

IntechOpen

# Optimization Algorithms

## Methods and Applications

*Edited by Ozgur Baskan*





---

# **OPTIMIZATION ALGORITHMS - METHODS AND APPLICATIONS**

---

Edited by **Ozgur Baskan**

## Optimization Algorithms - Methods and Applications

<http://dx.doi.org/10.5772/61426>

Edited by Ozgur Baskan

### Contributors

Hong Qi, Stefan Breban, Ramesh K. Agarwal, Arif Basgumus, Mustafa Namdar, Gunes Yilmaz, Ahmet Altuncu, Mary P. Zadeh, Fateme Pourhasanzade, Tahere Pourhasanzade, Andreas Rienow, Abraham Yosipof, Hanoeh Senderowitz, Simon Fong, Zhonghuan Tien, Christine Chan, Weihua Jin, Zhiying Hu, Hui Jiang, Yugal Kumar, Sumit Gupta, G. Sahoo, Amer M. Fahmy, Takeo Ishikawa, Duoc Phan Thanh, James B.P. Lim, Waqar Ahmed Khan, Nawaf Hamadneh, Surafel Tilahun, Jean M. T. Ngnotchouye

### © The Editor(s) and the Author(s) 2016

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](mailto: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 found at <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, 2016 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](mailto:orders@intechopen.com)

Optimization Algorithms - Methods and Applications

Edited by Ozgur Baskan

p. cm.

Print ISBN 978-953-51-2592-1

Online ISBN 978-953-51-2593-8

eBook (PDF) ISBN 978-953-51-5077-0

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

3,500+

Open access books available

111,000+

International authors and editors

115M+

Downloads

151

Countries delivered to

Our authors are among the  
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](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)





# Meet the editor



Ozgur Baskan is an Associate Professor at the Faculty of Engineering at the Pamukkale University in Turkey. He received his PhD degree in Transportation Engineering from Graduate School of Natural and Applied Sciences from the same university in 2009. In 2012, he was a visiting scholar at the Division of Transportation Engineering, Technical University of Bari, Italy. He is currently mainly interested in the fields of traffic and transportation planning, specifically road network design, traffic assignment, and nature-inspired optimization algorithms.



---

# Contents

---

## **Preface XI**

### **Section 1 Engineering Applications 1**

- Chapter 1 **Genetic Algorithm Optimization of an Energy Storage System Design and Fuzzy Logic Supervision for Battery Electric Vehicles 3**  
Stefan Breban
- Chapter 2 **The Future of Central European Cities – Optimization of a Cellular Automaton for the Spatially Explicit Prediction of Urban Sprawl 17**  
Andreas Rienow
- Chapter 3 **Inverse Geometry Design of Radiative Enclosures Using Particle Swarm Optimization Algorithms 43**  
Hong Qi, Shuang-Cheng Sun, Zhen-Zong He, Shi-Ting Ruan, Li-Ming Ruan and He-Ping Tan
- Chapter 4 **Shape Optimization of Busemann-Type Biplane Airfoil for Drag Reduction Under Non-Lifting and Lifting Conditions Using Genetic Algorithms 71**  
Yi Tian and Ramesh K. Agarwal
- Chapter 5 **Performance Analysis of the Differential Evolution and Particle Swarm Optimization Algorithms in Cooperative Wireless Communications 99**  
Arif Basgumus, Mustafa Namdar, Gunes Yilmaz and Ahmet Altuncu
- Chapter 6 **Genetic Algorithm-Based Approaches for Solving Inexact Optimization Problems and their Applications for Municipal Solid Waste Management 119**  
Weihua Jin, Zhiying Hu and Christine W. Chan

<b>Section 2</b>	<b>Applications in Various Areas</b>	<b>145</b>
Chapter 7	<b>Optimization Algorithms for Chemoinformatics and Material-Informatics</b>	<b>147</b>
	Abraham Yosipof and Hanoeh Senderowitz	
Chapter 8	<b>Optimization Algorithms in Project Scheduling</b>	<b>171</b>
	Amer M. Fahmy	
Chapter 9	<b>Survey of Meta-Heuristic Algorithms for Deep Learning Training</b>	<b>195</b>
	Zhonghuan Tian and Simon Fong	
Chapter 10	<b>Design and Characterization of EUV and X-ray Multilayers</b>	<b>221</b>
	Hui Jiang	
Chapter 11	<b>A Clustering Approach Based on Charged Particles</b>	<b>245</b>
	Yugal Kumar, Sumit Gupta, Dharmender Kumar and Gadadhar Sahoo	
Chapter 12	<b>Topology Optimization Method Considering Cleaning Procedure and Ease of Manufacturing</b>	<b>265</b>
	Takeo Ishikawa	
Chapter 13	<b>A Review and Comparative Study of Firefly Algorithm and its Modified Versions</b>	<b>281</b>
	Waqar A. Khan, Nawaf N. Hamadneh, Surafel L. Tilahun and Jean M. T. Ngnotchouye	

---

## Preface

---

During the past decades, optimization algorithms, especially nature-inspired algorithms, have attracted a great deal of attention for solving different type of problems arising in various areas such as engineering, computer science, operations research, and machine learning. The use of nature-inspired optimization algorithms has become very popular because of the fact that many real-world problems are significantly complex. Despite there is no guarantee of finding the optimal solution in these methods, they are strongly preferred because of their simplicity and less computational cost. On the other hand, although a large number of algorithms have been developed in this period, we need to review and summarize these optimization algorithms and their applications in order to introduce new developments to this area.

This book is written especially for researchers and practitioners who wish to improve their knowledge in possible applications for different optimization algorithms. It consists of 13 chapters divided into two parts — (I) Engineering applications and (II) Applications in various areas — and covers state-of-the-art methods and their applications in wide range.

In conclusion, I would like to thank to all authors for taking the time and efforts to prepare their chapters and also to acknowledge Ms. Romina Rován, InTech Publishing Process Manager, for her kind and professional assistance during the entire publishing process of this book.

**Ozgur Baskan**  
Pamukkale University,  
Faculty of Engineering,  
Department of Civil Engineering,  
Denizli, Turkey



---

# Engineering Applications

---



---

# **Genetic Algorithm Optimization of an Energy Storage System Design and Fuzzy Logic Supervision for Battery Electric Vehicles**

---

Stefan Breban

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62587>

---

## **Abstract**

This chapter presents a methodology to optimize the capacity and power of the ultracapacitor (UC) energy storage device and also the fuzzy logic supervision strategy for a battery electric vehicle (BEV) equipped with electrochemical battery (EB). The aim of the optimization was to prolong the EB life and consequently to permit financial economies for the end-user of the BEV. Eight variables were used in the optimization process: two variables that control the energy storage capacity and power of the UC device and six variables that change the membership functions of the fuzzy logic supervisor. The results of the optimization, using a genetic algorithm from MATLAB®, are showing an increase of the financial economy of 16%.

**Keywords:** Genetic algorithm optimization, battery electric vehicle, fuzzy logic, ultracapacitor, electrochemical battery

---

## **1. Introduction**

The humanity has to act on two major directions in order to reduce the pollution and the greenhouse effect of carbon dioxide released into atmosphere: on the one hand to increase the exploitation of renewable energy in the detriment of fossil fuel and on the other hand to increase the energy conversion efficiency in all the sectors of activity. Electrification of transportation sector will help reduce the pollution, mainly in the cities as there are mostly affected by this problem and will help reduce the greenhouse effect if the energy that powers the electric vehicles comes from renewable sources. The main advantages of electric vehicles

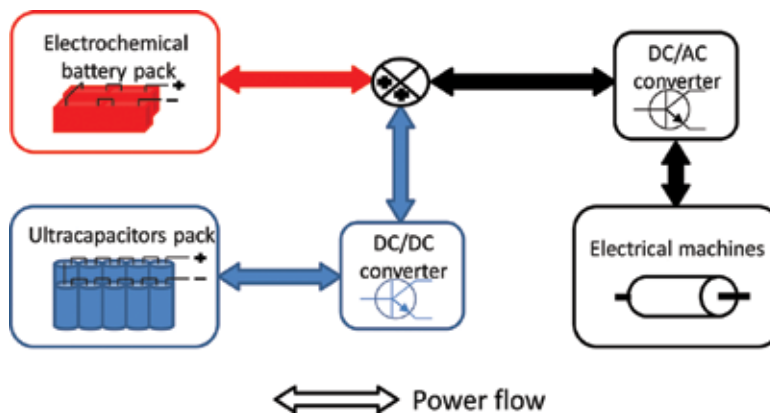
compared to those equipped with combustion engine are as follows: greater efficiency, increased reliability, better dynamics, and sometimes smaller costs [1].

Pure electric vehicles can be classified into non-autonomous vehicles and autonomous vehicles. The non-autonomous vehicles, represented by tramways, trolleybuses, metros, electric locomotives, and trains, depend on an external electric energy supply system: catenary lines or feeding rail. These types of vehicles are very clean and efficient solution to move people and goods on an established trail or route. In the future, these types of vehicles will be further improved and their use extended. The autonomous electric vehicles are needed where the routes are variated, for example, for personal small vehicles. These vehicles are usually depending on an electrochemical battery (EB) to be feed. The EBs are nowadays the most expensive part of the battery electric vehicles (BEVs), and thus, actions to optimize their operation and increase their lifespan should be taken. In [2], the authors are stating that for some LiFePO<sub>4</sub> batteries, “the cycle depth of discharge and relative fraction of low-rate galvanostatic cycling vs. acceleration/regenerative braking current pulses are not important even over thousands of driving days”; in conclusion, the only important factor in battery ageing is the energy processed. In this study, the authors are estimating an approximate capacity lost per normalized Wh of about  $-6 \times 10^{-3}\%$  for plug-in hybrid vehicle use and  $-2.7 \times 10^{-3}\%$  for vehicle to grid use, due to more rapid cycling found in driving conditions. In order to reduce the energy processed by the EB, a very well-known solution is to complement it with an ultracapacitor (UC) energy storage device that has opposite characteristics compared to EB, high-power and low energy density. Many papers are treating this combined energy storage system. The UC has usually the role to reduce the stress on the EB, by power peak shaving and braking energy recovering. In reference [3], a comparison between “current/voltage/power profiles of the batteries with and without UCs indicated the peak currents and thus the stress on the batteries were reduced by about a factor of three using UCs. This reduction is expected to lead to a large increase in battery cycle life”. The authors of reference [4] are proposing a strategy to design and supervise the battery and UC on a fuel-cell hybrid electric vehicle. The proposed strategy uses low-pass filters and some logical operations. In reference [5], a fuzzy logic strategy is presented, aiming at the reduction of power peaks on the EB with the help of a UC. In [6], a fuzzy logic control method is utilized to design an energy management strategy that enhances the fuel economy and increases the mileage of a vehicle by means of a hybrid energy storage power system consisting of fuel cell, EB, and UC. The authors of reference [7] are proposing a new battery/UC configuration that allows a reduced-size power converter. The braking energy is completely stored in the UC, having an important capacity of almost 1200 kJ.

Compared to state of the art, this chapter presents a methodology to optimize altogether the capacity and power of the UC energy storage device and also the fuzzy logic supervision strategy for a BEV equipped with EB. In Section 2, the power system architecture will be presented, in Section 3, the fuzzy logic supervision strategy, in Section 4, the BEV simulation, and in Section 5, the optimization using genetic algorithm.

## 2. Power system architecture

**Figure 1** presents the simplified diagram of the on-board power system [8] considered. The main power source consists of an EB that can be connected to the loads directly or by means of a power converter. The UC usually is connected through a buck–boost (DC/DC) converter to the DC link, due to low-voltage operation. The electric motors are supplied through power inverters (DC/AC converter).



**Figure 1.** Schematic of the BEV power system.

The EB used should be a rechargeable type. Li-based battery technology is nowadays the most used type of battery in electric vehicles due to its high energy-to-weight ratio, no memory effect, and low self-discharge, compared to other solutions like Ni/Cd or Ni/MH. Another important candidate that does not use toxic elements (like Lithium) is the EB with molten salts. This technology is maybe not as mature as Lithium technology but has similar energy density. The most important drawback of molten salt batteries is that they work at high temperature, around 300°C; thus, the thermal insulation of the battery should be very good in order to increase its efficiency.

UCs work in much the same way as conventional capacitors, in that there is no ionic or electronic transfer resulting in a chemical reaction, that is energy is stored in the electrochemical capacitor by simple charge separation. The main advantage of the UCs is the high-power capability that makes them highly suitable to be used in conjunction with the EBs. The energy stored ( $E$ ) in UCs varies linearly with the equivalent capacity ( $C$ ) and with the square of the voltage ( $U$ ):

$$E = \frac{1}{2} * C * U^2 \quad (1)$$

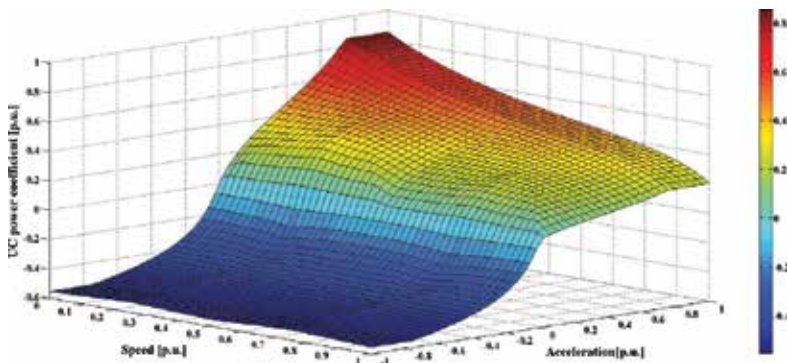
### 3. Fuzzy logic supervision strategy

The fuzzy logic supervision strategy is considered appropriate to create an overall energy flow management between electrical machines or equipment and energy storage devices. The main idea behind this supervision strategy is to vary the UCs level of charge considering the BEV operation point. Thus, it has been considered that when the BEV is at stop, the UC should have a high state of charge (SoC), to be able to provide power when BEV starts moving. During the increase of speed, the UCs should reduce their energy storage and when arriving at high speeds should be discharged to be able to recover the most or all of the energy generated when braking. More details are given in Breban and Radulescu [8].



**Figure 2.** Fuzzy logic supervision strategy methodology.

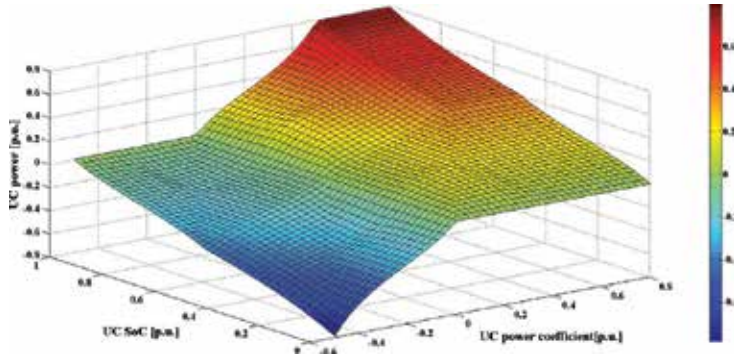
The fuzzy logic supervision strategy is divided into two levels. Each level of supervision has two inputs and one output. The input variables of the first supervision level are the BEV speed and acceleration. The output is a power coefficient of the UC (**Figure 2**). The second level of supervision has also two inputs, that is the output of the level one and the SoC of the UC, and one output, the UC power. All variables are expressed in p.u. values. These are representing the ratio of each considered parameter to its nominal value. The *Gain* multiplier makes the passing from p.u. to real power system values. This multiplier can also be used to increase or decrease the dynamics of the supervision strategy.



**Figure 3.** First level fuzzy logic supervision response surface.

For each level of supervision, a 3D response surface can be plotted (**Figures 3 and 4**). The outputs have a variation between  $-0.55$  and  $0.85$  p.u. for the first level and between  $-0.8$  and

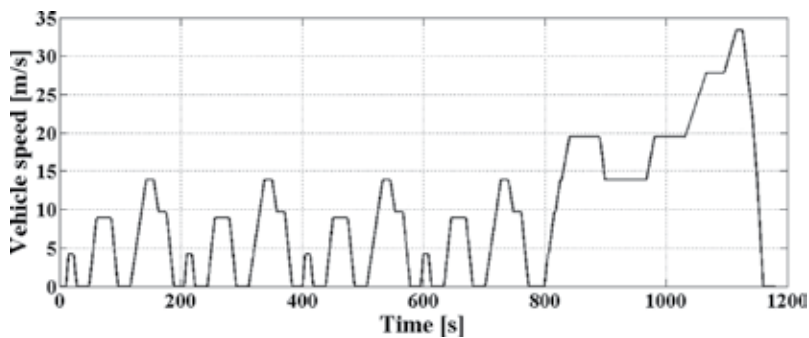
0.8 p.u. for the second level. This is due to the centroid defuzzification method. Thus, the UC power coefficient input of second-level supervision was developed with a variation between  $-0.5$  and  $0.8$  p.u. to increase the supervisor dynamic response at the limits of variation. More details are presented in Breban and Radulescu [8].



**Figure 4.** Second level fuzzy logic supervision response surface.

#### 4. BEV simulation

In order to obtain the power absorbed or produced by the BEV, three different simulations for two driving cycles were considered: the New European Driving Cycle (NEDC) that consists of four ECE-15 cycles followed by one EUDC cycle (**Figure 5**), and the Urban Dynamometer Driving Schedule (UDDS) as presented in **Figure 6**. The first two simulations are considering the EUDC and UDDS cycles with slopes (**Figures 7 and 8**) and in the third simulation, the UDDS cycle without slopes. The simulated BEV has a total mass of  $1400$  kg, the equivalent frontal area is  $2.2$  m<sup>2</sup>, and the aerodynamic drag coefficient is  $0.25$ . The air density was considered  $1.2$  kg/m<sup>3</sup> and the air mass speed, zero. The BEV is equipped with a  $16$  kWh EB.



**Figure 5.** BEV speed (NEDC cycle).

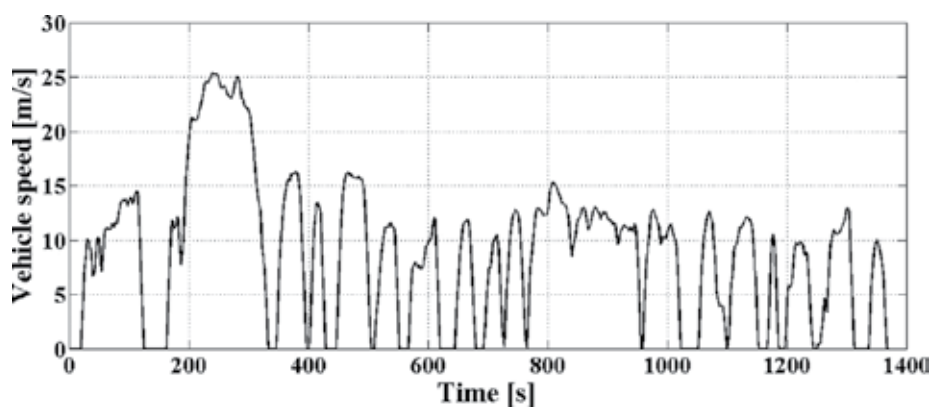


Figure 6. BEV speed (UDDS cycle).

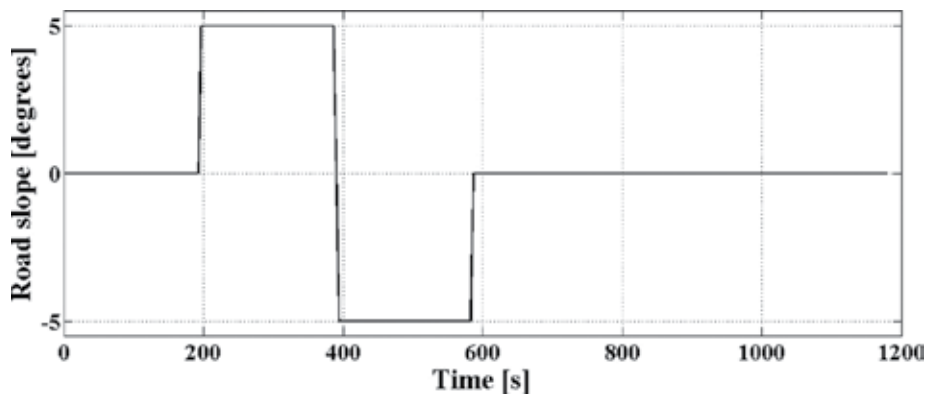


Figure 7. Road gradients (NEDC cycle).

