensemble weight*, which is calculated as

$$w_i = \frac{\varepsilon_i^{-1}}{\sum_{j=1}^{n_m}\varepsilon_j^{-1}}, \tag{4}$$

and the overall ensemble prediction is then given by

http://dx.doi.org/10.5772/66455

$$\psi(x) = \sum_{i=1}^{n_m} w_i m_i(x), \tag{5}$$

where n_m is the number of participating metamodels in the ensemble and $m_i(x)$ is a metamodel that has been trained with the new training sample.

- The prediction accuracy of the ensemble is estimated based on its RMSE on the testing set

$$\varepsilon = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\psi(x_i) - f(x_i)\right)^2}, \tag{6}$$

where x_i, $i = 1 \ldots k$, are the vectors in the testing set.

Step 2.3. After repeating the process with all the candidate topologies, the one having the lowest RMSE, as defined in Eq. (6), is selected for the current optimization iteration. A new ensemble is then trained based on the selected topology and on all the evaluated vectors. This ensemble will be used in the following step.

It is emphasized that during the above process, the true expensive function is not used, and hence the process requires negligible computational resources.

For the specific implementation in this chapter, three well-established metamodels were considered for the ensembles: RBF, RBFN and Kriging that are all described in Appendix 1, while **Table 2** presents the candidate ensemble topologies. Also, the split ratio between the training and testing vectors was calibrated by numerical experiments, as described in Appendix 2.

Step 3. *Optimization trial step*: The proposed algorithm now seeks an optimum based on the ensemble prediction in the bounded TR around the current best solution (x_b), namely

$$\mathfrak{I} = \{x : \|x - x_b\|_2 \le \Delta\}, \tag{7}$$

where Δ is the TR radius. The search is performed by using a hybrid approach [19], which is composed of an initial search performed by a SA algorithm, and followed by a localized refinement of the solution with a sequential quadratic programming (SQP) algorithm. The main settings of the SA algorithm were based on existing studies [20–22], as follows:

- Initial temperature: $T_{max} = 1000 \cdot d$ where d is the function dimension. This was done to increase the annealing schedule for higher dimensional problems, which require a more extensive search.
- Final temperature (stopping condition): $T \le 10^{-8}$.
- Temperature decrease function: A 5% decrease, namely

$$T(t) = 0.95 \cdot T(t-1) \tag{8}$$

Metamodels participating in the ensemble			
Index	**RBF**	**RBFN**	**Kriging**
1	•		
2		•	
3			•
4	•	•	
5	•		•
6		•	•
7	•	•	•

Table 2. Candidate ensemble topologies.

where t is the current time-step counter. This way the temperature initially decreases at a slow rate, which assists the exploration search, but the reduction is much faster in later stages when the search is localized.

- Acceptance probability function: A decaying exponent, namely

$$p(T) = \exp\left(-\frac{f(x_n) - f(x_c)}{T(t)}\right) \tag{9}$$

where x_n is the new vector being examined and x_c is the current vector.

During the entire process, objective values are obtained only from the ensemble at a negligible computational cost, and hence the SA and SQP were able to evaluate a large number of candidate vectors.

Step 4. *TR updates*: The best vector found in the previous step ($x^\star$) is evaluated with the true expensive function, and the following updates are performed [14]:

- If $f(x^\star) < f(x_\mathrm{b})$: The trial step was successful since the vector found was indeed better than the current best one. This indicates that the ensemble prediction is accurate, and accordingly the TR radius is doubled.
- If $f(x^\star) \geq f(x_\mathrm{b})$ *and* there are sufficient vectors in the TR: The search was unsuccessful since the solution found is not better than the current best one. However, since there are enough vectors in the TR to train the metamodels the failure is attributed to the TR being too large, and accordingly the TR radius is halved.
- If $f(x^*) \geq f(x_\mathrm{c})$ *and* there are insufficient vectors in the TR: As above but now the failure is attributed to having too few vectors in the TR to train the metamodels with. Therefore, a new vector is sampled in the TR, as explained below.

The implementation described above differs from the classical TR framework in two main aspects:

- The TR is contracted only if the number of vectors in the TR is above a threshold n, which is done to avoid premature convergence. The threshold parameter was calibrated through numerical experiments, as described in Appendix 2.
- In Step 4, a new vector may be sampled in the TR to improve the prediction accuracy. This vector should be located in a region that is sparse with vectors, namely away from existing TR vectors. To accomplish this efficiently, the proposed algorithm generates a Latin hypercube design (LHD) sample of vectors in the TR and selects the one having the largest minimal distance (max-min criterion) from the existing TR vectors.

To conclude the description, Algorithm 2 gives the pseudocode of the proposed algorithm.

4. Performance analysis

4.1. Benchmark tests based on mathematical test functions

To evaluate its effectiveness, the proposed algorithm was applied to the established set of mathematical functions [23]: Ackley, Griewank, Rastrigin, Rosenbrock, Schwefel 2.13 and Weierstrass in dimensions 5–40, as listed in **Table 3**. These test functions represent challenging features such as high multimodality, deceptive landscapes and noise, and are therefore adequate for the testing purpose.

For a comprehensive evaluation, the proposed algorithm was benchmarked against the following four reference algorithms:

- *V1*: A variant of the proposed algorithm, which is identical to it in operation, except that it used a *single metamodel* (RBF), but no ensembles.
- *V2*: A variant of the proposed algorithm, which is identical to it in operation, except that it used a *fixed ensemble*, which consisted of the RBF, RBFN and Kriging metamodels, but without any topology selection.

Function	Definition, $f(\boldsymbol{x})=$	Domain
Ackley	$-20\exp\left(-0.2\sqrt{\sum_{i=1}^{d}x_i^2/d}\right)-\exp\left(\sum_{i=1}^{d}\cos(2\pi x_i)/d\right)+20+e$	$[-32,32]^d$
Griewank	$\sum_{i=1}^{d}\{x_i^2/4000\}-\prod_{i=1}^{d}\{\cos(x_i/\sqrt{i})\}+1$	$[-100,100]^d$
Rastrigin	$\sum_{i=1}^{d}\{x_i^2-10\cos(2\pi x_i)+10\}$	$[-5,5]^d$
Rosenbrock	$\sum_{i=1}^{d-1}\{100(x_i^2-x_{i+1})^2+(x_i-1)^2\}$	$[-10,10]^d$
Schwefel 2.13	$\sum_{i=1}^{d}\left\{\sum_{j=1}^{d}[\left(a_{i,j}\sin(\alpha_j)+b_{i,j}\cos(\alpha_j)\right)-\left(a_{i,j}\sin(x_j)+b_{i,j}\cos(x_j)\right)]\right\}^2$	$[-\pi,\pi]^d$
Weierstrass	$\sum_{i=1}^{d}\left\{\sum_{k=0}^{20}0.5^k\cos\left(2\pi 3^k(x_i+0.5)\right)\right\}-d\sum_{k=0}^{20}0.5^k\cos(\pi 3^k)$	$[-0.5,0.5]^d$

Table 3. Mathematical test functions.

- Evolutionary algorithm with periodic sampling (EA-PS): The algorithm leverages on the concepts in references [24, 25]. It uses a Kriging metamodel and a real-coded evolutionary algorithm (EA). The accuracy of the metamodel is maintained by periodically evaluating a small subset of the EA population with the true objective function, and using these sampled vectors to update the metamodel.
- *Expected improvement with covariance matrix adaptation evolutionary strategy* (CMA-ES) (EI-CMA-ES) [26]: The algorithm combines a covariance matrix adaptation evolutionary strategy (CMA-ES) algorithm with a Kriging metamodel, and uses the expected improvement (EI) framework to select new vectors for evaluation based both on the response and uncertainty in the metamodel prediction. This way the algorithm balances between a local search around the current best solution, and an explorative search in less-visited regions of the search space.

This testing set-up was used for several reasons: (i) any gains brought by using ensembles are highlighted through the comparisons to the V1 and V2 algorithms and (ii) the performance of the proposed algorithm is benchmarked against representative metamodel-assisted algorithms from the literature, namely, the EA-PS and EI-CMA-ES algorithms. Each algorithm-objective function combination was tested over 30 runs to support a valid statistical analysis, and the number of evaluations of the true objective function was limited to 200, to represent the tight optimization budget in expensive problems.

Tables 4 and **5** give the resultant test statistics of mean, standard deviation (SD), median, minimum (best) and maximum (worst) objective value for each algorithm-objective function combination. It also gives the statistic α that indicates the significance level at which the results of the proposed algorithm were better than those of the competing algorithms, measured at either the 0.01 or 0.05 levels, while an empty entry indicates no such statistically significant advantage. The α statistic was obtained by using the Mann-Whitney non-parametric test [27].

Test results show that the proposed algorithm performed well, as it obtained the best mean and median statistics in 8 out of 12 cases. Its results also had a statistically significant advantage over the other algorithms in 30 out of 48 cases. The performance advantage of the proposed algorithm was particularly pronounced in the high-dimensional cases, where it obtained the best mean and median in five out of six test functions. In terms of repeatability of performance, its SD was often slightly higher but was competitive with the best SD in each test case.

Overall, the proposed algorithm consistently outperformed the V1 and V2 variants, which shows that selecting the ensemble topology during the search improved the search effectiveness, both when compared to using a fixed metamodel or a fixed ensemble topology. Also, the proposed algorithm consistently outperformed the two reference algorithms from the literature, which shows that it compares well with existing approaches.

The analysis of the experiments also examined how the ensemble topology was updated, to examine if either a single or multiple topologies were predominantly selected. Accordingly, **Figure 3** shows representative plots of the ensemble topologies selected during a run with the Ackley-10D function and another with the Rosenbrock-20D function. It follows that in both

http://dx.doi.org/10.5772/66455

		Proposed	V1	V2	EA-PS	EI-CMA-ES
Ackley-10	Mean	7.576e+00	1.415e+01	1.320e+01	**5.241e+00**	1.796e+01
	SD	8.218e+00	4.522e+00	7.843e+00	5.590e-01	1.529e+00
	Median	**2.274e+00**	1.547e+01	1.855e+01	5.408e+00	1.797e+01
	Min(best)	8.677e-02	2.362e+00	3.381e+00	4.098e+00	1.443e+01
	Max(worst)	1.779e+01	1.778e+01	2.046e+01	6.010e+00	1.988e+01
	α			0.01		0.01
Griewank-10	Mean	**1.253e-01**	1.920e-01	8.625e-01	9.579e-01	9.338e-01
	SD	1.759e-01	1.643e-01	1.401e-01	1.076e-01	2.435e-01
	Median	**7.507e-02**	1.267e-01	8.834e-01	9.862e-01	1.007e+00
	Min(best)	9.163e-03	3.485e-02	5.304e-01	7.146e-01	2.441e-01
	Max(worst)	6.194e-01	5.401e-01	1.022e+00	1.046e+00	1.050e+00
	α			0.01	0.01	0.01
Rastrigin-5	Mean	**6.259e+00**	9.031e+00	7.828e+00	7.631e+00	2.131e+01
	SD	3.695e+00	7.483e+00	8.286e+00	4.811e+00	4.890e+00
	Median	5.844e+00	7.273e+00	**4.246e+00**	7.226e+00	2.139e+01
	Min(best)	1.950e+00	1.005e+00	3.224e+00	1.621e+00	1.353e+01
	Max(worst)	1.189e+01	2.653e+01	3.046e+01	1.456e+01	3.006e+01
	α					0.01
Rosenbrock-5	Mean	**1.443e+01**	3.247e+01	1.358e+02	2.074e+02	3.701e+02
	SD	3.839e+01	7.480e+01	2.784e+02	1.640e+02	2.320e+02
	Median	**2.508e+00**	3.522e+00	6.636e+00	1.796e+02	3.498e+02
	Min(best)	2.465e-02	1.730e+00	4.085e+00	1.368e+01	7.677e+01
	Max(worst)	1.236e+02	2.388e+02	8.787e+02	5.617e+02	6.719e+02
	α			0.01	0.01	0.01
Schwefel-5	Mean	5.379e+02	3.809e+02	**3.663e+02**	5.598e+02	3.333e+02
	SD	8.894e+02	9.297e+02	6.567e+02	4.995e+02	3.227e+02
	Median	7.371e+01	**2.749e+00**	1.715e+02	4.804e+02	2.050e+02
	Min(best)	2.508e-02	5.862e-02	4.835e+01	5.685e+01	3.426e+01
	Max(worst)	2.202e+03	2.955e+03	2.215e+03	1.817e+03	1.080e+03
	α					
Weierstrass-10	Mean	6.620e+00	8.936e+00	8.168e+00	**3.706e+00**	5.909e+00
	SD	1.970e+00	1.731e+00	2.042e+00	5.593e-01	2.777e+00
	Median	6.395e+00	9.023e+00	8.494e+00	**3.702e+00**	5.805e+00
	Min(best)	3.786e+00	6.875e+00	5.234e+00	2.787e+00	1.657e+00
	Max(worst)	1.052e+01	1.236e+01	1.108e+01	4.627e+00	9.409e+00
	α		0.01			

Table 4. Test statistics: test functions—low dimension.

		Proposed	V1	V2	EA-PS	EI-CMA-ES
Ackley-20	Mean	**6.446e+00**	8.795e+00	1.947e+01	6.814e+00	1.863e+01
	SD	5.712e+00	6.302e+00	2.836e-01	2.461e-01	1.921e+00
	Median	**3.885e+00**	5.524e+00	1.944e+01	6.744e+00	1.934e+01
	Min(best)	2.683e+00	3.574e+00	1.909e+01	6.468e+00	1.493e+01
	Max(worst)	1.802e+01	1.804e+01	1.998e+01	7.203e+00	2.044e+01
	α		0.05	0.01	0.05	0.01
Griewank-40	Mean	**1.045e+00**	1.270e+00	8.192e+00	1.461e+00	1.102e+00
	SD	2.942e-02	1.185e-01	1.151e+00	6.031e-02	3.032e-02
	Median	**1.040e+00**	1.246e+00	8.068e+00	1.454e+00	1.096e+00
	Min(best)	1.013e+00	1.120e+00	6.567e+00	1.387e+00	1.071e+00
	Max(worst)	1.114e+00	1.475e+00	1.035e+01	1.595e+00	1.157e+00
	α		0.01	0.01	0.01	0.01
Rastrigin-20	Mean	**6.490e+01**	6.501e+01	1.492e+02	1.223e+02	2.105e+02
	SD	3.753e+01	1.703e+01	2.754e+01	1.219e+01	3.914e+01
	Median	**4.808e+01**	6.683e+01	1.449e+02	1.230e+02	2.296e+02
	Min(best)	3.924e+01	4.203e+01	1.144e+02	1.046e+02	1.395e+02
	Max(worst)	1.568e+02	8.807e+01	1.983e+02	1.429e+02	2.507e+02
	α			0.01	0.01	0.01
Rosenbrock-20	Mean	**5.653e+02**	1.005e+03	9.013e+03	8.435e+02	3.967e+03
	SD	2.067e+02	5.794e+02	5.142e+03	3.012e+02	9.406e+02
	Median	**5.807e+02**	8.295e+02	8.112e+03	7.782e+02	3.685e+03
	Min(best)	2.042e+02	5.281e+02	4.165e+03	4.676e+02	3.141e+03
	Max(worst)	8.756e+02	2.497e+03	2.271e+04	1.439e+03	6.144e+03
	α		0.01	0.01	0.05	0.01
Schwefel-40	Mean	**7.503e+05**	8.786e+05	2.322e+06	1.774e+06	1.667e+06
	SD	2.173e+05	2.506e+05	5.317e+05	2.509e+05	6.520e+05
	Median	**7.074e+05**	8.420e+05	2.369e+06	1.744e+06	1.528e+06
	Min(best)	4.989e+05	5.815e+05	1.666e+06	1.415e+06	8.933e+05
	Max(worst)	1.126e+06	1.348e+06	3.186e+06	2.104e+06	2.871e+06
	α			0.01	0.01	0.01
Weierstrass-40	Mean	2.730e+01	4.048e+01	5.105e+01	**2.343e+01**	3.598e+01
	SD	4.414e+00	4.106e+00	2.138e+00	1.265e+00	1.463e+01
	Median	2.445e+01	4.100e+01	5.135e+01	**2.304e+01**	2.597e+01
	Min(best)	2.365e+01	3.351e+01	4.817e+01	2.214e+01	2.100e+01
	Max(worst)	3.419e+01	4.685e+01	5.333e+01	2.567e+01	5.817e+01
	α		0.01	0.01		

Table 5. Test statistics: test functions—high dimension.

http://dx.doi.org/10.5772/66455

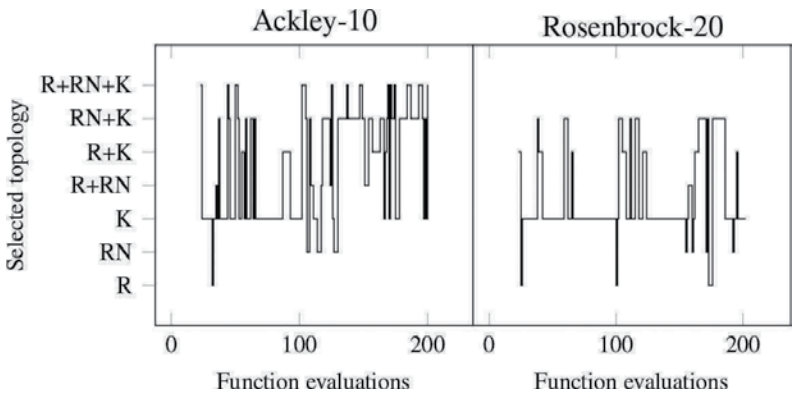

Figure 3. Ensemble topologies selected during two test runs. Abbreviations: K: Kriging, R: RBF, RN: RBF network.

cases, different topologies were selected, which indicate that no single topology was the overall optimal, and further justifies the proposed approach.

4.2. Engineering test problem

The proposed algorithm was also applied to a representative simulation-driven engineering problem, where the goal is to find an airfoil shape, which maximizes the lift L and minimizes the drag (aerodynamic friction) D at some prescribed flight conditions. In practise, the design requirements for airfoils are specified in terms of the non-dimensional *lift and drag coefficients*, c_l and c_d, respectively, defined as

$$c_l = \frac{L}{\frac{1}{2}\rho V^2 S} \tag{10a}$$

$$c_d = \frac{D}{\frac{1}{2}\rho V^2 S} \tag{10b}$$

where L and D are the lift and drag forces, respectively, ρ is the air density, V is the aircraft speed, and S is the reference area, such as the wing area. The relevant flight conditions are the aircraft altitude, speed and angle of attack (AOA) that is the angle between the aircraft velocity and the airfoil *chord line*. **Figure 4** gives a schematic layout of the airfoil problem.

Candidate airfoils were represented with the Hicks-Henne method [28], such that an airfoil profile was defined as

$$y = y_b + \sum_{i=1}^{h} \alpha_i b_i(x) \,, \tag{11}$$

where y_b is a baseline profile taken here to be the NACA0012 symmetric profile, and b_i are the shape basis functions [29].

$$b_{i(x)} = \left[\sin\left(\pi x^{\frac{\log((0.5)}{\log(i/(h+1))}}\right)\right]^4, \tag{12}$$

where $\alpha_i \in [-0.01, 0.01]$ are the variables, as shown in **Figure 4**. Ten basis functions were used for the upper and lower airfoil profiles, respectively, which resulted in a total of 20 variables per airfoil. Also, for structural integrity the thickness of an airfoil between 0.2 and 0.8 of its chord line was required to be greater than a critical thickness $t^\star = 0.1$. The lift and drag coefficients of candidate airfoils were obtained by using *XFoil*, an aerodynamics simulation code for subsonic isolated airfoils [30]. Each airfoil evaluation required up to 30 s on a desktop computer, so evaluations were not prohibitively expensive and the tests could be completed within a reasonable time.

Based on the above discussion, the objective function used was

$$f = -\frac{c_l}{c_d} + p\,, \quad p = \begin{cases} \frac{t^\star}{t} \cdot |\frac{c_l}{c_d}| & \text{if } t < \quad t^\star \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where p is the penalty for violation of the thickness constraint. The flight conditions were an altitude of 30,000 ft, a speed of Mach 0.7, namely 70% of the speed of sound, and an AOA of 2°.

Tests were performed along the set-up of Section 4.1, and **Table 6** gives the resultant test statistics. The trends are consistent with those of the test functions, and the proposed algorithm

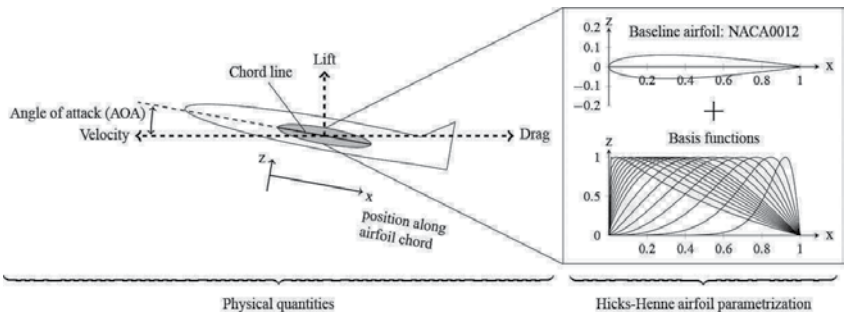

Figure 4. The layout of the airfoil problem: main components (left) and parameterization (right).

	Proposed	**V1**	**V2**	**EA-PS**	**EI-CMA-ES**
Mean	**-3.376e+00**	-3.279e+00	-3.368e+00	-3.231e+00	-3.278e+00
SD	1.242e-01	6.683e-02	1.008e-01	7.164e-02	9.597e-02
Median	**-3.355e+00**	-3.274e+00	-3.352e+00	-3.227e+00	-3.267e+00
Min(best)	-3.624e+00	-3.393e+00	-3.533e+00	-3.335e+00	-3.395e+00
Max(worst)	-3.158e+00	-3.161e+00	-3.214e+00	-3.134e+00	-3.098e+00
α		0.01		0.01	0.05

Table 6. Test statistics for the airfoil problem.

outperformed the other candidate algorithms also here. It obtained the best mean and median statistics, and had a competitively low SD.

5. Conclusion

While computer simulations can improve the efficiency of the engineering design process, they also introduce new optimization challenges. Metamodels aim to alleviate the challenge of a high evaluation cost by providing computationally cheaper approximations of the true expensive function.

While metamodels can significantly improve the search effectiveness, they also introduce various challenges, such as identifying an optimal combination of metamodel variants, and effectively searching the metamodel landscape. To address these issues, this chapter has proposed a hybrid algorithm that uses SA to perform a global search, and it then refines the solutions with an SQP local search. To further enhance its effectiveness, the proposed algorithm uses ensembles of metamodels and selects the most suitable ensemble topology during the search. Lastly, to ensure convergence to an optimum of the true expensive function in the light of the inherent metamodel inaccuracies, the proposed algorithm operates within a TR approach such that the optimization is performed through a series of trial steps.

In an extensive performance analysis, the proposed algorithm was benchmarked against two implementations without selection of the ensemble topology, and two reference algorithms from the literature, which also do not use topology adaption. Analysis of the results shows that the proposed algorithm consistently outperformed the other algorithms: it achieved better results in 30 out of 48 cases with mathematical test functions, and also performed well with a simulation-driven problem. Its performance advantage was evident from the superior mean and median statistics that was obtained across the tests and was particularly pronounced in the high-dimensional problems.

The analysis also showed that during the optimization search the optimal topology continuously varied during the search, and that no single topology was the overall optimal. This observation further supports the approach proposed of selecting the ensemble topology during the search.

Overall, the solid performance of the proposed algorithm shows the merit of the hybrid SA +SQP algorithm proposed. It also suggests that the proposed algorithm could be applied to problems from a variety of academic domains, such as scheduling, systems engineering and model calibration, to name a few.

Appendix 1: Candidate metamodels

This appendix describes the metamodels used, as follows:

- Kriging: A statistical metamodel that combines a global ‘drift’ function and a local adjustment based on the correlation between the sample vectors. The metamodel replicates the

observed responses precisely (Lagrangian interpolation). For a constant drift function, the metamodel becomes

$$m(\boldsymbol{x}) = \beta + \kappa(\boldsymbol{x}), \tag{14}$$

where $\kappa(\boldsymbol{x})$ is the local correction. The latter is defined by a stationary Gaussian process with mean zero and covariance

$$Cov[\kappa(\boldsymbol{x})\kappa(\boldsymbol{y})] = \sigma^2 c(\theta,\boldsymbol{x},\boldsymbol{y}), \tag{15}$$

where $c(\theta,\boldsymbol{x},\boldsymbol{y})$ is a user-prescribed correlation function. With the Gaussian correlation function [3]

$$c(\theta,\boldsymbol{x},\boldsymbol{y}) = \prod_{i=1}^{d} \exp\left(-\theta(x_i - y_i)^2\right), \tag{16}$$

the above metamodel becomes

$$m(\boldsymbol{x}) = \hat{\beta} + \boldsymbol{r}(\boldsymbol{x})^T R^{-1}(\boldsymbol{f} - \mathbf{1}\hat{\boldsymbol{\beta}}) \tag{17}$$

where $\hat{\beta}$ is the estimated drift coefficient, R is the symmetric matrix of correlations between all interpolation vectors, $\boldsymbol{f}$ is the vector of objective values and $\mathbf{1}$ is a vector with all elements equal to 1. $\boldsymbol{r}^T$ is the correlation vector between a new vector $\mathbf{x}$ and the sample vectors, namely,

$$\boldsymbol{r}^T = [c(\theta,\boldsymbol{x},\boldsymbol{x}_1), \ldots, c(\theta,\boldsymbol{x},\boldsymbol{x}_n)] \tag{18}$$

The estimated drift coefficient $\hat{\beta}$ and variance $\hat{\sigma}^2$ are calculated as

$$\hat{\beta} = (\mathbf{1}^T R^{-1}\mathbf{1})^{-1}\mathbf{1}^T R^{-1} f, \tag{19a}$$

$$\hat{\sigma}^2 = \frac{1}{n}[(\boldsymbol{f} - \mathbf{1}\hat{\beta})^T R^{-1}(f - \mathbf{1}\hat{\beta})] \tag{19b}$$

For an isotropic (single correlation parameter) Kriging metamodel the optimal value of the parameter is obtained by maximizing the metamodel likelihood [31]

$$\theta^\star : \max\{|R|^{1/n}\hat{\sigma}^2\} \tag{20}$$

- *Radial basis functions* (RBF): The metamodel approximates the true objective function with a set of basis functions, namely

http://dx.doi.org/10.5772/66455

$$m(x) = \alpha_i \sum_{i=1}^{n} \phi_i(x) + c \tag{21}$$

where $\phi_i(\boldsymbol{x})$ are basis functions of the form

$$\phi_i(\boldsymbol{x}) = \phi(\|\boldsymbol{x}-\boldsymbol{x}_i\|_2), \tag{22}$$

where $\boldsymbol{x}_i$ is a sampled vector and c is a constant. The coefficients α_i and c are determined from the interpolation conditions

$$m(\boldsymbol{x}_i) = f(\boldsymbol{x}_i), i = 1\ldots n, \tag{23a}$$

$$\sum_{i=1}^{n} \alpha_i = 0\,. \tag{23b}$$

In this study, the widely used Gaussian basis function [32] was used:

$$\phi_i(\boldsymbol{x}) = \exp\left(-\frac{\boldsymbol{x}-\boldsymbol{x}_i}{\tau}\right), \tag{24}$$

where τ controls the width of the Gaussians, and is determined by cross-validation [33, 34].

- *Radial basis function network* (RBFN): A variant of the RBF approach but in which the number of basis functions is smaller than the sample size, which can improve the prediction accuracy in certain scenarios. The metamodel is given by

$$m(\boldsymbol{x}) = \sum_{j=1}^{\hat{n}} \alpha_j \phi_j(\boldsymbol{x}), \tag{25}$$

where the coefficients α_i are determined from the least-squares interpolation conditions

$$\Phi^T \Phi \alpha = \Phi^T \boldsymbol{f} \tag{26}$$

where

$$\Phi_{i,j} = \phi_j(\boldsymbol{x}_i) \tag{27}$$

and $\boldsymbol{x}_i$, $i = 1\ldots n$, are the sample vectors, $\mathbf{f}$ is the vector of corresponding objective function values, and $\phi_j(\boldsymbol{x})$, $j = 1\ldots\hat{n}$ are the basis functions, which in this study were taken as the Gaussian functions described above. The basic function centres $\boldsymbol{x}_j$ are obtained by clustering the sampled vectors and using the resultant cluster centres.

Appendix 2: Parameter sensitivity analysis

As described in Section 3, the proposed algorithm relies on two main parameters: (i) the minimum number (n) of TR vectors needed to allow a TR contraction, and (ii) the split ratio (s) between the training and testing subsets, as used in estimating the accuracy of candidate metamodel and ensembles.

To calibrate these parameters, a set of numerical experiments were performed where different parameter settings were used, and for each setting the algorithm was tested with the Rastrigin-10D, Rosenbrock-10D, Rastrigin-20D and Rosenbrock-20D functions. The parameter settings examined were:

- n: $0.1d$, $0.5d$, d, where d is the dimension of the objective function.
- s: 80–20, 60–40, 40–60, in percent.