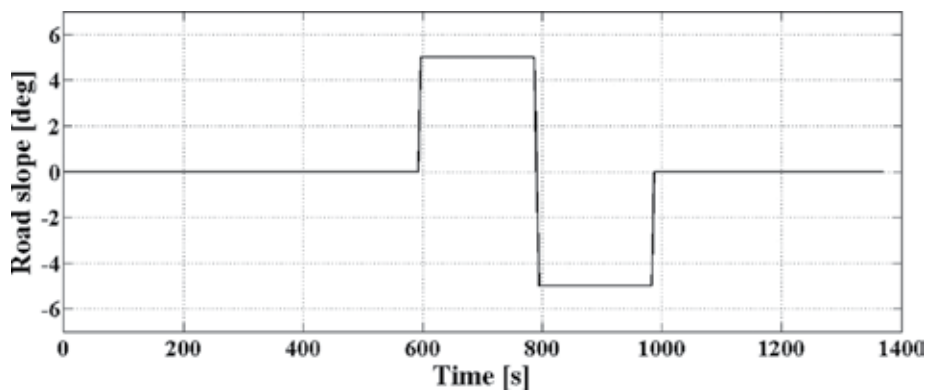


Figure 8. Road gradients (UDDS cycle).

## 5. Optimization using genetic algorithm

The optimization was made using Global Optimization toolbox and Optimization tool interface from Matlab®. Eight variables were used in the optimization process: The UC capacity (kJ), the *Gain* that passes the UC power to real values (kW) and six variables that change the membership functions of the fuzzy logic supervisor, that is the membership functions of the two inputs and one output for each level of supervision. The limits of variation for the eight variables are presented in **Table 1**. In the third line of **Table 1**, the empiric choice of variables is presented, which is used in the first phase of development of the fuzzy logic supervision strategy, with the results presented in Breban and Radulescu [8].

Variables	Gain (kW)	UC capacity (kJ)	First input; first level	Second input; first level	Output; first level	First input; second level	Second input; second level	Output; second level
Limits of variation	1÷100	1÷1000	0÷0.25	0÷0.375	0÷0.1	0÷0.25	0÷0.25	0÷0.2
Empiric choice	25	250	0	0	0	0	0	0

**Table 1.** Variables and limits of variation during optimization and empiric choice of variables.

The number of individuals used in the optimization algorithm is 800. This number was empirically chosen considering the fact that eight optimization variables were used and in order to allow a good initial spreading of individuals in the eight dimensions domain of search. In other words, 100 individuals were consider for each optimization variable. The number of generations was set to 25 as it was observed that the optimization function was converging. Two individuals are guaranteed to survive to each next generation, 80% of the individuals are generated by crossover and the remaining by mutation. The crossover function creates a random binary vector in order to select the genes from two parents and from a child. The mutation randomly generates new individuals considering that the limits of variation (presented in **Table 1**) are satisfied. At every five generations, 20% of the individuals of  $n^{\text{th}}$  subpopulation are migrating toward the  $(n + 1)^{\text{th}}$  subpopulation. This percent is calculated considering the smaller of the two subpopulations that moves.

The optimization function Eq. (3) to be minimized is the difference between the cost of the UC and the financial economy due to the increase in the lifespan of the EB. This financial economy Eq. (2) is calculated considering the product between the EB cost and the reduction of the energy processed by the EB, with the optimum UC capacity and control variables, compared to the case when no UCs are used, expressed in percentage.

$$\text{Economy}_{\text{EB life increase}} = \text{EB}_{\text{cost}} \times E_{\text{reduction}} \quad (2)$$

$$f = SC_{\text{cost}} - \text{Economy}_{\text{EB life increase}} \quad (3)$$

The EB considered cost is 500 dollars/kWh, and the UC cost is 2.85 dollars/Wh. The empiric choice of the variables gives for  $f = 1942$  dollars of economies and the optimized variables  $f = 2256.4$  dollars (Figure 9), thus an increase of around 16% is achieved.

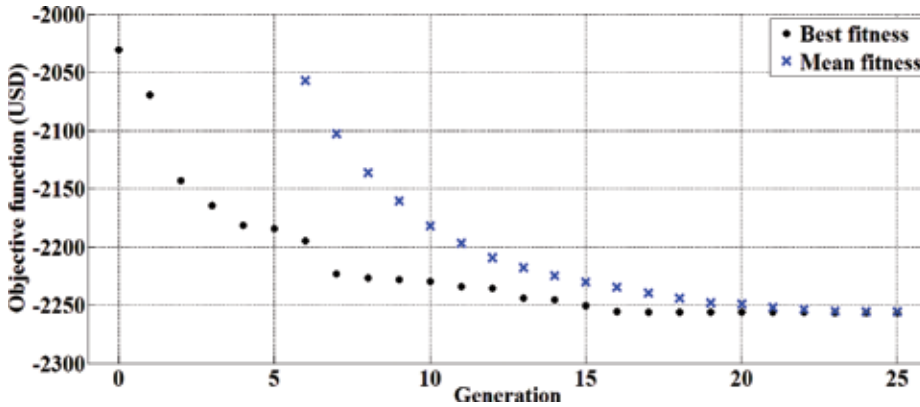


Figure 9. Best and mean fitness for the optimization function.

The optimum values for the eight variables used in the optimization algorithm are given in Table 2. As can be seen, the first two variables and the outputs of each level of supervision variables are changing considerable from the empiric choices, and the other four variables have only slight modifications or none. Also, the *Gain* value increases from 25 to reach almost the maximum permissible value, that is an increase of UC power from 20 to 80 kW. UC capacity increases from 250 kJ to more than 600 kJ. It should be noted that the mass of the BEV was considered constant whatever the value of the UC was used.

Variables	Gain (kW)	UC capacity (kJ)	First input; first level	Second input; first level	Output; first level	First input; second level	Second input; second level	Output; second level
Optimum	99.544	614.247	0.001	0.001	0.095	0	0	0.168

Table 2. Optimum values for the optimization variables.

As an example of membership function modifications with the change of an optimization variable, in Figures 10 and 11, the membership functions are presented for the output of the first level of supervision in the case of empiric choice of variable, respectively, optimum variable.

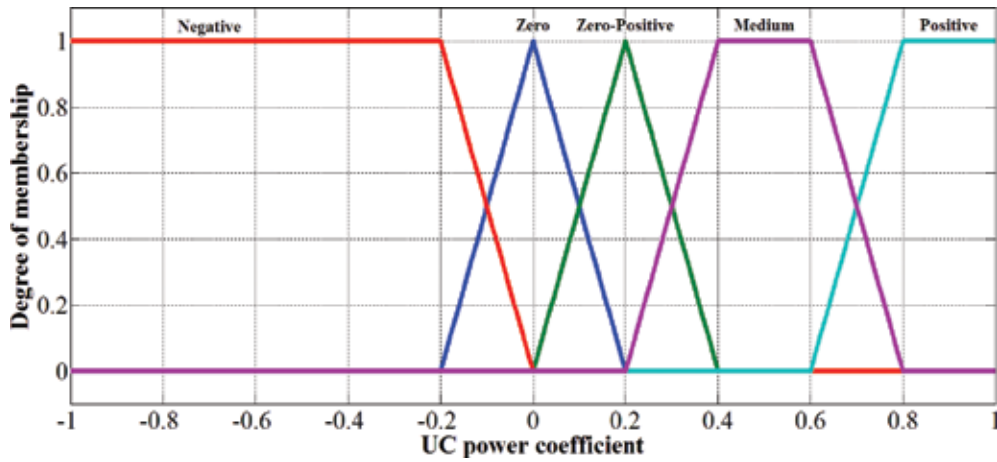


Figure 10. Membership functions for empiric choice of variable.

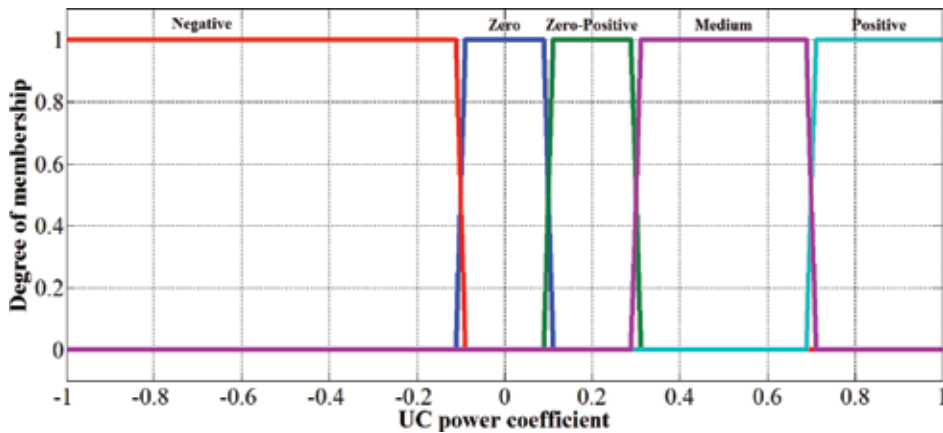


Figure 11. Membership functions for optimum variable.

From the point of view of BEV operation, the EB power and the UC power for each of the three road simulation conditions are presented as follows. **Figures 12** and **13** present the EB and UC powers for NEDC having the characteristics presented in **Figures 5** and **7**. **Figures 14** and **15** present the EB and UC powers for UDDS cycle having the characteristics presented in **Figures 6** and **8** from 500 to 1200 s in order to better view the power variations during BEV operation on a road with slopes. **Figures 16** and **17** present the EB and UC powers for UDDS cycle having the characteristics presented in **Figure 6** (no slopes) for the first 500 s.

As expected, the EB power, when optimum variables are used, decreases in certain periods of BEV operation, compared to the cases where the empiric variables were used, thus the energy processed by the EB reduces, as the SC power increases. Finally, this would lead to a lifespan extension for the EB and financial economies for the end-user.

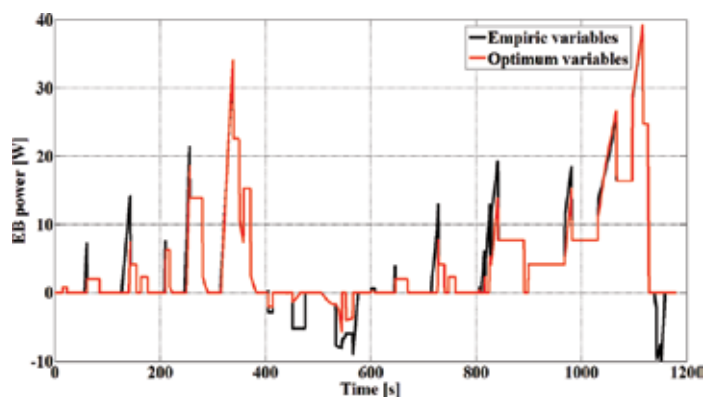


Figure 12. EB power for EUDC cycle.

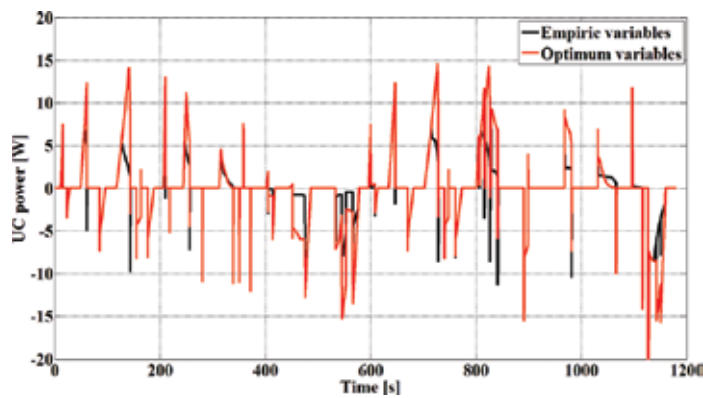


Figure 13. UC power for EUDC cycle.

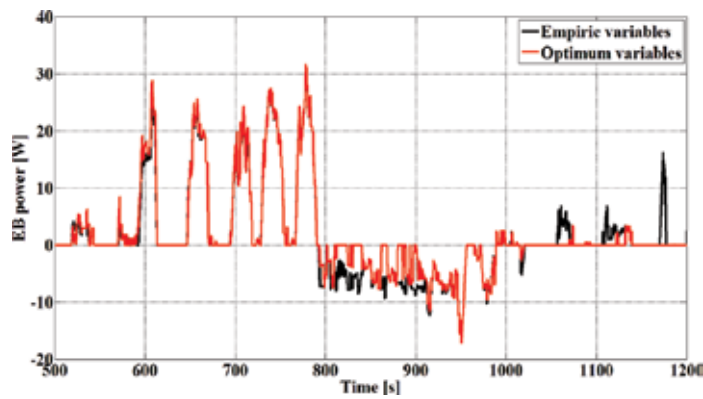


Figure 14. EB power for UDDS cycle with road gradients.

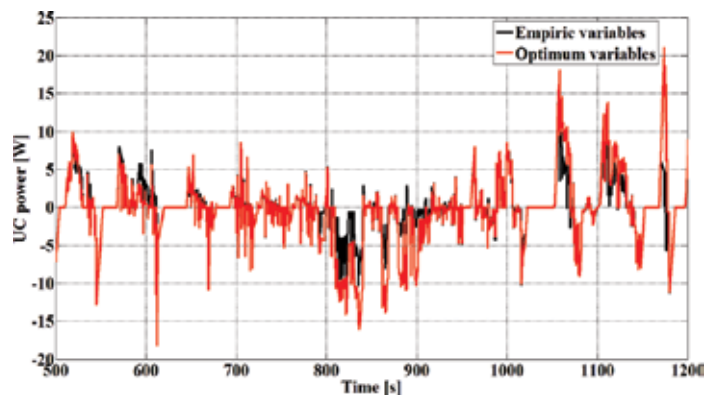


Figure 15. UC power for UDDS cycle with road gradients.

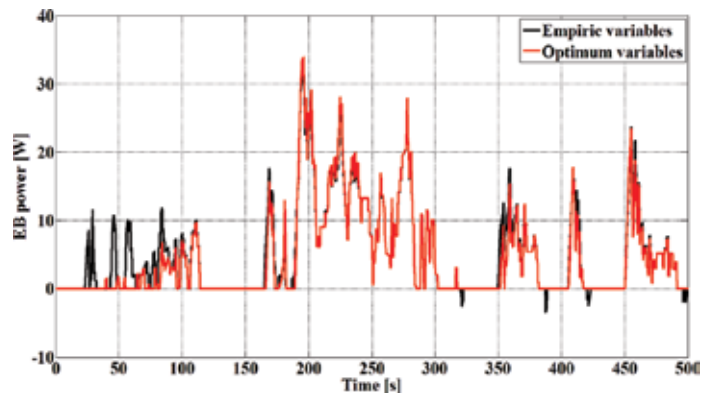


Figure 16. EB power for UDDS cycle without road gradients.

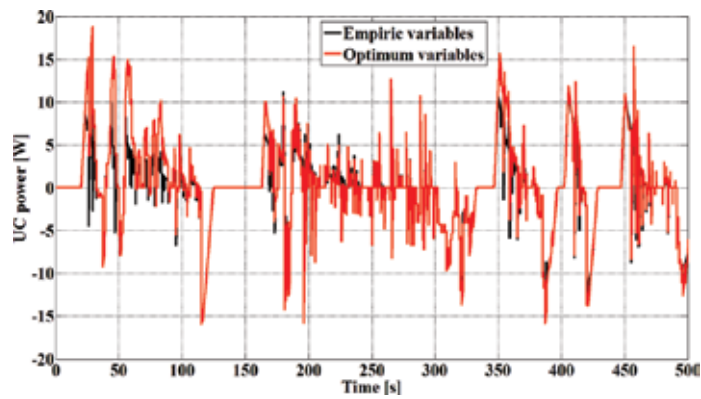


Figure 17. UC power for UDDS cycle without road gradients.

## 6. Conclusion

A methodology to optimize the capacity and power of the UC energy storage device and also the fuzzy logic supervision strategy for a BEV equipped with EB was presented. The results are showing that important financial economies could be made if an UC energy storage device is used with the aim to reduce the energy processed by the EB. The optimization algorithm maximizes these economies, in this study, an increase of around 16% is achieved, proving that optimization is an essential part of any product and system development.

## Acknowledgements

This work was supported by the project “Development and support of multidisciplinary postdoctoral programmes in major technical areas of national strategy of Research—Development—Innovation” 4D-POSTDOC, contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectoral Operational Programme Human Resources Development 2007–2013.

## Author details

Stefan Breban

Address all correspondence to: Stefan.Breban@emd.utcluj.ro

Technical University of Cluj-Napoca, Cluj-Napoca, Romania

## References

- [1] Zhang H, Saudemont C, Robyns B, Petit M. Electrical features comparison between more electric aircrafts and hybrid electric vehicles. *Electromotion* 2009;16(3):111-120.
- [2] Peterson SB, Apt J, Whitacre JF. Lithium-ion battery cell degradation resulting from realistic vehicle and vehicle-to-grid utilization. *Journal of Power Sources* 2010;195:2385–2392. doi:10.1016/j.jpowsour.2009.10.010
- [3] Burke H, Miller M, Zhao H. Lithium batteries and ultracapacitors alone and in combination in hybrid vehicles: Fuel economy and battery stress reduction advantages. In: 25th World Battery, Hybrid and Fuel Cell Electric Vehicle Symposium & Exhibition; 5–9 Nov. 2010; Shenzhen, China. 2010.

- [4] Schaltz E, Khaligh A, Rasmussen PO. Influence of battery/ultracapacitor energy-storage sizing on battery lifetime in a fuel cell hybrid electric vehicle. *IEEE Transactions on Vehicular Technology* 2009;58(8):3882–3891. doi:10.1109/TVT.2009.2027909
- [5] Wang G, Yang P, Zhang J. Fuzzy optimal control and simulation of battery-ultracapacitor dual-energy source storage system for pure electric vehicle. In: *International Conference on Intelligent Control and Information Processing*; 13–15 Aug. 2010; Dalian, China. 2010. doi:10.1109/ICICIP.2010.5564185
- [6] Li Q, Chen W, Li Y, Liu S, Huang J. Energy management strategy for fuel cell/battery/ultracapacitor hybrid vehicle based on fuzzy logic. *International Journal of Electrical Power & Energy Systems* 2012;43(1):514–525. doi:10.1016/j.ijepes.2012.06.026
- [7] Cao J, Emadi A. A new battery/ultracapacitor hybrid energy storage system for electric, hybrid, and plug-in hybrid electric vehicles. *IEEE Transactions on Power Electronics* 2012;27(1):122–132. doi:10.1109/TPEL.2011.2151206
- [8] Breban S, Radulescu MM. Fuzzy-logic supervision strategy for battery-powered electric vehicles. *UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science* 2014;76(2):187–196.



---

# **The Future of Central European Cities – Optimization of a Cellular Automaton for the Spatially Explicit Prediction of Urban Sprawl**

---

Andreas Rienow

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62424>

---

## **Abstract**

The quantitative and qualitative measurement, prediction and evaluation of urban sprawl have come to play a central role in land-system science. One of the most important and most implemented artificial intelligence (AI) techniques in terms of urban systems simulation is cellular automata (CA) like SLEUTH. SLEUTH models the physical urban expansion by accomplishing four simple growth rules with every modeling step. Simultaneously, SLEUTH also reflects main drawbacks of CA since they contain a higher degree of stochastic variation leading to a simulation uncertainty. This chapter will explain how the simulation power of CA can be optimized by combining them with the machine learning algorithm support vector machines (SVMs). Conceptually in SVMs, input vectors are projected in a higher-dimensional feature space in which an optimal separating hyperplane can be constructed for separating the input data into two or more classes. In the comparative analysis, the integrated modeling approach is carried out for a unique postindustrial European agglomeration: The Ruhr Area. It will be demonstrated how the AI learning approach is implemented, calibrated, validated and applied for the prediction of the regional urban land-cover pattern between 1975 and 2005. Finally, the probability effects will be visualized with the concept of urban DNA.

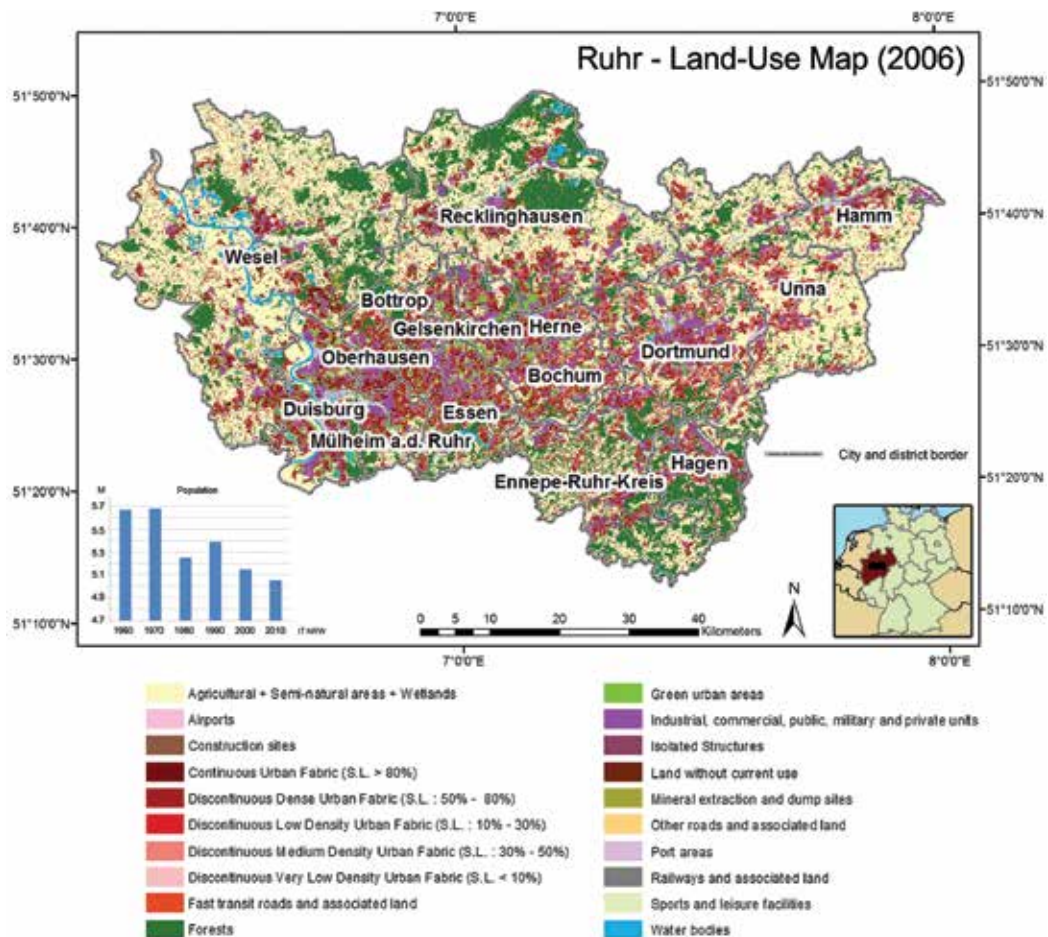
**Keywords:** support vector machines, Cellular automata, SLEUTH, urban growth, probability maps

## 1. Introduction

The American landscape designer Earle Draper initiated a term describing the unaesthetic and uneconomic settlement structure of American cities in his 1937 lecture on town planning. In reflecting the obliteration of rural and urban spaces, this term grew more and more in popularity: “perhaps diffusion is too kind of word. in bursting its bounds, the city actually sprawled and made the countryside ugly, uneconomic [in terms] of services and doubtful social value [1]”. Ever since the term ‘sprawl’ emerged in a geographical context, nearly 80 years have passed and the specific patterns and processes of growing cities have developed from an American to a European problem in general and a German problem in particular [2–5]. Being one of the most challenging land-use and land-cover changes implicating several consequences for the anthropogenic and geobiophysical spheres, urban sprawl has become an inherent part of the international sustainability discourse in the context of global change [6–10]. The same holds true for geographical research with its long history of diverse theories and models of land-use change in general and urban sprawl in specific [11–16]. One of the most important streams of urban models understands urban areas as complex systems. Those geosimulation techniques make use of artificial intelligence (AI) to model the micro processes being responsible for the macro pattern of urban systems [11, 12, 17]. Following the famous quote by Aristotle “the whole is more than the sum of its parts”, they model urban systems in a bottom-up manner. Cellular automata (CA) are very popular geosimulation tools. SLEUTH is a well-known urban CA [18]. It is a purely growth-oriented model and as a bottom-up approach it is not dependent on intensive prestudies about the general causes of urban growth or the location-specific determining factors. Based on the principles of neighborhood effects and spatial autocorrelation, the simulation rules are relatively simple. However, SLEUTH is able to capture complex emergences of urban patterns so that it has been applied in several urban growth studies all over the world [19]. Although the performance of CA for spatially explicit urban land-use simulation is very high, it gives no direct insight into the relationship between nonspatial human and ecological driving forces, spatial determining factors and the emergence of urban growth. The semistatistical method called support vector machines (SVMs) [20] is able to avoid this disadvantage: developed for solving nonlinear classification problems, SVMs are also suitable for analyzing the spatial driving factors of urban land-use change. They are a machine learning concept based on statistical learning theory. The basic idea is to project input vectors on a higher dimensional feature space, in which an optimal separating hyperplane can be constructed for separating the data into two or more classes. By using a specific feature selection, neglectable features can be separated and important features can be identified.

The objective of this contribution is to demonstrate how the simulation skills of CA on the one hand and SVM on the other hand can be integrated in order to achieve an optimized modeling approach. The anchor point for joining AI and machine learning physique is the exclusion layer of SLEUTH. Instead of the standard input map of the CA defining restriction areas where no urbanization is allowed to take place, a probability map derived from SVM will be utilized. In a nutshell, the coupling of SLEUTH and SVM is undertaken in order to answer the following research questions:

1. Which driving factors influence the physical urban growth in the Ruhr and how do they take effect according to a SVM feature selection?
2. Do SVM guide SLEUTH and do they optimize its modeling ability?
3. How well do SVM perform in comparison to standard SLEUTH implementations?



**Figure 1.** Land-use map (2006) with cities and districts of the Ruhr (North Rhine-Westphalia, Germany) [2].

The main study area which SLEUTH-SVM is set up for lies in the western part of Germany and in the central part of NRW: the Ruhr (**Figure 1**). With a polycentric and administratively fragmented structure but a homogenous and extensive urban area, the Ruhr is a worldwide unique urban entity. In general, 11 cities and 4 districts form the biggest agglomeration (1150 people per km<sup>2</sup>) in Germany, and with its 443,969 ha it is the fifth largest urban region in Europe. Concerning the geosimulation of urban systems, the Ruhr proves to be suitable due to two principal aspects: Firstly, it exhibits the highest absolute rates of urban sprawl in

Germany. Between 1975 and 2005, the agglomeration grew around 37,022 ha with a total urban area of 94,990 to 132,012 [21]. Compared to other sprawling cities in Germany, the Ruhr's urban expansion can be described as a 'metropolitan suburban sprawl' type with high values of new land consumption and with low fragmentations and dispersion patterns in the core area coexisting with lower densities and patched open spaces in the suburban area [5]. Secondly, the Ruhr acts as a 'hero' in the scientific discourse of demographic decline and structural transformations in old industrialized cities. Like other members of the 'rusty fellowship', the Ruhr has to struggle with archetypical problems of the former monofunctional manufacturing cities depending on mining and heavy engineering: a demographic decline, an aging population, high unemployment rates, an incipient brain drain and a lack of incentives to attract prosperous companies of the service sector, especially the 'new economy' [22, 23]. Before the CA is optimized, a brief overview of the implications of urban sprawl, the complexity of urban systems and important urban models in geography, as well as an introduction into the geosimulation with CA is given.

## 2. Geosimulation of urban sprawl

### 2.1. The complexity of urban systems

Urban sprawl is often treated as a specific of urbanization and urban growth. The classification of those processes is not homogenous and their definitions oscillate between the pure expansion of impervious surfaces and the distribution of urban activities and life styles in addition [2, 4, 24]. The study focuses on the land consumption aspects of urban sprawl. That is the conversion from nonurban to urban land cover and land uses for the main part indicating a certain amount of newly built-up, impervious areas [25]. The natural impacts of urban sprawl are manifold and concern several natural spheres. Cities consist by more than 50% of impervious land cover [26]. The sealing of even fruitful and agricultural valuable soils leads to a loss of fertility, transformation, filtering and buffering services [27]. The infiltration is disturbed and the surface runoff can be increased by five times with a sealing fracture of more than 90% [28]. The ecological footprint of a street can be estimated by 2 km and that of a European agglomeration by 1000 times of its area [9]. For a long time, the natural increase of people and migration flows into the cities' core areas as well as the economic expansion of the industrial age constituted the most important factors responsible for the growth of German cities. During the last 50 years, however, the trends of land consumption and population growth have moved apart: While the German population has grown around one fifth, the amount of settlement and traffic areas has nearly doubled [4]. In the literature, the following aspects are mentioned regularly: economic wealth; separation of individuals; the persuasion of the wish for owning a house on greenfield sites; new supply types; demographic change; aging of population; parish-pump politics and dominance of the automobile [2, 29–34]. Those causes build the main underlying driving forces of urban sprawl in Germany. But how do they interact? Which direct impacts do they evoke and what are their spatial outcomes? Why do they induce the emergence of sprawling urban areas even in regions affected by demographic and economic shrinkage?

In order to measure, model and understand urban sprawl, one has to think of it as a kind of urban land-use change embedded in the global land system [35]. The changes of land use and land cover and their driving factors construct nonlinear, complex systems including irrational behavior of their human actors. Therefore, urban areas must be treated as open and dynamic systems in which macro level patterns are a result of behavioral driven processes of micro level [36, 37]. Urban systems exhibit characteristics of hysteresis so that future developments and changes of urban systems are not only influenced by the current environment but also by the past one [32, 38, 39]. The initial configuration of its states affects the future decision making of its actors. Exemplary, road expansions do not only improve the infrastructural development but also change the spatial pattern affecting the circulatory system of a region's economy and feeding back to road improvements [40]. Thus, the observation scales of time (1) and space (2) are fundamental elements for the analysis of urban systems: (1) Technological innovations or new policies are exogenous drivers and affect sprawling urban growth in a short term. Due to coevolutionary interaction between states and actors, they become endogenous and are affected by urban dynamics in a long term [32]. (2) While on an aggregate level, residential areas may be clustered resulting in a positive spatial autocorrelation, and on an individual level, a certain range of distance may be kept resulting in negative spatial autocorrelation [41, 42].

## 2.2. Geosimulation – Using the AI of cells

The new wave of urban simulation often goes under the name of geosimulation. They shifted the modelling paradigm from macrostatics to microdynamics, from aggregation to disaggregation, from homogeneity to heterogeneity and from equilibrium to disequilibrium in urban modeling. Mandl defines geosimulation as a simulation where the modeled system processes, compartments and applications are spatially characterized and exhibit spatial relations [43]. Benenson and Torrens further comprehend geosimulation as “catch title that can be used to represent a very recent wave of research in geography ... [which] is concerned with the design and construction of object-based high resolution spatial models ... to explore ideas and hypotheses about how spatial systems operate [12]”. The implementation of AI methods for urban simulation purposes was heavily influenced by technical progresses in terms of computation, land-use data acquisition, geographic information systems (GISs), complexity studies, as well as the development and use of AI approaches in natural and social sciences outside of geography [17, 44]. Indeed, “geographers arrived somewhat late to the party” of addressing individual behavior alteration as design strategy of simulation applications [45]. The displacement of equations by code offered the possibility to explain the pattern and processes of complex open system dynamics on multiple organizational levels theoretically and experimentally [46]. Hence, the research object of urban sprawl could leave the suggestions of order, stability, linearity and rationality rooted in general system theory.

One of the most important and most implemented geosimulation techniques in terms of urban systems simulation is CA. The invention of CA is attributed to the mathematicians von Neumann (1951) and Ulam (1952), and “one can say that the ‘cellular’ comes from Ulam and the ‘automata’ comes from von Neumann” [47–49]. The final breakthrough of CA came with

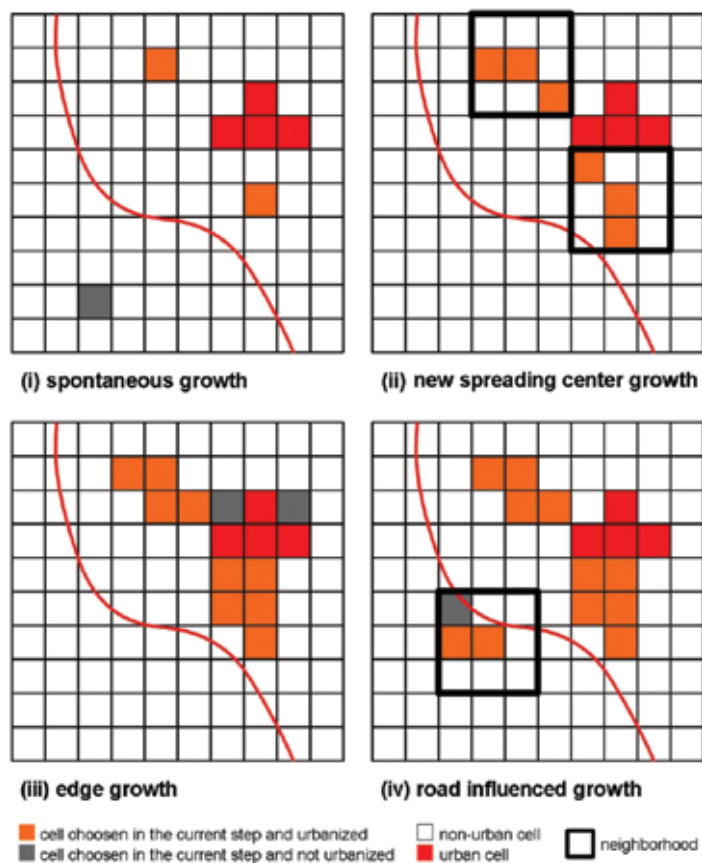
John Conway's 'The Game of Life' in 1970. Urban CA is often defined by (1) a raster lattice representing the spatial context, (2) a set of states associating a cell with a certain land-use type, (3) neighborhoods influencing the spatial configuration and (4) transition rules regulating the conversion of a cell state with every (5) time step [12]. The gridded two-dimensional character of a CA environment makes it well suited for the simulation of urban land-use and land-cover conversion. Here, the most popular CA modeling urban growth was developed by the authors mentioned in references [48–55]. The progress of CA implementations profited heavily by innovations in remote sensing techniques. The world's radiances are recorded from a bird's eye view and stored regularly pixel-by-pixel. Hence, the world's surface is represented in a two-dimensional raster lattice facilitating a paradigm of modeling from the pixel. Natural or administrative borders are completely neglected so that land-use and land-cover patterns become the only things that matter. While maps always depend on a more or less subjective semiotics and imply reduced content, remotely sensed images provide the unmediated biophysical context of coupled human–environment systems [56]: By classifying a data set of radiances with the help of specific spectral characteristics, a continuous spatial texture is turned into a discrete spatial pattern. Accordingly, a classified time series of LANDSAT data of 1975, 1984, 2001 and 2005 is applied for his study. The data sets were classified using a hybrid approach of supervised classification algorithms and knowledge-based decision trees. The resultant classification simply identifies 'urban' and 'nonurban' areas, where an "urban" area is defined as having a surface imperviousness of a minimum of 25%. A validation analysis of the classification documented an accuracy of >85%. In order to balance the spatial resolution and the spatial extent of the Ruhr, a grid resolution of 100 m was used. This classification procedure is described in detail in references [21, 61]. For the calibration of SLEUTH, the 1984 data comprise the base year and the 2001 data constitute the reference year. For the validation of SLEUTH, the 1975 data serve as the base year and the 2005 data constitute the reference year. Finally, the urban growth detected in the classified LANDSAT data between 1984 and 2001 is used to train the SVM model.

### 3. Modelling urban growth with an optimized cellular automaton

#### 3.1. SLEUTH – An urban cellular automaton

Clarke's urban growth model (UGM), mostly identified as SLEUTH, understands urbanization as a diffusion process where complex urban patterns spread as a whole. It has been applied in several urban growth studies all over the world [19] and offers many set screws for an enhancement of its performance. SLEUTH is an acronym of its initial input factors, which are slope, land-use, exclusion, transport and hillshade. The base data consist of the urban land-use configuration and the mandatory slope and transportation layer. The exclusion layer is optional, but recommended because it prevents the location of urban cells in, for example, conservation areas or water bodies. Additionally, the exclusion layer can be combined with a probability map to enhance the simulation performance. Five growth coefficients (dispersion, breed, spread, slope and road gravity) define the four growth rules of SLEUTH: spontaneous growth, representing the random emergence of new urban areas;

new spreading center growth; edge growth depicting extensive urban sprawl and road-influenced growth (**Figure 2**). The last one is SLEUTH-specific and relocates a temporary cell along the road to its final position. Space and time are treated discretely in the CA. Thus, one growth cycle represents 1 year of urban growth and consists of the four aforementioned successive growth simulations. Each selected new urban cell is tested against the local slope and exclusion information as well as a random value during each growth cycle. One model step consists of all growth cycles (years) between the starting and the end date of the calibration phase.



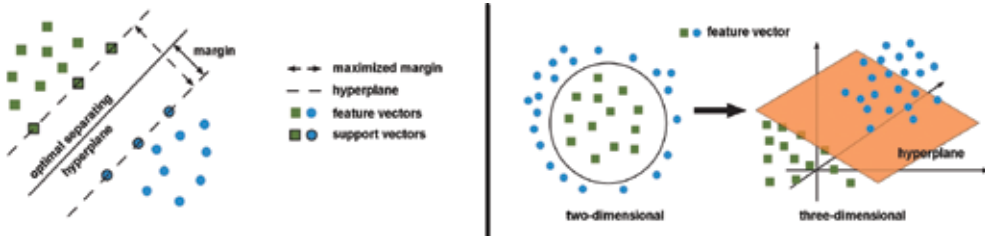
**Figure 2.** Growth types of a growth cycle in SLEUTH.

The calibration process of SLEUTH is based on the brute-force method. Every parameter combination of the particular growth coefficients between values of 0 and 100 is tested until the optimal balance is assessed. Every modeling process for each parameter combination is run several times by using Monte-Carlo (MC) iterations. Goetzke modified SLEUTH and implemented it into XULU (eXtendable Unified Land-use Modeling Platform), a modeling environment developed at the University of Bonn [57, 58]. The standard calibration evaluation

method of SLEUTH has been replaced by the multiple resolution validation (MRV) [59]. The MRV procedure compares an observed simulated map with a validation map at different spatial resolutions. High-resolution maps are weighted more than maps at lower resolution. The basic idea behind MRV is to attenuate the impact of localization errors by extending the conventional cell-by-cell comparison and considering the similarity of the entire neighborhood of a cell. Thus, the ‘fuzziness’ of categorical maps is addressed [58], and spatial patterns can be simulated quite precisely by accurately classifying only a relatively few map cells. Additionally, urban land-use calibration with MRV requires only two maps: a map of the initial calibration year and one of the final year.

### 3.2. Support Vector Machines

In their contemporary form, SVM was firstly formulated by Cortes and Vapnik [20]. Along with artificial neural networks and genetic programming they represent a new generation of machine learning algorithms. With their robust operation architecture and their valid classification results, SVM is a very popular classification technique for remote sensing data [60]. Fundamentally, SVM is a linear binary classifier that labels a sample of empirical data by constructing the optimal separating hyperplane (**Figure 3**, left). Traditional machine learning methods try to minimize the empirical training error, causing a tendency to overfit [61, 62]. They are strongly tailored to the training data, so extending them to additional data becomes difficult. According to the principles of structural risk minimization, SVM tries to minimize the upper bound of the expected generalization error through maximizing the margin between the separating hyperplane and the data. The margin concept is the key element in the SVM approach for it is an indicator of its generalization capability [63, 64].