Table 7 gives the resultant test statistics of mean objective value, rank per objective function and the overall rank based on each setting where a lower score is better. From these results it follows:

(a) Statistics: different TR vectors threshold (n)

	$n = 0.1d$		$n = 0.5d$		$n = d$	
Function	**Mean**	**Rank**	**Mean**	**Rank**	**Mean**	**Rank**
Ras-10	4.598e+01	02	4.827e+01	03	3.530e+01	01
Ros-10	1.233e+02	02	4.393e+01	01	1.317e+02	03
Ras-20	9.256e+01	03	8.656e+01	02	7.997e+01	01
Ros-20	4.503e+02	01	5.361e+02	02	8.518e+02	03
Overall		08		08		08

(b) Statistics: different split ratios (s)

	80–20		**60–40**		**40–60**	
Function	**Mean**	**Rank**	**Mean**	**Rank**	**Mean**	**Rank**
Ras-10	4.598e+01	02	3.981e+01	01	5.414e+01	03
Ros-10	1.233e+02	02	1.266e+02	03	4.929e+01	01
Ras-20	9.256e+01	03	8.899e+01	02	7.970e+01	01
Ros-20	4.503e+02	01	5.578e+02	03	5.318e+02	02
Overall		08		09		**07**

Note: ratios are in percent.

Table 7. Parameter sensitivity analysis results.

- n: Performance was similar across the different settings and accordingly the intermediate setting of $n = 0.5d$ was selected.
- s: The best performing split ratio was 40–60 between the training and testing subsets.

The above settings were then used during the numerical experiments described in Section 4.

Author details

Yoel Tenne

Address all correspondence to: y.tenne@ariel.ac.il

Department of Mechanical and Mechatronic Engineering, Ariel University, Ariel, Israel

References

[1] Tenne Y, Goh CK, editors. Computational intelligence in expensive optimization problems. Vol. 2 of Evolutionary Learning and Optimization. Berlin: Springer; 2010.

[2] Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker KP. Surrogate-based analysis and optimization. Progress in Aerospace Science. 2005;41:1–28.

[3] Forrester AIJ, Keane AJ. Recent advances in surrogate-based optimization. Progress in Aerospace Science. 2008;45(1–3):50–79.

[4] Acar E. Various approaches for constructing an ensemble of metamodels using local measures. Structural and Multidisciplinary Optimization. 2010;42(6):879–896.

[5] Zhou X, Ma Y, Cheng Z, Liu L, Wang J. Ensemble of metamodels with recursive arithmetic average. In: Luo Q, editor. Proceedings of the International Conference on Industrial Mechatronics and Automation–ICIMA 2010. Piscataway, NJ: IEEE; 2010. pp. 178–182.

[6] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Edward T. Equation of state calculations by fast computing machines. Journal of Chemical Physics. 1953;21 (6):1087–1092.

[7] Černy V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications. 1985;45 (1):41–51.

[8] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983;220(4598):471–480.

[9] Xiang Y, Gubian S, Suomela B, Hoeng J. Generalized simulated annealing for global optimization: The gensa package. The R Journal. 2013;5(1):13–28.

[10] Belloni A, Liang H Tengyuan Narayanan, Rakhlin A. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. The Journal of Machine Learning Research. 2015;40:240–265.

[11] Rodríguez DA, Oteiza PP, Brignole NB. Simulated annealing optimization for hydrocarbon pipeline networks. Journal of Industrial Engineering and Chemical Research. 2013;52 (25):8579–8588.

[12] Sivasankaran P, Sornakumar T, Panneerselvam R. Design and comparison of simulated annealing algorithm and grasp to minimize makespan in single machine scheduling with unrelated parallel machines. Intelligent Information Management. 2010;2:406–416.

[13] Jin Y, Olhofer M, Sendhoff B. A framework for evolutionary optimization with approximate fitness functions. IEEE Transactions on Evolutionary Computation. 2002;6(5):481–494.

[14] Conn AR, Scheinberg K, Toint PL. On the convergence of derivative-free methods for unconstrained optimization. In: Iserles A, Buhmann MD, editors. Approximation Theory and Optimization: Tributes to M.J.D. Powell. Cambridge, New York: Cambridge University Press; 1997. pp. 83–108.

[15] Simpson TW, Poplinski JD, Koch PN, Allen JK. Metamodels for computer-based engineering design: Survey and recommendations. Engineering with Computers. 2001;17:129–150.

[16] Jin Y, Sendhoff B. Reducing fitness evaluations using clustering techniques and neural network ensembles. In: Deb K, et al., editors. Proceedings of the Genetic and Evolutionary Computation Conference–GECCO 2004. Berlin, Heidelberg: Springer-Verlag; 2004. pp. 688–699.

[17] Viana FAC, Venter G, Balabanov V. An algorithm for fast optimal Latin hypercube design of experiments. International Journal of Numerical Methods in Engineering. 2009;82 (2):135–156.

[18] Burnham KP, Anderson DR. Model Selection and Inference: A Practical Information-Theoretic Approach. New York: Springer; 2002.

[19] Mininno E, Neri F. A memetic differential evolution approach in noisy optimization. Journal of Soft Computing. 2010;2:111–135.

[20] Yang B, Guang L, Säntti T, Plosila J. Parameter-optimized simulated annealing for application mapping on networks-on-chip. In: Hamadi Y, Schoenauer M, editors. Learning and Intelligent Optimization: 6th International Conference, LION 6, Paris, France, January 16–20, 2012, Revised Selected Papers. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. pp. 307–322.

[21] Frausto-Solis J, Alonso-Pecina F. Analytically tuned parameters of simulated annealing for the timetabling problem. WSEAS Transactions on Advances in Engineering Education. 2008;5(5):272–281.

[22] Park MW, Kim YD. A systematic procedure for setting parameters in simulated annealing algorithms. Computers and Operations Research. 1998;25(3):207–217.

[23] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, et al. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Nanyang Technological University, Singapore and Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India; 2005.KanGAL2005005.

[24] Ratle A. Optimal sampling strategies for learning a fitness model. In: The 1999 IEEE Congress on Evolutionary Computation–CEC 1999. Piscataway, New Jersey: IEEE; 1999. pp. 2078–2085.

[25] de Jong KA. Evolutionary Computation: A Unified Approach. Cambridge, Massachusetts: MIT Press; 2006.

[26] Büche D, Schraudolph NN, Koumoutsakos P. Accelerating evolutionary algorithms with Gaussian process fitness function models. IEEE Transactions on Systems, Man, and Cybernetics–Part C. 2005;35(2):183–194.

[27] Sheskin DJ. Handbook of Parametric and Nonparametric Statistical Procedures. 4th ed. Boca Raton, Florida: Chapman and Hall; 2007.

[28] Hicks RM, Henne PA. Wing design by numerical optimization. Journal of Aircraft. 1978;15(7):407–412.

[29] Wu HY, Yang S, Liu F, Tsai HM. Comparison of three geometric representations of airfoils for aerodynamic optimization. In: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2003. pp. 1–11. AIAA 2003-4095.

[30] Drela M, Youngren H. XFOIL 6.9 User Primer. Cambridge, MA: Massachusetts Institute of Technology; 2001.

[31] Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. Statistical Science. 1989;4(4):409–435.

[32] Powell MJD. Radial basis function methods for interpolation of functions of many variables. In: Lipitakis EA, editor. Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA-01). Athens, Hellas: LEA Press; 2001. pp. 2–24.

[33] Golberg MA, Chen CS, Karur SR. Improved multiquadric approximation for partial differential equations. Engineering Analysis with Boundary Elements. 1996;18:9–17.

[34] Benoudjit N, Archambeau C, Lendasse A, Lee J, Verleysen M. Width optimization of the gaussian kernels in radial basis function networks. In: Verleysen M, editor. Proceedings of the 10th European Symposium on Artificial Neural Networks–ESANN 2002.Evere D-Side; 2002. pp. 425–432.

Chapter 4

Simulated Annealing of Constrained Statistical Functions

Barry Smith and Augustine Wong

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66069

Abstract

In 1987, Corana et al. published a simulated annealing (SA) algorithm. Soon thereafter in 1993, Goffe et al. coded the algorithm in FORTRAN and showed that SA could uncover global optima missed by traditional optimization software when applied to statistical modeling and estimation in economics (econometrics). This chapter shows how and why SA can be used successfully to perform likelihood-based statistical inference on models where likelihood is constrained by often very complicated functions defined on a compact parameter space. These constraints arise because likelihood-based inference involves comparing the maxima of constrained versus unconstrained statistical optimization models. The chapter begins with a review of the relevant literature on SA and constrained optimization using penalty techniques. Next, a constrained optimization problem based in maximum likelihood stress-strength modeling is introduced, and its statistical and numerical properties are summarized. SA is then used to solve a sequence of penalty-constrained optimization problems, and the results are used to construct a confidence interval for the parameter of interest in the statistical model. The chapter concludes with a brief summary of the results and some ways we were able to enhance the performance of SA in this setting.

Keywords: SA, constrained optimization, penalty, likelihood

1. SA and penalty-constrained optimization

Several chapters in this book consider the foundations and development of the simulated annealing (SA) algorithm. In this chapter, we focus on just one version of the algorithm given in Ref. [1] and one FORTRAN implementation of the algorithm presented in Ref. [2]. The

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

reasons for this are efficiency and familiarity. The algorithm in Ref. [1] is tight and effective and the FORTRAN implementation in [2] is dependable, easily extensible and sufficiently fast that it can be applied to both complicated small-scale problems and to very large Monte Carlo studies such as the one in Ref. [3].

1.1. Background

The SA algorithm in Ref. [1] was developed as an approach to find the unconstrained global optimum of functions with a potential multiplicity of optima, some of which may lie on the boundaries of the function's domain. The function being optimized need not be differentiable or even continuous, but it must be bounded. As well, the domain of the function must be compact. The search algorithm in Ref. [1] depends upon function evaluations. Derivatives play no role. The reader is referred to the careful statement of the algorithm on pages 266–269 in Ref. [1]. Here we provide an overview of the algorithm and its implementation.

In a multivariate setting, the domain of the function is initially defined as a (perhaps very) large hypercube. Random paths of individual choice variable values are searched at each stage. The number of such searches is controlled by the user. The step sizes that partially define the random paths are part of the algorithm and will in general be different for each choice variable at each stage. The algorithm cycles through and individually changes (increases or decreases) all choice variables. Each time a variable change leads to a "better" value of the function being optimized, this point is accepted. Sometimes "worse" points are temporarily accepted. That is, sometimes, the algorithm deliberately allows a search path at a stage to contain "worse" points. By allowing the paths to meander through better and worse points in the domain, the algorithm allows the domain to be searched for better optima that can only be reached by first passing through the worse regions. In this way, the search process can escape from local optima that are dominated by one or more global optima. This distinguishes SA from traditional hill-climbing algorithms that use the local properties of the function to move always in better directions. Movements in worse directions are governed by a Metropolis decision. In a sense, the Metropolis decision can be thought of as the algorithm giving permission to the search process to move in a worse direction, depending upon the roll of a (weighted) die. As the algorithm progresses through subsequent ("cooling") stages, the chance that a worse point will be permitted/accepted decreases. In addition, as the algorithm progresses, the effective domain of the function (that part of the domain to which the search is effectively restricted) is contracted. In part, the search evolves in a fashion consistent with the overall (global versus local) topography of the surface of the function being optimized. Asymptotically, the algorithm converges to a domain that contains the best optimum encountered and which has a user-specified (small) volume. Convergence criteria and the number and length of searches at each stage are determined by parameters set by the user.

The FORTRAN implementation in Ref. [2] is faithful to the algorithm in Ref. [1], but it does include some features and suggestions that tend to help in deciding whether a global optimum has been reached and, in the initial stages of optimization, the features allow the researcher to search individual subsets of the domain of the function. The researcher can control the number

of searches, the initial "temperature" of the model and the rate at which temperature decreases. Uphill and rejected downhill moves are balanced by changes to the upper and lower bounds on parameter changes. In Ref. [2], the suggestion is made that starting the SA algorithm from a variety of randomly selected domain points may provide information about whether a global optimum has been found. This is balanced, to some extent, by the realization that the properties of SA that make it a global optimizer also tend to make it independent of initial values of the choice variables. Any indication of sensitivity to starting values is, therefore, a strong suggestion that the SA user-determined parameters should be changed to provide a more thorough search.

As with any mathematical tool, becoming adept at using SA to solve optimization problems requires practice. FORTRAN code for the SA implementation in Ref. [2] is widely available. This code, written in FORTRAN 77, is carefully documented and contains an example problem that illustrates many of the features of SA. These features, such as convergence criteria, cooling rate, lengths of search paths, and the like can be adjusted to study how the implementation works. It is straightforward to code and solve new problems. The advice in Ref. [2] and in the code is helpful in finding the set of search parameters that solves the problem at hand.

Finally, Goffe et al. [2] address the issue of the speed of SA and recommend the ways that the user can tune the implementation to run more quickly. The implementation was published in 1994, and since then multicore very fast processors with at least 64-bit single precision have become the norm. At the same time, though, there are optimization problems that are now being addressed at the limit of current technology.

1.2. Penalty-constrained optimization

The foregoing discussion has provided an overview of SA and how it can be applied to optimize a function. We now turn to a discussion of how SA can be used to find the global optimum of functions when there are constraints on the choice variables. Our principal concern is how to find the optimum of the statistical functions when the choice variables must satisfy an integral equality constraint. Our approach, however, is quite general and it can be adapted to solve optimization problems subject to several inequality as well as equality constraints that may or may not involve integrals. At this point, it is helpful to introduce some notation.

Our choice variables are represented by the vector: θ. The objective function to be maximized is given by: $l(\theta)$. As well, there is a constraint given by: $R(\theta) = \psi_0$. In the statistical problem considered in the next section, we examine the unconstrained problem.

1.2.1. (Unconstrained problem) U: choose θ to maximize l(θ)

(Unconstrained Problem) U: Choose θ to maximize $l(\theta)$ is an important part of the analysis. It is almost always the case in statistical settings that the unconstrained problem can be solved in a straightforward manner using SA.

1.2.2. (Constrained problem) C: choose θ to maximize l(θ) subject to R(θ) = ψ_0

(Constrained Problem) *C*: Choose θ to maximize $l(\theta)$ subject to $R(\theta) = \psi_0$ can also be solved using simulated annealing. Indeed, as we will see, SA is a very natural approach to solving problem *C*.

Within a statistical setting, substitution and Lagrange multiplier techniques tend not to work well when attempting to solve *C*. Typically, the objective function, $l(\theta)$, and the constraint, $R(\theta) = \psi_0$ will not have properties that guarantee a straightforward solution to the optimization problem. For example, the constraint can reasonably be expected to be a highly nonlinear function of the choice variables, θ. In both the substitution and Lagrange approaches, it is necessary at each iteration to solve the constraint equation to express one choice variable in terms of all of the others. If one or more of the choice variables takes on an extreme value (very large or very small) during the iteration process, then this can lead in turn to an extreme constraint solution and such extreme values tend to perpetuate themselves through subsequent iterations. The traditional absence of some form of textbook concavity or convexity on the objective function and/or the constraint function typically results in a failed optimization attempt. Standard derivative-based optimizers often get lost when derivatives take extreme values or when the Hessian of the function is indefinite. In part, this is due to their strong dependence upon local properties of the function being optimized.

The penalty function approach provides an alternative way of dealing with the constraint, which does not require exact satisfaction of the constraint equation at each "iteration." In fact, the constraint equation only holds asymptotically. To clarify this, we begin by introducing the penalty function *PL*:

$$PL(\theta, \psi_0) = l(\theta) - k\left[R(\theta) - \psi_0\right]^2, \quad k > 0. \tag{1}$$

This is the penalty function associated with the constrained optimization problem *C* introduced above. In this case, *k* is a positive parameter controlled by the researcher.

In a recent paper, Byrne [4] studied how the penalty functions like *PL* could be used to solve constrained optimization problems such as *C*. The following three conditions are introduced:

1. θ is chosen from a compact set.
2. The functions $l(\theta)$ and $R(\theta)$ are continuous.
3. Let θ_k^* be the vector that corresponds to the global maximum of *PL* in Eq. (1) when the parameter multiplying the squared term is *k*. We assume that each element in the sequence $\left\{\theta_k^*, k = 1, 2, \ldots\right\}$ exists.

Then, based on these conditions, it is proved in Ref. [4] that the sequence $\left\{\theta_k^*, k = 1, 2, \ldots\right\}$ converges to the θ^* that solves problem *C*.

This result is extremely important for applying simulated annealing to solve constrained optimization problems. First, SA chooses candidate optimizers from a compact set. Second, continuity of the penalty function *PL* is more than what is required for SA to reach a global

optimum. The complicating point is that the result in Ref. [4] is expressed in terms of the limit of a sequence of SA optimizations. But, in our experience, this is not a complication of considerable practical importance. In the first place, even for large values of k, SA is not helped by starting the iterations for the $(k + 1)$st solution at the optimal values from the kth solution. In practice, given that SA searches ("cools") sufficiently slowly and follows long enough paths in the domain, it tends to find the global solution of the $(k + 1)$ problem regardless of the starting values it is given. Second, there is a practical issue of how much accuracy can be expected. The SA algorithm terminates when successive improvements in the value of the objective function are all less that a user-specified threshold. This means that the contribution of the term $k[R(\theta) - \psi_0]^2$ must also be small in absolute value. In all of the problems we have considered, setting $k = 100,000$ is certainly enough to get a high-quality estimate of θ^*. That is, it is reasonable to truncate the sequence at this value of k.

In concluding this section, we reconsider the question: "Why does a penalty approach work when the Lagrange approach fails?" In our experience, the Lagrange approach, which requires solutions of an equation to a given level of accuracy, is prone to problems of numerical accuracy and their propagation. Alternatively, the penalty approach never requires the constraint to be exactly satisfied. Instead, it increasingly discourages large squared deviations of the constraint function $R(\theta)$ from its constraint value ψ_0 as k increases. As a final point, when conditions are satisfied for the Lagrange multiplier formally to exist, it can be obtained as the limit of the partial derivative of the penalty function with respect to the parameter ψ_0 as k increases.

2. Modeling reliability using SA penalized likelihood

2.1. Background on reliability

We consider two variables X and Y. We refer to them respectively as Strength and Stress. For example, Strength could refer to the "time before failure" of a component such as a digital storage device. Alternatively, Stress might measure the total time that the device is used. From the standpoint of a manufacturer, X and Y are both random variables with distributions that can, in principle, be estimated from available breakdown and usage data. Reliability, R, is defined as the probability that the component will withstand the stress it faces in use. In particular,

$$R = P(Y < X). \tag{2}$$

A variety of distributions have been used for X and Y in the literature. The actual choice depends upon the process being studied. It is standard and reasonable to suppose that X and Y are independent.[1]

[1]That is, the probability distribution of Y does not depend upon any realized value of X and vice versa. If dependence is possible in a given setting, it is easily accommodated. Formally, (Eq. (2)) will continue to hold, but additional parameters, associated with the interdependence, may appear in both the objective function and the constraint. In a formal sense, the penalty approach is unchanged.

In this section, we suppose that both X and Y are distributed as exponentiated exponential distributions. Exponentiated exponential distributions, $EE(\alpha, \beta)$, have two parameters: $\alpha > 0$ controls shape and $\beta > 0$ controls scale. Adopting the notation introduced in Ref. [5], the cumulative distribution function is:

$$F(x;\alpha,\beta) = \left(1-e^{-\beta x}\right)^{\alpha}, \qquad \alpha > 0, \beta > 0, x > 0, \tag{3}$$

As in Ref. [6], we assume that X is distributed as $EE(\alpha_1, \beta_1)$ and Y is distributed as $EE(\alpha_2, \beta_2)$. We do not, however, constrain the scale parameters, β_1 and β_2, to be equal. As a result, the expression for reliability is:

$$R = P(Y < X) \quad = \int_0^{\infty} \alpha_1\beta_1 \ \left(1-e^{-\beta_1 x}\right)^{\alpha_1-1} \ e^{-\beta_1 x} \ \left(1-e^{-\beta_2 x}\right)^{\alpha_2} \ dx. \tag{4}$$

There is no known closed-form solution for this integral. As noted in Ref. [7], introducing the change of variables: $z = \beta_1 x$ allows one to see that R is homogeneous of degree 0 in (β_1, β_2). The contours of R are, therefore, all constant along a line in a parameter space defined by $\beta_2 = \beta_1$.

2.2. The unconstrained EE likelihood equation and it properties

Following Ref. [7], we let $x = (x_1, \ldots, x_n)'$ and $y = (y_1, \ldots, y_m)'$ denote the realizations of random samples from $EE(\alpha_1, \beta_1)$ and $EE(\alpha_2, \beta_2)$, respectively. The log-likelihood function of the above model can be written:

$$\begin{aligned} l(\alpha_1,\beta_1,\alpha_2,\beta_2;x,y) \quad &= n\log\alpha_1 + n\log\beta_1 + (\alpha_1-1)\sum_{i=1}^{n}\log\left(1-e^{-\beta_1 x_i}\right) - \beta_1\sum_{i=1}^{n} x_i \\ &+ m\log\alpha_2 + m\log\beta_2 + (\alpha_2-1)\sum_{j=1}^{m}\log\left(1-e^{-\beta_2 y_j}\right) - \beta_2\sum_{j=1}^{m} y_j. \end{aligned} \tag{5}$$

We denote the parameter vector as $\theta = (\alpha_1, \alpha_2, \beta_1, \beta_2)'$.

The Appendix in Ref. [7] contains a derivation of the properties of $l(\theta) = l(\alpha_1, \beta_1, \alpha_2, \beta_2; x, y)$. In particular, $l(\theta)$ is not a concave function of θ nor is it quasi- or pseudo-concave. There is a small region around the point where the gradient of $l(\theta)$ vanishes and in that region, the Hessian matrix is negative definite. Elsewhere in the parameter space, the determinant of the Hessian matrix changes sign frequently. Thus, extreme care must be taken in trying to maximize $l(\theta)$, using a derivative-based algorithm. We found that a variable-metric algorithm would work as long as an approximate Hessian matrix, constrained to be negative definite, is used over a restricted parameter space.

One example, which we consider in greater detail later in the paper, uses the following data with sample sizes of 11 and 9: x= (2.1828, 0.5911, 1.0711, 0.9007, 1.7814, 1.3616, 0.8629, 0.2301, 1.5183, 0.8481, 1.0845) and y= (0.8874, 1.1482, 0.8227, 0.4086, 0.5596, 1.1978, 1.1324, 0.5625, 1.0679). Our SA program quickly and easily solved the associated unconstrained maximum likelihood optimization problem.

http://dx.doi.org/10.5772/66069

2.3. Constrained likelihood maximization

As will be discussed in the next section, inference for the "parameter" $R(\theta)$ in our reliability model requires that the likelihood function $l(\theta)$ be maximized subject to the constraint $R(\theta) = \psi_0$ for a range of values of the constraint parameter ψ_0. These constrained optimization problems are all solved using the penalty function approach introduced in Section 1.2 and using the penalty function $PL(\theta, \psi_0)$ given in Eq. (1). The functions $l(\theta)$ and $R(\theta)$ can be thought of now as the unconstrained *EE* likelihood function and the reliability function, respectively.

3. Likelihood-based inference and penalty functions

In Eq. (5), we introduced the statistical log-likelihood function primarily as an example of a function that needs to be maximized (with and without constraint) in a statistical setting. In this section, we provide more details about likelihood functions and inference. Our brief discussion is not intended as a complete explanation of the underlying statistical notions. Rather, it is intended only to motivate some importance of constrained and unconstrained optimization within statistics. Our discussion is rooted in the example of Eq. (5).

3.1. Background on likelihood models in statistics

Likelihood is akin to probability. The difference in the notions for our purposes is that likelihood is measured in terms of the probability density governing the realizations of a continuous random variable. Technically, the probability of any one outcome, say x_0, of a continuous random variable is 0. The value of the density, say $h(x)$, evaluated at x_0 and multiplied by dx, that is, $h(x_0)\ dx$, can be thought of as approximately the probability that there will be a realization of the random variable X in a very small interval containing x_0. It is common to have situations where the realized (observed) values of a random variable X arise from a process of random sampling where the outcomes are independent of each other yet are governed by identical probability density functions, $h(x)$. The likelihood of a given sample of realized values is defined as the product of the densities corresponding to each of the outcomes in the sample; so the likelihood of a given sample is, up to a scaling factor, a notion similar to the probability of the sample. The likelihood of a sample of realizations of X and Y is, given our assumptions, the product of the likelihoods of the X and the Y samples. For a variety of reasons, it is often easier to work with a positive monotonic transformation of the sample likelihood. In particular, we work with the log-likelihood of the sample. In Eq. (5), we are given the log-likelihood of a sample where the densities come from possibly different exponentiated exponential distributions.

The derivation of the log-likelihood associated with a sample of realizations is just the beginning of the modeling process. Extensions include forecasting the next realization of a random variable or perhaps finding an interval where one can be 95% confident that the next realization of a function of the random variables will fall. For example, we could ask for a 95% confidence interval of the measure of reliability in Eq. (4), incorporating the information in the

sample given at the end of Section 2.2. These are questions of statistical inference. We answer these questions by solving the optimization problems.

The likelihood function given in Eq. (5) can be combined with the sample of 11 realizations of X and 9 realizations of Y given in Section 2.2. We can use the information in the data to estimate the unknown vector of parameters: $\theta = (\alpha_1, \alpha_2, \beta_1, \beta_2)$. One set of estimates of the parameters of the model is obtained by maximizing the likelihood function with the choice variables being the parameters. These maximum likelihood parameter estimates can be thought of as the parameter values that yield specific density functions that are most likely to have generated the data. Of course, 20 observations are not enough to achieve certainty, so there is a related theory about where the true (population) parameters lie in relation to their estimates. Indeed, there are probability distributions associated with the maximum likelihood parameter estimation process, and the parameter values that maximize the log-likelihood for a given sample of data realizations are themselves just realizations. These probability distributions or their approximations allow us to estimate how close the parameter realizations are to the true parameter values.

There is also a probability distribution for the maximized value of the log-likelihood function. This allows us to ask questions such as the following: do I induce a "significant" change in the maximized likelihood value when I constrain the parameter estimates (choice variables) to satisfy an additional condition or set of conditions. This leads back to the constrained optimization problem C in Section 1.2, and the penalty function in Eq. (1).

In the subsection that follows, we present the process of inference for our reliability model. The presentation is more technical.

3.2. Inference in the reliability model

Given $l(\theta)$ is the log likelihood function, we denote the unconstrained maximum likelihood estimator $\widehat{\theta}$ when $l(\theta)$ alone is maximized. As well, we define

$$j\left(\widehat{\theta}\right) = -\frac{\partial^2 l(\theta)}{\partial\theta\partial\theta'}\bigg|_{\widehat{\theta}} \tag{6}$$

as the observed information matrix evaluated at $\widehat{\theta}$. Finally, we let $\widehat{\theta}_\psi$ be the constrained maximum likelihood estimator of θ given by maximizing $l(\theta)$ subject to $R(\theta) = \psi$. Formally, $\widehat{\theta}_\psi$ can be obtained for any ψ in the range of $R(\theta)$ by maximizing $l(\theta)$ subject to the constraint $R(\theta) = \psi$ using the penalty function approach within SA.

The aim next is to obtain inference concerning $R = R(\theta)$, where $dim(R) = 1$. Two widely used likelihood-based methods for obtaining confidence interval for R are based on the asymptotic distribution of the maximum likelihood estimator $\widehat{\theta}$ and the (log) likelihood ratio statistic.

Taking θ as the true population parameter vector, $\left(\widehat{\theta}-\theta\right)'\left[var\left(\widehat{\theta}\right)\right]^{-1}\left(\widehat{\theta}-\theta\right)$ is asymptotically distributed as chi-square with degrees of freedom equal to $dim(\theta)$, and variance-covariance

http://dx.doi.org/10.5772/66069

matrix $\widehat{var}\left(\widehat{\theta}\right) \approx \hat{j}^{-1}\left(\widehat{\theta}\right)$. Since $R = R(\theta)$ depends upon the entire vector of parameters, we approximate its variance by applying the Delta method to $\widehat{R} = R\left(\widehat{\theta}\right)$ and obtain:

$$\widehat{var}\left(\widehat{R}\right) \approx R'_{\theta}\left(\widehat{\theta}\right)\widehat{var}\left(\widehat{\theta}\right)R_{\theta}\left(\widehat{\theta}\right) \tag{7}$$

where

$$R_{\theta}\left(\widehat{\theta}\right) = \left.\frac{\partial R(\theta)}{\partial \theta}\right|_{\widehat{\theta}}. \tag{8}$$

Since $dim\left(\widehat{R}\right) = 1$, we have

$$\frac{\widehat{R} - R}{\sqrt{\widehat{var}\left(\widehat{R}\right)}} \tag{9}$$

asymptotically distributed as standard normal. An approximate $(1 - \alpha)100$ % confidence interval for R based on $\widehat{\theta}$ is

$$\left(\widehat{R} - z_{\alpha/2}\sqrt{\widehat{var}\left(\widehat{R}\right)}, \widehat{R} + z_{\alpha/2}\sqrt{\widehat{var}\left(\widehat{R}\right)}\right) \tag{10}$$

where $z_{\alpha/2}$ is the $(1 - \alpha/2)100^{th}$ percentile of the standard normal distribution. This is our first confidence interval.

Alternatively, with regularity conditions stated in Refs. [8, 9], the log likelihood ratio statistic:

$$W(\psi) = 2\left[\ell\left(\widehat{\theta}\right) - \ell\left(\widehat{\theta}_{\psi}\right)\right] \tag{11}$$

is asymptotically distributed as chi square with 1 degree of freedom. Therefore, an approximate $(1 - \alpha)100$ % confidence interval of R based on the likelihood ratio statistic is:

$$\left\{\psi : W(\psi) \leq \chi^2_{1,\alpha}\right\}. \tag{12}$$

The set of all constrained values ψ in the domain of R that cannot be rejected at the $(1 - \alpha)100$ % as the true value $R(\theta)$ is defined in Eq. (10). These values form our second confidence interval.

It should be noted that both methods have rates of convergence $O(n^{-1/2})$. While the MLE-based interval is often preferred because of simplicity in calculation, the log-likelihood ratio method has the advantage that it is invariant to reparameterization and the MLE-based method is not. The results presented in Ref. [10] suggest that, in terms of coverage, the confidence interval based on the log-likelihood ratio statistic should be preferred to the MLE-based interval. In particular, when 95% confidence intervals for both statistics are compared, the interval from the log-likelihood ratio statistic is shorter and therefore more precise.

	95% Confidence interval
MLE	(0.306, 0.934)
χ^2	(0.378, 0.823)

Table 1. Interval estimates of R.

The results are summarized in **Table 1**. We note that, consistent with Ref. [10], the χ^2 interval is indeed shorter than the MLE interval.

4. Conclusion

This chapter has considered how SA can play an important role as a global optimizer of constrained likelihood-based statistical models. SA is naturally paired with the penalty function approach to constrained optimization. SA and the penalty approach both require compact domains and bounded functions. Penalty functions must be continuous and, within a statistical setting, this almost always holds. SA supplies the global optimization property that guarantees that the penalty function approach converges to the global constrained optimum. Even though our implementation of SA does not make use of derivatives, the converged penalty function will often be differentiable and Lagrange multipliers, gradients, and Hessian matrices can be calculated. The extension of these results to multiple constraints is computationally straightforward.

In this chapter, we have motivated the pairing of SA and penalty functions in a statistical setting. But the approach can be used to solve many other types of numerical constrained optimization problems. Over time, we have accumulated a considerable amount of experience solving constrained problems using SA and the penalty functions. One lesson stands out: SA is a global optimizer and, for the most part, it should be independent of initial conditions such as starting values of parameters (choice variables). If you find that you get a different optimum after changing the starting values, then it is likely that neither solution is the true global optimum. You can usually remedy this by increasing the initial temperature, slowing the rate of cooling, and/or increasing the length and number of search paths.

Author details

Barry Smith[1] and Augustine Wong[2]*

*Address all correspondence to: august@yorku.ca

1 Department of Economics, York University, Toronto, Canada

2 Department of Mathematics and Statistics, York University, Toronto, Canada

http://dx.doi.org/10.5772/66069

References

[1] Corana, A., Marchese, M., Martini, C. and Ridella, C. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. ACM Transactions on Mathematical Software. 1987; 13: 262–280.

[2] Goffe, W.L., Ferrier, G.D. and Rogers, J. Global optimization of statistical functions with simulated annealing. Journal of Econometrics. 1994; 60: 65–99.

[3] Li, X., and Smith, B. Diagnostic analysis and computational strategies for estimating discrete time duration models—a Monte Carlo study. Journal of Econometrics. 2015; 187 (1): 275–292.

[4] Byrne, C.L. Sequential unconstrained minimization: a survey. [Internet]. 2013. Available from: http://faculty.uml.edu/cbyrne/SUM.pdf.

[5] Gupta, R.D. and Kundu, D. Exponentiated exponential family: an alternative to gamma and Weibull distributions. Biometrical Journal. 2001; 43(1): 117–130.

[6] Kundu, D. and Gupta, R.D. Estimation of P[$X < Y$] for generalized exponential distributions. Metrika. 2005; 61: 291–308.

[7] Smith, B., Wang, S., Wong, A. and Zhou, X. A penalized likelihood approach to parameter estimation with integral reliability constraints. Entropy. 2015; 17: 4040–4063. DOI:10.3390/e17064040.

[8] Severeni, T. Likelihood Methods in Statistics. New York: Oxford University Press; 2000.

[9] Cox, D.R. and Hinkley, D.V. Thoeretical Statistics. London: Chapman and Hall; 1974.

[10] Doganaksoy, N. and Schmee, J. Comparisons of approximate confidence intervals for distributions used in life-data analysis. Technometrics. 1993; 35: 175–184.

Section 2

Applications & Exploitations

Chapter 5

Fitting Truncated Mode Regression Model by Simulated Annealing

Maoxi Tian, Jian He and Keming Yu

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66070

Abstract

Like mean, median, and standard deviation, mode as the value that appears most often in a set of data is an important feature of a distribution. The numerical value of the mode is the same as that of the mean and median in a symmetric distribution but may be very different in a highly skewed distribution. Mode regression, which models the relationship between the mode of a dependent variable and some covariates, was first introduced by Lee in terms of truncated dependent variables. Some modifications of the truncated mode regression have been proposed recently. However, little progress is made on the computation or algorithm of fitting a mode regression due to an NP-hard optimization problem. In this paper we first introduce the popular simulated annealing (SA) to solve the truncated mode regression optimization. Experiments with simulations compare favorably to SA. Then, a mode regression with the proposed algorithm is applied to explore the typical income structure of China. We also compare the income returns to gender, education, experience, job sector, and district between the majority of workers with typical income and the workers with mean, middle income via comparison between mode regression, mean regression, and median regression.

Keywords: income inequality, Lee's estimate, median regression, mode, mode regression fitting, simulated annealing algorithm, truncated variable

1. Introduction

Mode, the most likely value of a distribution, has wide applications in biology, astronomy, economics, and finance. For example, in the archeology, many practical questions often focus on "Which era the biological species most likely to survive in according to the biological fossils?" In such cases, mode regression provides a convenient summary of how the repressors

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

affect the conditional mode. Income inequality is of growing concern to people in rich as well as poor countries. This inequality may result in problem in social stability in a region or a country. But using average income as a statistical measurer is largely affected by a small number of richest who earned a high proportion of all income. Mode is the right statistic to measure the typical income over the whole population, or mode-based measurement represents the income of majority workers. The causes of income inequality have been attracting a lot of attention in literature and public. Education and experience are often cited as important factors for income. Gender pay gap has been an issue for many regions, particularly in some developing countries. The difference in wage between public sector staff and private workers changes from country to country and region to region. The mode regression or mode-based regression analysis provides a direct and powerful tool to explore the typical income and the return to education, experience, gender, sector, and so on. Mode-based clustering techniques have also been developed [1].

The mode regression models the relationship between the conditional mode of the dependent variable y^* and covariates x as

$$y^* = x'\beta + \varepsilon \tag{1}$$

where vector $x = (1, x_1, \ldots, x_p)'$, $\beta = (1, \beta_1, \ldots, \beta_p)'$ is the unknown parameter vector, and ε is the model error. Let mode($y^* \mid x$) be the mode of y^* conditional on x and then mode($y^* \mid x$) = $x'\beta \Leftrightarrow$ mode($\varepsilon \mid x$) = 0. Usually, one assumes that the density of ε is wider than $[-w, w]$ for a $w > 0$ suitably chosen.

Lee [2] first considers truncated model regression where y^* is truncated from below at c by y or y^* is observed only when $y^* > c$. That is, $y = \max(y^*, c)$. Examples of truncated regression include (1) candidates of students who want to nominate a university prize are required to have a minimum examination mark of 70 out of 100 to qualify for the entry. Thus, the sample is truncated at an examination mark of 70. (2) A researcher has data for a sample of British citizens whose income is above the poverty line. Hence, the lower part of the distribution of income is truncated. Truncated regression cannot be fitted by ordinary least squares (OLS) regression, as OLS regression will not adjust the estimates of the coefficients to take into account the effect of truncation, so that the estimated coefficients may be severely biased. This can be conceptualized as a model specification error [3].

Under model (1), given observations on $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, we aim at estimating β. This is the main task of fitting the model. Lee's [2] mode regression estimates β by

$$\hat{\beta} = \arg\max_{\beta} \left(n^{-1} \sum_{i=1}^{n} I\left[\left| y_i - \max(x'\beta, c + w) \right| \le w \right] \right) \tag{2}$$

where $I[\ ]$ is the indicator function, which takes the value 1 if the condition inside [ ] is satisfied and 0 otherwise.

http://dx.doi.org/10.5772/66070

But little progress on mode regression has been made due to computation difficulty although some modifications of the proposed mode regression have been developed (e.g., see [4–7]). The difficulty in computing these estimators arises because the objective function consists of indicator function $I[\ ]$ and involves in an absolute function and maximum operator. So that the objective function in Eq. (2) is neither convex nor differentiable but may result in a large number of local maxima. See further details in Section 2. Therefore, the estimation in Eq. (2) is an NP-hard problem, so that standard optimization algorithms will perform poorly if they tend to get trapped in local maxima, or they may not be applicable if analytical gradients are required, because standard optimization tools, which require the objective function to be differentiable and/or convex, may fail to discover the true mode regression function.

Currently, rather than dealing with the NP-hard problem directly, some attempts were made to solve the computation of mode regression via replacing rectangular kernel in Eq. (2) by a "smooth" version [4, 5]. However, these smooth versions of mode estimators, in spite of their improved asymptotical properties, may not estimate mode but estimate something else. This big issue is often ignored in literature.

For example, the smoothing version of mode estimator of Kemp and Silva [5] is to maximize $Q_n(\beta) = n^{-1}\sum_{i=1}^{n} K_h\left(y_i - x'\beta\right)$ with $K_h = \frac{1}{h}K\left(\frac{\cdot}{h}\right)$ and the standard normal density as kernel function $K(\ )$, and then the estimator is closer to mean than to mode due to the quadratic property of normal density function. Also, a careful selection of bandwidth h which requires tending to zero is not available.

Lee [4] employed a quadratic kernel (QME) to smoothing the rectangular kernel and estimated β by

$$\begin{aligned}\hat{\beta} &= \arg\max_{\beta}\left(n^{-1}\sum_{i=1}^{n}\left[w^2 - \{y_i - \max(x'\beta, c+w)\}^2\right] I\left[\left|y_i - \max(x_i'\beta, c+w)\right| \le w\right]\right)\\ &\Leftrightarrow \arg\max_{\beta}\left(n^{-1}\sum_{i=1}^{n}\max\left[w^2 - \{y_i - \max(x'\beta, c+w)\}^2, 0\right]\right)\end{aligned} \tag{3}$$

This QME is quite similar to the symmetrically trimmed least squares (STLSs) estimation in Powell [8], which, instead of maximization, minimizes

$$\hat{\beta} = \arg\min_{\beta} n^{-1}\sum_{i=1}^{n}\left[y_i - \max(0.5y_i + 0.5c, x'\beta)\right]^2 \tag{4}$$

However, QME is quite sensitive to the choice of w whose optimal value is difficult to derive in practice. And the STLS, which strongly depends on the symmetric requirement of y conditional on, needs the symmetry up to $\pm x'\beta$. That is, STLS requires global symmetry if x is unbounded.

The other way to develop efficient algorithms for the truncated mode regression objective function (2) could be the emulation algorithms (EA) [9, 10], which compute the truncated mode regression estimator by checking every critical point and solving maximum score estimation as a nonlinear programming problem. However, EA exhibits a high degree of complexity in its implementation. EA may achieve convergence to local minima, whereas obtaining a global minimum requires a heavy computational load, something that renders its use in solving real problems impractical.

This paper aims to introduce meta-heuristic methods for computing the elegant rectangular mode regression estimator so that one could not only fit the mode regression but also improve computation efficiency. The recommended heuristic method for dealing with complicated objective function with many local optima could be the popular simulated annealing (SA) [11], because SA provides a means to avoid getting stuck in local optima by accepting worse neighbors in hopes of finding a global optimum. However, SA has not been used in mode regression fitting.

The paper is organized as follows. In Section 2, we outline the issue of many local optima of Lee's estimator and SA computation. Section 3 compares different algorithms or estimation methods such as QME, STLS, EA, and SA via (Monte Carlo) simulation study. In Section 4, we fit a multiple mode regression model for the analysis of a real income data via SA algorithm. The final section concludes with brief remarks.

2. Lee's estimator with many local optima and SA algorithm

As mentioned earlier, truncated mode regression fitting problem may be seen as a global optimization problem (GOP) with many local optima. SA is applied to optimize Eq. (2), which represents a nonlinear objective function. Then, the estimation equation can be formulated as follows:

$$f(\beta) = n^{-1}\sum_{i=1}^{n} I\left[\left|y_i - \max(x'\beta, c+w)\right| \le w\right] \tag{5}$$

Note that $f(\beta)$, as a function of regression coefficient β, is an unconstrained nonlinear and non-smooth function; it is difficult to calculate its optimizer. Moreover, $f(\beta)$ is not convex in β. Therefore, an effective tool for solving global optimization problems is required. Let us demonstrate this issue via a simple example and graphical representation of the objective function. Draw an independent sample size of 10 from a standard normal distribution, i.e., $x_i \sim N(0, 1)$, $i = 1, 2, \ldots, 10$, and let the i.i.d error term ε also have a standard normal distribution. So the dependent variable y^* can be generated from the mode regression $y_i^* = \beta x_i + \varepsilon_i$. Then we obtain observations of y_i from y_i^* via truncated point $y_0 = 0$. This gives a 52% heavy truncated on average. The objective function $f(\beta)$ against β's values from this model can take a form as plotted in **Figure 1**. Clearly, the objective function under $w = 1$ has many local maxima, which

http://dx.doi.org/10.5772/66070

lie in the range between −0.5 and 0.25. When $w = 1.5$, this objective function still has many localities but in the range between −0.75 and 0.25. However, **Figure 1** shows that these localities under $w = 0.5$ fall in a much narrower range than those under $w = 1$ and $w = 1.5$.

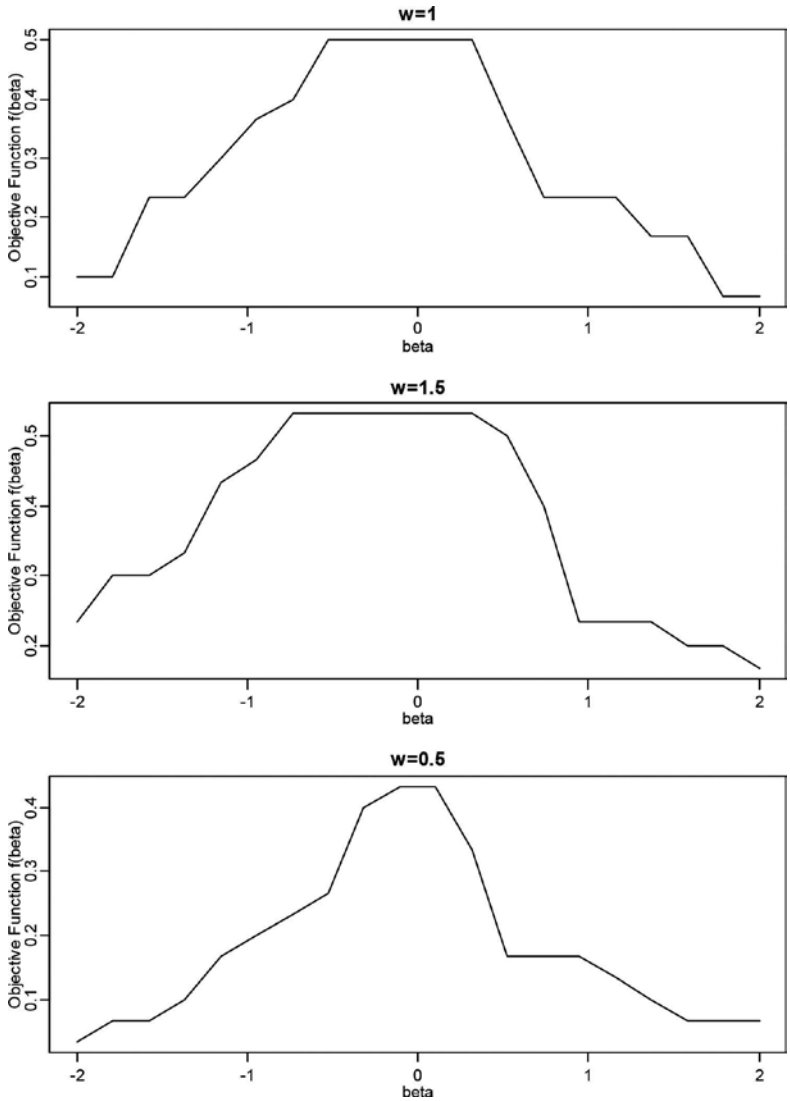

Figure 1. The objective function of a simple example with $w = 1$, $w = 1.5$, and $w = 0.5$.

The SA algorithm proposed by Kirkpatrick, Gelatt, and Vecchi [11], as a local search meta-heuristic, is characterized by an acceptance criterion for neighboring solutions that adapts itself at run time. In this chapter, we process the data by the R software. One of the packages to implement SA in R is *stats* with *sann* as an option of *optim* function [12]. An alternative SA is the *GenSA* function in specific R-package *GenSA*. The SA flow diagram is presented in **Figure 2**.

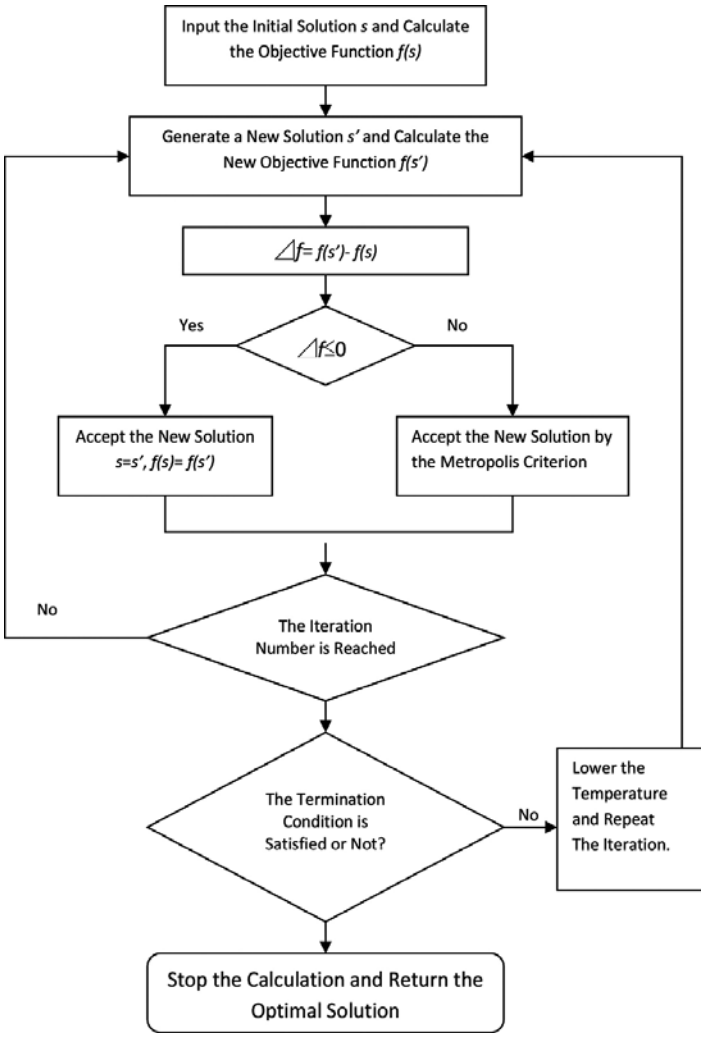

Figure 2. The flow diagram of SA.

3. Numerical comparison

Following Lee ([2, 4]) consider the mode regression model

$$y_i = 0 + x_i + \varepsilon_i, i = 1, 2, \cdots, n \tag{6}$$

where sample size $n = 30$, x_i follows a standard normal random variable, and the model error ε_i is generated from a standard normal distribution.

The four methods—QME (quadratic kernel smoothing method in Ref. [2]), STLS (trimmed least square method in Ref. [8]), emulation algorithm (EA in Ref. [9, 10]), and SA [11]—are implemented for numerical comparison of optimization of $f(\beta)$ in terms of β, where β simply stands for the slope coefficient whose true value is 1.

The data generated from the model are under four different distributions (normal, Cauchy, logistic, and gamma) for model error ε and two different numbers of truncated schemes: 25 and 50%. For each of these five different distributions of ε and each truncated scheme, we implement all four methods to estimate the slope coefficient 200 times, respectively, via simulation.

Then the performance criteria of each method consist of bias of the estimator of slope coefficient, standard deviation (STD), root mean square errors (RMSE), lower quartile (LQ), median (MED), and upper quartile (UQ) of estimation. At each simulation of 200 times, the bias is the difference between estimate and the true value; then the bias we collect for comparison is a simple average of 200 times of replications. The computation of STD, RMSE, LQ, MED, and UQ are based on 200 times of replications.

Table 1 reports the results by different designs which are the combination of truncation rate and distribution of ε.

	BIAS	SE	RMSE	LQ	MED	UQ
Design 1: 50% truncation, standard normal						
w = 0.5	0.535	1.451	1.563	0.910	1.341	1.811
w = 1.0	0.502	1.011	1.217	0.925	1.244	1.930
w = 1.5	0.523	1.345	3.492	0.696	1.112	1.748
w = 2.0	0.591	1.876	1.729	0.997	1.270	1.840
Design 2: 25% truncation, standard normal						
QME	0.085	0.331	0.345	0.839	1.042	1.269
STLS	0.023	0.210	0.208	0.873	0.989	1.136
EA	0.067	0.023	0.022	0.891	0.993	1.542
SA	0.065	0.381	0.412	0.893	1.114	1.458
Design 3: 50% truncation, standard normal						
QME	0.515	1.139	1.248	0.895	1.213	1.677
STLS	0.200	0.502	0.542	0.913	1.090	1.354
EA	0.206	0.489	0.512	0.807	1.051	1.431
SA	0.215	0.431	0.387	0.801	1.098	1.398
Design 4: 50% truncation, standard Cauchy						
QME	0.470	1.812	1.872	0.851	1.270	1.878

	BIAS	SE	RMSE	LQ	MED	UQ
STLS	0.396	6.390	6.949	0.823	1.109	1.605
EA	0.401	1.978	2.231	0.901	1.123	1.589
SA	0.393	1.759	2.217	0.801	1.012	1.598
Design 5: 50% truncation, standard normal						
QME	0.767	2.338	2.479	0.858	1.488	2.620
STLS	0.494	3.765	3.798	0.776	1.189	1.966
EA	0.506	2.634	2.634	0.884	1.052	1.890
SA	0.509	2.031	3.012	0.765	1.175	1.935
Design 6: 50% truncation, gamma (2, 1) mode						
QME	0.615	3.147	3.209	0.879	1.402	2.710
STLS	0.556	3.403	3.450	1.048	1.474	2.168
EA	0.598	3.479	3.581	0.881	1.057	1.111
SA	0.501	2.549	3.000	0.893	1.231	1.786
Design 7: 50% truncation, gamma (3, 1) mode						
QME	0.688	2.361	2.460	0.771	1.613	2.761
STLS	0.705	3.712	3.778	0.731	1.335	2.486
EA	0.676	2.476	3.247	0.790	0.895	1.431
SA	0.617	3.001	3.223	0.801	1.112	1.524

Table 1. $y_i = 0 + x_i + \varepsilon_i$, $i = 1, 2, \cdots, n$, $x_i \sim N(0, 1)$, $n = 30$, and 200 replications; only the slope is reported.

In Design 1, with 50% truncation and the standard normal distribution as the model error, we first check the effect of choosing w on the performance of SA algorithm. We note that $w = 0.5 - 2$ appears to be the range. So we try SA algorithm corresponding to w = 0.5, 1, 1.5, and 2, respectively. The algorithm gives quite "stable" results with w = 0.5, 1, and 1.5 but provides big variation for w = 2 under different replicates. This fact not only concludes that the selection of w may not be necessarily unique but also indicates that larger w may not have better outcome. So in the real data analysis of Section 4, we suppose that w follows a uniform distribution over the interval of $[w_1, w_2]$, where $w_1 = 0.5$ and $w_2 = 1.5$.

Because of the results from Design 1, we use w = 1 (a value between 0.5 and 1.5) in Designs 2 and 3 for SA algorithm but use $w = 0.1$, 0.5, 0.9, and 1.3 to construct a weighted version of QME (WQME) which is due to Lee's [2] suggestion that an average of the estimates for several ws may work along with WQME applied to the reasonable range of w. Actually, WQME sometimes performs better than a specific QME. STLS and EA seem to do well in both Designs 2 and 3. This, however, will change for distributions with thicker tails. Furthermore, STLS and EA get much more estimation variation than SA.

http://dx.doi.org/10.5772/66070

In Designs 4–7, we check the sensitivity to the underlying distribution, particularly to the tail behavior of the distribution. We consider the standard Cauchy, standard logistic, and gamma distributions with 50% truncation.

In Designs 4 and 5, except STLS, all methods perform similar. This is also true in Design 7. STLS method, in spite of the small bias on average, gets big estimation variation for most of cases except the case with a normal distribution. For example, among methods, STLS gives the biggest variance and RMSE for fat-tailed distribution (Cauchy) and skewed distributions (logistic and gamma). So STLS is not very reliable for practical application.

In Designs 4 and 5, EA and SA show slightly better than others in some of cases.

A final comment goes to algorithm speed, measured by CPU time. We found that CPU times for all algorithms used in **Table 1** are comparable, typically in the range between 5 and 10 s.

4. Returns of human capital in China

China has experienced rapid economic growth in the past 30 years, but the increase in wage inequality is gradually a serious problem and should be given more attention. We will analyze the important factors, which affect the incomes for most of Chinese workers. This study consists of an analysis of annual income of 1967 Chinese citizens with ages between 18 and 55 in 2008. The data is provided by Chinese Social Survey Open Database (CSSOD) (http://www.cssod.org/index.php in Chinese). We aim to check how education, experience, sex, district, and job sector affect the typical income. One may use mean regression to carry out the analysis, but average income is largely affected by small number of high-income receivers, so that the resulting analysis does not represent the majority people or the typical case. For example, majority workers often see their income as far lower than the reported average income. Therefore, mode regression is an ideal model to carry out the analysis.

We use a standard log-linear Mincer formulation:

$$\log Y = \beta_1 + \beta_2 Edu + \beta_3 Exp + \beta_4 Exp^2 + \beta_5 Gender + \beta_6 Dis + \beta_7 Sector + \varepsilon \tag{7}$$

where log Y is the logarithm transform of annual income (in 1000 Yuan) and truncated by 2, as majority annual income is more than 8000 Yuan. *Edu* is the number of years of schooling, *Exp* is potential experience (approximated by the age minus years of schooling minus 7), *Gender* is equal to 1 for male and 0 otherwise, *Dis* is equal to 1 for workers from the eastern China and 0 otherwise, *Sector* is equal to 1 for private sector workers and 0 otherwise, and ε is the model error. The estimate coefficient β_i means that a unit increase in the predictor variable results in an increase in $(100(\exp(\beta_i)-1))\%$.

We now carry out the mean and median regression analysis. That is, we fit the model (7) by the mean regression and the median regression model, respectively. By implementing the *lm*

and *rq* (in R-package *quantreg*) function in the R software, we obtain the fitted mean and median results as shown in **Table 2**.

	Mean regression	**Median regression**	**Mode regression**
Intercept	113.884***	134.376***	132.179***
	(9.755)	(11.486)	(228.825)
Education	9.381***	8.501***	3.519***
	(14.418)	(12.614)	(63.185)
Experience	2.103***	1.879***	3.887***
	(3.630)	(3.314)	(56.280)
Experience^2	−0.049***	−0.045***	−0.192***
	(−3.569)	(−3.173)	(−47.839)
Gender	27.919***	25.912***	20.921***
	(8.347)	(7.585)	(232.146)
District	44.463***	43.874***	42.571***
	(13.249)	(12.714)	(447.565)
Sector	7.024*	−2.418	−3.291***
	(1.919)	(0. 520)	(−23.626)

Note: Asymptotic standard errors are in parentheses for the mean and median regression.
* $p < 0.1$, ** $p < 0.05$, and *** $p < 0.01$ (two tailed)

Table 2. The results of three different regressions (%).

Table 2 shows that each additional year of education increases the conditional-mean income by a factor of exp(0.09381) = 1.09835, which indicates a 9.835% increase. And the fitted median regression model gives a coefficient of 0.08501, which indicates that one more year of education increases the conditional-median income by exp(0.08501) = 1.08873 or a 8.873% increase. That is, for small values of the estimated coefficient β_i, this is approximately 100 β_i%.