**Figure 3.** An optimal hyperplane constructed by the support vectors separates the training data (left). To solve a non-linear classification problem, the input data is projected onto a higher-dimensional Hilbert space (right) [65].

The principal advantage of SVM is the option to transform the model in order to solve a nonlinear classification problem without any a priori knowledge. Using the so-called ‘kernel trick’ (Eqs 7–9), the input vectors are reprojected to a higher dimensional space in which they can be classified linearly (**Figure 3**, right).

Considering the scenario of a set of training vectors  $T$  belonging to two classes:

$$T = \{(x_i, y_i) : i = 1, \dots, n\} x_i \in \mathbb{R}^n, y_i \in \{-1, 1\} \quad (1)$$

where

$y_i$  = the class label (here urban growth and no urban growth)

$x_i$  = a given data point in the  $n$ -dimensional feature space.

In this scenario, the dimension of the input space is determined by the range of urban growth driving forces. A hyperplane needs to be found which separates the positive from the negative feature vectors. The 'separating hyperplane'  $H$  can be parameterized linearly by  $w$  and  $b$

$$H : \langle w, x \rangle + b = 0 \quad (2)$$

where  $w$ , element of  $\mathbb{R}^d$ , is a normal to  $H$  and  $b$ , element of  $\mathbb{R}$ , the bias. The classification problem can be formalized as the decision function

$$\text{sgn}(f_{(x)}) = \text{sgn}(\langle w, x \rangle + b) \quad (3)$$

For the linearly separable case, two hyperplanes  $H_+$  and  $H_-$  are constructed by the closest positive and negative, respectively, examples – the so-called support vectors:

$$H_+ : \langle w, x \rangle + b = 1 \text{ and } H_- : \langle w, x \rangle + b = -1 \quad (4)$$

Note that  $H_+$  and  $H_-$  are parallel because they have the same normal and no training points fall between them. Through the perpendicular distances from the origin of  $H_+$  and  $H_-$  it can be shown that the distance between the optimal separating hyperplanes  $H_+$  and  $H_-$ , respectively,  $H_+$  and  $H$ , is  $1/\|w\|$  where  $\|w\|$  is the Euclidean norm of  $w$ . So the margin between  $H_+$  and  $H_-$  is  $2/\|w\|$ . The optimal separating hyperplane can be found where the margin between  $H_+$  and  $H_-$  is the largest. Hence  $\|w\|$  has to be minimized. The formulation of the constrained optimization problem is

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{subject to } & y_i (\langle w, x_i \rangle + b) - 1 \geq 0 \text{ for } i = 1, \dots, n \end{aligned} \quad (5)$$

The constant  $C$  is called penalty parameter and  $\xi_i$  is a slack variable representing the error in the classification. The first part of the objective function tries to maximize the margin between the two classes and the second part minimizes the classification error. The optimization problem is solved by formulating it in a dual form derived by constructing a Lagrange function according to the Karush-Kuhn-Tucker optimality condition [63]. The resulting classification rule is

$$sgn(f_{(x)}) = sgn \sum_{i=1}^n \alpha_i y_i (\langle x_i, x \rangle + b) \quad (6)$$

where  $\alpha_i$  values are the corresponding Lagrange multipliers to  $x_i$  and  $b$  is the constant. The support vectors are all  $x_i$  where  $\alpha_i \neq 0$ . If the classification problem is not separable linearly, then the decision function cannot be solved simply with the separation approach described above. The data set must be transferred or projected respectively into a higher dimension: the Hilbert space. This process extends the methods of vector algebra from two-or three-dimensional spaces to spaces depicting any finite or infinite number of dimensions (**Figure 3**). By using the function  $\phi$  with  $d1 < d2$ , the number of possible linear separations is increased:

$$\mathbb{R}^{d1} \rightarrow \mathbb{R}^{d2}, x \rightarrow \phi(x) \quad (7)$$

SVMs are well suited for this operation since the training data  $x_i$  emerge only in scalar products in terms of the optimization problem. The scalar product  $x_i, x$  is calculated in the higher dimensional space  $\phi(x_i), \phi(x)$ . The transfer is performed with the use of a kernel function  $k$  according to Mercer's theorem [65].

$$k_{(x_i, x)} = \langle \phi_{(x_i)}, \phi_{(x)} \rangle \quad (8)$$

The Gaussian radial basis kernel function is a reasonable first choice ([64, 71]):

$$k_{(x_i, x)} = e^{-\gamma \|x - x_i\|^2} \quad (9)$$

The parameter  $\gamma$  defines the width of the Gaussian kernel function. After the kernel trick the decision function becomes

$$sgnf_{(x)} = sgn \sum_{i=1}^n \alpha_i y_i k(x_i, x + b) \quad (10)$$

Instead of predicting the label directly, the class probability is calculated (Eq 11) which delivers the basis for the probability maps of urban growth. Platt approximates the probabilities for binary SVMs using a sigmoid function

$$P(y=1 | x) = \frac{1}{1 + e^{A + f_{(x)} B}} \quad (11)$$

where  $A$  and  $B$  are parameters estimated by minimizing the negative log-likelihood function [67].

### 3.3. Optimizing SLEUTH

In this study, SVM is applied to raster layer stack consisting of different geophysical, socioeconomic as well as demographic driving factors of urban growth in order to optimize the CA SLEUTH. The selection is based on recent empirical studies dealing with urban sprawl in Central Europe [2, 5, 24]. Most studies of SVM in the context of urban growth modeling employ ordinary distance variables to represent proximity [66, 68]. In some cases, varying accessibility has a significant effect on the forces driving land use decisions. Hence, the effects of accessibility to markets or important infrastructure facilities are measured by weighting distances with a road variable that was derived using a categorized road network data set. Demographic and socioeconomic data that are exclusively available at the district level have been disaggregated to dasymetric maps [69]. For the other socioeconomic variables that are not related to population, statistical data were projected to the center points of each district and inverse distance weighting was used for interpolation. To minimize spatial autocorrelation effects, a stratified sampling method was applied [70]. This procedure generated a training data set containing 4000 image pixels for each urban growth and no urban growth class, with a 1 km minimum separation distance between equal pixels. For the construction of the SVM model one can use the software tool imageSVM® implemented in the EnMAP Toolbox®. Originally, imageSVM was created for solving classification problems in the context of multi- and hyperspectral satellite imagery [71]. The output of a SVM classification with imageSVM is not only a classified binary image but also a probability image based on the principles of (Eq 11). The crucial step for constructing a SVM model is determining optimal parameter settings, including appropriate values for the penalty parameter  $C$  (Eq 5) and the kernel parameter  $\gamma$  defining the width of the RBF kernel (Eq 9). An effective method for balancing the accuracy results of ‘known’ training data with ‘unknown’ testing data is the  $n$ -fold cross validation procedure [70]. According to the ‘curse of dimensionality’ and the Hughes’ phenomenon, which describes the degradation of the classifier performance when increasing the number of features, it is additionally advisable to select the optimal feature combination [71, 72]. A common element of SVM feature selection is a forward feature selection, which initially trains each feature of the input feature set. The best performing feature is selected and the remaining features are used for training in combination with the initially selected one. The procedure is repeated until all features have been selected.

The driving forces (**Table 1**) form the feature space (Eq 2) and build the base for training the 1984–2001 SVM urban growth model. The rank of a variable within the SVM feature selection (**Table 1**) is a reliable indicator for assessing the influence of the urban growth driving forces that were selected as model input parameters [70]. Following several subsequent feature selections, nine variables were eliminated. It can be stated that the characteristic attributes of the distance-related variables [73] and of the number of jobs are more suitable for constructing the SVM model than other socioeconomic or demographic variables. The distance to the next railway station in particular seems to be a very good indicator for a possible urbanization. This

might be a link to the industrial past traversing the urban area by a freight transport network. Beside elevation – and slope, which already is a part of SLEUTH – no other geophysical variable is usable for the selection of areas suitable for urban growth with SVM. The parameters “Jobs” related to the labor market and “NetDwellArea” variable which is a descriptor of living conditions are other important urban growth drivers.

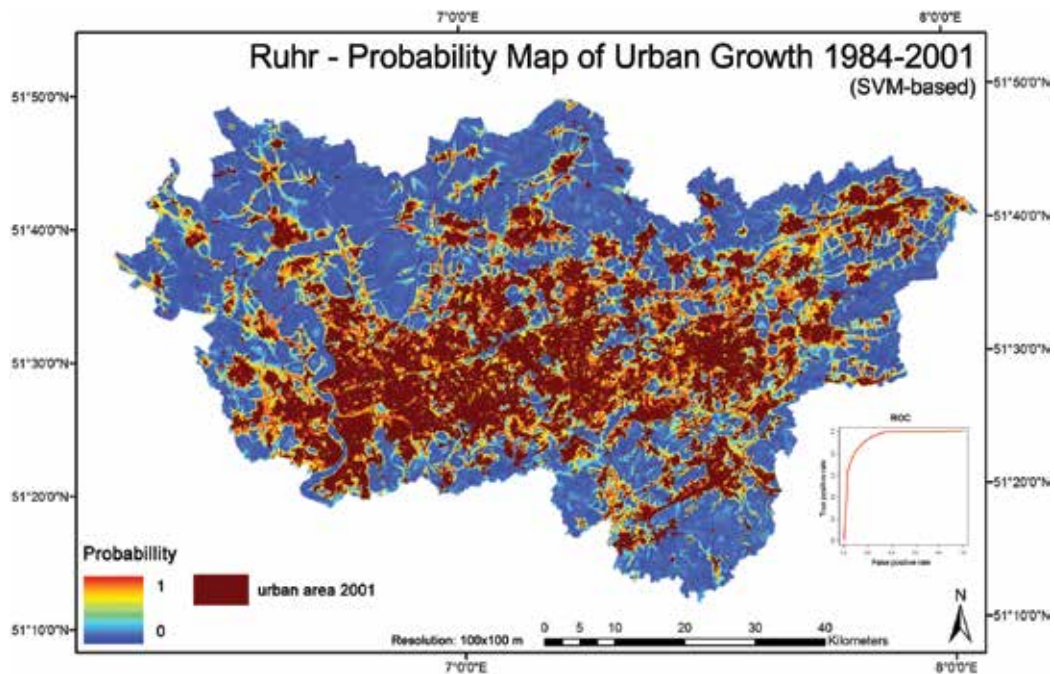
Name	Description	Rank*
Distance-related variables		
Dist Airport	Cost-weighted distance (CWD) to next international airport	5
Dist City	CWD to next city >25.000 inhabitants	3
DistHighway	CWD to next highway exit	2
Dist Railway	CWD to next railway station	1
Dist River	Euclidian distance to next river	6
Highway Buffer	500 m buffer to highways	n.i. <sup>x</sup>
Geophysical variables		
Elevation	Elevation above sea level (m)	11
Soil depth°	Vertical extent of soil layer (cm)	n.i.
Soil type°	Soil type defined by grain size (nominal)	n.i.
Soil quality°	Agricultural appropriateness (from [temporary] ‘not usable’ to ‘very good agricultural location’)	n.i.
Waterlogging°	Waterlogging type (from ‘low’ to ‘very high’)	n.i.
Water table	Depth of complete water saturation below ground (cm)	n.i.
Socioeconomic variables		
Income	Inverse distance-weighted (IDW) average income per month in district 1991	n.i.
Jobs	IDW number of jobs 1991	4
Land Price	IDW land value 1990	7
NetDwellArea	IDW per capita net dwelling area 1990	8
Unemployment	IDW unemployed per population 1991	9
Demographic variables		
Cars	Number of cars in district; density function (10 km kernel) DF	n.i.
Migration25–50	Difference between in- and out-migration per settlement of the group aged 25 to 50	n.i.
PopDens	Population density 1984; DF	10

\*Rank according to the forward feature selection.

°Not included.

<sup>x</sup>Dummy coded.

**Table 1.** Variables selected for SVM model.



**Figure 4.** SVM probability map of urban growth from 1984 to 2001.

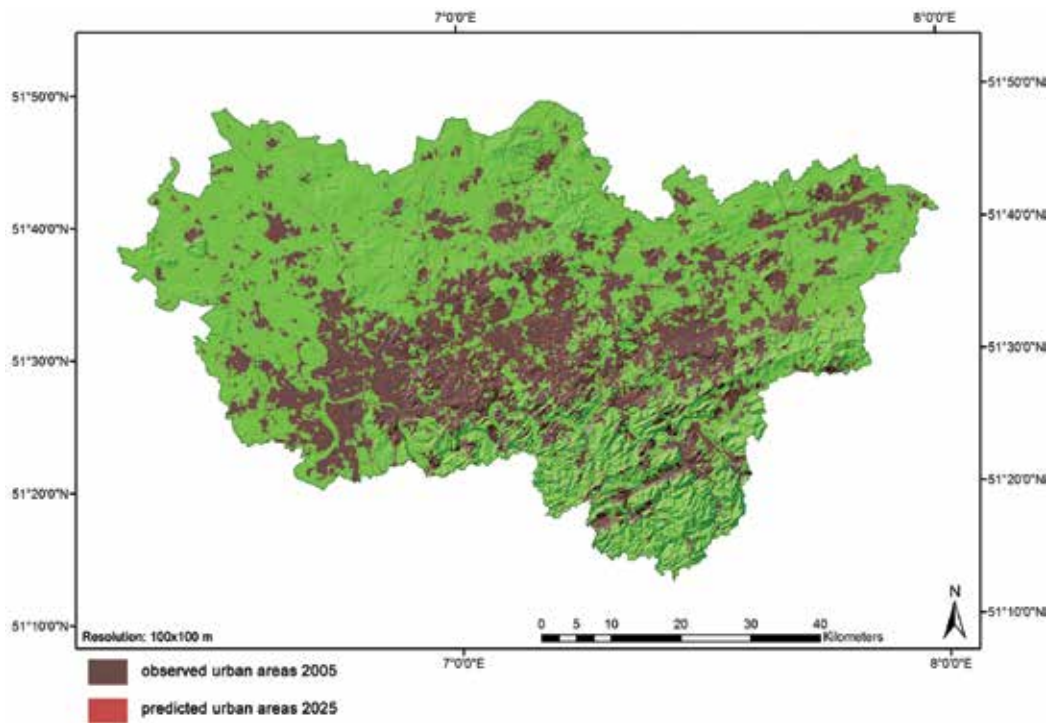
The output of the imageSVM® classification is not only a classified land-cover image but also an image containing the probabilities of each pixel for 'urban growth' and 'nonurban growth' [66]. The receiving operator characteristic (ROC) is an approved index for the accuracy assessment of binary categorical probability estimations [74]. The ROC divides the probability outcomes into percentile groups from high to low probability and compares the individual probability groups with the cumulative real values. The ROC only considers the positive values estimated by the model; in our case all urban growth cells. To define the ROC, true positive and false positive rates are plotted for every percentile group. The result is a curve where the area under the curve (AUC) is the measure that represents the ROC statistic. If a model acts randomly, then the curve will be a line through the origin with a slope of 1 and an AUC of 0.5. If a model acts perfectly, then the AUC is 1. **Figure 4** shows the ROC of the SVM model as well as the created probability map and the observed urban growth between 1984 and 2001. The curve of the SVM model clearly reaches a stable level already at low percentile groups. The resulting AUC values confirm the visual impression. The SVM model achieves a value of 0.94 – an outstanding performance for the AUC.

### 3.4. Calibration and validation of SLEUTH-SVM

Together with the SVM-based allocation probability map of urban growth, SLEUTH-SVM is calibrated using the produced urban land-cover maps 1984 and 2001 as the start and the testing year. For the validation of the model, the predicted urban growth is compared with the

observed urban growth of 2005, starting the simulation in 1975. It is clearly distinguished between the data sets used for calibration, and the data set used for validation of the model. The SVM probability map based on various driving forces of urban growth is combined with the exclusion layer of SLEUTH. In total, the three different versions of SLEUTH are compiled:

- SLEUTH: without exclusion layer.
- SLEUTH-AR: exclusion layer (areas where urban growth is restricted such as water bodies, natural reserves, etc.).
- SLEUTH-SVM: exclusion layer and probabilities for urban growth based on the SVM analysis.



**Figure 5.** Agreement results of SLEUTH-SVM: comparison map of observed and predicted urban areas 2005.

The validation of complex system models such as CA not only includes careful error estimation but also addresses the uncertainty of the model results [75]. Unwin defines uncertainty in the context of spatial simulations as a measure of doubt and distrust in results which can be seen as a certain kind of vagueness due to stochastic variability [76]. Aerts et al. [77] demonstrated that a detailed examination of the outcomes after 100 MC iterations is a valid procedure for assessing the uncertainty of CA like SLEUTH. The resulting map shows how often a certain cell has been depicted by the model to be urbanized. In order to transform it into a binary land-use map, a probability of 33% as cut-off value is used which was elaborated via a standard

histogram frequency method [78, 79]. With this value the best balance between location and quantification performance could be achieved.

**Figure 5** contains the comparison map of the predicted and the observed urban growth for 2005. It is important to emphasize that SLEUTH is a purely growth-oriented model. The CA exclusively models the transformation from nonurban to urban cell states. There is no ability to model in a spatially explicit manner the reverse process of urban contraction due to demolition or removal of sealed surfaces. In the near future, however, regional planning will have to deal with this process of extensive urban demolition (or urban perforation). For the 1975–2005 study period, spatial urban growth occurred concurrently with a contracting population [34]. There are only few areas where the simulation acts ‘false-positively’, meaning the model predicted urban growth where no growth has occurred [76]. These are nearly always located in open-spaced inner city areas or clustered along recreational parks of the river Emscher between Bottrop and Essen. In contrast, there are more areas where the model simulates ‘false-negatively’, which is predicting persistence where urban areas have spread in reality. Admittedly, it is virtually impossible to allocate greenfield development with industrial estates or the emergence of new traffic areas with a spatial certainty of 100 m without any local planning knowledge. Indeed, the urban areas in the district of Ennepe-Ruhr should have been actually captured. It seems that the slope layer of SLEUTH suppressed urbanization in this hilly region during the simulation run. However, in total, 108,220 ha of built-up land were predicted in the Ruhr for 2005. This is a growth of nearly 14% since 1975. In comparison, the observed urban growth rate is 39%. The underestimation of the quantity of change is specific for spatially explicit land-use change models [59, 80, 81]. For this reason, the model is validated regarding its overall simulation performance (overall agreement) in comparison with randomness (Cohen’s Kappa), the quantity estimations of urban growth ( $\kappa_{\text{histo}}$ ), the allocation ability of urban growth ( $\kappa_{\text{loc}}$ ), as well as the fuzziness of urban growth ( $F_i$ ) (**Table 2**).

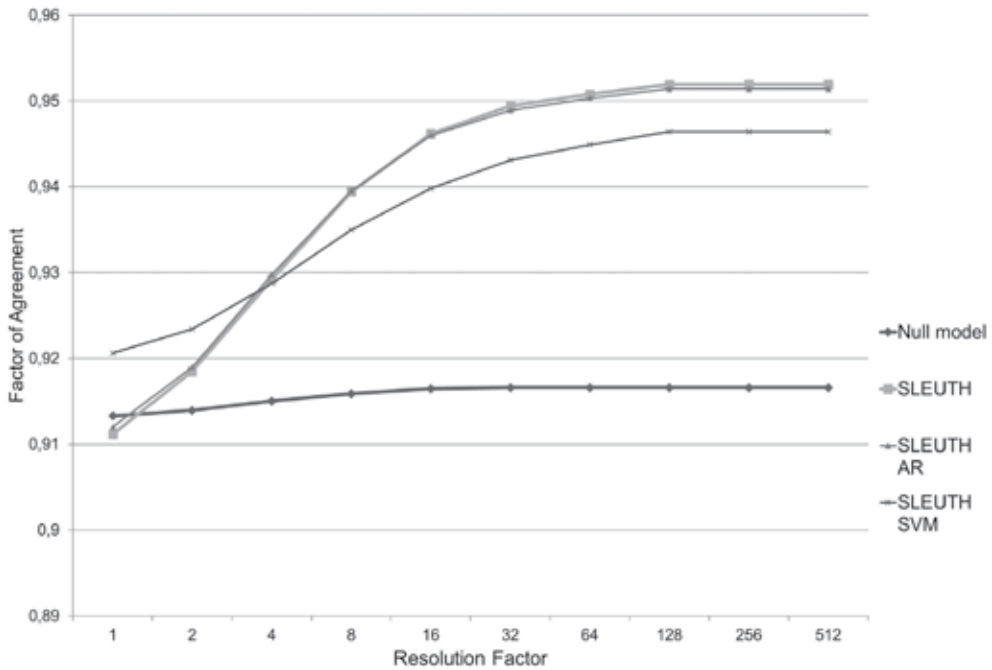
	SLEUTH	SLEUTH-AR	SLEUTH-SVM
New urban cells (ha)	110,668	108,220	110,445
Overall Agreement	0.911	0.912	0.914
Kappa	0.777	0.779	0.782
$F_i^*$	0.923	0.924	0.926
$\kappa_{\text{loc}}$	0.884	0.887	0.911
$\kappa_{\text{histo}}$	0.879	0.878	0.859

\* $F_i$  is the mean factor of agreement over all resolutions of the MRV.