The experience entering the model in quadratic form is due to the most popular version of the Mincer equation [13], which includes a quadratic function in years of potential experience to capture the fact that on-the-job training investments decline over time in a standard life cycle human capital model. The estimated coefficients of years of experience and its square both are significant in the mean and the median regression. That is, the Chinese worker's income shows an inverted U-shaped relationship with years of experience and reaches the maximum at years of 21.459 and 20.878, respectively. This means that experience within about 20 years does make difference on the income but experience longer than 20 years would not add anything more for the conditional-mean or the conditional-median income.

http://dx.doi.org/10.5772/66070

Compared with being female, being male increases the conditional-mean income by 100[exp(0.27919)−1]% = 32.206% according to the mean regression results but by 100[exp(0.25912)−1]% = 29.579% according to the median regression results. In other words, the conditional-mean income for male is 32.206% higher than it is for female, and male's conditional-median income is 29.579% higher than female's, with other covariates held constant.

Similarly, the conditional-mean and conditional-median income difference between workers from the west and the east of China both is about 100[exp(0.44)−1]% = 55.271%.

For the dummy variable sector, the results of the mean regression indicate that the conditional-mean income of workers from private sectors is greater than from public sectors by a factor of exp(0.07024) = 1.072766, that is, a 7.277% increase in conditional-mean income. However, according to the result of the median regression model, the coefficient of sector is not significant, which means that there is no difference between the public and private sectors for the conditional-median income.

As is known that the selection of w may not be unique for the mode regression, we suppose that $w \sim$ U[0.5, 1.5] and draw 300 different random ws from this uniform distribution. Based on the 300 different ws, we implement the *GenSA* function in the R-package *GenSA* to fit the multivariate mode regression and obtain 300 different vectors of the coefficients $\widehat{\beta}$. Following the law of large numbers, the mean of these estimated $\widehat{\beta}_i$ must approach to the real β_i. Then we can use the two-tailed T test for checking if hypotheses about $\widehat{\beta}_i = 0$ are true or not. The mean and the two-tailed T test results of these seven coefficients β_i based on the 300 different mode regressions are shown in **Table 2**.

According to the results of mode regression based on the SA algorithm, the estimated returns to an additional year of education are about 3.519%, and the education has a much smaller effect on the conditional-mode income. Maybe this is one of the reasons why the typical income is lower than the mean and median income.

The coefficients of years of experience and its square both are significant. That is, the relationship between the conditional-mode income and the years of experience is shown in the inverted U-shaped relationship too, but the income for most workers reaches the maximum at 10.122 years of experience. The return to experience, i.e., the derivative of the typical value of log income with respect to experience, is therefore given by a combination of coefficients (3.887 − 2 × 0.192 × experience). The derivative needs to be evaluated at some specified level of experience. Two points were chosen: 5 years of experience, representing fairly new entrants, and 15 years of experience, representing experienced workers. And these two points give the combination of coefficients 1.967 and −1.873. That is, the return to experience of 5 years is 1.967% and experience of 15 years is −1.873%. In contrast with the results of the mean and median regression model, the return to experience of 15 years both is positive. The relationship between income and experience based on these three different regression models is represented in **Figure 3**.

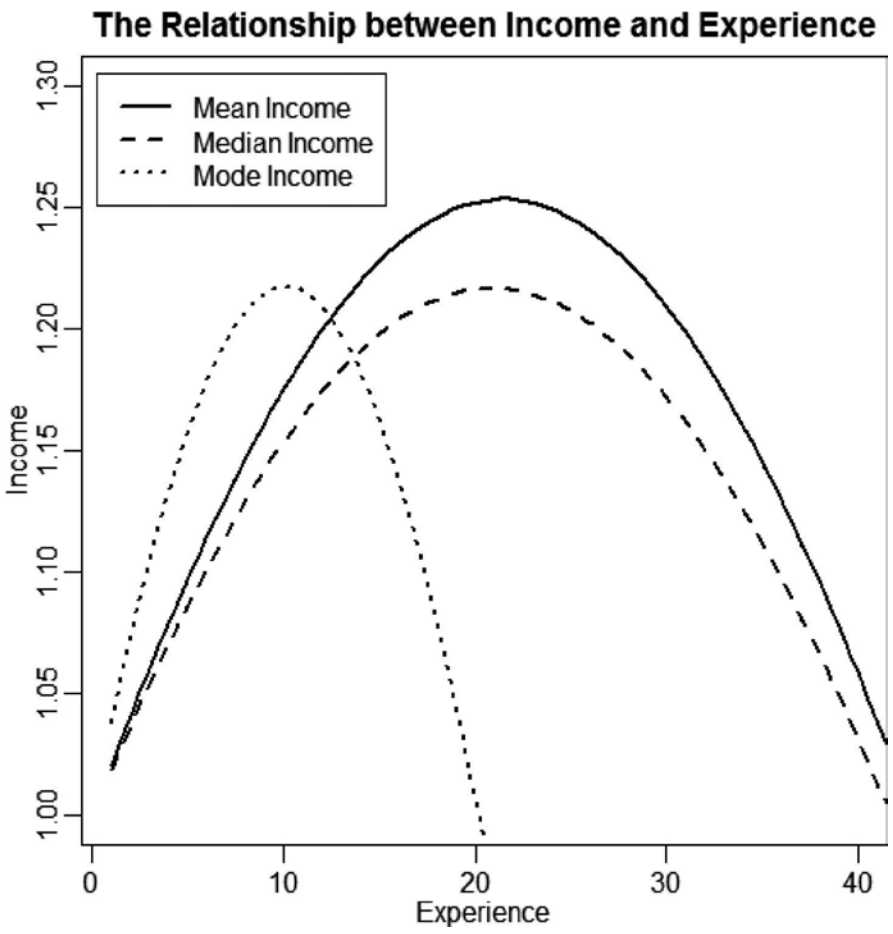

Figure 3. The relationship of income and experience.

As shown in **Figure 3**, we conclude that experience does make a great contribution to the conditional-mode income within 10 years, but experience longer than 10 years would not add anything more for it. That is, the conditional-mode income increases rapidly due to experience and decreases rapidly too after reaching the peak income. However, experience returns are positive until about 20 years to the conditional-mean and conditional-median income. Maybe this is another important reason why the typical income is lower than the mean and median income.

The conditional-mode income from a man is about 20.921% more than from a woman, which means that the gender pay gap is slightly less serious for majority workers compared with the mean and median regression. Similarly, the conditional-mode income from workers of eastern China is about 42.571% more than from worker of western China.

The private sector does not have positive return to mode income; this is in sharp contrast with the results of the mean regression. Concretely speaking, the conditional-mode income for the

http://dx.doi.org/10.5772/66070

public sector worker is 3.291% higher than for the private sector worker. This result is the opposite of what is shown in the mean regression model.

The results from the analyses of both mean and the mode regression models are inconsistent with the economic intuitions. In China, private sector includes private enterprises operated by local Chinese; Sino foreign joint ventures; Hong Kong-, Macao-, and Taiwan-funded enterprises; and foreign-funded enterprises. Workers in the private enterprises which account for the vast majority of the private sectors do not have the "compilation" or sign the labor contracts with these enterprises; thus, their rights and interests cannot be guaranteed and their wage is low. Although the workers in Sino foreign joint ventures and Hong Kong-, Macao-, and Taiwan-funded enterprises do not have the "compilation," they gain an attractive income because of these enterprises' relatively good efficiency. Workers in the foreign enterprises have the highest wages, especially for those CEOs. So the average wage level of workers in the private sector is relatively high. Most workers in the public sectors protected by the "compilation," often sign the labor contracts with the stated own enterprises, tend to gain a higher wage than the workers in the private enterprise, and the wage gap is very relatively small. Thus, the majority of workers have a lower wage in the private sector than in the public sector, but on average, the opposite is the case.

5. Conclusion

While mode regression has been found very useful in practical regression analysis, the main goal of this article is to introduce SA algorithm to fit mode regression models. The most popular mode regression model is introduced by Lee [2]. There is no doubt on the elegance of rectangular mode regression estimator of Lee [2], but the main problem with this estimator lies in computation. While the difficulty in obtaining reliable mode regression coefficients limits the application of the mode regression, we propose the SA algorithm for the mode regression estimation and then compare the proposed SA algorithm with other existing methods. To sum up, SA algorithm for fitting truncated mode regression does not require any liberalization or modification of Lee's estimator and solves the corresponding nonlinear optimization problem more efficiently and robustly as a rule.

As an application of the SA algorithm fitting a real data-based mode regression and the illustration of the sensible interpretation of fitted mode regression coefficients by the algorithm, we apply the mode regression model in income inequality analysis of China and some meaningful conclusions are obtained which are different from the mean regression and the quantile regression.

Acknowledgements

The research is supported in part by the National Sciences Foundation of China (11261048) and National Bureau of Statistics of China (2013LY022).

Author details

Maoxi Tian[1,2], Jian He[1] and Keming Yu[3*]

*Address all correspondence to: Keming.Yu@brunel.ac.uk

1 Department of Statistics, Shihezi University, Xinjiang, China

2 Jinhe Center for Economic Research, Xi'an Jiaotong University, Shaanxi, China

3 Department of Mathematics, Brunel University, London, UK

References

[1] Li J., Ray S., Lindsay B. G. A nonparametric statistical approach to clustering via mode identification. Journal of Machine Learning Research. 2007; 8(4):1687–1723. DOI: 10.1007/s10846-007-9145-xLi

[2] Lee M. J. Mode regression. Journal of Econometrics. 1989; 42(3):337–349. DOI: 10.1016/0304-4076(89)90057-2

[3] Heckman J. J. Sample selection bias as a specification error. Econometrica. 1979; 47(1): 153–161. DOI:10.2307/1912352

[4] Lee M. J. Quadratic mode regression. Journal of Econometrics. 1993; 57:1–19. DOI: 10.1016/0304-4076(93)90056-B

[5] Kemp G. C. R., Silva J. M. C. S. Regression towards the mode. Journal of Econometrics. 2012; 170(1):92–101. DOI:10.1016/j.jeconom.2012.03.002

[6] Yu K., Aristodemou K. Bayesian mode regression. Technical report (2012). arXiv: 1208.0579v1

[7] Yao W., Li L. A new regression model: modal linear regression. Scandinavian Journal of Statistics. 2013; 41(3):656–671. DOI: 10.1111/sjos.12054

[8] Powell J. L. Symmetrically trimmed least squares estimation for Tobit models. Econometrica. 1986; 54(6):1435–1460. DOI:10.2307/1914308

[9] Pinkse C. A. P. On the computation of semiparametric estimates in limited dependent variable models. Journal of Econometrics. 1993; 58(1–2):185–205.

[10] Fitzenberger B. A guide to censored quantile regressions. Handbook of Statistics. 1997; 15(97):405–437. DOI:10.1016/S0169-7161(97)15017-9

[11] Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing. Science. 1983; 220(4598):671–680.

http://dx.doi.org/10.5772/66070

[12] Theussl S., Borchers H. CRAN Task View: Optimization and Mathematical Programming. [Internet]. 2015. Available from: https://cran.r-project.org/web/views/Optimization.html

[13] Mincer J. Schooling, Experience, and Earnings. New York: Columbia University Press; 1974.

Chapter 6

Facility Layout Problem for Cellular Manufacturing Systems

Maral Zafar Allahyari and Ahmed Azab

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67313

Abstract

Good layout plan leads to in improve machine utilization, part demand quality, efficient setup time, less work-in-process inventory and material handling cost. Cellular Manufacturing (CM) is an application of GTCM is the combination of job shop and/or flow shop. Facility Layout Problem (FLP) for CMS includes both inter-cell layout and intra-cell layout. A bi-level mixed-integer non-linear programming continuous model has been formulated to fully define the problem and the relationship between intra-cell and inter-cell layout design. Facilities are assumed unequal size; operation sequences, part demands, overlap elimination, aisle are considered. The problem is NP-hard; hence, a simulated annealing meta-heuristic employing a novel constructive radial-based heuristic for initialization have been designed and implemented. For the first time, a novel heuristic algorithm has been designed to allocate and displace facilities in radial direction. In order to improve the search efficiency of the developed SA algorithm, the cell size used in the initialization heuristic algorithm is assumed twice as that of the original size of the cells. A real case study from the metal cutting inserts industry has been used. Results demonstrate the superiority of the developed SA algorithm against rival comparable meta-heuristics and algorithms from the literature.

Keywords: facility layout problem, cellular manufacturing, mathematical modelling, simulated annealing, aisle constraint

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

1. Introduction

Facility layout problem (FLP) is the arrangement of a given number of non-equal-sized facilities within the given space. Good layout plan leads to improve machine utilization, part demand quality, efficient setup time, less work-in-process inventory and material handling cost. Generally speaking, efficient layout design provides two main advantages: (1) Reduction of between 30% to 70% in the total material handling cost (MHC) and (2) designing layout is the long term-plan, hence, any changes in layout impose some expenditure such as shutting down production or service line, losing process time and so on. Thus, designing proper facility layout plan would prevent lots of costs [1].

Several algorithms have been developed for FLP problem. The traditional approach to FLP called discrete representation often addressed by quadratic assignment problem (QAP) with the objective of minimizing a given function cost. There are two main assumptions in QAP: firstly, all facilities are equal size and shape; secondly, the location of facilities is known in a priori. However, these kinds of assumptions are not applicable in real-world case studies. This approach to FLP is not suited to represent the exact location of facilities and cannot formulate FLP especially when facilities are unequal size and shape or if there are different clearances between the facilities. The more suitable approach to such a kind of cases is continuous representation rather than discrete. There are two ways to solve this problem. Chronologically, the first one attempts was to divide each facility into smaller size unit blocks, where the total area of those blocks is approximately equal to the area of the facility. There are two drawbacks to this method: firstly, the problem size is growing as the total number of blocks increase, and secondly, the exact shapes of facilities are ignored. The second approach to continuous problem assumes the exact shape and dimensions of the facilities (**Table 1**).

Approach	Plant site	Distance	Facilities	Mathematical formulation
Discrete	Divided in rectangular blocks with same size and shape; i.e., predetermined locations	Parameters Meller et al., [2]	Equal-sized	QAP
Continuous	No predetermined location, i.e., no blocks	Variable	Unequal-sized	MIP

Table 1. FLP discrete approach versus FLP continuous approach.

The design of a cellular manufacturing system (CMS) includes: (1) cell formation (CF), (2) group layout, (3) group scheduling and (4) resource allocation. FLP to CMS is focusing on the second step of design of CMS which by itself is twofold: inter-cell and intra-cell layouts. The main objective of group layout is minimizing material handling cost (MHC) by arranging facilities in their corresponding cells and cells in floor. In this chapter, both demand and operation sequencing have been considered in optimizing the layout both at inter- and intra-cellular levels. However, this was not the case with the literature; there is a dearth of papers that happened to take a discrete approach which really did address those factors. Moreover, in this chapter, a continuous approach has been adopted.

http://dx.doi.org/10.5772/67313

Here, a bi-level mixed-integer non-linear programming continuous model has been developed for both intra-cell and inter-cell layout design sequentially. The problem is to arrange facilities that are machine tools in the leader problem and cells in the follower problem on the continual planar site. The objective function of leader and follower problems is minimizing the material handling cost at intra- and inter-cellular levels, respectively. The developed mathematical model has some main novelties. Firstly, a continuous approach has been adopted; i.e., facilities take unequal size and their locations are not predetermined. Secondly, operation sequences and part demands are taken into consideration. Thirdly, the model has the ability to consider certain restrictions or preferences for cells and floors such as aisle. Finally, CMS design of disjoint cells is considered; hence, the overlapping elimination constraint is presented. Since the model is NP-hard, a novel heuristic has been developed to solve the problem at two different levels (intra- and inter-cellular) in a similar fashion to that used for developing the mathematical model. The developed heuristic is very different from its counterparts in the literature in the sense that it places the facilities radially, while dividing the production floor area into four quadrants. A real case study from the metal cutting industry has been used, where multiple families of inserts have been formed, each with its distinguished master plan.

2. Literature review

The block facility layout problem that was originally formulated by Armour and Buffa [3] is concerned with finding the most efficient arrangement of m indivisible departments with unequal area requirements within a facility [4]. As defined in the literature, the objective of the block layout design problem is to minimize the material handling costs by considering the following two sets of constraints: (a) department and floor area requirements; i.e. departments cannot overlap, must be placed within the facility, and some must be fixed to a location or cannot be placed in specific regions; see Refs. [1, 3, 5, 6].

Cellular layout is considered as one of the special cases of the general FLP. There is an increasing interest in solving the block layout problem by taking a continuous approach [6]. Alfa et al., [9] have developed a model to simultaneously solve group formation and intra-cell. The objective function is the summation of both inter-cell and intra-cell flow times based on distance. They develop SA/heuristic algorithm to solve their model. SA has been used to find the initial solution, and then a heuristic approach based on the penalty model developed to improve the solution. The main limitation of this model is that the cell locations are predetermined.

Bazargan-Lari and Kaebernick published few papers about design of cellular manufacturing [10–13]. Bazargan-Lari and Kaebernick [11] present a continuous plane approach where different constraints such as cell boundaries, non-overlapping, closeness relationships, location restrictions/preferences, orientation constraints and travelling distances have been considered. They develop a hybrid method which combined a non-linear goal programming (NLGP) and simulated annealing for machine layout problem. They have combined all constraints as goals using goal programming (GP) formulas. Generally speaking, GP divides

those constraints into two main categories such as absolute or hard and goal or soft constraints. Hard constraints are those that have to be satisfied absolutely. It means that violation of any of them would yield to infeasibility. However, soft constraints can be compromised and be offset from desired set goals. Those constraints are considered as three separate sets of objectives. The first priority level includes all set of absolute or hard objectives which have to be absolutely satisfied such as non-overlapped and cell boundary constraints. The second and third priority levels are preferences. The second priority is devoted to minimizing the area of the cells/shop floor, satisfying closeness relationship and orientation. Finally, the third priority is to minimize the total travelling cost. Overall, the approach of Bazargan-Lari and Kaebernick is a combination of the NLGP and SA. They use the pattern search to solve their NLGP based on those three priorities. Since a pattern search is finding the local minimum, then they have been using SA to exit from the trap of local minimum. The core of their model is that they are generating alternative layout design by changing the order of priority levels 2 and 3 in each outer loop of SA algorithm. In other words, the starting point of new outer loop of SA is generated by the patter search algorithm. By changing the goal priority levels, huge pools of efficient solutions are generating. To solve this issue, they used what they called the filtering process to choose which sets of solutions have more different with the other ones. The logic behind this is giving decision-makers the chance to consider how changing preferences' priorities would impact the solutions.

The other important piece of research was written by Imam and Mir [14, 15]. Imam and Mir [14] introduce a heuristic algorithm to place unequal-sized rectangular facilities in continuous plane by introducing the new concept of 'controlled coverage' by using 'envelop blocks'. In the initial solution, facilities are randomly placed in plane in the envelop block the size of which is much larger than the actual size of facility and is calculated by multiplying magnification factor with the facilities' actual dimensions. Afterwards, during the heuristic iterations, the sizes of envelop blocks are gradually decreased by decreasing the magnification factor until the dimensions of envelopes will became equal to the dimensions of their corresponding facilities. By this approach, they were controlling the coverage of facilities together. The improvement iteration is based on the univariate search method. In this method, only one of the $2n$ design variables where n is the number of facilities is changing at time. This change means moving facility horizontally or vertically along the x-axis or y-axis, respectively. There are three drawbacks to their method. Firstly, each iteration cycle is repeated $2n$ times, n times to move facilities horizontally and then another n more times to move them vertically. The other drawback is that facilities are just allowed to move horizontally or vertically, there is no diagonal movement. Thirdly, there are no borders for the assumed continuous plane. However, in real world, there is no plane without borders. The last drawback is related to magnification factor, they have not specified how large this factor has to be originally and by which fraction it has to be reduced in each iteration cycle.

Mir and Imam [15] have mentioned the second drawback above is addressed and try to improve their primary procedure. They develop a hybrid model by using SA for gaining the sub-optimal initial feasible solution and then they improved it using a steepest descent approach. As they also noted that the number of optimization iterations depends of the magnification factor by which the size of the envelope blocks reduces when the magnification

http://dx.doi.org/10.5772/67313

factor was being reduced. The algorithm stopped when the magnification factor is equal to one. So it is obvious that the computational cost and time are quite dependent on magnification factor.

On the other hand, there are various papers that considered alternative as a discrete approach. QAP is an NP-complete problem, which means that when the size of the problem is increasing it cannot be solved by exact algorithm [16]. Hence, lots of efforts have been made to develop and apply heuristic and meta-heuristic algorithm for this kind of problem. Wilhelm and Ward [16] have applied simulated annealing (SA) to solve QAP. Their results have been compared with the computerized relative allocation of facilities technique (CRAFT), biased sampling and revised Hillier problem and showed better quality solutions.

Baykasoğlu and Gindy [17] have applied SA for dynamic layout problem, discrete approach. They claim their proposed algorithm finds better solution. They compared their proposed algorithm to the three works done [18–20]. In the first comparison, their SA approach found optimum solution and revealed better solution than dynamic programming algorithm of Rosenblatt [18]. The second comparison has two experiments: first one carried out with no shifting cost and the SA algorithm found optimum solution and outperforms that Conway and Venkataramanan [19] genetic algorithm. In this experiment, relocation costs are included. The optimum solution was not found; however, the results of SA showed a slight improvement over that of Rosenblatt [18]. Finally, in the third comparison the data set obtained from Balakrishnan and Cheng [20]. They develop non-linear genetic algorithm (NLGA). The comparison between the SA-based approach and NLGA reveals the superiority of SA algorithm when the size of the problems is large. Since they have taken discrete approach to FLP, the only operator has been used in neighbourhood generation algorithm is the swap operator.

Tavakkoli-Moghaddam et al., [21] are developed a non-linear mathematical modelling to solve the cell formation in dynamic environment in which demand varies in each time horizon. The strength point of their model is that it is a multi-objective model, i.e. considering more than one objective such as machine cost, operating cost, inter-cell material handling cost and machine relocation cost. Three meta-heuristic models, such as genetic algorithm (GA), simulated annealing (SA) and tabu search (TS), have been used to solve this problem. The results show SA outperforms compare to the two meta-heuristics.

Safaei et al., [22] have developed a mixed integer programming which tries to minimize machine constant and variable costs, inter- and intra-material handling cost and reconfiguration costs. They present a hybrid model called mean field annealing and simulated annealing (MFA-SA) to solve the problem. MFA stands for mean field annealing which used to find the feasible initial solution for SA. Most of the developed heuristics in the literature have taken a discrete approach to FLP than a continuous one. Developing heuristics for the discrete problem is easier, because locations are predetermineda priori; hence, the only operator that is usually used is the swap operator, to shuffle the different facilities locations. Moreover, in the discrete approach no overlap would happen between facilities. On the other hand, it is harder to design heuristics for the continuous formulation of FLP since overlap takes place. It is usually the case that repeated repairs and checks of validity of the generated solutions have to take place.

3. Mathematical modelling

The problem is to arrange facilities that are cells in the leader problem and machine tools in the follower problem in their respective space. The site has a rectangular shape with specified length (L) and width (W). Moreover, there is a horizontal aisle in the site by the same length as of site, however, with two different vertical dimensions $\mathbf{Y_{VALU}}$ and $\mathbf{Y_{VALL}}$. Aisle divides the site into two sections, upper and lower. No facilities are allocated to the aisles. The objective is to minimize the total travel-flow cost by considering shape, size and geometric characteristic of the different facilities. Facilities have a rectangular shape. The position of each facility is determined by the coordinates of its centroid as well as its predetermined length and width. Facilities are not allowed to overlap each other and have to be assigned in their related boundary areas, which is the overall site's boundaries for the follower problem and that of the cell for the leader problem. The traditional Cartesian coordinate system, shown in **Figure 1**, represents the scheme used in this chapter. The following model has represented by Allahyari and Azab [7, 8] and Allahayri [7]. The problem is formulated under the following assumptions [6]:

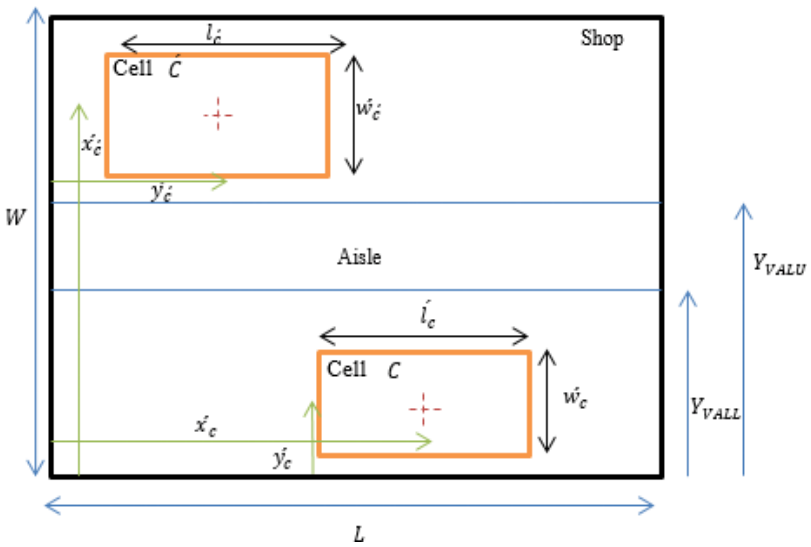

Figure 1. Scheme of shop.

The problem is formulated under the following assumptions:

1. CF is known in advanced.
2. Machines are not in the same size.
3. Machines must be located within a given area.
4. Machines are not allowed to overlap each other.
5. Cell's dimensions and orientation are predetermined.

http://dx.doi.org/10.5772/67313

6. Each part type has a number of operations that must be processed based on its operation sequence readily available from the route sheet of parts. It should be noted that the process sequence of each part is different.
7. The demand for each part type in known and is constant.
8. Material handling devices moving the one part between machines.
9. Inter- and intra-cell movements related to the part types have different costs that are related to the distance travelled. We assume the rectangular distance between each pair of machines' centroid.
10. In determining machine size and dimensions, the workspace required for operator usage and that needed to enforce between the different machines have been taken into account.

The mathematical formulation represented as below

Sets:

$P = \{1, 2, 3, \ldots, P\}$ Index set of part types

$M = \{1, 2, 3, \ldots, M\}$ Index set of machine types

$C = \{1, 2, 3, \ldots, C\}$ Index set of cell types

$O_p = \{1, 2, 3, \ldots, O_p\}$ Index set of operations indices for part p

Parameters:

L Horizontal dimension of shop floor

W Vertical dimension of shop floor

Y_{VALU} Vertical dimension of upper side of aisle

Y_{VALL} Vertical dimension of lower side of aisle

X_{HALLF} Horizontal dimension of left side of aisle

X_{HALRT} Horizontal dimension of right side of aisle

l_i Length of machine i

w_i Width of machine i

l_c Length of cell c

w_c Width of cell c

$\boldsymbol{CA_j}$ Intra-cellular transfer unit cost for part j

$\boldsymbol{CE_j}$ Inter-cellular transfer unit cost for part j

$\boldsymbol{D_j}$ Demand quantity for part j

U_{joi} 1, if operation o of part j is done by machine i, otherwise 0

U'_{joi} 1, if operation o of part j is done by machine i which is located in cell c, otherwise 0

Q_{ic} 1, if machine i is assigned in cell c

Decision variables:

x_i Horizontal distance between centre of machine i and vertical reference line

y_i Vertical distance between centre of machine i and horizontal reference line

x'_c Horizontal distance between centre of cell c and vertical reference line

y'_c Vertical distance between centre of cell c and horizontal reference line

Z_{iu} 1, if machine u is arranged in the same horizontal level as machine i, and 0 otherwise

$W_{cc'}$ 1, if cell c is arranged in the same horizontal level as cell c' and 0 otherwise

Z_c 1, if cell c is arranged in out of aisle horizontal boundaries and 0 otherwise

W_c 1, if cell c is arranged in out of aisle vertical boundaries and 0 otherwise

The continuous bi-level programming problem is defined as: the intra-cell layout mathematical formulation to layout the different machines (machines here are the facilities) of every cell c at a time is as follows:

$$\text{Min} \sum_{j=1}^{P} \sum_{o=1}^{o_p-1} \sum_{\substack{i,u=1 \\ i \neq u}}^{M} U_{joi}\, U_{jo+1u}(|x_i - x_u| + |y_i - y_u|)\, CA_j D_j \tag{1}$$

s.t.

$$x_i + \frac{l_i}{2} \leq l_C \qquad i = 1,..,M \tag{2}$$

$$x_i - \frac{l_i}{2} \geq 0 \qquad i = 1,..,M \tag{3}$$

$$y_i + \frac{w_i}{2} \leq w_c \qquad i = 1,..,M \tag{4}$$

$$y_i - \frac{w_i}{2} \geq 0 \qquad i = 1,..,M \tag{5}$$

$$|x_i - x_u| \geq Z_{iu}(l_i + l_u)/2 \qquad i,u = 1,..,M \tag{6}$$

$$|y_i - y_u| \geq (1 - Z_{iu})(w_i + w_u)/2 \qquad i,u = 1,..,M \tag{7}$$

$$x_i,\ y_i \geq 0, Z_{iu} \text{ are binary} \qquad i,u = 1,..,M \tag{8}$$

Equation (1) declares the objective function of leader problem, which is minimizes the total intra-cell transportation cost of parts. Equations (2)–(5) are within site constraints that ensure each machine tool is assigned within the boundaries of its corresponding cell. Equations (6)

http://dx.doi.org/10.5772/67313

and (7) force the overlap elimination for machine tools. Equation (8) represents the nature of the decision variables which are binary and non-negative.