**Table 2.** Validation results 1975–2005 of SLEUTH.

The achieved results are on a very good level. They show that the optimized CA SLEUTH-SVM outperforms the SLEUTH model without any exclusion information as well as the SLEUTH model exclusively containing the restriction areas. The lower quantification per-

formance is due to the simulation algorithm of SLEUTH: if SLEUTH is run without a probability map or an exclusion layer, then nearly every cell has the same probability of urbanization. Utilizing probability maps increase the possibility that a particular cell is defined as not suitable for urbanization and the resulting growth rate decreases. In several urban land-use change studies, a low signal of urban growth in comparison to a very high signal of persistent nonurban cells was discussed [59, 81].  $\kappa_{loc}$  is a suitable measurement to assess the allocation accuracy of a land-use model. Additionally, the model's ability of allocating newly urbanized cells should be tested. The null model comparison is a rational choice. A null model is a map containing the initial land-use pattern; it can therefore also be thought of as a pure persistence-map. Thus, null models regularly achieve better results at high resolutions than the actual land-use model; no change predicted means there are no allocation errors of urbanized areas. At a certain spatial resolution, the quantity error of a null model increases so that the predicted map achieves better results. The resolution level where the agreement factor of the land-use model outperforms the null model for the first time is called 'null resolution'.

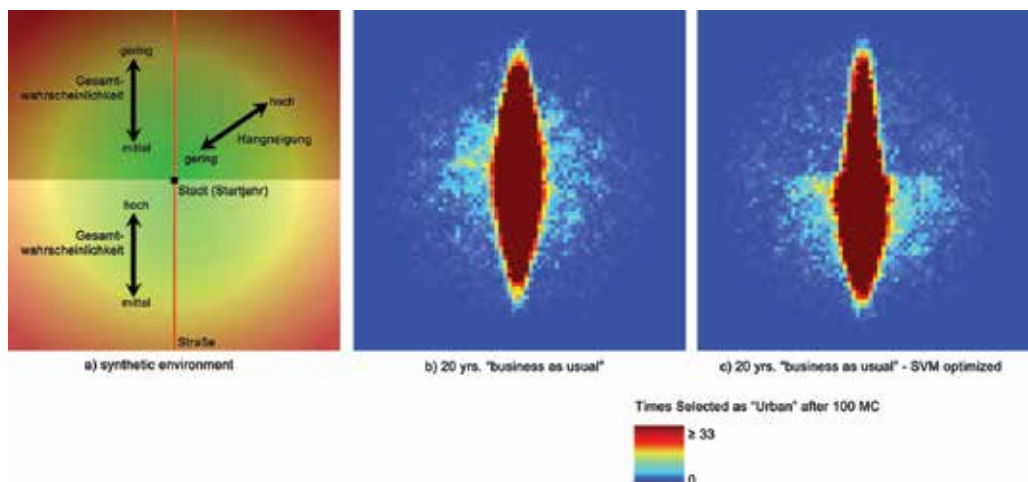


**Figure 6.** MRV curves of the applied models.

The MRV results show a consistently excellent level of overall accuracy. This can be attributed to the ability of all three models to almost perfectly predict the location and the quantity of nonurban growth cell states. SLEUTH-SVM is the only model exceeding the level of the null model already at the resolution level 1 of 100 m (**Figure 6**). The agreement curve of the null model forms a straight line, reflecting the low signal of urban growth in comparison to the very high signal of persistent nonurban growth cells. Again, the optimized version of SLEUTH

shows a higher agreement between the predicted and the observed urban growth on the level of high resolutions up to 400 m than SLEUTH and SLEUTH-AR. As soon the resolution gets coarser, the impact of allocation errors decreases and the performance of SLEUTH-SVM stagnates.

The question of how the SVM-probability map influences the spatial extension of the Ruhr's urban growth will be analyzed with the concept of urban DNA [82]. Analogous to the biological DNA, it postulates fundamental elements that are common to each urban area and determine their future growth pattern [83]. Gazulis and Clarke apply the concept to an abstract space representation mimicking the variable input of SLEUTH [84]. Hence, this grid image reflects a kind of digital petri dish consisting of an urban area, a slope layer, transport information as well as exclusion areas (**Figure 7**). The urban input is just a single urban cell in the middle of the image whereas all other cells are defined as nonurban. The slope has a minimum value of 0% and increases concentric-radially to a maximum value which is equal to the maximum slope value to be found in the Ruhr (70%). The transport network is represented by a single road crossing the center of the image from north to south. In this study, the exclusion layer is exchanged by the SVM probability map of urban growth of the Ruhr. For estimating the spatial impact of the SVM optimization procedure, the particular probability map is allocated with a linear transition from high to low probabilities equivalent to their particular value range. In the south of the urban center, the probabilities decrease from 1 to the particular medium value. This medium level continues northwards from the urban center and decreases to zero. Thus, the maps are divided lengthwise from west to east.



**Figure 7.** Urban simulation in a digital petri dish consisting of the fundamental elements of the Ruhr's urban area.

The models SLEUTH and SLEUTH-SVM are run with the calibrated growth coefficients and 100 MC iterations. Hence, one can observe the allocation behavior of the CA under the SVM-defined conditions of the Ruhr's urban areas. The border of high and medium probabilities of urban growth is distinct in the SVM-optimized version of SLEUTH. The CA is clearly guided

whereas the regular version allocates the cells more randomly in the surroundings of already built-up areas. In the northern part of the synthetic region, the urban pattern is slim while in the southern part it broadens. All in all, the DNA of the Ruhr's urban areas reveals a tendency to edge growth and a high influence of the road network. The SVM-based probability optimizes the allocation performance of the urban CA SLEUTH.

## 4. Conclusion and outlook

The aim of this study was to optimize the CA SLEUTH by using SVM and to assess its performance in comparison to its standard configuration. A SVM-based probability map including the impact of driving forces on the allocation of new urban areas was combined with the exclusion layer of SLEUTH. Hence, the model could be guided and its stochastic variability regarding the emergence of urban cells depressed. The application of SVM additionally delivered insights into the most important factors driving the local urbanization suitability. Thus, the CA provided a theoretical foundation. The importance of distance-related variables over socioeconomic or demographic variables in the SVM model was clear. Geophysical variables were not useful to select areas suitable for urban growth with SVM. The models have been applied to the polycentric agglomeration of the Ruhr. The region's specific settlement structure is characterized by large urban areas solely divided through administrative borders. The scattered rural districts complete the image of a challenging study region in terms of organizational hierarchies, migration flows and heterogenic growth conditions. The accuracy of SLEUTH-SVM has been assessed regarding the overall agreement, the fuzziness, the ability to allocate new urban cells, the performance in comparison with the randomness as well as the quantity and the allocation ability of urban growth. The calibration and the validation of the model have been separated carefully. The validation indices for SLEUTH-SVM showed good values around 0.91 (kappa) and 0.93 (MRV). As a reliable result, it can be stated that the allocation performance of SLEUTH is optimized clearly when coupling it with a SVM-based probability map. Its spatial impacts are visualized with the concept of urban DNA and a digital petri dish. Hence, the generic growth elements of the Ruhr's urban area were uncovered.

As a next step, coupling different modeling techniques for prediction and process analyses of urban land-use and land-cover change should be pushed beyond the physical restriction of pixels. To incorporate the irrational human component impacting on all spatial and temporal levels between the household and the global scale, pixels must be coupled with people. This can, for instance, be done by extending the enhanced version of SLEUTH with a multiagent system. That way, one can study emergence phenomena resulting from complex behavioral interaction processes on the micro scale. Hence, the simulation of urban change could be extended from analyzing growth processes to the estimation of urban decline. Coupling actors with factors would be a chance to overcome formal, differential equations. Therefore, the present ride on the surface of an ocean of driving forces, pressures, states, impacts and responses would be optimized into diving right into it.

## Author details

Andreas Rienow

Address all correspondence to: [a.rienow@geographie.uni-bonn.de](mailto:a.rienow@geographie.uni-bonn.de)

Remote Sensing Research Group, Department of Geography, University of Bonn, Bonn, Germany

## References

- [1] Wassmer RW. An economist's perspective on urban sprawl, Part 1 – Defining excessive decentralization in California and other western states. Sacramento: California Senate Office of Research; 2002. 33 p.
- [2] EEA. Urban sprawl in Europe: The ignored challenge. Luxembourg: Office for Official Publications of the European Communities; 2006. 60 p.
- [3] Kasanko M, Barredo JI, Lavalle C, McCormick KN, Demicheli L, Sagris V. Are European cities becoming dispersed? A comparative analysis of 15 European urban areas. *Landsc Urban Plan.* 2006;77(1–2):111–130. DOI:10.1016/j.landurbplan.2005.02.003.
- [4] Hoymann J, Dosch F, Beckmann G. Trends der Siedlungsflächenentwicklung. Status quo und Projektion 2030. Bonn: BBSR-Analysen KOMPAKT; 2012. 20 p.
- [5] Siedentop S, Fina S. Monitoring urban sprawl in Germany: Towards a GIS-based measurement and assessment approach. *J Land Use Sci.* 2010;5(2):73–104. DOI: 10.1080/1747423X.2010.481075.
- [6] GLP. Science plan and implementation strategy. Stockholm: IGBP Secretaria; 2005. 74 p.
- [7] IPCC. Renewable energy sources and climate change mitigation: Special report of the Intergovernmental Panel on Climate Change. Cambridge: Cambridge University Press; 2012. 1088 p.
- [8] Millenium Ecosystem Assessment. Ecosystems and human well-being: Synthesis. Washington: Island Press; 2005. 155 p.
- [9] Ramankutty N, Graumlich L, Achard F, Alves D, Chhabra A, DeFries RS. Global land-cover change: Recent progress, remaining challenges. In: Lambin EF, Geist H, editors. *Land-Use and Land-Cover Change*. Springer Berlin Heidelberg; 2006, p. 9–39.
- [10] UNEP. Global environment outlook: GEO4: Environment for development. Nairobi: United Nations Environment Programme; 2012, 572 p.

- [11] Batty M. Fifty years of urban modeling: macro-statics to micro-dynamics. In: Albeverio S, Andrey D, Giordano P, Vancheri A, editors. *The dynamics of complex urban systems*. Springer, London; 2008, p. 1–20.
- [12] Benenson I, Kharbush V, Xie Y, Brown, DG. Geographic automata: from paradigm to software and back to paradigm. *Proc 8th Int Conf GeoComputation Univ Mich U S Am* July–3 August 2005. University of Michigan, 2005, p. 1–12.
- [13] Silva E, Wu N. Surveying models in urban land studies. *J Plan Lit* 2012;27:1–14. DOI: 10.1177/0885412211430477.
- [14] Steven DP, Hoffman M, Parker DC, Manson SM, Manson SM, Janssen MA. Multi-agent systems for the simulation of land-use and land-cover change: A review. *Ann Assoc Am Geogr* 2002;93:314–337. DOI: 10.1111/1467-8306.9302004.
- [15] Schwarz N, Haase D, Seppelt R. Omnipresent sprawl? A review of urban simulation models with respect to urban shrinkage. *Environ Plan B Plan Des*. 2010;37(2):265–283. DOI: 10.1068/b35087.
- [16] Matthews RB, Gilbert NG, Roach A, Polhill JG, Gotts NM. Agent-based land-use models: A review of applications. *Landsc Ecol*. 2007;22(10):1447–1459. DOI: 10.1007/s10980-007-9135-1.
- [17] Couclelis H. Computational human geography. In: Kitchin R, Thrift N, editors. *International Encyclopedia of Human Geography*. Oxford: Elsevier; 2009. p. 245–250.
- [18] Clarke KC. Mapping and modelling land use change: an application of the SLEUTH model. In: Pettit C, Cartwright W, Bishop I, Lowell K, Pullar D, Duncan D, editors. *Landscape analysis and visualisation*. Springer Berlin Heidelberg; 2008. p. 353–366.
- [19] Chaudhuri G, Clarke KC. The SLEUTH land use change model: A review. *Int J Environ Resour Res*. 2013;1(1):88–104.
- [20] Cortes C, Vapnik V. Support-vector networks. *Mach Learn*. 1995;20(3):273–297.
- [21] Goetzke R, Over M, Braun M. A method to map land-use change and urban growth in North Rhine-Westphalia (Germany). *Proc 2nd Workshop EARSeL SIG Land Use Land Cover*, ZFL Bonn, 2006, p. 102–10.
- [22] Blotevogel HH. Gemeindetypisierung Nordrhein-Westfalens nach demographischen Merkmalen. In: Danielzyk R, Kilper H, editors. *Räumliche Konsequenzen des Demographischen Wandels Teil 8: Demographischer Wandel in ausgewählten Regionaltypen Nordrhein-Westfalens – Herausforderungen und Chancen für die regionale Politik*. Hannover: Akademie für Raumforschung und Landesplanung – ARL; 2006. p. 17–33.
- [23] Couch C, Karecha J, Nuissl H, Rink D. Decline and sprawl: An evolving type of urban development – Observed in Liverpool and Leipzig. *Eur Plan Stud*. 2005;13(1):117–136. DOI: 10.1080/0965431042000312433.

- [24] Antrop M. Landscape change and the urbanization process in Europe. *Landsc Urban Plan.* 15. 2004;67(1–4):9–26. DOI: 10.1016/S0169-2046(03)00026-4.
- [25] Ulmer F, Renn O, Ruther-Mehlis A, Jany A, Lilienthal M, Malburg-Graf B. Erfolgsfaktoren zur Reduzierung des Flächenverbrauchs in Deutschland – Evaluation der Ratsempfehlungen “Mehr Wert für die Fläche: das Ziel 30ha“. Eine Studie im Auftrag des Rates für Nachhaltige Entwicklung. Stuttgart: Rat für Nachhaltige Entwicklung; 2007. 66 p.
- [26] Hostert P. Advances in urban remote sensing: Examples from Berlin (Germany). In: Netzband DM, Stefanov DWL, Redman PC, editors. *Applied remote sensing for urban planning, governance and sustainability*. Berlin: Springer; 2007. p. 37–51.
- [27] Blum WEH. Flächenverbrauch und Auswirkungen auf die Ökologische Bodennutzung. In: Umweltbundesamt, editors. *Versiegelt Österreich? Der Flächenverbrauch und seine Eignung als Indikator für Umweltbeeinträchtigungen*, Wien, 15. März 2001, p. 74–79.
- [28] Arnold CL, Gibbons CJ. Impervious surface coverage: The emergence of a key environmental indicator. *J Am Plann Assoc.* 1996;62(2):243–258.
- [29] Mielke B, Münter A. Demographischer Wandel und Flächeninanspruchnahme. In: Danielzyk R, Meyer C, Grüber-Töpfer W, editors. *Demographischer Wandel in Nordrhein-Westfalen*. 2<sup>nd</sup> ed. Dortmund: Institut für Landes- und Stadtentwicklungsforschung und Bauwesen des Landes Nordrhein-Westfalen (ILS NRW); 2008. p. 58–64.
- [30] Sieverts T. Zwischenstadt: zwischen Ort und Welt, Raum und Zeit, Stadt und Land. 3<sup>rd</sup> ed. Berlin: Bertelsmann Fachzeitschriften; 2001, 191 p.
- [31] Kistenmacher H. Ursachen und räumliche Wirkungen der Suburbanisierung. In: ARL Arbeitsmaterial 276. 2001. p. 17–30.
- [32] Geist H, McConnell W, Lambin EF, Moran E, Alves D, Rudel T. Causes and trajectories of land-use/cover change. In: Lambin EF, Geist H, editors. *Land-use and land-cover change*. Berlin: Springer; 2006. p. 41–70.
- [33] Klijn JA. Driving forces behind landscape transformation in Europe, from a conceptual approach to policy options. In: Jongman RHG, editor. *The new dimensions of the European landscape*. Dordrecht; 2004. p. 201–218.
- [34] Siedentop S. Urban sprawl – verstehen, messen, steuern. Ansatzpunkte für ein empirisches Mess- und Evaluationskonzept der urbanen Siedlungsentwicklung. *DISP.* 2006;160:23–35.
- [35] Verburg PH, Eck JRR van, Nijs TCM de, Dijst MJ, Schot P. Determinants of land-use change patterns in the Netherlands. *Environ Plan B Plan Des.* 2004;31(1):125–150. DOI: 10.1068/b307.

- [36] Veldkamp A, Lambin EF. Predicting land-use change. *Agric Ecosyst Environ.* 2001;85(1):1–6. DOI: 10.1016/S0167-8809(01)00199-2.
- [37] Verburg PH. Simulating feedbacks in land use and land cover change models. *Landsc Ecol.* 2006;21(8):1171–1183. DOI: 10.1007/s10980-006-0029-4.
- [38] Alcamo J, Kok K, Busch G, Priess JA, Eickhout B, Rounsevell M. Searching for the future of land: Scenarios from the local to global scale. In: Lambin EF, Geist HJ, editors. *Land-use and land-cover change: Local processes and global impacts.* Berlin: Springer; 2006. p. 137–157.
- [39] Lambin EF, Geist HJ. Introduction: Local processes with global impacts. In: Lambin EF, Geist HJ, editors. *Land-use and land-cover Change: local processes and global impacts.* Berlin: Springer; 2006. p. 1–8.
- [40] Verburg PH. Modeling land-use and land-cover change. In: Lambin EF, Geist HJ, editors. *Land-use and land-cover change: Local processes and global impacts.* Berlin: Springer; 2006. p. 117–136.
- [41] Lesschen JP, Verburg PH, Staal SJ. Statistical methods for analysing the spatial dimension of changes in land use and farming systems. Wageningen: LUCC Report Series 7 (IGBP); 2005. 81 p.
- [42] Overmars KP, de Koning GHJ, Veldkamp A. Spatial autocorrelation in multi-scale land use models. *Ecol Model.* 2003;164(2–3):257–270. DOI: 10.1016/S0304-3800(03)00070-X.
- [43] Mandl P. Geo-Simulation-Experimentieren und Problemlösen mit GIS-Modellen. In: Strobl J, Blaschke T, Griesebner G, editors. *Beiträge zum AGIT-Symposium Salzburg.* Heidelberg: Wichmann; 2000. p. 345–356.
- [44] Miller EJ, Douglas Hunt J, Abraham JE, Salvini PA. Microsimulating urban systems. *Comput Environ Urban Syst.* 2004;28(1–2):9–44. DOI: 10.1016/S0198-9715(02)00044-3.
- [45] Benenson I, Torrens PM. *Geosimulation: Automata-based modeling of urban phenomena.* West Sussex: John Wiley & Sons; 2004. 287 p.
- [46] Bethell T. The search for artificial intelligence. *Am Spect.* 2006;39(6):26–35.
- [47] Rucker R. *Seek! Selected Nonfiction.* New York: Four Walls Eight Windows; 1999. 356 p.
- [48] Ulam S. Random processes and transformations. In: *Proceedings of the International Congress on Mathematics.* Cambridge: Cambridge University Press; 1952. p. 264–275.
- [49] von Neumann J. The general and logical theory of automata, *Cerebral Mechanisms in Behavior.* The Hixon Symposium, John Wiley & Sons, Inc., New York, NY; Chapman & Hall, Ltd., London, 1951, p. 1–31.

- [50] Barredo JI, Lavalle C, Kasanko M, Demicheli L, McCormick N. Sustainable urban and regional planning: The MOLAND activities on urban scenario modelling and forecast. Luxemburg: EC; 2003. 54 p.
- [51] Landis J. CUF, CUF II, and CURBA: A family of spatially explicit urban growth and land-use policy simulation models. In: Brail R, Klostermann R, editors. Planning support systems: integrating geographic information systems, models, and visualization tools. Redlands: Esri Press; 2001. p. 159–200.
- [52] Batty M, Xie Y. Possible urban automata. *Environ Plan B Plan Des*. 1997;24(2):175–192.
- [53] Clarke KC, Hoppen S, Gaydos L. A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area. *Environ Plan B Plan Des* 1997;24:247–262.
- [54] Hilferink M, Rietveld P. Land use scanner: An integrated GIS based model for long term projections of land use in urban and rural areas. *J Geogr Syst*. 1999;1(2):155–177.
- [55] Tobler W. Cellular geography. In: Gale S, Ollson G, editors. *Philosophy in geography*. Dordrecht: D. Reidel Publishing Company; 1975. p. 379–386.
- [56] White R, Engelen G. Cellular automata and fractal urban form: A cellular modelling approach to the evolution of urban land-use patterns. *Environ Plan A*. 1993;25(8):1175–1199.
- [57] Goetzke R. Entwicklung eines fernerkundungsgestützten Modellverbundes zur Simulation des urban-ruralen Landnutzungswandels in Nordrhein-Westfalen. Hamburg: Disserta Verlag; 2012. 296 p.
- [58] Schmitz M, Bode T, Thamm H-P, Cremers AB. XULU – A generic JAVA-based platform to simulate land use and land cover change (LUCC). MODSIM 2007 Int Congr Model Simul Model Simul Soc Aust N Z Dec 2007. 2007; p. 4–7.
- [59] Pontius RG, Boersma W, Castella J-C, Clarke K, Nijs T, Dietzel C. Comparing the input, output, and validation maps for several models of land change. *Ann Reg Sci*. 2008;42(1): 11–37. DOI: 10.1007/s00168-007-0138-2.
- [60] Costanza R, Maxwell T. Spatial ecosystem modelling using parallel processors. *Ecol Model*. 1991;58(1–4):159–183.
- [61] Schoettker B. Monitoring statewide urban development using multitemporal multi-sensoral satellite data covering a 40-year time span in north Rhine-Westphalia (Germany). *Proc SPIE 10th Int Symp Remote Sens 8-12th Sept 2003 Barc*. 2003; p. 252–61.
- [62] Mountrakis G, Im J, Ogole C. Support vector machines in remote sensing: A review. *ISPRS J Photogramm Remote Sens*. 2011;66(3):247–259. DOI: 10.1016/j.isprsjprs. 2010.11.001.
- [63] Vapnik V. Statistical learning theory. New York: John Wiley & Sons; 1998. 736 p.

- [64] Xie C. Support vector machines for land use change modeling. Calgary: UCGE Reports; 2006. 128 p.
- [65] Burges CJC. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov* 1998;2:121–167.
- [66] Huang B, Xie C, Tay R. Support vector machines for urban growth modeling. *GeoInformatica*. 2010;14(1):83–99. DOI: 10.1007/s10707-009-0077-4.
- [67] Platt JC. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv Large Margin Classif*. 1999;61–74.
- [68] Okwuashi O, McConchie J, Nwilo P, Eyo E. Stochastic GIS cellular automata for land use change simulation: Application of a kernel based model. *Proc 10th Int Conf GeoComputation Univ New South Wales Syd Aust* 30 Nov – 02 Dec 2009. 2009; p. 1–7.
- [69] Langford M, Unwin DJ. Generating and mapping population density surfaces within a geographical information system. *Cartogr J*. 1994;31(1):21–26.
- [70] Congalton RG. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sens Environ* 1991;37:35–46.
- [71] Waske B, van der Linden S, Benediktsson JA, Rabe A, Hostert P. Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. *IEEE Trans Geosci Remote Sens*. 2010;48(7):2880–2889.
- [72] Hsu C-W, Chang C-C, Lin C-J. A practical guide to support vector classification. Taipei: Department of Computer Science, National Taiwan University; 2010. 16 p.
- [73] Hughes GF. On the mean accuracy of statistical pattern recognizers. *IEEE Trans Information Theory*. 1968;14(1):55–63.
- [74] Nguyen MH, de la Torre F. Optimal feature selection for support vector machines. *Pattern Recognit*. 2010;43(3):584–591. DOI: 10.1016/j.patcog.2009.09.003.
- [75] Koomen E, Borsboom-van Beurden J. Land-use modelling in planning practice. Dordrecht: Springer; 2011. 214 p.
- [76] Pontius RG, Schneider LC. Land-cover change model validation by an ROC method for the Ipswich watershed, Massachusetts, USA *Agric Ecosyst Environ* 2001;85(1–3): 239–248. DOI: 10.1016/S0167-8809(01)00187-6.
- [77] Aerts JCJH, Clarke KC, Keuper AD. Testing popular visualization techniques for representing model uncertainty. *Cartogr Geogr Inf Sci*. 2003;30(3):249–261. DOI: 10.1559/152304003100011180.
- [78] Rafiee R, Mahiny AS, Khorasani N, Darvishsefat AA, Danekar A. Simulating urban growth in Mashad City, Iran through the SLEUTH model (UGM). *Cities*. 2009;26(1):19–26. DOI: 10.1016/j.cities.2008.11.005.

- [79] Wu X, Hu Y, He HS, Bu R, Onsted J, Xi F. Performance evaluation of the SLEUTH model in the Shenyang metropolitan area of northeastern China. *Environ Model Assess.* 2008;14(2):221–230. DOI: 10.1007/s10666-008-9154-6.
- [80] Pontius RG, Malizia NR. Effect of category aggregation on map comparison. In: Egenhofer MJ, Freska C, Miller HJ, editors. *GIScience 2004*. Springer, Berlin 2004, p. 251–268.
- [81] Pontius RG. Quantification error versus location error in comparison of categorical maps. *Photogramm Eng Remote Sens.* 2000;66(8):1011–1016.
- [82] Silva EA. The DNA of our regions: Artificial intelligence in regional planning. *Futures.* 2004;36(10):1077–1094. DOI: 10.1016/j.futures.2004.03.014.
- [83] Silva EA, Clarke KC. Complexity, emergence and cellular urban models: Lessons learned from applying SLEUTH—To two Portuguese metropolitan areas. *Eur Plan Stud.* 2005;13(1):93–115.
- [84] Gazulis N, Clarke KC. Exploring the DNA of our regions: Classification of outputs from the SLEUTH model. In: Yacoubi SE, Chopard B, Bandini S, editors. *Cellular Automata*. Berlin: Springer; 2006.:462–471.



---

# **Inverse Geometry Design of Radiative Enclosures Using Particle Swarm Optimization Algorithms**

---

Hong Qi, Shuang-Cheng Sun, Zhen-Zong He,  
Shi-Ting Ruan, Li-Ming Ruan and He-Ping Tan

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62351>

---

## **Abstract**

Three different Particle Swarm Optimization (PSO) algorithms—standard PSO, stochastic PSO (SPSO) and differential evolution PSO (DEPSO)—are applied to solve the inverse geometry design problems of radiative enclosures. The design purpose is to satisfy a uniform distribution of radiative heat flux on the designed surface. The design surface is discretized into a series of control points, the PSO algorithms are used to optimize the locations of these points and the Akima cubic interpolation is utilized to approximate the changing boundary shape. The retrieval results show that PSO algorithms can be successfully applied to solve inverse geometry design problems and SPSO achieves the best performance on computational time. The influences of the number of control points and the radiative properties of the media on the retrieval geometry design results are also investigated.

**Keywords:** Particle Swarm Optimization algorithm, inverse geometry design, radiative heat transfer, SPSO, DEPSO

---

## **1. Introduction**

Radiative heating devices are encountered in various industrial fields, such as industrial boilers, spacecraft, infrared reflecting ovens, metallurgical equipment, and so on and the design of radiative enclosure has a direct impact on security issues [1]. Inverse design technique is a new method in recent years, whose solving process is to establish an objective function according to the design requirements at first, and then optimize the objective function by some optimization methods, and achieve the pre-specified purpose finally. Inverse design

technique has the advantages of simple process, short design circle, good optimization results, etc., and it has got more and more attentions and applications.

Inverse design problems can be divided into two categories according to the design prerequisites [2]. One is inverse boundary design problems, in which the geometry shape of the radiative enclosures is fixed and the boundary conditions are need to be deigned [3–6]. The other is inverse geometry design problems, where the boundary conditions are predetermined and the geometry shape of the design surface needs to be designed [7, 8]. For the reason that the geometry shapes of the boundaries of radiative enclosures are different at each iteration and the grids in the computational domain must be re-meshed in every iterative calculation, the inverse geometry design of radiative enclosures is the most complex inverse radiative problem [2, 9].

During the last few decades, some inverse design techniques have been successfully used for solving inverse geometry design problems. Howell et al. [10, 11] proposed inverse design ideas and applied inverse Monte Carlo techniques, Tikhonov method, truncated singular value decomposition (TSVD), Modified TSVD (MTSVD), artificial neural networks (ANN) and conjugate gradient method (CGM) to solve the inverse design problem of a three-dimensional industrial furnace, which greatly improved the practical design of thermal and environmental systems. Franca and Howell [12] studied a transient inverse design problem that finding the optimal location of a heater on the top surface of a three-dimensional enclosure to produce a prescribed time-dependent temperature distribution on the bottom surface of the enclosure, the TSVD method is used to regularize the ill-conditioned system of linear equations the pre-specified temperature curve is obtained with an error of less than 1.0%. Tan et al. [13] applied meshless method to solve the coupled conductive and radiative heat transfer problem in heating devices, in which a series of nodes are used for discretizing the computational domain to overcome the tedious re-mesh works, and CGM is adopted to optimize the height of the adiabatic diffuse reflection surfaces and the geometry shape of the heating surface to satisfy the required total heat flux on the pre-appointed region of the low temperature heated surface. Sarvari and Mansouri [14] used CGM to minimize the objective function which is expressed as the sum of square residuals between estimated and desired heat fluxes on the design surface to satisfy the specified temperature and heat flux distributions. The radiative heat transfer problem in the two-dimensional irregular enclosure filled with gray participating media with uniform absorption coefficient is solved by discrete transfer method, the effect of optical depth and angular refinement on the inverse design results are also investigated.

However, it can be found that most of the above researches about inverse geometry design problems are solved based on the gradient-based methods. All these methods have the common disadvantages that the computational process of getting the gradient is complicated and the retrieval results are strongly depended on the initial guessed values. On the contrast, intelligent algorithms can abolish the complex calculations about gradient and randomly generate the potential solutions in the search space to overcome the drawback of depending on the initial value. In recent years, some intelligent algorithms have been successfully applied for solving inverse radiative problems, including PSO, genetic algorithm (GA), ant colony optimization (ACO), difference evolution (DE), fruit fly optimization algorithm (FOA), to

name a few [15–22]. Compared with conventional techniques, intelligent algorithms can obtain much more potential solutions at each iteration and all searches are executed in parallel, which greatly improved the computational efficiency, especially for solving some high dimensional problems. Intelligent algorithms also have been successfully used to solve inverse design problems. For example, Moparthy et al. [23] solved the coupled radiative and conductive heat transfer problem in one-dimensional planar system based on finite volume method (FVM) and lattice Boltzmann method (LBM) and applied GA to optimize the heater temperature to produce the desired heat flux and temperature distributions on the design surface. The retrieval results show that the temperature or heat flux of the heater surface has a significant effect on the design surface condition, the medium properties and the distance between the two surfaces. Amiri et al. [24] adopted modified discrete ordinate method to solve the radiative transfer equation (RTE), the micro genetic algorithm (MGA) is employed to optimize the objective function which is defined as the sum of the square of the difference between estimated and desired heat fluxes on the design surfaces. The design purpose is finding the best number and locations of the discretized heaters to meet the desired temperature and heat flux distributions on the design surface. Sarvari et al. [25] discretized the design surface into a series of control points and used *B*-spline to approximate the geometry shape of the boundary, the MGA is employed to optimize the locations of the control points to produce a desired heat flux distribution on the temperature-specified surface. The effects of corresponding parameters on the inverse design results are also investigated and the angular meshes are recommended as  $N_\theta \times N_\varphi = 10 \times 10$ . In addition, the results indicate that optimizing the weights corresponding to the control points can improve the quality of designed shape. However, the MGA has the obvious drawback that the convergence velocity is relatively low.

The PSO algorithm is a kind of biologically inspired algorithm whose search process is similar to foraging of birds and it was proposed in 1995 by Eberhart and Kennedy [26]. The physical model of PSO is very simple and the computational program is easy to be implemented, it also has strong robustness and achieves good performance on computational efficiency and accuracy. In addition, PSO algorithm can well balance the global and local search of particles, which enhance the global convergence of the algorithm. Farahmand et al. [27] investigated the inverse geometry design of two-dimensional radiative enclosures with diffuse gray surfaces based on the PSO and the retrieval results show that PSO algorithm obtains better performance in satisfying the design goal based in terms of computational accuracy and CPU time compared with MGA. However, the standard PSO algorithm also suffers from easily trapping into local optima in solving high dimensional problems. In order to strengthen the applicability of PSO, some improvements have been proposed and widely applied, including Stochastic PSO (SPSO), Differential Evolution PSO (DEPSO), Multi Phase PSO (MPPSO), and so on. However, to the authors' best knowledge, there are few reports concerning about the applications of improved PSO algorithms for solving inverse geometry design problems of radiative enclosures.

In this chapter, the application of PSO algorithms in solving inverse geometry design problems of two-dimensional radiative enclosures filled with participating media is investigated. The design goal is to satisfy a uniform distribution of radiative heat flux on the designed surface.

The discrete ordinate method (DOM) with a body-fitted coordinate system is used to solve the RTE. The standard PSO, SPSO and DEPSO algorithms are applied to optimize the locations of the control points, and Akima cubic interpolation is adopted to obtain the boundary geometry shape through these points. A typical inverse geometry design test is studied to demonstrate the good performance of PSO algorithms and the effects of corresponding parameters are also discussed.

The remainder of this chapter is organized as follows: the theoretical principles of PSO algorithms are introduced in Section 2. The feasibility of PSO algorithms by four famous benchmark functions is verified in Section 3. The inverse geometry design of two-dimensional radiative enclosures and the influences of the number of control points and the radiative properties of media on the inverse design results are investigated in Section 4. The main conclusions of the researches in this chapter are summarized in Section 5.

## 2. Theoretical overview of PSO algorithms

Bird individuals will communicate with each other to share their information about food when they are foraging, which can help birds find food faster. The advantages of cooperation of bird swarm are much greater than the disadvantages of competition among bird individuals. Based on the features of bird foraging behavior, Kennedy and Eberhart proposed PSO algorithm in 1995 [26]. The solving process of PSO algorithm is similar to the foraging behavior of birds, and the corresponding relationships are shown in **Table 1**.

Bird foraging behavior	PSO algorithm
Foraging domain of bird individual	Searching space of each particle
Bird individual	Particle
Flight speed of bird	Moving speed of particle
Location of bird	Location of particle, which represents a solution of optimization problems
Location of food	The best solution of optimization problems

**Table 1.** Corresponding relationships between bird foraging and PSO algorithm.

There are two dominant parameters in the PSO algorithm, namely, the speed and the location of particles. The moving speed decides the direction and distance of the particles, and every location of particles can be considered as the potential a solution of optimization problems. PSO adopts a combination of local and global searches and shares the evolutionary information among particle individuals to find the optimal solution.

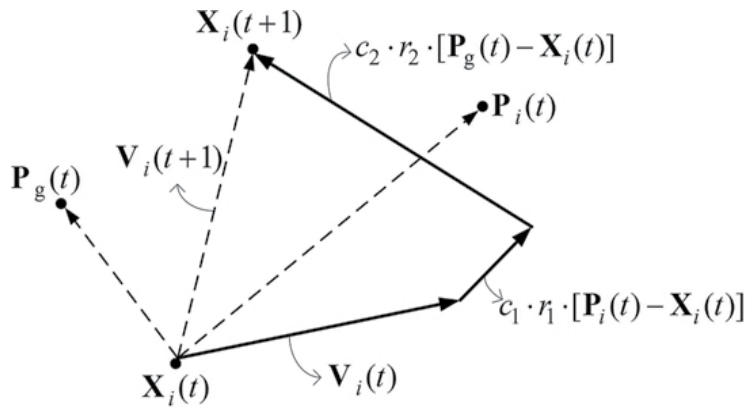
## 2.1. Basic PSO algorithm

At the beginning of the optimization of PSO, every location and velocity of particles are randomly generated. During each iteration, there will be two extreme values. One is best location that an individual particle found so far, which is called local best location. Another is the best location that the whole particle swarm found so far, which is called global best location. The velocity and location of each particle are stochastically accelerated according to these two extremes and the evolutionary formula can be expressed as follows [26]

$$\mathbf{V}_i(t+1) = \mathbf{V}_i(t) + c_1 \cdot r_1 \cdot [\mathbf{P}_i(t) - \mathbf{X}_i(t)] + c_2 \cdot r_2 \cdot [\mathbf{P}_g(t) - \mathbf{X}_i(t)] \quad (1)$$

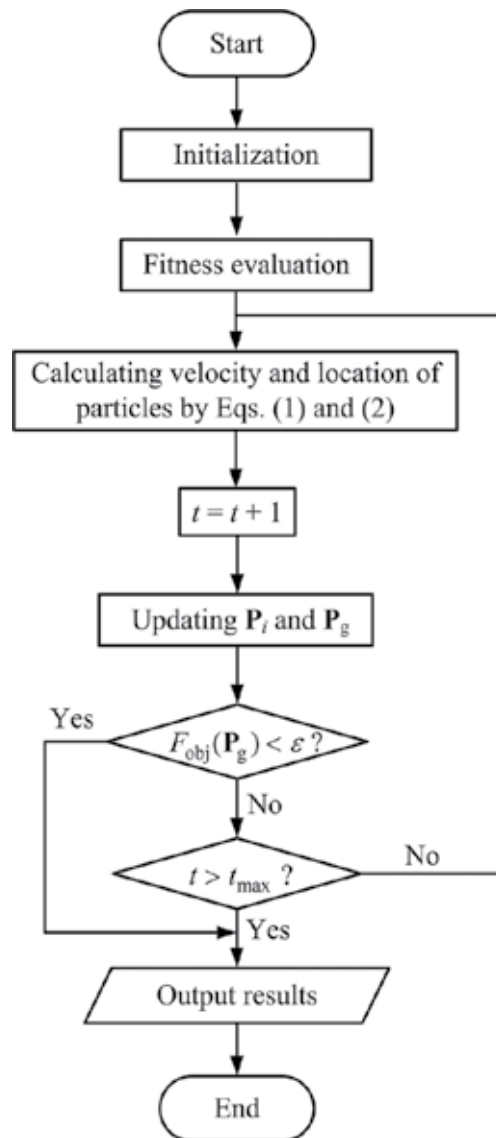
where  $\mathbf{V}_i(t)$  and  $\mathbf{V}_i(t+1)$  represent the velocity of  $i$ th particle at iterations  $t$  and  $t+1$ , respectively, and  $\mathbf{V}_i \in [-V_{\max}, V_{\max}]$ .  $c_1$  and  $c_2$  are two positive constants called acceleration coefficient.  $r_1$  and  $r_2$  are two uniform random values in the range of  $[0, 1]$ .  $\mathbf{P}_i(t)$  and  $\mathbf{P}_g(t)$  indicate the local best location and global best location, respectively.  $\mathbf{X}_i(t)$  denotes the location of  $i$ th particle, which is depending on the search experience of  $i$ th particle and surrounding particles. The evolutionary formula of  $i$ th particle's location is defined as [26]

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \quad (2)$$



**Figure 1.** The evolution of particle's location.

According to Eq. (1), we can find that the velocity of  $i$ th particle consists of three parts: the first part on the right side is the current velocity of  $i$ th particle, which can counterpoise the local search and the global search; the second part on the right side denotes the influence of the search memory of  $i$ th particle, which makes individual particle has the ability of global search; the third part on the right side indicates the influence of the cooperation among particles. The updating process of the  $i$ th particle' location is shown in **Figure 1**.