Finally, the inter-cell layout problem tries to layout the different cells (cells here are the facilities) of the entire shop floor is as follows:

$$\text{Min}\sum_{j=1}^{p}\sum_{o=1}^{o_p-1}\sum_{\substack{c,c'=1\\ c\neq c'}}^{C} U'_{joc}\,U'_{jo+1c'}\,(|x'_c - x'_{c'}| + |y'_c - y'_{c'}|)CE_jD_j \tag{9}$$

s.t

$$x'_c + \frac{l'_c}{2} \le L \qquad c = 1,..,C \tag{10}$$

$$x'_c - \frac{l'_c}{2} \ge 0 \qquad c = 1,..,C \tag{11}$$

$$y'_c + \frac{w'_c}{2} \le W \qquad c = 1,..,C \tag{12}$$

$$y'_c - \frac{w'_c}{2} \ge 0 \qquad c = 1,..,C \tag{13}$$

$$|x'_c - x'_{c'}| \ge W_{cc'}(l'_c + l'_{c'})/2 \qquad c,c' = 1,..,C \tag{14}$$

$$|y'_c - y'_{c'}| \ge (1 - W_{cc'})(w'_c + w'_{c'})/2 \quad c,c' = 1,..,C \tag{15}$$

Aisle constraints:

Horizontal aisle:

$$(\boldsymbol{y'_c} + \boldsymbol{w'_c}/\mathbf{2}) - \boldsymbol{Y_{VALL}} \le \mathrm{M}\,\boldsymbol{Z_c} \tag{16}$$

$$\boldsymbol{Y_{\mathbf{VALU}}} - (\boldsymbol{y'_c} - \boldsymbol{w'_c}/\mathbf{2}) \le \mathrm{M}\,(\mathbf{1} - \boldsymbol{Z_c}) \tag{17}$$

Vertical aisle:

$$(\boldsymbol{x'_c} - \boldsymbol{l'_c}/\mathbf{2}) - \boldsymbol{X_{\mathbf{HALRT}}} \le \boldsymbol{M W_c} \tag{18}$$

$$\boldsymbol{X_{HALLF}} - (\boldsymbol{x'_c} + \boldsymbol{l'_c}/\mathbf{2}) \le \boldsymbol{M}\,(\mathbf{1} - \boldsymbol{W_c}) \tag{19}$$

$$\boldsymbol{x'_c, y'_c} \ge \boldsymbol{0}, \mathbf{W_{cc'}}, \boldsymbol{Z_c}, \boldsymbol{W_c} \text{ are binary } \boldsymbol{c} = \boldsymbol{1},..,\boldsymbol{C} \tag{20}$$

Equation (9) represents the objective function of follower program. The objective function minimizes the inter-cell transportation cost of parts. The within-site constraints are enforced by the set of constraints 10–13; i.e. these constraints ensure that cells are assigned within the boundaries of shop floor. Moreover, overlap elimination constraints are defined by constraints (14) and (15) which enforce the overlap elimination among cells. Equations (16) and (19) in the follower problem ensure that no cells would be assigned in the aisle boundaries. Finally, Eq. (20) specifies that the decision variables are binary and positive.

4. Simulated annealing

Simulated annealing is a stochastic neighbourhood search technique, which was initially developed by Metropolis and applied to combinatorial problems by Kirkpatrich et al. [25] for the first time.

To begin with, the basic of SA is based on statistical mechanics and comes from the similarity between the annealing of solids process and the solving method of combinatorial problem. If each feasible solution to the combinatorial optimization problem as a configuration of atoms and the objective function value of corresponding feasible solution as the energy of the system, then the optimal solution of combinatorial optimization problem is as like as the lowest energy state of the physical system [23]. The core of heuristic algorithms for solving the combinatorial problem is based on continual improvement, moving from one solution to another one in order to decrease the objective function from one iteration to next one. The same procedure is taking in quenching the system from high to low temperature in order to reach the required quality.

4.1. The elements of an SA algorithm

The core of SA algorithm is Metropolis algorithm, which allows uphill moves sometimes. Metropolis algorithm has four main elements [24, 25]. **Figure 2** represents the simulated annealing steps.

1. Initial solution and description of system configuration

It is the starting point of SA algorithm. There are two main approaches for generating initial solution. One is generating initial solution randomly; by taking this approach feasibility of initial solution has to be considered. The second approach is getting feasible initial solution by adapting greedy algorithms or another heuristic algorithm. It has to be noted that initial solution should not be too good because escaping from its local optimum is hard.

2. Configuration changes

By moving from one configuration to another one, new neighbourhood solution is generated. These changes occurred by defining some operators which are responsible to make changes in the current solution.

3. Objective function that represent the quantitative measurement of goodness of a system

After finding any neighbour, the difference between objective value of new solution (E_{n+1}) and of the current solution (E_n) is calculated. If ($\Delta E < 0$), it means that the objective value of neighbourhood solution is showing improvement in comparison to the objective value of the current solution found so far ($\Delta E < 0$). Hence, the current one will be accepted as the new best solution. On the other hand, if ($\Delta E \geq 0$) the new solution is accepted with a certain probability. Using this approach, SA tries to exit from the local optima region in which it trap. The probability is based on the so-called Boltzmann probability distribution

$$Prob(\Delta E) \sim exp(-\Delta E / k_b T) \tag{21}$$

http://dx.doi.org/10.5772/67313

where T is the parameter and k_b is the Boltzmann's constant which is not required when Metropolis algorithm is applying to combinatorial problems [16]. The acceptance probability of new solution depends on two factors, one is how large is this difference. The bigger the difference, the lesser the chance of accepting this new solution. The second criterion is a control parameter (temperature). It should be noted if the initial temperature is not large enough or it decreases dramatically the chances that the algorithm traps at local optima is high.

4. Annealing schedule/cooling schedule

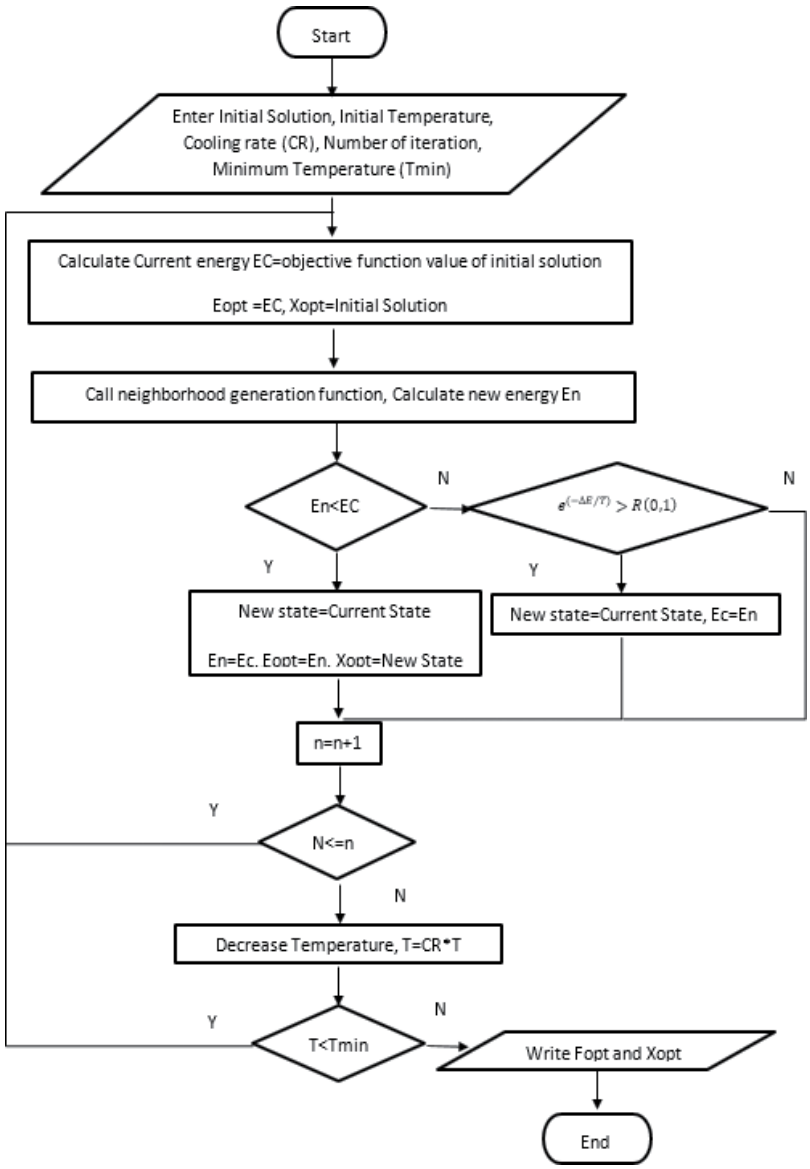

Figure 2. Flowchart of simulated annealing.

The annealing schedule determines four rules:

1. Initial temperature: Since the annealing of solids is the basic of the SA approach, initial temperature is the melting point of SA algorithm and it should be defined in such a way that the solutions generated by high acceptance probability approximately close to one. Kirkpatrick et al. [25] noted that the initial temperature has to be large enough that 80% of generated solutions are accepted. Kia et al., [26] and Baykasoğlu and Gindy [17] defined initial solution high enough in such a way that 95% of generated candidates can be accepted using the following equation:

$$T_0 = \frac{\text{Obj}v_j - \text{Obj}v_i}{\ln(0.95)} \tag{22}$$

Objv_j and Objv_i are the objective values of two random solution i and j, respectively. It should be noted initial solution T_0 is generated once at the beginning of SA algorithm.

2. Temperature length

3. Termination: There are different approaches for stopping criteria such as

- A specific number of iteration
- Exact final temperature
- No improvement for a number of iteration

Based on the literature review, there are different approaches for choosing SA parameters as explained briefly in **Table 2**.

Author	Initial temperature (T_0)	Cooling rate (α)	Temperature reduction	Loop length	
				Inner	Outer
Bazargan-Lari and Kaebernick [11]	10	0.9	$t_i = 10(0.9)^{i-1}$	$N' \times n$	K
Baykasoğlu and Gindy [17]	$T_{\text{in}} = \frac{f_{\min} - f_{\max}}{lnP_c = \ln(0.95)}$	$\propto = \left(\frac{\ln P_c}{\ln P_f}\right)^{1/(eL_{max}{}^{-1})}$	$T_{el+1} = \alpha T_{e1}$	I_L>LMC	$el_{\max}$ calculated
Heragu and Alfa [27]	999	0.90	$T = rT$	Epoch concept $N' \times n$	K
Wilhelm and Ward [16]	10	0.9	$t_i = 10(0.9)^{i-1}$	Epoch concept $N' \times n$	K

Epoch: Predetermined specific number of successful pairwise interchanges at each temperature.
N' : Predetermined integer.
n : Total number of facilities.
K : Predetermined integer- the total number of temperature steps.

Table 2. SA parameters.

http://dx.doi.org/10.5772/67313

4.2. Developed simulated annealing for FLP

4.2.1. Initialization

A unique heuristic is used to generate a feasible initial solution for SA algorithm [7, 8]. The explanation of the developed heuristic is provided in Section 4.2.1.1.

4.2.1.1. Initialization heuristic

The mechanics of the developed algorithm are very different than any of the available heuristics in the literature. The whole idea behind our algorithm is to place facilities radially along vectors $\vec{r_f}$ that are originated from the centroid of the space considered, where all facilities are to be placed as shown in **Figure 3**. The radial vectors along which all facilities are to be placed are distant radially by an angle $\theta = \frac{360^0}{M}$.

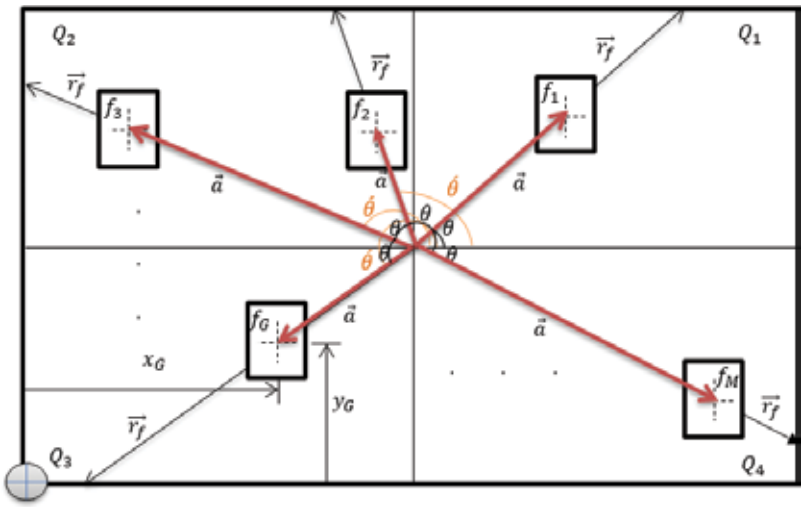

Figure 3. The mechanics of developed heuristics.

At the start of the heuristic method, at first the given area is first divided into four equal size quadrants; i.e. Q_1, Q_2, Q_3, and Q_4. Afterwards, all facilities are placed on top of each other in the middle of the given area. The developed heuristic algorithm consists of the two nested loops.

4.2.1.1.1. Outer loop

For each iteration of the outer loop, one random facility (called target facility) f_G is chosen and located radially along the radius (r_f), which is making an angle θ' with the abscissa, as shown in **Figure 3**.

$$\acute{\theta} = i \times \theta \qquad i = 1, 2, \ldots, M \tag{23}$$

Facilities are permitted to be placed within the boundaries of the given area. In order to satisfy this constraint, vector $\vec{a}$, which is a vector of random magnitude along vector's $\vec{r_f}$ direction, is taken,

and facility, f_G, is placed at the end of this vector. The length of vector $\vec{a}$ is a random number between $[0, |\vec{r_f}| - r]$, where r is the length of the diagonal of facility f_G. The next step is checking the possibilities for overlap between all facilities. If any overlap occurs between the target facility f_G and the given area's boundaries or between target facility f_G and the previously placed facilities, the inner loop is triggered. It should be noted that the facility coordinates for each is calculated based on an origin that is located at the bottom-left corner of the site as shown in **Figure 3**.

4.2.1.1.2. Inner loop

Different repair functions based on the type of overlap are being developed to eliminate overlap. Repair functions guarantee the elimination of overlap between facilities and allocation of the facility within the boundaries of its corresponding quadrant. However, if the corresponding quadrant is too congested, the overlapped facility can be placed partially in a different quadrant. Nevertheless, no facilities are allowed to violate the given area boundaries. The inner loop has two main steps: in the first step, the overlap between facility f_G and the overlapped facility f_j is repaired. Afterwards, overlap checking is performed for all facilities starting from the last placed facility to the first one to see if repair done in previous step has caused further overlaps or not. If no overlap takes place, the inner loop is ended and algorithm goes back to the outer loop to place another facility, given a facility is still left to be placed. However, if overlap is detected when checking for overlap between all the facilities, the second step of the inner loop is enacted.

The second step of the inner loop consists of few iterations. In each iteration, as explained one facility f_i is selected as target facility, and then the possibility of overlap between the target facility and rest of previously placed facilities is checked. If there is overlap between the target facility f_i and facility f_j, overlap elimination algorithms are enacted. The overlap has two main projections: one in the x-direction, Δx, and another in the y-direction, Δy. Δx represents the horizontal overlap between the two facilities f_i and f_j. In a similar fashion, Δy shows the vertical overlap between the two overlapped facilities as demonstrated in **Figure 4**. If $\Delta x \leq \Delta y$,

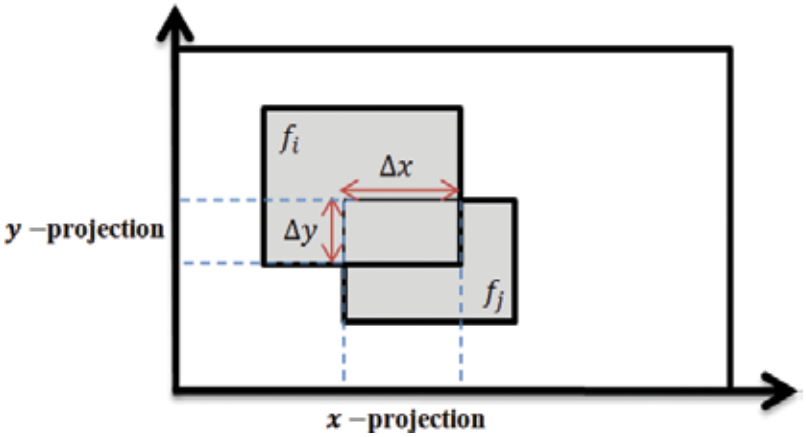

Figure 4. Scheme of overlap between two facilities f_i and f_j.

http://dx.doi.org/10.5772/67313

the overlap is fixed by removing overlap in the x-projection direction; otherwise, it does that in the y-direction. The repair mechanism starts by moving target facility f_i by the overlap distance Δ in appropriate direction.

Since no facility is allowed to violate the given area's boundaries, there is a need to know how much distance left between facility f_i and cell/floor (or quarter) boundaries. If the distance left is less than overlap Δ, then overlap elimination is carried out for the facility f_j. Moreover, if the distance left between the facility f_j and site (or quarter) boundaries is not less than overlap Δ, the overlap distance Δ should be applied to both facilities f_i and f_j. At the end of each iteration, the overlap is checked once again to tackle any possibility of newly occurred overlap. This loop is repeated until all overlap and intersection between facilities are repaired. The summary of the developed initialization heuristic is represented in **Figure 5**.

Step 1: Place all facilities on top of each in the centroid of the given area.

Step 2: Divide the given area into four quadrants and calculate angle between facilities. $\theta = \frac{360^o}{M}$

Step 3: Outer loop

- Step 3.1: Randomly Choose one facility as target f_G
- Step 3.2: Take radial movement:
  - Step 3.2.1: Calculate angle $\hat{\theta}$ of corresponding target facility f_G and specify radial r_f
  - Step 3.2.2: Find a vector $\vec{a}$ along vector $\overrightarrow{r_f}$
- Step 3.3: Place target facility f_G at the end of vector $\vec{a}$
- Step 3.4: overlap checking: if there is any overlap between target f_G and facilities that have already been placed go to step 4, otherwise go to step 5

Step 4: Inner loop

- Step 4.1: Specify the corresponding quarter of facilities f_i and f_j
- Step 4.2: Calculate the overlap Δ based on the comparision between Δx and Δy projections of the overlap between the two overlapped facilities f_i and f_j
- Step 4.3: Apply an appropriate repair function
- Step 4.4: Overlap checking: if there is any overlap go to step 4, otherwise go to step 5

Step 5: if all facilities placed on the floor (cell) go to step 6, otherwise go to step 3

Step 6: End

Figure 5. Summary of developed initialization heuristic algorithm.

4.2.2. Neighbourhood solution scheme

In order to generate new neighbourhood solution, two main operators, namely, move operator and swap operator, have been developed. The move operator tries to make facilities close to each other and also the swap operator switches the location of the two facilities. The details about these two operators explained below.

4.2.2.1. Move operator

The developed move operator tries to reduce distances between the facilities. The logic behind this algorithm is decreasing the distance between one facility called in-context facility, which is chosen randomly and the closest facility towards that. By moving the in-context facility towards its closest facility, the possibility of overlap between in-context facility and the rest of facilities is decreased. Main point here is that how much the maximum_movable_ distance is. Maximum_movable_ distance is the maximum length which if in-context facility moved towards its closest facility no overlap will happen between them. The steps of move operator algorithm are explained below:

1. Randomly choose one facility, called in-context facility f_G.

2. The Euclidean distance between the centroid of in-context facility f_G and the rest of facilities is calculated.

$$\text{Dis}_{Gi} = \sqrt{(X_G - X_i)^2 - (Y_G - Y_i)^2} \qquad \forall\, i = 1, 2, \ldots, M \text{ and } i \neq G \tag{24}$$

3. Facilities are sorted based on the distances found in step 2 in the descending order. The first one among the above set would be the closest facility f_C to the in-context facility f_G.

4. Divide the in-context facility f_G into four equal-sized quadrants by the origin of its centroid.

5. Find in which quadrant of in-context facility f_G the closest facility f_C is located.

6. At this point the maximum _movable_ distance $|\overrightarrow{CC'}|$ is calculated. For finding this distance, two points C and C' have to be found. C is the conjunction of vector $\vec{r'}$ and the closest boundary of in-context facility f_G to the closest facility f_C; and C' is the conjunction of vector $\vec{r''}$ and the closest boundary of closest facility to in-context facility. To do this, these concepts are defined:

$\overrightarrow{O'O''}$: Vector between centroids of in-context facility f_G and closest facility f_C.

$|\overrightarrow{CC'}|$: Maximum_movable_distance

θ_1: The angle between vector $\overrightarrow{O'O''}$ and horizontal line

http://dx.doi.org/10.5772/67313

θ_2: The angle between vector $\overrightarrow{O'O''}$ and vertical line

$\vec{r'}$: Vector from centroid of in-context facility O' to the closest boundary of in-context facility f_G towards the closet facility f_C.

$\vec{r''}$: Vector from centroid of the closest facility O'' to the closet boundary of the closest facility f_C toward the in-context facility f_G.

$$\theta_1 = \tan^{-1}\frac{|\text{Opposite side}|}{|\text{Adjacent side}|} = \tan^{-1}\frac{|Y_G - Y_C|}{|X_G - X_C|} \tag{25}$$

$$\theta_2 = \tan^{-1}\frac{|\text{Opposite side}|}{|\text{Adjacent side}|} = \tan^{-1}\frac{|X_G - X_C|}{|Y_G - Y_C|} \tag{26}$$

Also: $\theta_2 = 90 - \theta_1$

where X_G and Y_G are vertical and horizontal coordinates of centroid of in-context facility f_G, respectively. Similarly, X_C and Y_C are vertical and horizontal coordinates of centroid of in-context facility f_C, respectively.

It has to be noted, the length of both vectors $\vec{r'}$ and $\vec{r''}$ depends on their corresponding angles θ_1 and θ_2. **Figures 6** and **7** illustrate this topic.

$$|\vec{r'}| = \begin{cases} \dfrac{Adjacent\ side}{Cos\theta_1} = \dfrac{L_G/_2}{Cos\theta_1} & if \quad 0 \le \theta_1 \le 45^0 \\ \dfrac{Opposite\ side}{Sin\theta_1} = \dfrac{W_G/_2}{Sin\theta_1} & if \quad 45^0 \le \theta_1 \le 90^0 \end{cases} \tag{27}$$

$$|\vec{r''}| = \begin{cases} \dfrac{Adjacent\ side}{Cos\theta_2} = \dfrac{W_C/_2}{Cos\theta_1} & if \quad 0 \le \theta_2 \le 45^0 \\ \dfrac{Opposite\ side}{Sin\theta_2} = \dfrac{L_C/_2}{Sin\theta_1} & if \quad 45^0 \le \theta_2 \le 90^0 \end{cases} \tag{28}$$

where L_G and W_G are length and width of in-context facility f_G, respectively. Similarly, L_C and W_C are length and width of in-context facility f_C, respectively.

Based on in which quadrant closing facility is located, C and C' coordinates are calculating by equations shown in **Table 3**.

Hence, the length of vector $|\overrightarrow{CC'}|$ is

$$|\overrightarrow{CC'}| = \sqrt{(X_C - X_{C'})^2 - (Y_C - Y_{C'})^2} \tag{29}$$

7. At this point the length of the movement, called *ml* is the random number in interval $(0, |\overrightarrow{CC'}|]$. Furthermore, the direction of movement is along the vector $\overrightarrow{CC'}$.

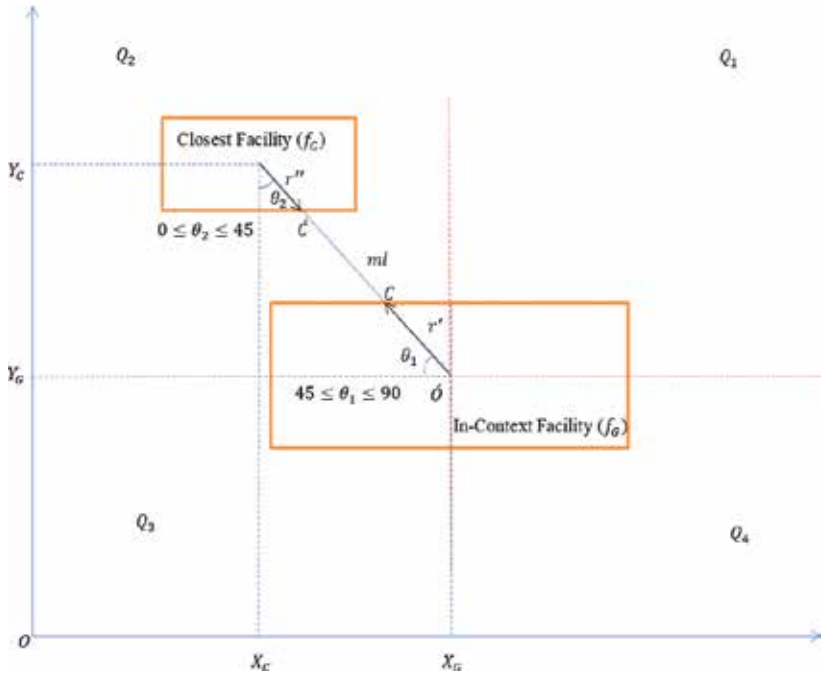

Figure 6. Angle calculation in move operator (I).

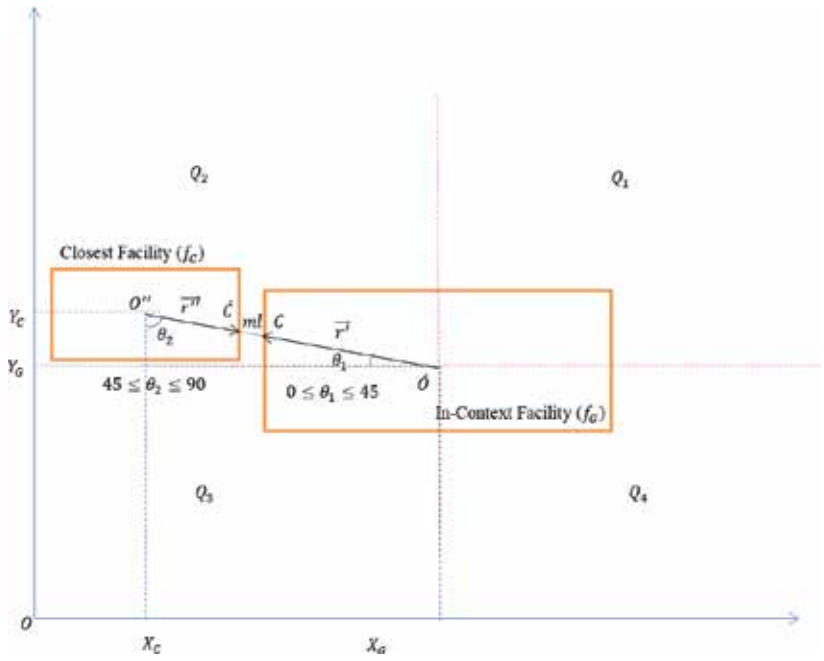

Figure 7. Concept of angle in move operator (II).

http://dx.doi.org/10.5772/67313

8. If the closest facility is adjacent to the facility f_G, find the other closest facility and go to step 5, otherwise go to step 9.

9. Finally, new coordinates of in-context facility f_G are calculated and shown in **Table 4**.

	Coordinates	
Quadrant	c	c'
1	$(X_G + r'\mathbf{Cos}\theta_1,\ Y_G + r'\mathbf{Sin}\theta_1)$	$(X_i - r''\mathbf{Cos}\theta_2,\ Y_i - r''\mathbf{Sin}\theta_2)$
2	$(X_G - r'\mathbf{Cos}\theta_1,\ Y_G + r'\mathbf{Sin}\theta_1)$	$(X_i + r''\mathbf{Cos}\theta_2,\ Y_i - r''\mathbf{Sin}\theta_2)$
3	$(X_G - r'\mathbf{Cos}\theta_1,\ Y_G - r'\mathbf{Sin}\theta_1)$	$(X_i + r''\mathbf{Cos}\theta_2,\ Y_i + r''\mathbf{Sin}\theta_2)$
4	$(X_G + r'\mathbf{Cos}\theta_1,\ Y_G - r'\mathbf{Sin}\theta_1)$	$(X_i - r''\mathbf{Cos}\theta_2,\ Y_i + r''\mathbf{Sin}\theta_2)$

Table 3. C and C' coordinates.

	New coordinates of target facility	
Direction	X_G	Y_G
Quadrant 1	$X_G + ml * \mathrm{Cos}\theta_1$	$Y_G + ml * \mathrm{Sin}\theta_1$
Quadrant 2	$X_G - ml * \mathrm{Cos}\theta_1$	$Y_G + ml * \mathrm{Sin}\theta_1$
Quadrant 3	$X_G - ml * \mathrm{Cos}\theta_1$	$Y_G - ml * \mathrm{Sin}\theta_1$
Quadrant 4	$X_G + ml * \mathrm{Cos}\theta_1$	$Y_G - ml * \mathrm{Sin}\theta_1$

Table 4. New coordinate of f_G after move.

4.2.2.2. Swap operator

The second operator of the developed SA is the swap operator which is switching positions of two facilities. The point here is how swap two facilities together that with the minimum possibility of overlap. To do that, the new concepts called free zone is defined. To apply this concept, a random facility called f_G is chosen and the available free space around this facility called FZ_G is determined by applying the maximum_movable_distance concept introduced in move operator. It has to be noted the centroid of free zone FZ_G is the same as centroid of the facility f_G. If there is any facility whose area is greater than the area of the facility f_G and less than the area of free zone FZ_G then that facility is qualified for swapping. By swapping this facility with facility f_G the possibility of occurrence of overlap is minimized. Moreover, if there is more than one facility which are qualified to swap with the facility f_G, one facility is chosen randomly. **Figure 8** shows the scheme of free zone concept. The algorithm below explained swap operator's steps in detail:

1. One facility is chosen randomly, called facility f_G.

2. The closest facility to the f_G is determined-details mentioned in move operator.
3. Maximum_movable_distance is calculated.
4. Free zone FZ_G of facility f_G is determined.
5. Areas of facility f_G and FZ_G are calculated.
6. Among the rest of facilities those ones whose areas are greater than the area of facility f_G and less than the area of free zone FZ_G are found.
7. Randomly one facility among those facilities is found in step 6 is chosen, call it f_i.
8. Swap facility f_G to the facility f_i.
9. Calculated the new coordinates of both f_G and f_i.
10. End

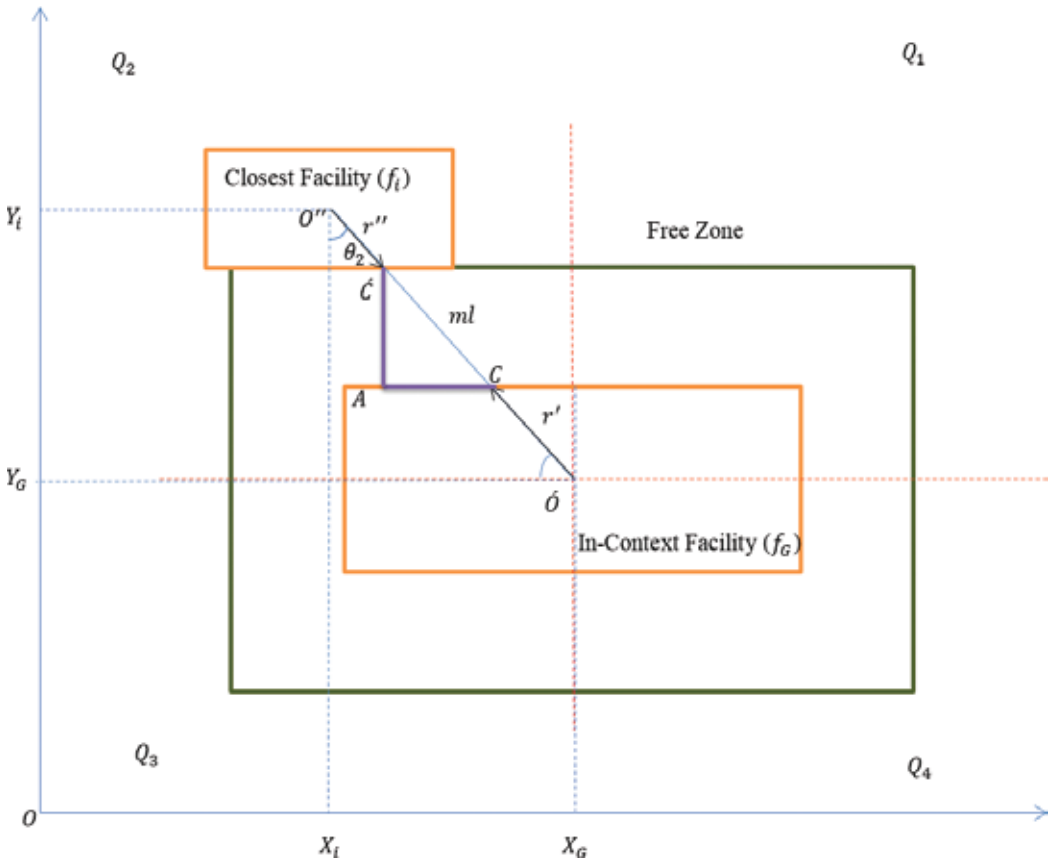

Figure 8. Free zone concept.

http://dx.doi.org/10.5772/67313

Assume:

L_G: Length of the f_G

W_G: Width of the f_G

ml: Maximum movable distance

LFZ: Length of the *FZ*

WFZ*:* Width of the *FZ*

AFZ*:* Area of *FZ*

$$AC = \min((X_G - {}^{L_G}/_2), ml \times \mathrm{Cos}\theta_1) \tag{30}$$

$$AC' = \min((Y_G - {}^{W_G}/_2), ml \times \mathrm{Sin}\theta_1) \tag{31}$$

$$LFZ = L_G + 2AC \tag{32}$$

$$WFZ = W_G + 2AC' \tag{33}$$

$$\mathrm{AFZ} = \mathrm{LFZ} \times \mathrm{WFZ} \tag{34}$$

4.2.3. Aisle constraints

In case of aisle, the operators move and swap vary. The details are presented in the below section.

4.2.3.1. Move operator

The move operator has the same procedure as the move operator developed in case of no aisle. Hence, in case of aisle one facility is chosen randomly f_G and moves to its closest facility f_C. Afterwards, the possibility of overlap between aisle and new position of facility f_G called $f_G^{'}$ is considering. If any overlap happened, it has to be fixed. To do that, two repair functions have been developed.

4.2.3.2. Before-aisle repair function

The idea behind this function is if there is any overlap between $f_G^{'}$ and aisle happens, the facility $f_G^{'}$ moves back exactly before the aisle. To illustrate, $f_G^{'}$ backs to the back of boundary of aisle which it passed over. **Figures 9** and **10** represent the overlap conditions in both cases of vertical and horizontal aisle.

The steps of the move operator with aisle constraints are explained as follows:

Step 1. Move facility f_G towards its closest facility. Calculate new coordinates of facility f_G and call it facility $f_G^{'}$.

Step 2. Check overlaps possibility between $f_G^{'}$ and aisle.

Step 3. If there is any overlap, take appropriate repair function.

Step 4. Find the coordinates of $f_G^{'}$- —details are shown in **Tables 5** and **6**.

Step 5. End

Repair function-horizontal aisle

- Facility f_G is lower side of the aisle is

$$\text{Rep} = \left(\left(y_G^{'} + W_G/2\right) - \left(Y_A - w_A/2\right)\right)/\text{Sin}\theta \tag{35}$$

- Facility f_G is upper side of the aisle:

$$\text{Rep} = \left(\left(Y_A + W_A/2\right) - \left(y_G^{'} - w_G/2\right)\right)/\text{Sin}\theta \tag{36}$$

Repair function-vertical aisle

- Facility f_G is in the left side of the aisle:

$$\text{Rep} = \left(\left(x_G^{'} + l_G/2\right) - \left(X_A - L_A/2\right)\right)/\text{Cos}\theta \tag{37}$$

- Facility f_G is in the right side of the aisle:

$$\text{Rep} = \left(\left(X_A + L_A/2\right) - \left(x_G^{'} - l_G/2\right)\right)/\text{Cos}\theta \tag{38}$$

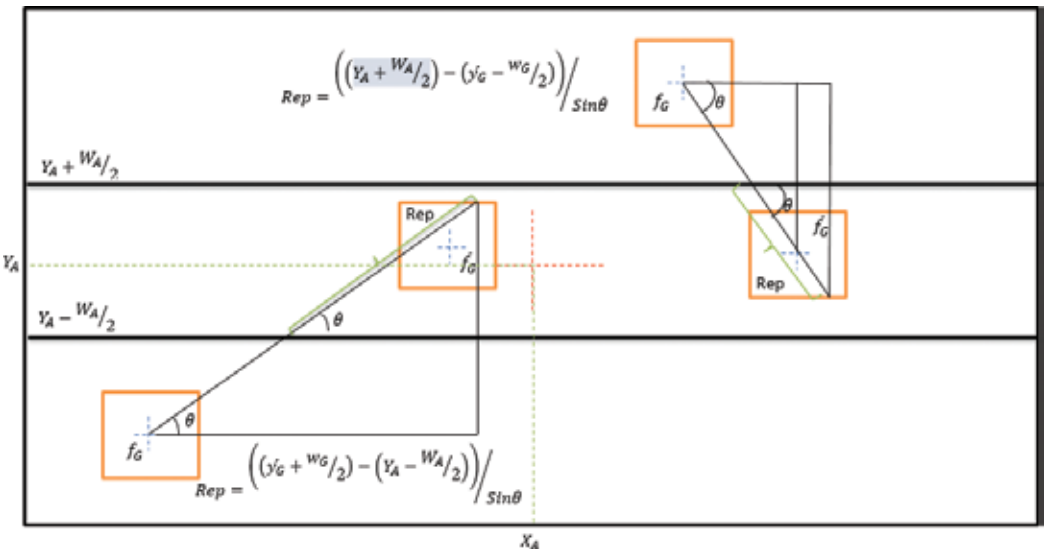

Figure 9. Before-aisle move operator for horizontal aisle.

http://dx.doi.org/10.5772/67313

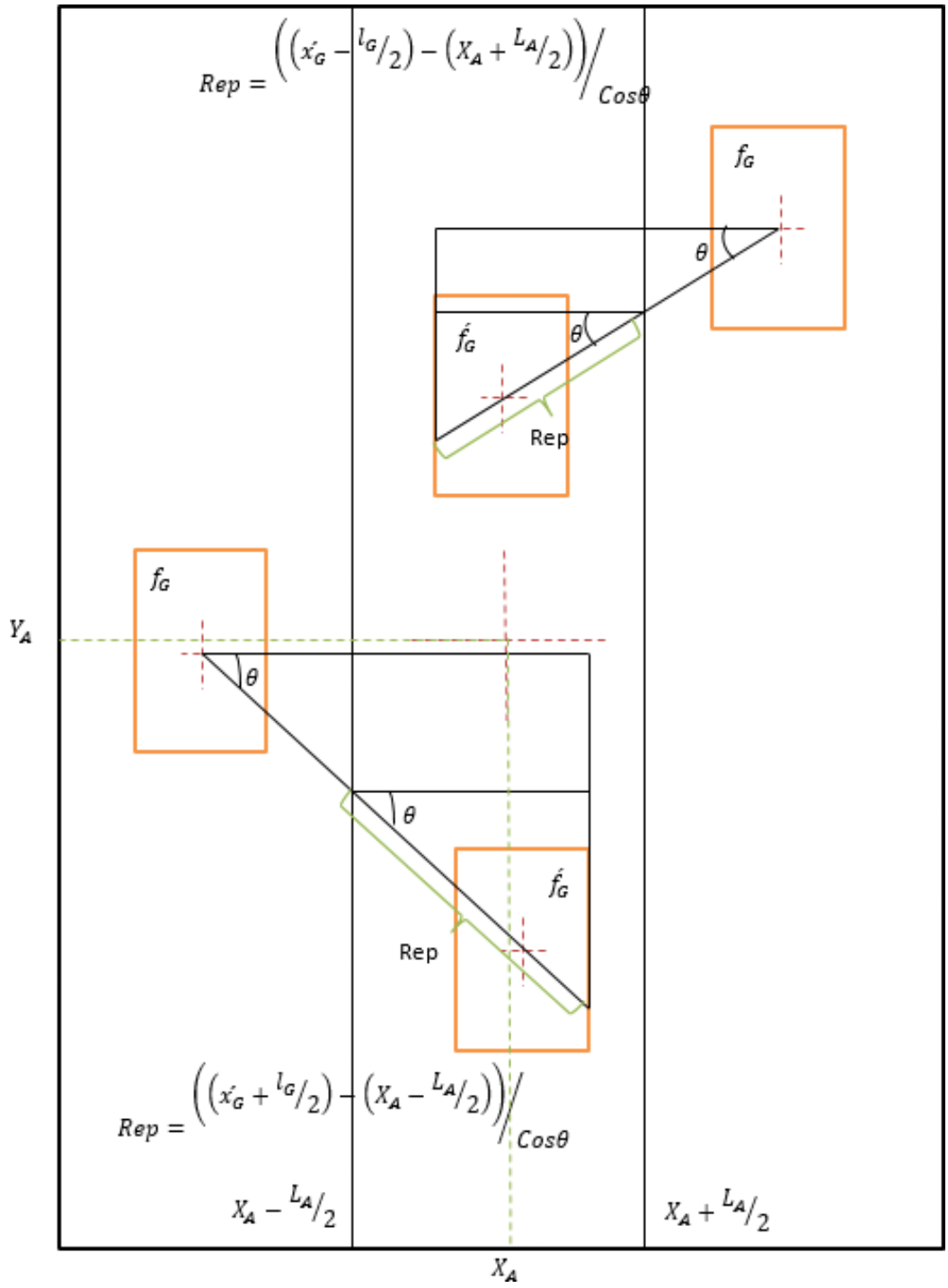

Figure 10. Before-aisle move operator for vertical aisle.

Horizontal Aisle	$x_{f_G} < x_{f_G}$	$x_{f_G} \geq x_{f_G}$
$y_{f_G} < Y_L$	$x_{f_G} = x_{f_G} - \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} - \text{Rep} \times \sin\theta$	$x_{f_G} = x_{f_G} + \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} - \text{Rep} \times \sin\theta$
$y_{f_G} > Y_L$	$x_{f_G} = x_{f_G} - \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} + \text{Rep} \times \sin\theta$	$x_{f_G} = x_{f_G} + \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} + \text{Rep} \times \sin\theta$

Table 5. Revised coordinate based on before-aisle repair function-horizontal aisle.

Vertical Aisle	$y_{f_G} < y_{f_G}$	$y_{f_G} \geq y_{f_G}$
$x_{f_G} < X_L$	$x_{f_G} = x_{f_G} - \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} - \text{Rep} \times \sin\theta$	$x_{f_G} = x_{f_G} - \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} + \text{Rep} \times \sin\theta$
$x_{f_G} > X_L$	$x_{f_G} = x_{f_G} + \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} - \text{Rep} \times \sin\theta$	$x_{f_G} = x_{f_G} + \text{Rep} \times \cos\theta$ $x_{f_G} = x_{f_G} + \text{Rep} \times \sin\theta$

Table 6. Revised coordinate based on before-aisle repair function-vertical aisle.

4.2.4. Developed SA algorithm

In this chapter, the parameters taken by Bazargan-Lari and Kaebernick [10] have been used in the developed SA algorithm:

1. Initial temperature: 10
2. Cooling rate: 0.9
3. Temperature reduction: $t_i = 10(0.9)^{i-1}$
4. Outer loop: 25
5. Inner loop: $100 \times M$, M is the total number of facilities

5. Case study

Carbide Tool Inc. manufactures and distributes metalworking tools. The company is dedicated to developing specialized carbide, polycrystalline diamond (PCD) and cubic boron nitride (CBN) inserts, as well as multi-task tooling for the aerospace, automotive and mould-die industries. The company currently has a process layout configuration. Five different kinds of family cutting insert tools are produced. Each part has specific monthly demand. There are different kinds of unequal sized grinding machine tools. Some of the machine tools have identical copies on the shop floor to increase productivity. Therefore, the demand is being

http://dx.doi.org/10.5772/67313

shared among the different copies of those machine tools. Different operations are performed on inserts with different sequences. The list of operations of each insert and the machine tools used for those operations are shown in **Table 7**.

The company's shop floor has a rectangular shape. There is no special material handling device for transforming unfinished products among machine tools. As demonstrated in **Table 7**, it is obvious that the number of operations performed on each part insert tool is large enough; hence, the amount of travel taking place every day on the production floor is quite significant. Additionally, as per their original layout, all the raw materials are transported from the back side of the shop to the front to start operation. This causes extra unnecessary travel, and hence extra material handling cost. The inspection and shipping stations which are two of the last steps as per the sequence of operations are not properly positioned in the current layout, because they are located in front of the floor. Since the current layout is process layout, similar machine tools are grouped together and located on one side of the floor. The original layout is

		Dimension		**Cutting insert tools**						
ID	**Machine**	**Length**	**Width**	**Dog bone**	**S Shape**	**Triangular**	**Top notch**	**Diamond type 1**	**Diamond type 2**	**Diamond type 3**
M1	Double disk (1)	12.67	5		O1	O2				
M2	Blanchard (2)	6	9.07			O1	O1	O1	O1	
M3	Wendt (3)	8.5 6.8	6.1 9.45	O1	O2	O4	O2	O2	O2	
M4	Polish (1)	6	5			O3				
M5	EWAG (1)	4.3	7.3					O7		O3
M6	Surface grinding (2)	7	6	O4	O5	O5	O3			
M7	Surface grinding (1)	6	7.54					O3	O3	
M8	Swing fixture (1)	8	6	O2	O3					
M9	V-bottom (1)	7	6	O3	O4					
M10	Wire cutting (2)	7.8 7.4	6.7 5.7					O4	O4	
M11	Laser M/C (1)	7.6	9.74						O6	
M12	Brazing (1)	4	1.8					O6	O5	O1
M13	ETCH (1)	3	4	O5	O5	O6	O4	O8	O7	O3
ST1	Inspection (1)	4	3	O6	O6	O7	O5	O9	O8	O4
ST2	Wash (1)	5	3	O7	O7	O8	O6	O10	O9	O5
ST3	Packing (1)	16	8	O8	O8	O9	O7	O11	O10	O6
		Part demand		1200	900	500	500	600	600	200

Table 7. Machine tool characterizations and parts' operations sequence.

causing too much traffic, since it is not taking into account the sequence of processing of parts. For an example, the primary operations of all insert tools are performed by the combination of three machine tools: Double Disk, Blanchard and Wendt. All Wendt machines are located in upper side of hall, while Blanchard and Double Disk machines are arranged in the lower side. Therefore, it could be concluded that there are too much back and forth travel being done between the two sides of the floor just for performing the first couple of operations.

After having several meetings with the plant manager and executive board of the company, cellular layout was chosen as the best layout plan. Group formation was performed by the plant manager. Four cells with specific types of machine tools were designed as given in **Table 8**. The problem was solved using both the developed mathematical model and heuristic [7].

Cell name	Machine tools/work station				
Primary	M1 (1)	M2 (2)	M4 (1)	M3 (3)	
Grinding	M6 (2)	M8 (2)	M9 (1)		
Diamond	M10 (2)	M7 (1)	M5 (1)	M12 (1)	M11 (1)
Final	M13 (1)	ST1 (1)	ST2 (1)	ST3 (1)	

Table 8. GF results.

5.1. Mathematical model

5.1.1. Intra-cellular layout

For the leader problem the layout of the different machine tools and work stations in their respective cells are being solved. The intra-cellular travel cost per unit distance per one unit of each respective part are ¢10, ¢10, ¢15, ¢12 and ¢20, respectively for Dog Bone, S Shape, Triangular, Top Notch and all types of Diamond. The results for intra-cellular layout are summarized in **Table 9**.

Cells	Dimension		Centroid		MHC (material handling cost)
	Length	Width	*X*	*Y*	
Primary	35	25	42.5	13.5	$1191.550
Grinding	26	20	74	50	$520.588
Diamond	30	20	45	59.22	$764.580
Final	30	20	75	8	$1056.350
Aisle	90	60	45	32.5	

Table 9. Intra-cell material handling costs and inter-cell dimensions of cells.

5.1.2. Inter-cellular layout

In the follower problem, the four cells with exact dimensions are located on the 90″ × 60″ shop floor. The inter-cellular travel cost per unit distance for each unit of Dog Bone, S shape,

http://dx.doi.org/10.5772/67313

Triangular, Top Notch, and Diamond are ¢12, ¢12, ¢18, ¢15, and ¢20, respectively. Material handling cost for the inter-cellular layout is $7520.42. **Table 11** shows the coordinates of cells based on inter-cellular layout plan. The final sketch of inter-cellular and intra-cellular layout is shown in **Figure 11**.

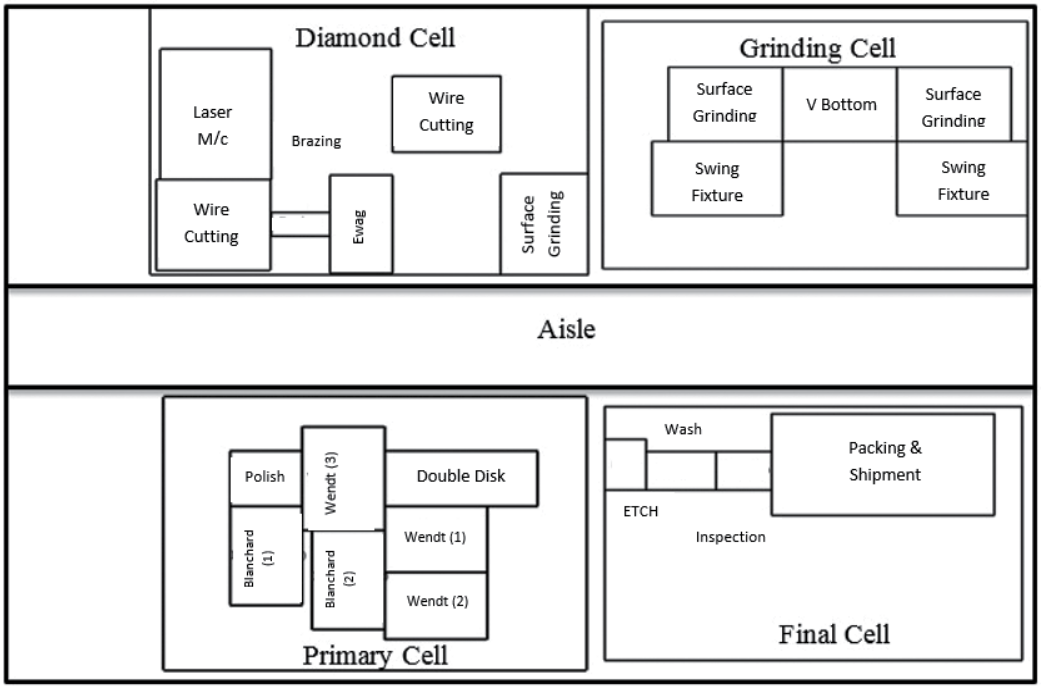

Figure 11. Inter-cell and intra-cell layout plan.

5.2. Heuristic

The heuristic is applied to solve the intra-cellular layout problems for each respective cell. The results obtained are provided in **Table 10** and plotted in **Figure 12**. The material handling cost for the inter-cellular layout is $6134.50 [6].

5.3. Simulated annealing (SA)

5.3.1. To validate the proof of the efficiency of the developed SA algorithm, the developed SA was applied for 10 runs for each cells [8]

5.4. Discussion

The comparison between the solutions provided non-linear, linear model and simulated annealing is represented in **Table 11**. The linear model gives the exact optimum solution, however simulated annealing provides near optimum solution. The results also prove this fact. In both leader and follower problem, i.e. intra- and inter-cell, respectively, the total

Cell	Machine	Coordinates	
		X	Y
Primary	Blanchard	14.5	18.30
	Blanchard	21.27	17.16
	Polish	14.5	5.98
	Wendt	6.58	7.64
	Wendt	23.34	4.45
	Double Disc	23.83	10
	Wendt	7.25	17.72
	MHC	$734.581	
Grinding	Surface grinding	16.79	4
	Surface grinding	21.05	16
	Swing fixture	18.97	10
	Swing fixture	5.72	15.26
	V-bottom	8.55	6.76
	MHC	$669.480	
Diamond	Wire cutting	10.39	3.8
	Wire cutting	5.50	10
	Surface grinding	18	5.64
	Brazing	26	10
	Ewag	11.53	16.31
	Laser M/c	18.8	14.87
	MHC	$808.640	
Final	ETCH	26.19	9
	Wash	15	12.42
	Inspection	4.89	9
	Packing	15	5
	MHC	$2410.760	

Table 10. Machine coordinates based on heuristic.

material handling cost is less than costs provided by non-linear mixed integer programming and simulated annealing.

The follower problem solved by simulated annealing has just assumed aisle.

Generally speaking, the linearized model obviously has yielded exact optimal results which proved to be better than those obtained by both the simulated annealing and the original non-

http://dx.doi.org/10.5772/67313

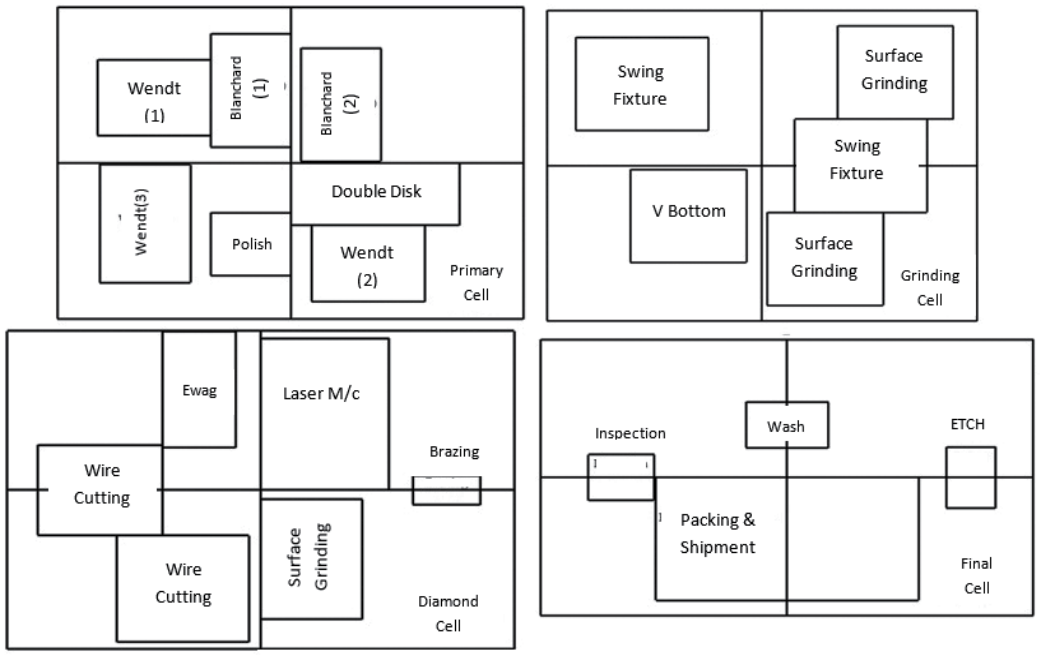

Figure 12. Heuristic results showing layout presented at intra-cell level for different cells (note: quadrant have been plotted demonstrating how facilities were spread around the different quadrants as per the working of the algorithm).

	Leader problem				**Follower problem**
Method	***Primary cell***	***Grinding cell***	***Diamond cell***	***Final cell***	***Shop***
NLMIP	$ 1191.550	$520.588	$764.580	$1056.350	$7520.420
LMIP	$503.024	$399.750	$360.800	$685.200	$2838.6
SA	$701.592	$526.004	$787.940	$856.508	$6167.6

Table 11. Comparisons between mathematical modelling and simulate annealing.

linear model. This was quite expected; in most cases simulated annealing resulted in better solutions than the non-linear model; however, there were cases where the non-linear model results were slightly better than those obtained by simulated annealing. The exception was for grinding cell and diamond cell where the non-linear model outperformed slightly than simulated annealing.

Table 12 summarizes the results from both leader and follower problems. Both mean and SDV from the performed 10 runs are being provided. Standard deviation is good except for inter-cell layout problem. For inter-cell, we believe the algorithm is yet to be improved, and variance as shown in **Table 12** is relatively high.

Cell	Average	SDV
Primary	$633.86	$11.19
Grinding	$492.44	$15.63
Diamond	$759.790	$22.315
Final	$902.62	$32.23
Inter-cell	$5474.61	$423.97

Table 12. Mean and standard deviation of SA solutions.

6. Conclusion

Cellular manufacturing system (CMS) layout has recently begun to receive heightened attention worldwide. The design of a CMS includes: (1) cell formation (CF), (2) group layout, (3) group and (4) resource allocation. An effective CMS implementation help any company improve machine utilization and quality; it also makes reduction in setup time, work-in-process inventory, material handling cost, part makespan and expediting costs.

There are two main approaches to FLP such as the discrete and continuous approaches. The discrete approach holds two main assumptions: one is all facilities are equal size and shape; the other one is predetermined locations of facilities. However, these kinds of assumptions are not realistic. The discrete approach is not suited to represent the exact locations of facilities. Moreover, this approach is not applicable for FLP with unequal size and shape facilities. The appropriate approach to this kind of FLP is continuous representation.