**Figure 2.** The flowchart of the basic PSO algorithm.

The main procedure of PSO algorithm for solving optimization problems can be carried out according to the following steps:

**Step 1:** Initialization: Randomly initialize the location and the velocities of particle of every particle in the searching space, input the number of particle swarm, the maximum iteration numbers  $t_{\max}$  and the stop criterion  $\varepsilon$ , etc. Set the current iteration as  $t=1$ .

**Step 2:** Fitness evaluation: Evaluate each particle's fitness according to its location and determine the local best location  $\mathbf{P}_i(t)$  and global best location  $\mathbf{P}_g(t)$ .

**Step 3:** Updating: Update the velocity and the location of each particle according to Eqs. (1) and (2), respectively. Calculate the new objective function value of each particle and update the local and global best locations  $\mathbf{P}_i(t)$  and  $\mathbf{P}_g(t)$ .

**Step 4:** Comparison: Compare the objective function value of each newly obtained particle with the corresponding values at the last iteration. If the new objective function value is better than the one in the last generation, then the new location and velocity of this particle is updated. Otherwise, the new location is abandoned.

**Step 5:** Repeating: Check whether one of the following two stop criterion is reached: (1) the objective function value is less than the value of  $\varepsilon$ , and (2) the iteration number reaches the maximum iteration number. If so, go to the next step; otherwise, go to **Step 3**.

**Step 6:** Update of iteration number: Update the number of iteration from  $t$  to  $t+1$ .

**Step 7:** Termination of iteration: Output the global best optima and its corresponding results of optimization problems and then stop the calculation.

The flowchart of the basic PSO algorithm is shown in **Figure 2**.

However, there are some obvious shortcomings in the basic PSO algorithm, such as slow convergence, easy to fall into local optimum, and even the velocities tend to be infinity in some occasions. Thus, many modifications have been proposed to overcome these drawbacks.

## 2.2. Standard PSO algorithm

There are two important capabilities in PSO algorithm, namely exploration and exploitation of particles. Exploration is the phenomenon that particles leave the original orbit and search for new space. Exploitation is the phenomenon that particles look for better locations along the original track. In order to better take advantage of these two search way, Shi and Eberhart put forward the standard PSO algorithm on the basis of basic PSO in 1998 [28], in which an inertia weight coefficient  $w$  is introduced to control the impact of the current velocity on the next velocity. The velocity formula of  $i$ th particle can be expressed as [28]

$$\mathbf{V}_i(t+1) = w \cdot \mathbf{V}_i(t) + c_1 \cdot r_1 \cdot [\mathbf{P}_i(t) - \mathbf{X}_i(t)] + c_2 \cdot r_2 \cdot [\mathbf{P}_g(t) - \mathbf{X}_i(t)] \quad (3)$$

where  $w$  is the inertia weight coefficient, which can directly affect the balance between the global and local exploration abilities. At the initial stage of the search process, a big inertia weight coefficient is recommended to improve the global exploration ability in the relatively large space; whereas the inertia weight should be reduced with the iteration number increases to strengthen the local exploitation ability. It is worth pointing that a linearly decreasing inertia weight coefficient can successfully prevent particles from oscillating near the global best location [29]. Therefore, the inertia weight coefficient can be defined as

$$w = w_{\max} - \frac{t}{t_{\max}} \cdot (w_{\max} - w_{\min}) \quad (4)$$

Comparing Eq. (1) with Eq. (3) we can find that basic PSO algorithm is a special circumstance of standard PSO algorithm that inertia weight is set as  $w=1$ . The searching efficiency is significantly improved by introducing inertia weight  $w$ . However, the proportional relation of particle velocities at every generation is not all the same, and the standard PSO algorithm can't be successfully applied for solving some complicated optimization problems. For breaking through the limitation of PSO, many modified techniques have been developed out and widely applied in engineering fields.

### 2.3. Stochastic PSO algorithm

In order to overcome the drawback that PSO algorithm converges too early and make sure to reach the goal of global convergence, Zeng and Cui proposed SPSO algorithm in 2004 [30, 31], in which a stopped changing particle is utilized to improve the global searching ability of particle swarms.

In SPSO algorithm, the inertia weight coefficient is set as  $w=0$ . Hence, the velocity of  $i$ th particle at  $t+1$  iteration is determined by three parameters of  $t$  iteration, namely  $\mathbf{X}_i(t)$ ,  $\mathbf{P}_i(t)$  and  $\mathbf{P}_g(t)$ . The new velocity of  $i$ th particle can be expressed as [30]

$$\mathbf{V}_i(t+1) = c_1 \cdot r_1 \cdot [\mathbf{P}_i(t) - \mathbf{X}_i(t)] + c_2 \cdot r_2 \cdot [\mathbf{P}_g(t) - \mathbf{X}_i(t)] \quad (5)$$

According to Eq. (5), it can be found that the local searching ability of SPSO is increased compared with standard PSO. However, the global searching ability is reduced significantly. In order to further strengthen the global searches of SPSO, the algorithm randomly generates a particle in the searching space whose location is  $\mathbf{X}_j(t+1)$ , and other particles' locations are updated based on the Eq. (5). The whole amending process can be expressed by the following equations

$$\begin{aligned} \mathbf{P}_j &= \mathbf{X}_j(t+1) \\ \mathbf{P}_i &= \begin{cases} \mathbf{P}_i, & F(\mathbf{P}_i) < F[\mathbf{X}_i(t+1)] \\ \mathbf{X}_i(t+1), & F(\mathbf{P}_i) \geq F[\mathbf{X}_i(t+1)] \end{cases} \\ \mathbf{P}'_g &= \arg \min \{F(\mathbf{P}_i) | i = 1, \dots, M\} \\ \mathbf{P}_g &= \arg \min \{F(\mathbf{P}'_g), F(\mathbf{P}_g)\} \end{aligned} \quad (6)$$

After above updates, the following criterions are executed:

(1) If  $\mathbf{P}_g = \mathbf{P}_j$ , which demonstrates that the random location  $\mathbf{X}_j$  is the global best location. In this situation the  $j$ th particle will not be updated on the basis of Eq. (5) and the algorithm will randomly generate a location  $\mathbf{X}_j$  in the searching space at the next iteration. At the same time, the velocities and locations of other particles are updated according to Eqs. (5) and (2) after  $\mathbf{P}_g$  and  $\mathbf{P}_j$  are updated, respectively.

(2) If  $\mathbf{P}_g \neq \mathbf{P}_j$  and  $\mathbf{P}_g$  has not been updated, which indicates that there is no better global best location found, compared with the last iteration, then all the particles' the velocities and locations are updated based on the Eqs. (5) and (2), respectively.

(3) If  $\mathbf{P}_g \neq \mathbf{P}_j$  and  $\mathbf{P}_g$  has been updated, which demonstrates that there is a location of  $k$ th particle ( $k \neq j$ ) meets the requirement  $\mathbf{X}_k(t+1) = \mathbf{P}_k = \mathbf{P}_g$ , namely the local best location of  $k$ th particle is the global best location and it is better than the global best location at the last iteration. In this situation, the  $k$ th particle stops evolving and it is used for storing global optima. Other particles' velocities and locations of other particles are updated according to Eqs. (5) and (2) after  $\mathbf{P}_g$  and  $\mathbf{P}_j$  are updated, respectively.

Through the above analysis we can find that there is at least one particle's location reaches the global best location at a particular iteration, which indicates that at least one particle is randomly generated at each iteration. Therefore, SPSO algorithm has been proved with strong global search ability.

#### 2.4. Differential evolution PSO algorithm

Differential Evolution (DE) algorithm adopts simple differential operation among potential solutions to produce new candidate solution, which is a parallel, direct and stochastic searching technique. It was first proposed for solving Chebyshev polynomials and global optimization problems over continuous spaces by Storn and Price in 1995 [32]. Taking a cue from DE, the mutation operation is introduced into PSO algorithm to overcome the drawback of trapping in local optima, which is called DEPSO.

In DEPSO algorithm, the differential evolution operator is introduced to increase the diversity of particle swarms which is defined as

$$\beta = \chi [\mathbf{X}_{r_1}(t) - \mathbf{X}_{r_2}(t)] \quad (7)$$

where  $\chi$  is the differential mutation operator which controls the magnification of differential variation  $\mathbf{X}_{r_1}(t) - \mathbf{X}_{r_2}(t)$  and it is usually set as a constant in the interval  $[0, 2]$ .  $r_1$  and  $r_2$  are two integers which are randomly chosen in the interval  $[1, M]$  and they are not equal to the index  $i$ . Hence, the location evolution of  $i$ th particle can be expressed as

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t) + \beta(C - F_{\min}) \quad (8)$$

where  $C$  represents a preset constant value which satisfies  $C \leq F_{\min}$ .  $F_{\min}$  indicates the minimum objective function value at the current iteration. The differential evolution term can force the particle to change if only  $C \neq F_{\min}$ , which can effectively prevent PSO algorithm from falling into local optima.

However, the location obtained by mutation operation maybe a worse result which will cause a bad influence on the search of other particles. In order to make sure the rapidity and stability

of DEPSO algorithm, the following judgment should be executed before the location is updated:

(1) If the fitness value of the new location is better than the fitness value of the earlier location, which demonstrates the mutation is successful. Then the location of  $i$ th particle is updated according to Eq. (8).

(2) If the new fitness value is worse than before, which indicates this mutation is failed. Then the location of  $i$ th particle is updated according to Eq. (2) and the mutation operation of  $i$ th will continue at the next iteration until the mutation is successful.

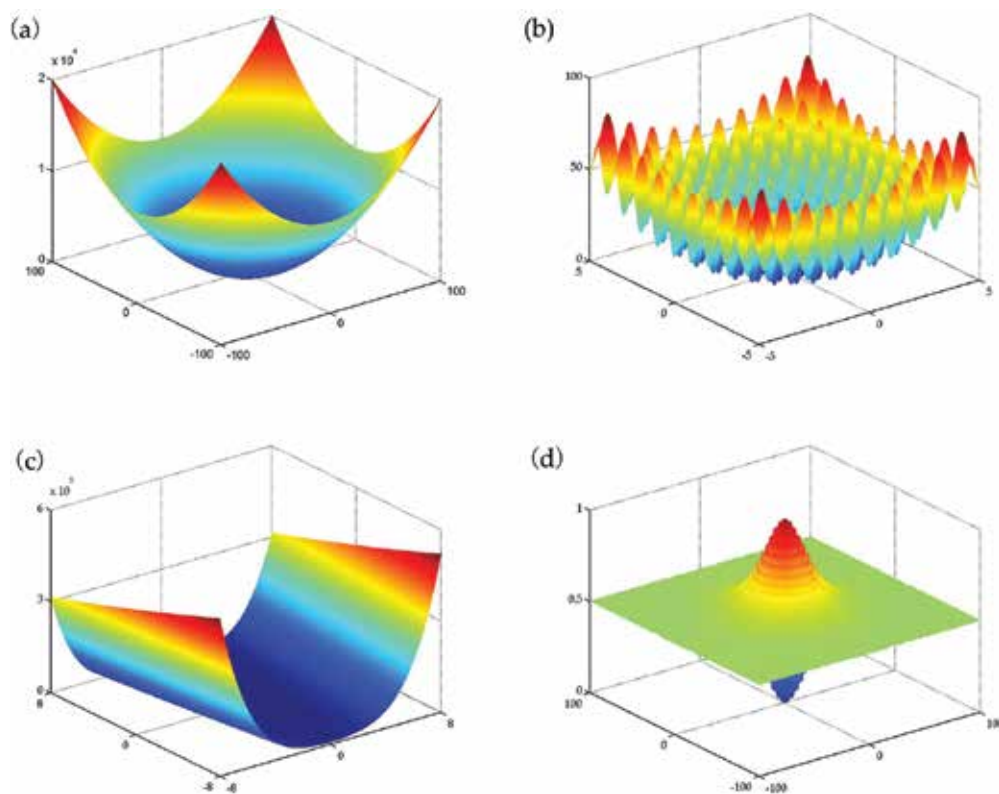
In addition, there is a significant difference between DEPSO and basic PSO that the velocities of particles are not limited in the searching process, which can increase the convergence rate.

### 3. Simulation test of PSO algorithms

In order to test the performance of the above PSO algorithms, four benchmark optimization functions are used for verification whose details are shown in **Table 2**. The parameters in PSO algorithms are set as follows: the number of particle swarm population is set as  $M=50$ , the maximum velocity is set as  $V_{\max}=3.0$ , the inertia weight coefficient is chose according to Eq. (4). The acceleration constants are set as  $c_1=1.2$  and  $c_2=0.8$  in SPSO and DEPSO algorithms, whereas  $c_1=2.0$  and  $c_2=2.0$  in standard PSO algorithm, respectively. The objective functions are set as the same as the test functions and the iteration stops upon the fitness value reaches the maximum iteration number  $t_{\max}=1000$  or the objective function value is less than  $10^{-30}$ . The test results of the four functions with dimension  $n=2$  are shown in **Figure 3**. **Table 3** lists the average objective function values and the corresponding mean squared errors for these four benchmark functions with 1000 independent runs. As shown, both the SPSO and DEPSO algorithms achieve better performance than the standard PSO algorithm in terms of computational accuracy.

Function	Expression	Dimension	Search Space
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$n \leq 30$	$[-100, 100]^n$
Rastrigin	$f_2(x) = \sum_{i=1}^n [x_i^2 - \cos(2\pi x_i) + 10]$	$n \leq 30$	$[-5, 5]^n$
Rosenbrock	$f_3(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$n \leq 10$	$[-5, 5]^n$
Schaffer	$f_4(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2} + \frac{1}{2}$	$n \leq 2$	$[-100, 100]^n$

**Table 2.** Details of four benchmark functions.



**Figure 3.** The images of (a) Sphere function, (b) Rastrigin function, (c) Rosenbrock function and (d) Schaffer function with dimension  $n=2$ .

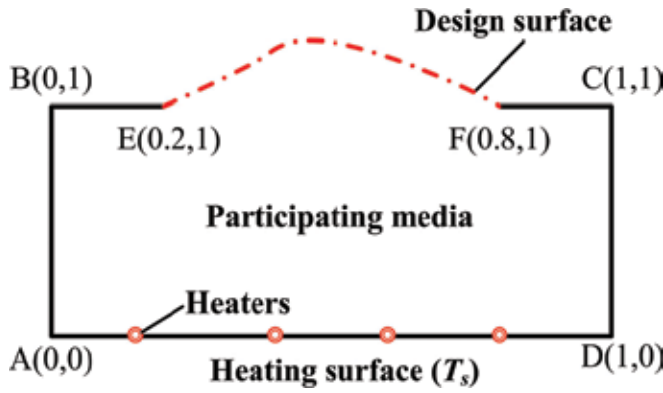
Function	Dimension	PSO	SPSO	DEPSO
Sphere	10	$5.94 \times 10^{-28} \pm 3.46 \times 10^{-27}$	$8.33 \times 10^{-31} \pm 1.67 \times 10^{-31}$	$8.25 \times 10^{-31} \pm 1.91 \times 10^{-31}$
	20	$2.82 \times 10^{-13} \pm 7.76 \times 10^{-13}$	$4.80 \times 10^{-17} \pm 3.63 \times 10^{-16}$	$2.98 \times 10^{-25} \pm 1.54 \times 10^{-24}$
	30	$5.27 \times 10^{-8} \pm 8.69 \times 10^{-8}$	$1.23 \times 10^{-6} \pm 3.97 \times 10^{-6}$	$7.30 \times 10^{-15} \pm 2.34 \times 10^{-14}$
Rastrigin	10	$4.82 \times 10^{-28} \pm 1.82 \times 10^{-27}$	$8.35 \times 10^{-31} \pm 1.62 \times 10^{-31}$	$6.87 \times 10^{-31} \pm 2.16 \times 10^{-31}$
	20	$2.68 \times 10^{-13} \pm 5.52 \times 10^{-13}$	$7.22 \times 10^{-17} \pm 9.23 \times 10^{-16}$	$2.03 \times 10^{-25} \pm 1.46 \times 10^{-24}$
	30	$5.39 \times 10^{-8} \pm 7.56 \times 10^{-8}$	$1.48 \times 10^{-6} \pm 4.34 \times 10^{-6}$	$7.08 \times 10^{-15} \pm 6.66 \times 10^{-14}$
Rosenbrock	3	$3.36 \times 10^{-8} \pm 8.39 \times 10^{-8}$	$3.53 \times 10^{-17} \pm 1.06 \times 10^{-16}$	$2.95 \times 10^{-16} \pm 9.21 \times 10^{-16}$
	5	$5.62 \times 10^{-2} \pm 6.02 \times 10^{-2}$	$4.12 \times 10^{-2} \pm 1.99 \times 10^{-2}$	$1.37 \times 10^{-2} \pm 2.85 \times 10^{-2}$
	10	$3.18 \times 10^0 \pm 1.56 \times 10^0$	$4.62 \times 10^0 \pm 3.45 \times 10^0$	$5.78 \times 10^0 \pm 2.61 \times 10^0$
Schaffer	2	$5.11 \times 10^{-2} \pm 5.61 \times 10^{-2}$	$9.87 \times 10^{-4} \pm 2.92 \times 10^{-3}$	$4.66 \times 10^{-4} \pm 2.60 \times 10^{-3}$

**Table 3.** The retrieval results of three test functions with 1000 independent runs.

## 4. Inverse geometry design of two-dimensional radiative enclosures

### 4.1. Description of the inverse geometry design problem

Considering a radiative equilibrium problem in two-dimensional irregular enclosures filled with participating media whose schematic diagram is shown in **Figure 4**. The curve EF represents the design surface and the design purpose is to produce a uniform distribution of radiative heat flux on the design surface. The bottom surface AD is the heating surface which can be considered as the radiative heat source and its temperature is fixed as  $T_s$ . The two side surfaces AB and CD are cold with temperatures of 0K.



**Figure 4.** Physical model of inverse geometry design.

In order to optimize the geometry shape of the design surface to meet the specified requirement, the objective function (being equal to the fitness function in PSO algorithms) is defined as the square residuals between the estimated and average dimensionless radiative heat flux values which can be expressed as

$$F_{\text{obj}} = \sum_{i=1}^N (\mathcal{Q}_{w_i} - \mathcal{Q}_{\text{av}})^2 \quad (9)$$

where  $N$  is the number of the computational node on the design surface, and  $\mathcal{Q}_{w_i}$  and  $\mathcal{Q}_{\text{av}}$  are the dimensionless radiative heat flux of the  $i$ th node and average value on the design surface, respectively. The iteration stop criterion of PSO is defined as

$$F_{\text{obj}} < \varepsilon \quad (10)$$

where  $\varepsilon$  is a small positive value. The smaller the value of  $\varepsilon$  is, the better the homogenization degree will be.