Generally speaking, the design of layout cannot be efficient if manufacturing attributes are not being considered in it. To illustrate, operation sequencing and parts' demand are the two factors that have significant impacts on the flow rate which minimizes the main objective of FLP. The majority of literature studies have not considered these factors in the design of layout plan. Besides those manufacturing attributes, the available area of the shop that can be used for locating facilities is the other factor that has to be considered.

The facility layout problem for cellular manufacturing system in both inter- and intracellular levels is considered in this chapter. The problem is to arrange facilities that are cells in the leader problem and machine tools in the follower problem in the continual planar site. Operation sequence and parts' demand are the two main manufacturing attributes considered in the developed model. The MIP has been presented for both leader and follower problems. The novel aisle constraints have been presented in the mathematical formulation. Since the model is non-linear, the linearized model has been developed. Additionally, a novel mathematical modelling has been developed for considering block constraints such as fixed departments and facilities. Since the FLP is an NP-hard problem, novel heuristics presented in this chapter.

A novel heuristic model developed for finding feasible initial solution for designed metaheuristic algorithm, simulated annealing. The initial solution is based on the radial movement.

In other words, the algorithm placed facilities along the specific radius with certain angle within site. The algorithm starts with dividing site into four equal-sized quadrants, start placing facilities into first quadrant to the fourth one. After placing any new facility, the overlap's possibility between facilities and between facility and site boundaries is being checked. The different repair functions have been designed for different cases.

The SA algorithm developed for both inter- and intra-cellular problem. The results of heuristic have used to initialize the developed SA algorithm. However, in order to have more efficient SA, the cell size used in heuristic algorithm is assumed two times of the original size of the cells. The two main operators used are move and swap operators. The move operator decreases the distance between facilities by moving the target facility towards the closest facility to it. Furthermore, the swap operator developed by defining the concept of the free zone.

Author details

Maral Zafar Allahyari[1] and Ahmed Azab[2]*

*Address all correspondence to: azab@uwindsor.ca

1 Industrial Engineering, University of Windsor, Windsor, Ontario, Canada

2 Production & Operations Management Research Lab, Faculty of Engineering, University of Windsor, Ontario, Canada

References

[1] Yaman R, Gethin DT, Clarke MJ. An effective sorting method for facility layout construction. International Journal of Production Research. 1993;**31**(2):413–427.

[2] Meller RD, Narayanan V, Pamela HV, Optimal facility layout design, 1999; **23**:117–127.

[3] Armour Gordon C, Buffa Elwood S. A heuristic algorithm and simulation approach to relative location of facilities. Management Science. 1963;**9**(2):294–309.

[4] Russell DM, Kai-Yin G. The facility layout problem: recent and emerging trends and perspectives. Journal of Manufacturing Systems. 1996;**15**(5):351–366.

[5] Kusiak A, Heragu SS. The facility layout problem. European Journal of Operational Research. 1987;**29**(3):229–251.

[6] Chiang W-C. Visual facility layout design system. International Journal of Production Research. 2001;**39**(9):1811–1836.

[7] Zafar A M, Ahmed A. Improved Bi-Level Mathematical Programming and Heuristics for the Cellular Manufacturing Facility Layout Problem. In: ASME 2015.

[8] Zafar AM. Bi-Level Mathematical Programming and Heuristics for the Cellular Manufacturing Facility Layout Problem. MASc Thesis, University of Windsor, 2014.

[9] Alfa Sule A, Mingyuan C, Sunderesh SH. Integrating the grouping and layout problems in Cellular manufacturing systems. Computers & Industrial Engineering. 1992;23(1):55–58.

[10] Kaebernick H, Bazargan-Lari M, Arndt G. An integrated approach to the design of cellular manufacturing. CIRP Annals—Manufacturing Technology. 1996;**45**(1):421–425.

[11] Bazargan-lari M, Kaebernick H. An approachto the machine layout problem in a cellular manufacturing environment. Production Planning & Control. 1997;**8**(1):41–55.

[12] Bazargan-Lari M. Layout designs in cellular manufacturing. European Journal of Operational Research. 1999;112(2):258–272.

[13] Bazargan-Lari M, Kaebernick H, Harraf A. Cell formation and layout designs in a cellular manufacturing environment a case study. International Journal of Production Research. 2000;**38**(7): 1689–1709.

[14] Imam M H, Mir M. Automated layout of facilities of unequal areas. Computers & Industrial Engineering. 1993;**24**(355–366):0360–8352.

[15] Mir M, Imam M H. A hybrid optimization approach for layout design of unequal-area facilities. Computers & Industrial Engineering. 2001;**39**(1–2):49–63.

[16] Wilhelm M R,Ward T L. Solving quadratic assignment problems by 'simulated annealing'. IIE Transactions. 1987;**19**(1):107–119.

[17] Baykasoğlu A, Gindy NNZ. A simulated annealing algorithm for dynamic layout problem. Computers & Operations Research. 2001;**28**(14):1403–1426.

[18] Rosenblatt M J. The dynamics of plant layout. Management Science. 1986;**32**(1):76–86.

[19] Conway D G, Venkataramanan MA. Genetic search and the dynamic facility layout problem. Computers & Operations Research. 1994;**21**(8):955–960.

[20] Balakrishnan J, Cheng C H. Genetic search and the dynamic layout problem. Computers & Operations Research. 2000;**27**(6):587–593.

[21] Tavakkoli-Moghaddam R., Aryanezhad M B, Safaei N, Azaron A. Solving a dynamic cell formation problem using metaheuristics. Applied Mathematics and Computation. 2005;**170**(2):761–780.

[22] Safaei N, Saidi-Mehrabad M, Jabal-Ameli M S. A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. European Journal of Operational Research. 2008;**185**(2):563–592.

[23] Golden, B L, Skiscim, CH. Using simulated annealing to solve routing and location problems. Naval, Research Logistics Qaurtely. 1986;**33** (2):261–279.

[24] Press W, Teukolsky S A, Vetterling W T, Flanner B P 2007. Numerical Recipes. The Art of Scientific Computing. WH Press. pp. 549–555.

http://dx.doi.org/10.5772/67313

[25] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing. Science. 1983;**220**:671–680.

[26] Kia R, Baboli A, Javadian N, Tavakkoli-Moghaddam R, Kazemi M, Khorrami J. Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routing, lot splitting and flexible reconfiguration by simulated annealing. Computers and operations research, 2012;**39**:2642–2658.

[27] Heragu S, Alfa AS, 1992. Experimental analysis of simulated annealing based algorithms for the layout problem, European Journal of Operational Research, **57**:190–202.

Chapter 7

Application of Simulated Annealing and Adaptive Simulated Annealing in Search for Efficient Optimal Solutions of a Groundwater Contamination related Problem

Mahsa Amirabdollahian and Bithin Datta

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66998

Abstract

Characterization of groundwater contamination sources is a complex inverse problem. This inverse problem becomes complicated, due to the nonlinear nature of the groundwater flow and transport processes and the associated natural uncertainties. The mathematical challenges arise due to the nonunique characteristics of this problem resulting from the nonunique response of the aquifer system to a set of stresses and the possibility of instead locating only local optimal solutions. The linked simulation-optimization model is an efficient approach to identifying groundwater contamination source characteristics. Efficiency and accuracy of the search for optimum solutions of a linked simulation-optimization depend on the utilized optimization algorithm. This limited study focuses on the application and efficiency of simulated annealing (SA) as the optimization algorithm for solving the source characterization problem. The advantages in using adaptive simulated algorithm (ASA) as an alternative are then evaluated. The possibility of identifying a local optimal solution rather than a global optimal solution when using SA implies failure to solve the source characterization inverse problem. The cost of such inaccurate characterization may be enormous when a remediation strategy is based on the model inferences. ASA is shown to provide a reliable and acceptable alternative for solving this challenging aquifer contamination problem.

Keywords: groundwater contamination, adaptive simulated annealing, source characterization, simulation, optimization

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

1. Introduction

One of the efficient methodologies for identifying groundwater contamination source characteristics is the linked simulation-optimization model [1]. Efficiency and accuracy of the optimal solutions for this type of inverse models, which are often complex, nonlinear, and large scale, depend on the efficiency and accuracy of the optimization algorithm. Simulated annealing (SA) and adaptive simulated annealing (ASA) are two efficient evolutionary optimization algorithms which have been recently applied for solving such a large-scale nonlinear and complex linked simulation-optimization models for optimal characterization of unknown contaminant sources in groundwater systems. This chapter discusses the application of these two optimization algorithms and their relative performances in identifying the characteristics of contamination sources in groundwater systems. Adaptive simulated annealing is a modified version of simulated annealing where the optimization parameters are tuned automatically [2]. The advantages of using the ASA algorithm are demonstrated for an illustrative groundwater contamination-related problem, and the relative efficiency and accuracy of these two optimization algorithms, SA and ASA, are compared. The adaptive algorithm, ASA, is shown to be computationally more efficient and more suitable for searching for a globally optimal solution for a complex nonlinear optimization model representing complex flow and contaminant transport process in a contaminated groundwater aquifer.

2. Background

Effective groundwater pollution management and remediation depend on identifying the unknown pollution source and reconstructing their release history [3, 4]. The optimal and accurate identification of the contaminant sources plays an important role in modeling of subsurface transport processes and in reducing the long-term contamination remediation cost. The source identification problem deals with the spatial and temporal variations of the location, activity duration, and the injection rate of the pollutant sources and is mostly inferred from the available sparse and sometimes erroneous concentration measurements at the site. Mainly, source identification includes a simulation problem, such as groundwater flow and pollutant transport models, used to estimate past phenomena or predict future scenarios.

A linked optimization simulation-based methodology is often the viable and efficient approach for source identification in a regional-scale aquifer. The unknown contamination source identification in a contaminated aquifer is generally a very complex, ill-posed, and nonunique problem [5]. The nonuniqueness can be caused by sparsity of field measurement data or due to the inefficiency of the optimization algorithm to reach a global optimal solution. Designed monitoring networks [6–9] can reduce the nonuniqueness related to data availability. However, the nonuniqueness related to the search for a single global optimal solution to the inverse problem depends on the efficiency of the optimization algorithm. Other approach for source identification consists of solving the differential equations backwards in time (inverse problem). The random walk particle method [10, 11], the quasi-reversibility technique [12], the minimum relative entropy method [13], the Bayesian theory and geostatistical techniques [14], and genetic algorithm [15, 16] are some examples of this approach.

http://dx.doi.org/10.5772/66998

A simulation-optimization methodology couples the forward time contaminant simulation model with optimization techniques. If an optimization problem is solvable and every minimization sequence converges to a unique answer, it is called stable [17]. This methodology avoids the problem of stability associated with formally solving the inverse problem, but the iterative nature of simulation models usually requires increased computational effort. Many techniques were proposed utilizing coupled simulation-optimization, and a few representative ones are: response matrix [18, 19], embedded optimization [3], and linked simulation and ptimization [3, 15, 20].

The two limitations of the response matrix approach are as follows: it is based on the premise that the superposition principle is approximately valid in terms of flow and contaminant transport in the aquifer, and the aquifer parameters must be known and the simulation model must be used to generate the response matrix prior to running the source identification model [3]. Mahar and Datta [3] showed that the embedding methods need large computer storage and computational time, for large aquifers. Gorelick and Evans [18] concluded that numerical difficulties are likely to arise for large-scale problems using the embedding technique.

To conduct unknown pollutant source characterization in large-scale aquifers and real areas, linked simulation-optimization methodology was proposed. In this methodology, the numerical models for simulation of the flow and transport process are internally linked to the optimization algorithm. Chadalavada and Datta [1] and Amirabdollahian and Datta [21] presented an overview of the pollution source identification simulation-optimization approaches.

The linked simulation-optimization model is an efficient and effective technique to characterize the contaminant sources by internal linkage between flow and contaminant transport simulation models and the selected optimization technique. This methodology can solve contaminant source problems in fairly large study areas.

Evolutionary optimization algorithms have made it possible to solve complex linked simulation-optimization models, which are difficult to solve, or difficult to even obtain feasible solutions, when utilizing classical optimization tools. Moreover, there is less limitation in mathematical definition of objective function and constraints compared to former optimization algorithms such as linear programming [22]. Finally, evolutionary algorithms can optimally identify relatively large number of decision variables [23], and utilization of the evolutionary optimization algorithms simplifies the linking process. Examples of the evolutionary optimization algorithms include: genetic algorithm (GA) [24], tabu search (TS) [25], simulated annealing (SA), adaptive simulated annealing (ASA) [26], and differential evolution algorithm [27]. Yeh [28] and Datta and Kourakos [22] presented an overview on various optimization methods coupled with simulation techniques utilized for groundwater quantity management, and quality management, respectively.

In a linked simulation-optimization approach, the optimization algorithm is used as an efficient search tool and the accuracy and efficiency of the methodology depend on the selected optimization algorithm. In groundwater contaminant source characterization problems, even when there are no errors or uncertainties associated with the inputs and parameter estimates for the physical process simulation model, there may not be a unique solution to the inverse problem due to nonunique physical response of the system. The ill-posed nature of the

inverse problem is a different issue and is predominantly due to sparsity of measurement data, which can be addressed by designing and implementing a suitable concentration monitoring network [29]. The ill-posed nature of the inverse problem and the plausibility of nonunique solutions can be interrelated as well. More efficient monitoring networks can reduce the plausibility of nonunique solutions, and therefore, optimal monitoring network design is a related issue [9, 22]. Also, only if the global optimum solution is found, it may represent the accurate solution to the source identification problem. Nonuniqueness in the system response may introduce alternate optimal solutions, although each globally optimal [5, 20, 21]. This is due to the fact that different sets of stresses (i.e., contaminant sources) can result in identical responses (resulting spatial and temporal concentrations). Therefore, nonuniqueness is inherent, independent of errors in parameter estimates or measurements. In addition, if the optimal solution is not the global solution, it itself introduces nonuniqueness due to local optimal solutions being wrongly identified as the optimal solution of the inverse problem. In the optimal contaminant source identification process, this global optimum generally represents the actual contaminant source characteristics. Failure to identify the global optimal and the plausible local optimal solutions introduces nonuniqueness in the solution space. Therefore, efficiency of the optimization algorithm to reach a global optimum solution, or near optimal solution, is crucial to accurate source identification. Efficiency of algorithms like ASA can be extremely useful especially when compared to SA which needs tuning of optimization parameters, hence rendering the search for a global optimal somewhat subjective or more uncertain.

In this chapter, two evolutionary optimization algorithms are described: SA and ASA. Simulated annealing (SA) approaches the optimization model like a bouncing ball, which bounces over mountains from valley to valley. The SA controlling parameter is temperature (T) which mimics the effect of fast moving particle in a hot object like hot molten metal; as the T decreases and reaches relatively colder states, the height of the ball bounce decreases and it settles gradually in the deepest valley. To reach the optimal solution, there are many parameters which need to be tuned, such as probability density function, acceptance probability density function, and re-annealing temperature schedule. ASA is a variant of SA in which the automated re-annealing temperature schedule and random step selection make the algorithm less sensitive to the user-defined parameters. One of the issues in selecting a suitable and efficient optimization algorithm for solution of an optimization model is the likelihood of reaching a global optimum solution. It has been shown that SA is relatively more efficient in reaching a better optimal solution compared to GA [2]. The added advantage of using ASA is the elimination of the requirement to choose all the relevant optimization parameters appropriately, a process very much dependent on the structure and nature of the optimization model to be solved. ASA also eliminated the need for several trial executions of the model, to adjust the parameters [23]. Therefore, the possibility of reaching a global optimal solution faster is also enhanced by utilizing ASA. Very fast simulated re-annealing (VFSR) developed in 1987 is the first version of ASA [30]. Ingber and Rosen [31] showed that VFSR is about one order of magnitude faster than GA in convergence speed and is more likely to find the global optimum.

http://dx.doi.org/10.5772/66998

This study investigates the applicability of ASA to unknown groundwater contaminant source release history reconstruction problem and compares its performance to SA-based solutions. The performance evaluation of linked simulation-optimization approach is based on a realistic scenario where contaminant concentration measurements are available a few years after the sources have ceased to exist. Apart from the convergence speeds, the two algorithms are compared for their performance in recovering accurate source release histories in terms of source location, magnitude, and duration of activity.

3. Methodology

The pollutant source characteristics which are required to be addressed in an identification procedure are: (1) source release history (time); (2) source locations; and (3) source flux magnitudes. The linked simulation-optimization model consists of an optimization algorithm which specifies the candidate source characteristics. A simulation model which is linked to the optimization algorithm uses the candidate characteristics to simulate the contaminant concentration at monitoring locations at various time intervals. The optimization algorithm is used to minimize the objective function representing the differences between measured concentrations and simulated ones. This process evolves until the algorithm reaches the optimal solution or a specified stopping criterion. This methodology for identification of unknown groundwater contamination sources has two major components: numerical groundwater flow and transport simulation models and linked simulation-optimization model.

3.1. Groundwater flow and transport simulation models

Groundwater flow simulation model used in this study is MODFLOW-2000 [32]. MODFLOW is a computer program that numerically solves the three-dimensional ground water flow equation for a porous medium by using a finite-difference method. The partial differential equation for transient groundwater flow utilized in MODFLOW is given by the following equation [33]:

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right)+W=S_s\frac{\partial h}{\partial t} \quad (1)$$

where K_{xx}, K_{yy} and K_{zz} are the hydraulic conductivities (L/T) along the x, y, and z coordinate axes which are assumed to be parallel to the principal axes of hydraulic conductivity, respectively. H, SS, and t are the potentiometric head (L), the specific storage of the porous material (L^{-1}), and time, respectively. W is the volumetric flux per unit volume representing sources and/or sinks of water, where $W<0$ for flow moving out of the groundwater system, and $W>0$ for flow moving in (T^{-1}). When combined with boundary and initial conditions, Eq. (1) describes transient three-dimensional groundwater flow in a heterogeneous and anisotropic medium, assuming that the principal axes of hydraulic conductivity are aligned with the coordinate directions.

The contaminant transport simulation model which is used in this study is chosen as MT3DMS [34]. The partial differential equation describing three-dimensional transport of contaminants in groundwater can be written as follows [35]:

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x_j}\left(D_{j,k}\frac{\partial C}{\partial x_k}\right) - \frac{\partial}{\partial x_j}(u_j C) + \sum_{i=1}^{N}\frac{q_i}{\theta} + \sum_{p=1}^{NR} R_p \quad (2)$$

where C is the solute concentration in groundwater (ML^{-3}); t is the time (T); j and k are the Cartesian coordinates along axes; u_j is the groundwater velocity (LT^{-1}); $D_{j,k}$ is the dispersion coefficient tensor (L^2T^{-1}); q_i is the flux of contaminant concentration for source i (MT^{-1}); and $\sum_{p=1}^{NR} R_p$ are chemical reaction terms ($ML^{-3}T^{-1}$).

The MT3DMS transport model uses a mixed Eulerian-Lagrangian approach to the solution of the three-dimensional advective-dispersive-reactive equation [34]. The groundwater velocity values (u_j), estimated by the flow model, are used by the transport model to estimate concentration values. The estimated concentrations (C) are transferred to the optimization model (in the linked simulation-optimization approach) to evaluate the objective function value.

3.2. Linked simulation-optimization method

For completeness, a brief description of the formulation for the linked simulation-optimization source identification framework is presented. More details can be found in Mahar and Datta [3]. Based on the available background information about the site, the set of potential contaminant source locations is assumed to be known. The optimization model estimates the optimal contaminant fluxes associated with each potential source location at each stress period. The objective function minimizes the weighted sum of normalized differences between temporal and spatial observed and simulated concentrations subject to upper and lower bounds on source fluxes. The optimal source identification model is defined by the objective function 3, subject to the constraints 4 and 5.

$$Min\ F = \sum_{k=1}^{nk}\sum_{iob=1}^{nob} (Ces\ t_{iob}^{k} - Cob\ s_{iob}^{k})^2\ /(Cob\ s_{iob}^{k} + \alpha)^2 \quad (3)$$

Subject to:

$$Cest = f(D, HC, \theta, x_i, y_i, z_i, q_i) \quad i = 1, ..., N \quad (4)$$

$$0 \le q_i \le q_{\max} \quad i = 1, ..., N \quad (5)$$

where nk, nob, and N are the total number of concentration observation time periods, available monitoring locations, and candidate source locations, respectively. $Ces\ t_{iob}^{k}$ and $Cob\ s_{iob}^{k}$ are the concentration estimated by the simulation model and observed concentration at observation location iob and at the end of time period k, respectively. D, HC, and θ are the dispersion coefficient, hydraulic conductivity, and porosity, respectively. x_i, y_i, z_i, and q_i are the Cartesian coordinates of candidate contaminant source i and the contaminant release flux for candidate location i, respectively. $q_{\max}$ is the upper bound for contaminant release fluxes.

The constraint set 4 represents the flow and contaminant transport simulation models, and it couples the simulation model and optimization algorithm. Eq. (5) limits the candidate contaminant flux values, at each potential location, to an upper bound. In Eq. (3), α is a constant,

http://dx.doi.org/10.5772/66998

which is sufficiently large to prevent any individual term in Eq. (3) becoming indeterminate due to the observed value of concentration becoming very small. Adding this parameter variable also prevents domination of the obtained solution by deviation between measured and simulated concentrations corresponding to low concentration measurement values [3]. **Figure 1** shows a schematic representation of the linked simulation-optimization source identification process using evolutionary optimization algorithms.

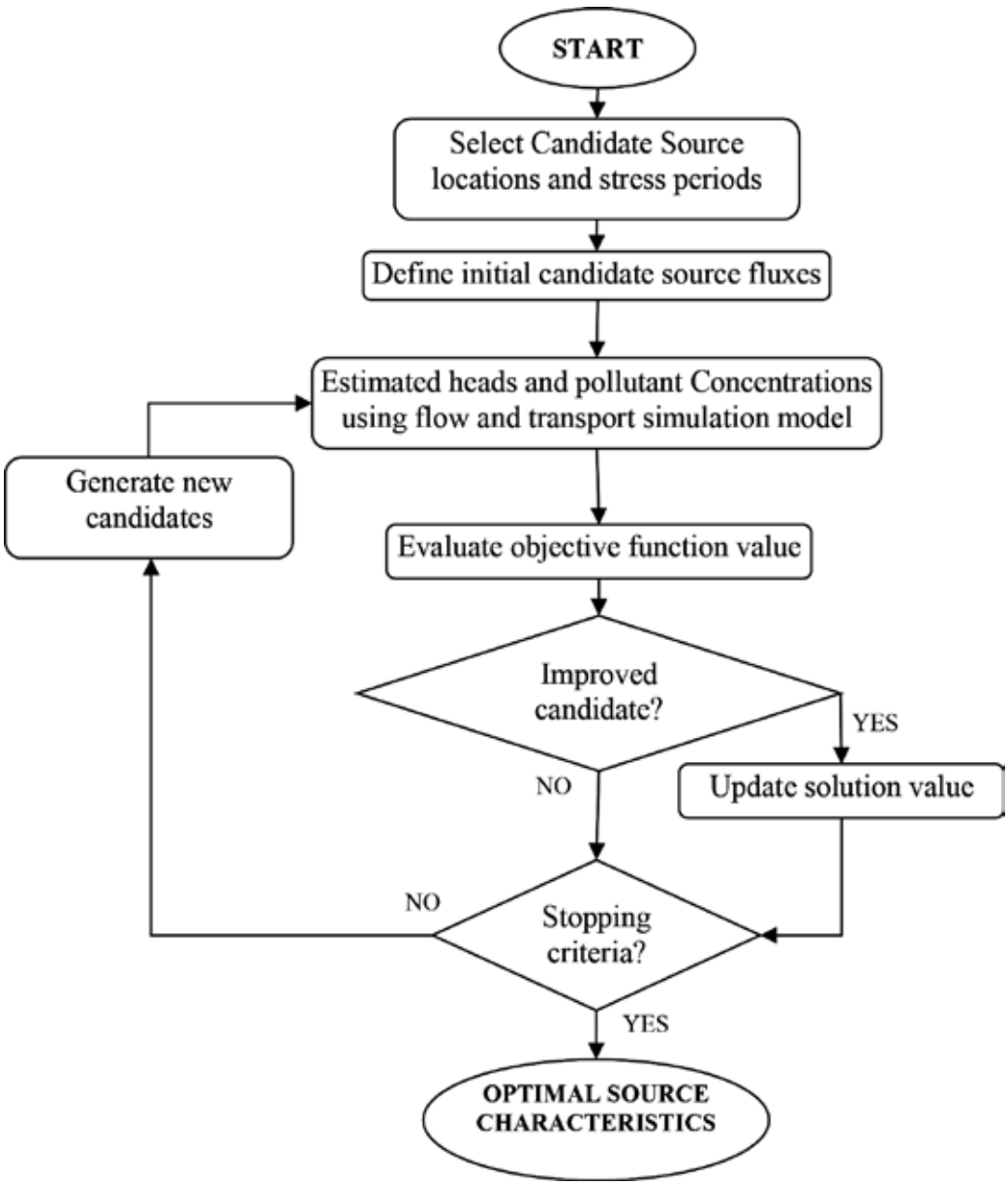

Figure 1. Schematic diagram of linked simulation-optimization contaminant source identification methodology.

4. Performance evaluation

A three-dimensional transient illustrative groundwater contamination problem is utilized to compare the efficiency and accuracy of SA and ASA optimization algorithm in the context of this research. First, the illustrative groundwater system is defined. Arbitrary monitoring locations are selected, and flow and transport simulation models are used to estimate contaminant concentrations at monitoring locations and times. Specifically for the performance evaluation purpose, these simulated concentration measurement values are to be used as observed concentration in the linked simulation-optimization source identification model. Using this performance evaluation procedure with synthetic data for an illustrative example facilitates the comparison between the application of SA and ASA optimization algorithms, without the need to consider the reliability of model properties, measurement accuracies, and parameters estimation errors.

4.1. Illustrative groundwater contamination problem

The performance of the proposed methodology is evaluated for a three-dimensional illustrative groundwater aquifer study area. **Figure 2** shows the plan view of the three-dimensional study area measuring 1500 m × 1000 m × 36 m and consisting of two unconfined layers. The top, bottom, and left side boundaries have a specified head, and the right-hand

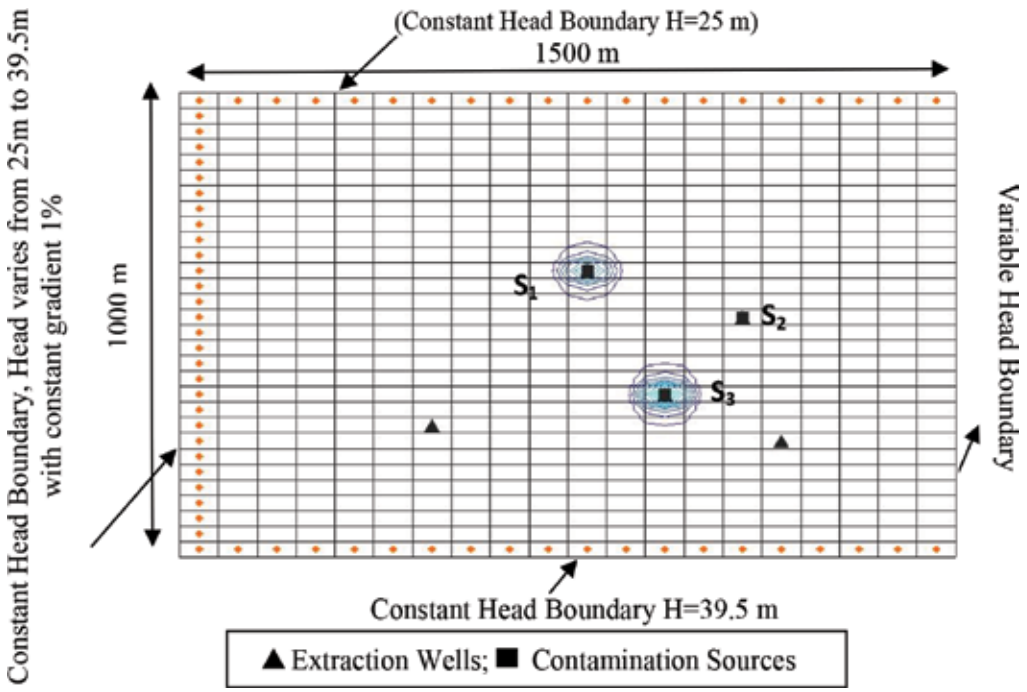

Figure 2. Plan view of the study area with location of contaminant sources.

http://dx.doi.org/10.5772/66998

side boundary has variable head boundary conditions. The triangular signs show the location of active extraction wells (sinks). The candidate contaminant source locations are shown by square signs. Two of them are actual sources and one is a dummy. The dummy (not actual) source is introduced to evaluate the accuracy of the proposed methodology in correctly identifying actual sources. Twelve concentration monitoring locations are specified. It is assumed that the contaminant source fluxes are the same in every stress period. The study period is divided into five stress periods. **Table 1** shows the length of stress periods and the extraction wells and contaminant sources' properties. The field hydrogeological parameters are given in **Table 2**.

	Location			Stress period				
	Row	**Column**	**Layer**	**1** **183 days**	**2** **183 days**	**3** **183 days**	**4** **183 days**	**5** **2196 days**
Contamination source	12	11	1	60	70	20	30	10
	15	15	1			Dummy source		
Flux (g/s)	20	13	1	30	50	70	80	0
Extraction well	22	7	1			100		
Flow rate (L/day)	23	16	1			500		

Table 1. Characteristics of the contamination sources and extraction wells.

Parameter	**Unit**	**Value**
Number of cells in x-direction	–	20
Number of cells in y-direction	–	30
Number of cells in z-direction	m	2
Horizontal hydraulic conductivity (1st and 2nd layers)	m/d	25, 18
Vertical hydraulic conductivity	m/d	3
Specific storage	1/m	0.2
Porosity	–	0.25
Longitudinal dispersivity	m	20
Horizontal transverse dispersivity	m	2
Vertical transverse dispersivity	m	1
Initial contaminant concentration	ppm	0
Diffusion coefficient	–	0
Upper and lower bounds for source fluxes	g/s	0–100

Table 2. Hydrogeologic parameters for the study area.

4.2. Contaminant source identification process

For the purpose of identifying contaminant source characteristics, all sources are considered to be active during all five stress periods, and the pollutant flux from each of the sources is assumed to be constant over a specified stress period. In order to evaluate the model performance, one dummy (not actual) source is also introduced as a potential source. Therefore, the source identification decision variables are the contaminant fluxes at three potential source locations for each stress period, and in total, there are 15 decision variables to be identified.

Since the performance is evaluated using illustrative problem, flow and transport simulation models are utilized to find contaminant concentration at monitoring locations using the actual source fluxes. These values are used in the linked simulation-optimization process as observed concentrations. The initial source fluxes are set to 0 for all sources and stress periods. In real-life contaminant source identification problems, the observed concentrations collected in the field will be used to find optimal source characteristics.

5. Results and discussion

The applicability of these two algorithms is compared in terms of efficiency and accuracy. The run time and number of generations are utilized to compare the efficiency of the algorithms. Moreover, the estimated source characteristics are compared with the actual properties in order to compare algorithms in terms of accuracy. To examine the capability of both models in terms of accuracy in estimating source fluxes, the normalized absolute error of estimation (NAEE%) is estimated using Eq. (6) [2]:

$$\text{NAEE}(\%) = \frac{\sum_{i=1}^{N} \left| q_{est}^{i} - q_{act}^{i} \right|}{\sum_{i=1}^{N} q_{act}^{i}} \times 100 \tag{6}$$

where N is the number of stress periods and q_{est}^{i} and q_{act}^{i} are the estimated and actual source fluxes for stress period i, respectively.

Table 3 presents the SA and ASA optimization parameters. Every iteration of SA- and ASA-based methods uses one run of the groundwater transport simulation model (MT3DMS). Irrespective of the optimization algorithm, the execution time for one transport simulation run depends on the computation platform. In order to have a comparison between methods,

Parameter	Unit	Value
Error tolerance for termination	–	0.01
Objective function multiplier	–	100
Lower bound for source fluxes	g/s	0
Lower bound for source fluxes	g/s	100
Initial source fluxes	g/s	0

Table 3. Optimization model parameters.

http://dx.doi.org/10.5772/66998

independent of the utilized computation platforms, both methods are compared based on the number of simulation runs which is directly proportional to the computational time.

In real-life groundwater contaminant source identification problems, the source characteristics are unknown. Therefore, there is no information available to measure the accuracy of linked simulation-optimization methods. The accuracy and efficiency of SA depend to a large extent on the selected SA optimization parameters. However, due to unknown source characteristics, sensitivity analysis and tuning SA parameters are not possible or very difficult. To compare SA- and ASA-based methods, SA with initial temperature (T) 1.0E8 and temperature reduction factor (TR) 0.5 is selected. **Figures 3** and **4** compare the estimated against actual fluxes for sources 1 and 3, respectively. NAEE% of the estimated fluxes using ASA and SA models (T = 1.0E8, RT = 0.5) is 22.5 and 75%, respectively. Both methods identified the dummy source (not an actual source but introduced as a potential source for evaluation purpose). As shown in **Figures 3** and **4**, this particular set of SA parameters represents one particular SA solution highlighting the comparative inefficiency of SA.

Figure 5 shows the convergence profiles for both SA- (T = 1.0E8, RT = 0.5) and ASA-based models. Minimum value of the objective function achieved is plotted against number of the transport process numerical simulation models. **Figure 5** shows that ASA converges faster to the smaller objective function values (in the minimization problem), compared to the utilized SA model. Although, as **Figure 5** shows, the SA-based model converges to very small objective function values, the corresponding estimated NAEE% (75%) is large. This shows that SA-based solutions seem to get trapped in a local optimum and did not find or get close to the global optimum. This may be due to the nonunique nature of the local optimal solutions

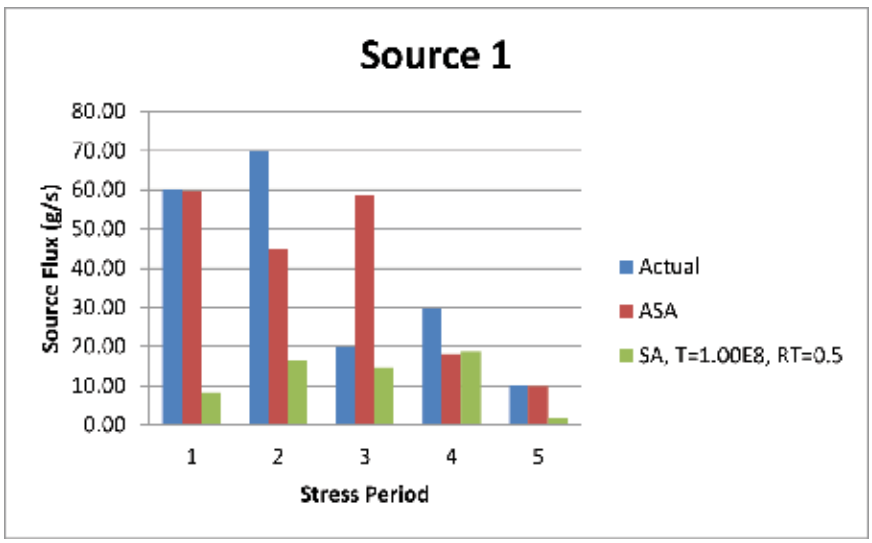

Figure 3. Source 1 release fluxes.

as well, that is, the obtained solution matches the observed and simulated concentrations for a different set of sources not representing actual sources. The objective function is very small,

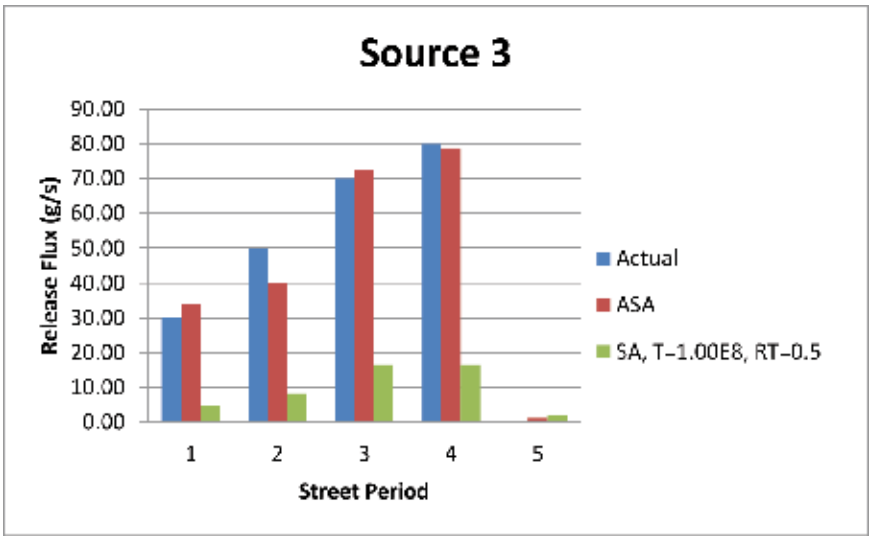

Figure 4. Source 3 release fluxes.

even though the accuracy of source estimates is very poor. **Figure 5** shows that the objective function improvement rate decreases after 10,000 simulation runs. Next, sensitivity analysis is conducted to find suitable SA parameters. A set of 10,000 simulation runs is selected as maximum number of simulation runs.

For the performance evaluation purposes, a sensitivity analysis is performed to find suitable SA parameters. This sensitivity analysis, so-called artificial, is not possible in real-life

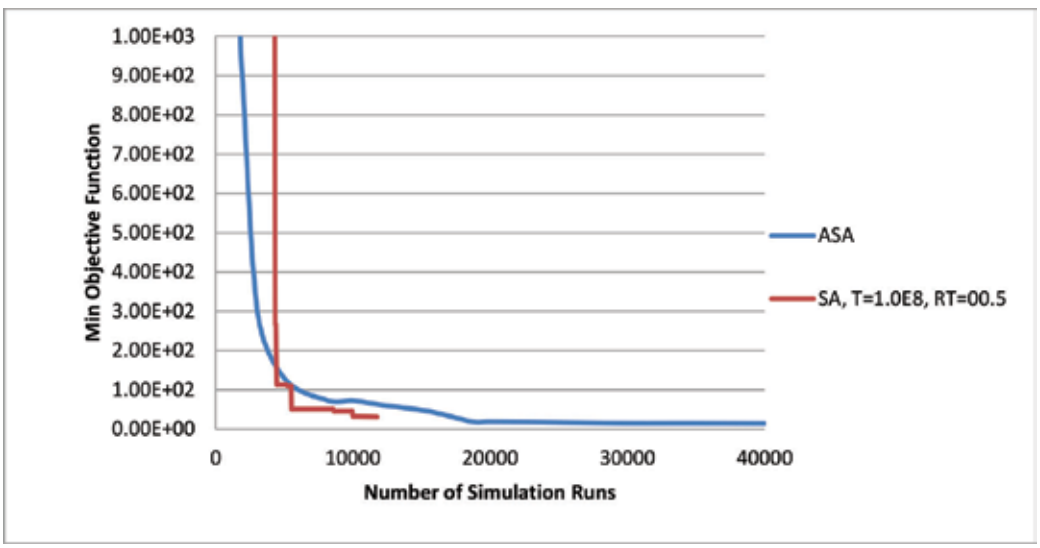

Figure 5. SA- (T = 1.0E8, RT = 0.5) and ASA-based model convergence profiles.

http://dx.doi.org/10.5772/66998

contamination problems. In this chapter, illustrative example study area is selected with synthetic aquifer data. Therefore, SA parameters are tuned by comparing estimated and actual release fluxes. This step is not possible in real-life scenarios where the source release fluxes are unknown. These evaluation results only show that SA can deliver solution results comparable to the ASA solutions if the SA parameters could be optimally tuned based on sequential matching of desired and obtained solutions, an impractical scenario.

Two parameters, initial temperature and temperature reduction factor, in order to find the sensitivity of SA models to the optimization parameters. Since the objective of this chapter is to compare the performance of SA- and ASA-based models, an initial execution of ASA is used to find desirable number of simulation runs. Using this initial model execution, 10,000 simulation runs are selected. **Figure 6** shows the resulted NAEE% using different SA parameters with the same number of maximum simulation runs (10,000). The least NAEE% is associated with 1000 as initial temperature (T) and 0.3 as the temperature reduction factor (TR).

In order to compare the accuracy and convergence of the tune SA- and ASA-based models, both models are executed with unconstrained time of run. Error tolerance of estimation is set as 0.01 for both methods. NAEE% of the estimated fluxes using ASA and tuned SA (T = 1.0E3, RT = 0.3) models is 22.5 and 16.5%, respectively.

It can be inferred from the results that both methods are able to correctly identify the dummy source. A zero or near zero estimation of the dummy source implies correct identification. Solution results obtained using tuned SA parameters recovered source 1 release fluxes more accurately compared to the other SA-based solutions. Solutions obtained utilizing the ASA algo-

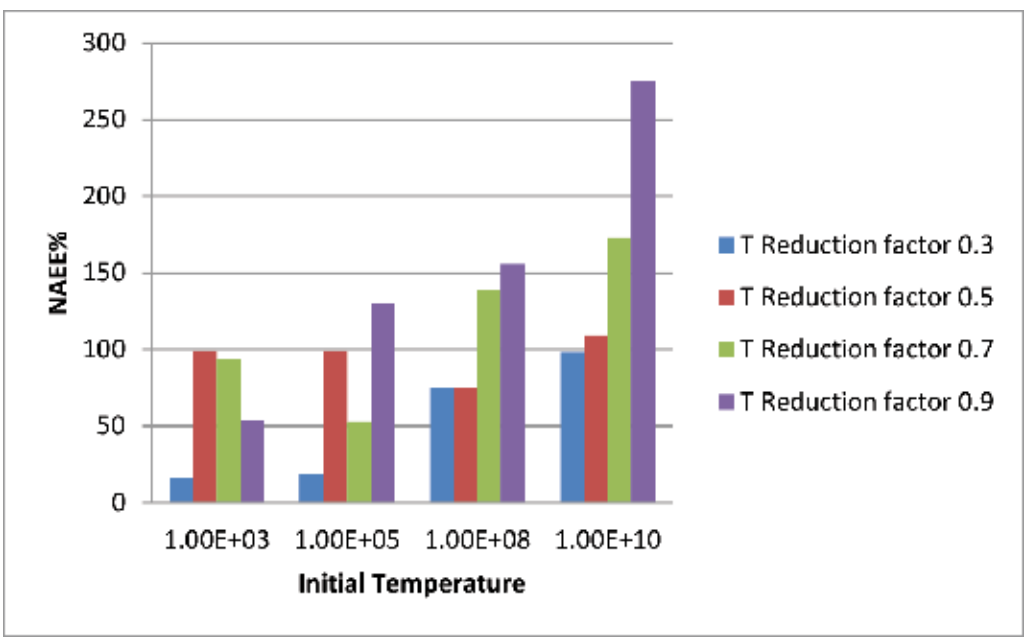

Figure 6. SA parameters sensitivity analysis.

rithm recovered source 3 release fluxes more accurately. Apparently, both methods resulted in relatively similar accuracy in recovering source release fluxes in terms of location and magnitudes. The tuned SA method solution results are slightly superior to ASA method considering accuracy, although the errors are similar in magnitude. As shown in **Figure 6**, all other sets of SA parameters (the ones tested in sensitivity analysis process) resulted in higher NAEE% compared with ASA method (25%). This shows that the performance of the SA-based method is highly reliant on the selection of its parameters which limits its applicability in real-life scenarios.

It can be argued that the poor performance of various SA-based models reported in **Figure 6** may have resulted from limited number of simulation runs. The purpose of this chapter is to compare ASA- and SA-based models based on both accuracy and convergence speed. Therefore, improved accuracy with the cost of relatively larger execution times is beyond the scope of this chapter. However, to give an insight to the readers, the SA model with T = 1.0E10 and RT = 0.9 is tested. The NAEE% associated with the estimated fluxes is 0.65 which was achieved after 103,876 simulation runs. **Figure 7** shows the T values and corresponding minimum objective function values.

Therefore, longer execution times can improve the results using non-tuned SA-based model. **Figure 7** shows that faster convergence happens when T reaches cooler states. This potentially demonstrates that very high T values would not have substantial positive effect on finding the optimal values. More rigorous studies are required to make a definitive conclusion about the SA parameters optimization without any time (computational costs) constraints.

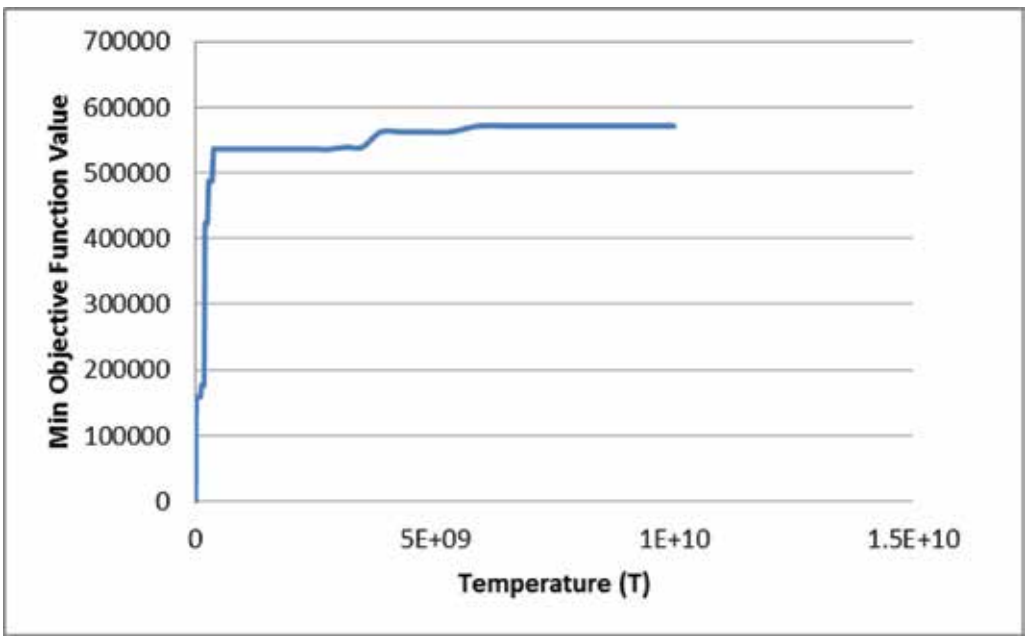

Figure 7. SA-based model convergence profile, T = 1.0E10, RT = 0.9.

http://dx.doi.org/10.5772/66998

6. Conclusion

Characterization of unknown sources of groundwater pollution, especially when field concentration measurement data are sparse and arbitrary, remains a challenging problem. The linked simulation-optimization method to solve the inverse problem of unknown groundwater contamination source characterization problem is utilized. The performance of two evolutionary optimization algorithms, SA and ASA, in terms of accuracy and convergence speed is evaluated. It is applied to an illustrative contaminated aquifer study area. Evaluation results show that suitable SA parameters need to be selected based on the nature of the problem to be optimized. Performance evaluation shows that the ASA-based method estimates the source release fluxes more accurately and convergences to a smaller objective function value (in a minimization problem) faster than non-tuned SA-based method.

In this limited study, an illustrative study area was selected with synthetic hydrogeology and contamination data specified for evaluation of the solution results. Therefore, just for comparison and performance evaluation purpose, SA parameters are tuned by comparing estimated and actual release fluxes. This practice is not possible in real-life scenarios where the source release fluxes are unknown. Without synthetic data, and simulation results to represent field concentration measurements, such tuning with prior knowledge will be impossible. Therefore, in real-life scenarios, the SA performance cannot be controlled by tuned SA parameters. This limits the application of SA-based models. The non-tuned SA model converged to poor results even with unconstrained computational time. Results demonstrated that non-tuned SA might not converge to near optimum results, even with a large number of iterations, and it may be trapped in the vicinity of local optimal solutions. This is due to nonunique nature of the groundwater contamination source characterization problems. Since tuning SA parameters in real-life scenarios is hard or impossible, utilizing SA may lead to wrong source flux estimation. The wrong estimation of source fluxes is not verifiable in real life and may lead to wrong decisions about management and remediation of the contaminated area.

The solution results obtained by an SA-based model with tuned parameters and solutions obtained by ASA-based model show that they have relatively similar performance concerning both accuracy and convergence speed. Moreover, the need to tune SA parameters will substantially limit its application in groundwater source identification problems. The tuning trial and error process increases the total computational costs of the linked simulation-optimization process. Therefore, the application of ASA in linked simulation-optimization-based groundwater source identification models results in substantial savings in computational time and potentially results in more accurate results. This inference is critical for designing, effective, and efficient contaminated aquifer remediation strategies which are often very costly, and cost of failed remediation strategies, caused by inaccurate characterization of unknown contamination sources, can be enormous. ASA is shown to provide a reliable and an acceptable alternative for solving this challenging aquifer contamination problem.

Author details

Mahsa Amirabdollahian[1]* and Bithin Datta[2]

*Address all correspondence to: mahsa.amirabdollahian@my.jcu.edu.au

1 James Cook University, Townsville, QLD, Australia

2 James Cook University and CRC-CARE, Callaghan, NSW, Australia

References

[1] Chadalavada S, Datta B, Naidu R. Optimisation approach for pollution source identification in groundwater: an overview. International Journal of Environment and Waste Management, 2011; **8**(1–2): 40–61. doi:10.1504/IJEWM.2011.040964

[2] Jha M, Datta B. Three-dimensional groundwater contamination source identification using adaptive simulated annealing. Journal of Hydrologic Engineering, 2013; **18**(3): 307–317. doi:10.1061/(ASCE)HE.1943-5584.0000624

[3] Mahar P S, Datta B. Optimal identification of ground-water pollution sources and parameter estimation. Water Resources Planning and Management-ASCE, 2001; **127**(1): 20–29.

[4] Datta B, et al. Efficient identification of unknown groundwater pollution sources using linked simulation-optimization incorporating monitoring location impact factor and frequency factor. Water Resources Management, 2013; **27**(14): 4959–4976. doi:10.1007/s11269-013-0451-8

[5] Datta B. Discussion of "Identification of Contaminant Source Location and Release History in Aquifers" by Mustafa M. Aral, Jiabao Guan, and Morris L. Maslia. Journal of Hydrologic Engineering, 2002; **7**(5): 399–400. doi:10.1061/(ASCE)1084-0699(2002)7:5(399)

[6] Dhar A, Datta B. Multiobjective design of dynamic monitoring networks for detection of groundwater pollution. Water Resources Planning and Management, 2007; **133**(4): 329–338. doi:10.1061/(ASCE)0733-9496(2007)

[7] Jha M, Datta B. Application of dedicated monitoring–network design for unknown pollutant-source identification based on dynamic time warping. Journal of Water Resources Planning and Management, 2015; **141**(11): 4015022. doi:10.1061/(ASCE)WR.1943-5452.0000513

[8] Jha M, Datta B. Application of unknown groundwater pollution source release history estimation methodology to distributed sources incorporating surface-groundwater interactions. Environmental Forensics, 2015; **16**(2): 143. doi:10.1080/15275922.2015.1023385

[9] Prakash O, Datta B. Optimal characterization of pollutant sources in contaminated aquifers by integrating sequential-monitoring-network design and source identification: methodology and an application in Australia. Hydrogeology Journal, 2015; **23**: 1089–1107. doi:10.1007/s10040-015-1292-8

[10] Bagtzoglou A C, Tompson A F B, Dougherty D E. Probabilistic simulation for reliable solute source identification in heterogeneous porous media. Water Resources Engineering Risk Assessment, 1991; **29**(NATO ASI Series): 189–201. doi:10.1007/978-3-642-76971-9_12

[11] Bagtzoglou A C, Tompson A F B, Dougherty D E. Application of particle methods to reliable identification of groundwater pollution sources. Water Resources Management, 1992; **6**: 15–23. doi:10.1007/BF00872184

[12] Skaggs T H, Kabala Z J. Recovering the release history of a groundwater contaminant plume—method of quasi-reversibility. Water Resources Research, 1995; **31**(11): 2669–2673. doi:10.1029/95WR02383

[13] Woodbury A D, Ulrych T J. Minimum relative entropy inversion: theory and application to recovering the release history of a groundwater contaminant. Water Resources Research, 1996; **32**(9): 2671–2681. doi:10.1029/95WR03818

[14] Snodgrass M F, Kitanidis P K. A Geostatistical approach to contaminant source identification. Water Resources Research, 1997; **33**(4): 537–546. doi:10.1029/95WR03818

[15] Aral M M, Guan J, Maslia M L. Identification of contaminant source location and release history in aquifers. Journal of Hydrologic Engineering, 2001; **6**(3): 225–234. doi:10.1061/(ASCE)1084-0699(2001)6:3(225)

[16] Mahinthakumar G K, Sayeed M. Hybrid genetic algorithm—local search methods for solving groundwater source identification inverse problems. Water Resources Planning and Management, 2005; **131**(1): 45–57. doi:10.1061/(ASCE)0733-9496(2005)131:1(45)

[17] Tikhonov A N. On the stability of the functional optimization problem. USSR Computational Mathematics and Mathematical Physics, 1966; **6**(4): 28–33. doi:10.1016/0041-5553(66)90003-6

[18] Gorelick S M, Evans B, Remson I. Identifying sources of groundwater pollution: an optimization approach. Water Resources Research, 1983; **19**(3): 779–790. doi:10.1029/WR019i003p00779

[19] Datta B, et al. Development of an expert system embedding pattern recognition techniques for pollution source identification. In: Project Completion Technical Report submitted to United States Geological Survey (U.S.G.S.). 1989. University of California, Davis: NTIS, Springfield, Virginia, USA.

[20] Datta B, Chakrabarty D, Dhar A. Simultaneous identification of unknown groundwater pollution sources and estimation of aquifer parameters. Journal of Hydrology, 2009; **376**(1–2): 48–57. doi:10.1016/j.jhydrol.2009.07.014

[21] Amirabdollahian M, Datta B. Identification of contaminant source characteristics and monitoring network design in groundwater aquifers: an overview. Journal of Environmental Protection, 2013; **4**(5A): 26–41. doi:10.4236/jep.2013.45A004

[22] Datta B, Kourakos G. Preface: Optimization for groundwater characterization and management. Hydrogeology Journal, 2015; **23**(6): 1043–1049. doi:10.1007/s10040-015-1297-3

[23] Ingber L. Adaptive simulated annealing (ASA): lessons learned. Control Cybernetics, 1996; **25**(1): 33–54.

[24] Singh R M, Datta B. Identification of groundwater pollution sources using GA-based linked simulation optimization model. Journal of Hydrologic Engineering, 2006; **11**(2): 101–109. doi:10.1061/(ASCE)1084-0699(2006)11:2(101)

[25] Yeh H-D, Chang T-H, Lin Y-C. Groundwater contaminant source identification by a hybrid heuristic approach. Water Resources Research, 2007; **43**(9): 1–16. doi:10.1029/2005WR004731

[26] Amirabdollahian M, Datta B. Identification of pollutant source characteristics under uncertainty in contaminated water resources systems using adaptive simulated annealing and fuzzy logic. International Journal of Geomate, 2014; **6**(1): 757–762.

[27] Gurarslan G, Karahan H. Solving inverse problems of groundwater-pollution-source identification using a differential evolution algorithm. Hydrogeology Journal, 2015; **23**(6): 1109–1119. doi:10.1007/s10040-015-1256-z

[28] Yeh W W G. Review: Optimization methods for groundwater modeling and management. Hydrogeology Journal, 2015; **23**(6): 1051–1065. doi:10.1007/s10040-015-1260-3

[29] Mahar P S, Datta B. Optimal monitoring network and ground-water-pollution source identification. Water Resources Planning and Management, 1997; **123**(4): 199–207. doi:10.1061/(ASCE)0733-9496(1997)123:4(199)

[30] Ingber L. Very fast simulated re-annealing. Mathematical and Computer Modelling, 1989; **12**(8): 967–973. doi:10.1016/0895-7177(89)90202-1

[31] Ingber L, Rosen B. Genetic algorithms and very fast simulated reannealing: a comparison. Mathematical and Computer Modelling, 1992; **16**(11): 87–100. doi:10.1016/0895-7177(92)90108-W

[32] Zheng C, Hill M C, Hsieh P A. MODFLOW-2000, The U.S. Geological Survey Modular Ground-Water Model-User Guide to the LMT6 Package, The Linkage with MT3DMS for Multi-Species Mass Transport Modeling, in Open File Report 01-82, U.S.G. SURVEY, Editor. 2001. Denver, Colorado.

[33] Zheng C, Wang P P. A field demonstration of the simulation optimization approach for remediation system design. Ground Water, 2002; **40**(3): 258–266. doi:10.1111/j.1745-6584.2002.tb02653.x

[34] Zheng C, Wang P P. A modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems, U.A.C.o. Engineers, Editor. 1999. Washington, DC.

[35] Zheng C, Bennett G D. Applied Contaminant Transport Modeling. 2002. New York: Wiley-Interscience.

Edited by Hossein Peyvandi

The purpose of optimization is to maximize the quality of lives, productivity in time, as well as interests. Therefore, optimization is an ongoing challenge for selecting the best possible among many other inferior designs. For a hundred years in the past, as optimization has been essential to human life, several techniques have been developed and utilized. Such a development has been one of the long-lasting challenges in engineering and science, and it is now clear that the optimization goals in many of real-life problems are unlikely to be achieved without resource for computational techniques. The history of such a development in the optimization techniques starts from the early 1950s and is still in progress. Since then, the efforts behind this development dedicated by many distinguished scientists, mathematicians, and engineers have brought us today a level of quality of lives.

This book concerns with the computational optimization in engineering and techniques to resolve the underlying problems in real life. The current book contains studies from scientists and researchers around the world from North America to Europe and from Asia to Australia.

ISBN 978-953-51-4862-3

Photo by e_zebolov / iStock

IntechOpen