To evaluate the optimization results, the relative error is defined as

$$\varepsilon_{\text{rel}} = \left| \frac{Q_{w_i} - Q_{\text{av}}}{Q_{\text{av}}} \right| \times 100\% \quad (11)$$

#### 4.2. Akima cubic interpolation

The design surface is discretized into a series of control points, and Akima cubic interpolation is used to approximate the geometry shape of the surface in optimization process. Akima interpolation is formulated by a cubic polynomial between two control points. First, the prerequisite is introduced as [33]

$$\begin{cases} Y(x_k) = y_k \\ Y(x_{k+1}) = y_{k+1} \\ Y'(x_k) = g_k \\ Y'(x_{k+1}) = g_{k+1} \end{cases} \quad (12)$$

where  $g_k$  is the slope of the curve at the position  $x_k$ , which can be defined as [33]

$$g_k = \begin{cases} \frac{l_{k-1} + l_k}{2} & l_{k+1} = l_k \text{ and } l_{k-1} = l_{k-2} \\ \frac{|l_{k+1} - l_k|l_{k-1} + |l_{k-1} - l_{k-2}|l_k}{|l_{k+1} - l_k| + |l_{k-1} - l_{k-2}|} & \text{else} \end{cases} \quad (13)$$

and the function  $l_k$  can be expressed as [33]

$$l_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \quad (14)$$

At the endpoints, the value of the function  $l_k$  can be defined as [33]

$$\begin{cases} l_0 = 2l_1 - l_2, & l_{-1} = 2l_0 - l_1 & \text{at the left endpoint} \\ l_{N_d} = 2l_{N_d-1} - l_{N_d-2}, & l_{N_d+1} = 2l_{N_d} - l_{N_d-1} & \text{at the right endpoint} \end{cases} \quad (15)$$

If the above equations are satisfied, then the cubic polynomial in the subinterval  $[x_k, x_{k+1}]$  can be determined as [33]

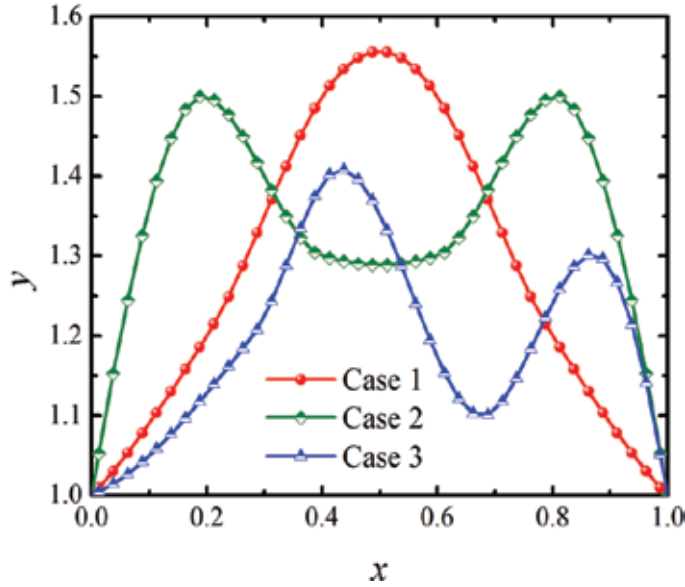
$$Y(x) = C_1 + C_2(x - x_k) + C_3(x - x_k)^2 + C_4(x - x_k)^3 \quad (16)$$

where  $C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$  are polynomial coefficients, which can be calculated as [33]

$$\begin{cases} C_1 = y_k \\ C_2 = g_k \\ C_3 = \frac{3(u_k - 2g_k - g_{k+1})}{x_{k+1} - x_k} \\ C_4 = \frac{g_k + 2g_{k+1} - 2u_k}{(x_{k+1} - x_k)^2} \end{cases} \quad (17)$$

$(x, y)$	$(x_1, y_1)$	$(x_2, y_2)$	$(x_3, y_3)$	$(x_4, y_4)$	$(x_5, y_5)$	$(x_6, y_6)$
Case 1	(0.0, 1.0)	(0.2, 1.2)	(0.4, 1.5)	(0.6, 1.5)	(0.8, 1.2)	(1.0, 1.0)
Case 2	(0.0, 1.0)	(0.2, 1.5)	(0.4, 1.3)	(0.6, 1.3)	(0.8, 1.5)	(1.0, 1.0)
Case 3	(0.0, 1.0)	(0.28, 1.2)	(0.41, 1.4)	(0.67, 1.1)	(0.88, 1.3)	(1.0, 1.0)

**Table 4.** Coordinates of control points of three test cases.



**Figure 5.** Curve fitting results by means of Akima cubic interpolation.

In order to test the performance of the Akima cubic interpolation, three interpolation cases are applied, in which six specified points are used as control points whose coordinates are shown

in **Table 4**. The curves in **Figure 5** indicate the Akima cubic interpolation can successfully applied for geometry shape fitting.

#### 4.3. Discrete ordinate method with a body-fitted coordinate

The boundary shape changes with the optimization of the design surface and the computational domain must be re-meshed, which greatly increase the solving difficulty of the inverse geometry design problems. In addition, the forward radiative heat transfer problem cannot be precisely solved by the normal numerical method. For the purpose of fitting the irregular boundary shape, the DOM with a body-fitted coordinate system is adopted to solve the RTE. For the participating media, the forward can be written as [34]

$$\frac{\partial I(s, \hat{s})}{\partial s} = -\beta_e I(s, \hat{s}) + \kappa_a I_b(s) + \frac{\kappa_s}{4\pi} \int_{4\pi} I(\hat{s}_i) \Phi(\hat{s}_i, \hat{s}) d\Omega_i \quad (18)$$

which is an integro-differential type, where  $I$  is the function of position  $s$  and direction  $\hat{s}$ .  $\beta_e$ ,  $\kappa_a$ , and  $\kappa_s$  are the extinction, absorption, and scattering coefficients of media, respectively. The  $\Phi(\hat{s}_i, \hat{s})$  is the scattering phase function between incoming direction  $\hat{s}_i$  and scattering direction  $\hat{s}$ , which can be defined as  $\Phi = 1.0 + a \cos(\hat{s}_i \cdot \hat{s})$ . The coefficient "a" are different for different scattering characteristics of media and values are  $a=0$ ,  $a=1$  and  $a=-1$  for the isotropic scattering, the forward scattering and the backward scattering of media. The forward radiative heat transfer problem in the irregular enclosures is solved by using the computationally feasible DOM with a body-fitted coordinate system in this research. The 2D RTE discretized by the DOM can be expressed as [34, 35]

$$\alpha^m \frac{\partial I^m}{\partial x} + \beta^m \frac{\partial I^m}{\partial y} = -\beta_e I^m + \kappa_a I_b(s) + \frac{\kappa_s}{4\pi} \left( \sum_{m'=1}^{N\Omega} w^{m'} I_w^{m'} \Phi^{m',m} \right) \quad (19)$$

where  $\alpha^m$  and  $\beta^m$  are the direction cosines of the discrete direction  $m$ .  $s$  refers to the spatial position,  $w$  is the quadrature weight. The radiative boundary condition can be directly imposed as follows [34]

$$I_w^m = \varepsilon_w I_{b,w} + \frac{1-\varepsilon_w}{\pi} \sum_{\mathbf{n}_w \cdot \mathbf{s}^{m'} < 0} w^{m'} I_w^{m'} \left| \mathbf{n}_w \cdot \mathbf{s}^{m'} \right|, \quad \mathbf{n}_w \cdot \mathbf{s}^{m'} > 0 \quad (20)$$

where  $\varepsilon_w$  is the emissivity of boundaries,  $I_{b,w}$  is radiative intensity of blackbody boundaries,  $\mathbf{n}_w$  represents the unit normal vector on the boundary, and  $\mathbf{s}$  denotes the direction of radiative transfer.

The Jacobian matrix is used for coordinate transformation to solve the radiative heat transfer in irregular enclosures

$$J = \frac{\partial(x, y)}{\partial(\xi, \eta)} \quad (21)$$

The spatially discretized RTE with a body-fitted coordinate system can be expressed as

$$\begin{aligned} & \left[ I^m J \left( \alpha^m \frac{\partial \xi}{\partial x} + \beta^m \frac{\partial \xi}{\partial y} \right) \right]_e \Delta \eta - \left[ I^m J \left( \alpha^m \frac{\partial \xi}{\partial x} + \beta^m \frac{\partial \xi}{\partial y} \right) \right]_w \Delta \eta + \left[ I^m J \left( \alpha^m \frac{\partial \eta}{\partial x} + \beta^m \frac{\partial \eta}{\partial y} \right) \right]_n \Delta \xi \\ & - \left[ I^m J \left( \alpha^m \frac{\partial \eta}{\partial x} + \beta^m \frac{\partial \eta}{\partial y} \right) \right]_s \Delta \xi = J \left[ -\beta_e I^m + \kappa_a I_b + \frac{\kappa_s}{4\pi} \left( \sum_{m'=1}^{N\Omega} w^{m'} I_w^{m'} \Phi^{m', m} \right) \right]_p \Delta \xi \Delta \eta \end{aligned} \quad (22)$$

where  $P$  represents the central node of the control volume. The subscripts  $e$ ,  $w$ ,  $n$ , and  $s$  represent the eastern, western, northern and southern boundaries around  $P$ , respectively.

The step scheme is applied to solve the above equations, and Eq. (22) can be expressed as

$$a_P^m I_P^m = a_E^m I_E^m + a_W^m I_W^m + a_N^m I_N^m + a_S^m I_S^m + b_P^m \quad (23)$$

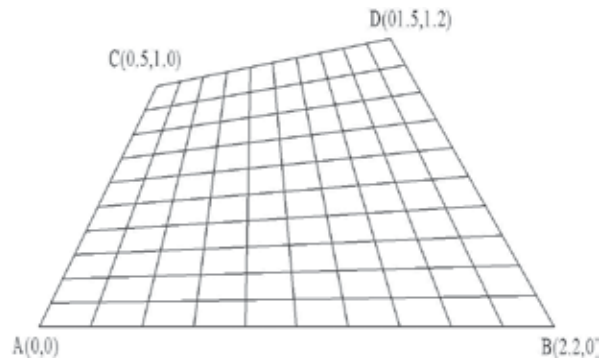
where the subscripts  $E$ ,  $W$ ,  $N$ , and  $S$  represent the central node of eastern, western, northern, and southern control volumes around control volume  $P$ , and

$$\begin{aligned} a_E^m &= \max \left\{ - \left[ J \left( \alpha^m \frac{\partial \xi}{\partial x} + \beta^m \frac{\partial \xi}{\partial y} \right) \right]_e \Delta \eta, 0 \right\} & a_W^m &= \max \left\{ \left[ J \left( \alpha^m \frac{\partial \xi}{\partial x} + \beta^m \frac{\partial \xi}{\partial y} \right) \right]_w \Delta \eta, 0 \right\} \\ a_N^m &= \max \left\{ - \left[ J \left( \alpha^m \frac{\partial \eta}{\partial x} + \beta^m \frac{\partial \eta}{\partial y} \right) \right]_n \Delta \xi, 0 \right\} & a_S^m &= \max \left\{ \left[ J \left( \alpha^m \frac{\partial \eta}{\partial x} + \beta^m \frac{\partial \eta}{\partial y} \right) \right]_s \Delta \xi, 0 \right\} \\ a_P^m &= \max \left\{ \left[ J \left( \alpha^m \frac{\partial \xi}{\partial x} + \beta^m \frac{\partial \xi}{\partial y} \right) \right]_e \Delta \eta, 0 \right\} + \max \left\{ - \left[ J \left( \alpha^m \frac{\partial \xi}{\partial x} + \beta^m \frac{\partial \xi}{\partial y} \right) \right]_w \Delta \eta, 0 \right\} \\ &+ \max \left\{ \left[ J \left( \alpha^m \frac{\partial \eta}{\partial x} + \beta^m \frac{\partial \eta}{\partial y} \right) \right]_n \Delta \xi, 0 \right\} + \max \left\{ - \left[ J \left( \alpha^m \frac{\partial \eta}{\partial x} + \beta^m \frac{\partial \eta}{\partial y} \right) \right]_s \Delta \xi, 0 \right\} \\ &+ J_P \beta_{e,P} \Delta \xi \Delta \eta \\ b_P^m &= J_P \left[ \kappa_a \frac{\sigma T_P^4}{\pi} + \frac{\kappa_s}{4\pi} \left( \sum_{m'=1}^{N\Omega} w^{m'} I_P^{m'} \Phi^{m', m} \right) \right]_p \Delta \xi \Delta \eta \end{aligned} \quad (24)$$

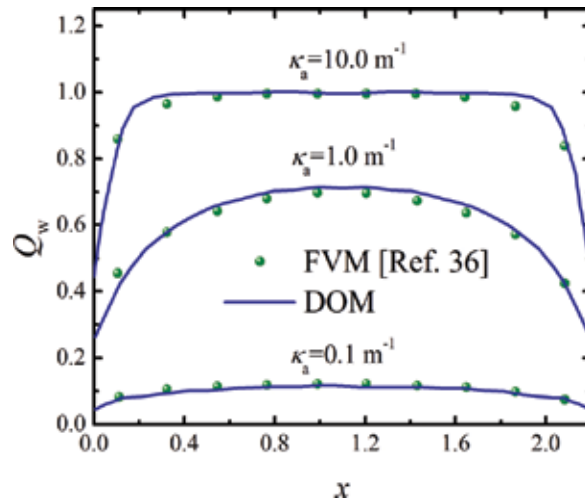
Eq. (24) can be expressed in matrix form

$$\mathbf{A} \Psi = \mathbf{B} \quad (25)$$

where  $\mathbf{A}$  represents the five-diagonal non-symmetric coefficient matrix,  $\Psi$  represents the vector that consists of the variables  $I^m$  at grid nodes, and  $\mathbf{B}$  represents the vector that consists of the variables  $b^m$  on the right side of Eq. (23). The conjugate gradients stabilized (CGSTAB) method is adopted to solve the final discretized RTE because of its stability and fast convergence rate [35].



**Figure 6.** The schematic diagram of grids in computational domains.



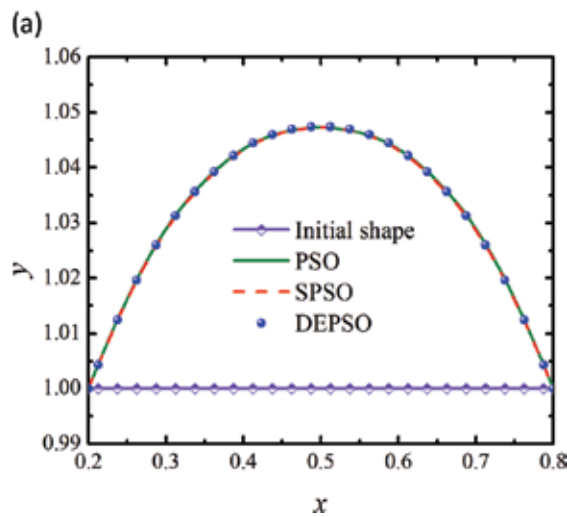
**Figure 7.** The dimensionless radiative heat flux distribution on the bottom boundary for different absorption coefficients.

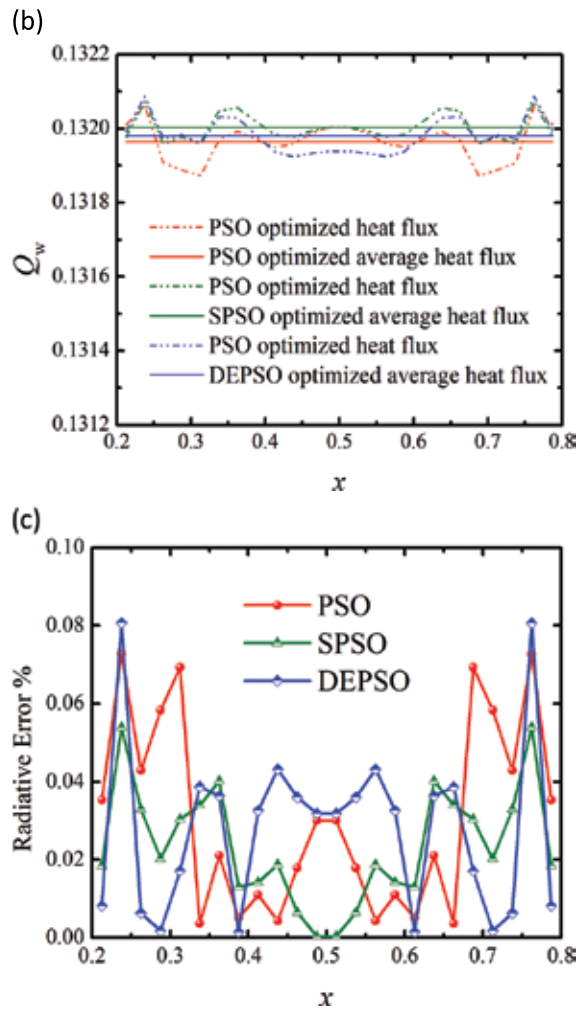
Consider a non-radiative equilibrium problem in the two-dimensional irregular enclosure filled with participating media. The four boundaries are cold surfaces whose temperature is

0K and all the boundaries are assumed as blackbody. The temperature of media is set as  $T_g$ , which can be considered as the heat source. The computational grids are meshed as  $10 \times 10$  and the schematic in the computational domain is shown in **Figure 6**. The radiative heat transfer problem for different absorption coefficients in the radiative enclosure is solved by the body-fitted DOM and the retrieval results are compared with the results obtained by FVM in Ref. [36] which is shown in **Figure 7**. The curves show that the retrieval results achieve good agreements with FVM, which verified the accuracy and reliability of the computational program for solving radiative problems in irregular enclosures.

#### 4.4. Inverse design results and discussions

The inverse geometry design model in Section 4.1 is considered and the standard PSO algorithm is abbreviated as PSO if there is no special instruction. The initial shape of the enclosure is rectangular, whose size is set as  $L_x \times L_y = 1.0 \times 1.0$  m. The absorption and scattering coefficients of media are set as  $\kappa_a = 2.0 \text{ m}^{-1}$  and  $\kappa_s = 0.5 \text{ m}^{-1}$ , respectively. The parameters in the PSO algorithms are set as the same as the ones in test cases of Section 3. The stopping criteria are set as follows: (1) the objective function value is less than  $10^{-7}$ , (2) the iteration times reach a maximum of 1000 and (3) the fitness value remains unchanged in consequent 100 iterations. The initial and final geometry shape of the design surfaces and their dimensionless radiative heat flux distributions are shown in **Figure 8(a)** and **8(b)**, respectively. The curves demonstrate that a uniform distribution of radiative heat flux on the design surface is obtained by means of PSO algorithms. **Figure 8(c)** shows the relative error distributions of the dimensionless radiative heat flux on the design surface. The average relative errors are 0.0310%, 0.0234% and 0.0279%, and the maximum relative errors are 0.0728%, 0.0540% and 0.0807% for PSO, SPSSO and DEPSO algorithms, respectively. Overall, the retrieved results clearly reveal that the specified requirement of producing a uniform radiative heat flux distribution on the design surface can be obtained by PSO algorithm.



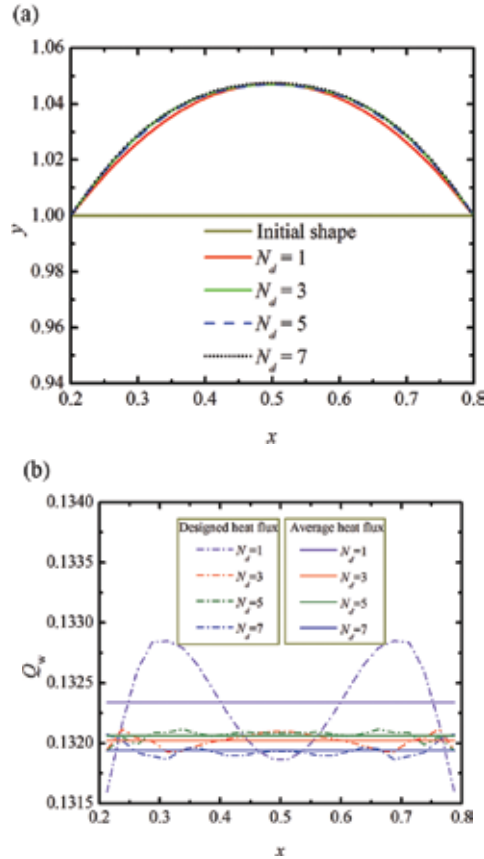


**Figure 8.** (a) Initial and final geometry shape of the design surface, (b) dimensionless radiative heat flux distribution on the design surface, and (c) relative error distributions of dimensionless radiative heat flux on the design surface, of a two-dimensional radiative enclosure.

Algorithms	CPU time (s)	Fitness values	Average relative error (%)	Maximum relative error (%)
PSO	40037	$7.28 \times 10^{-8}$	0.0313	0.0737
SPSO	31572	$4.05 \times 10^{-8}$	0.0244	0.0518
DEPSO	36944	$5.62 \times 10^{-8}$	0.0275	0.0669

**Table 5.** Comparison of inverse design results by PSO, SPSO and DEPSO algorithms.

In view of the random characteristic of intelligent algorithms, all the tests are repeated 50 trials to compare the performance of PSO algorithms. **Table 5** shows the comparison of the results obtained by PSO, SPSO and DEPSO algorithms. It can be found that all the PSO algorithms have reached the special design requirement and both SPSO and DEPSO achieve better performance than the initial PSO in terms of computational accuracy and efficiency.



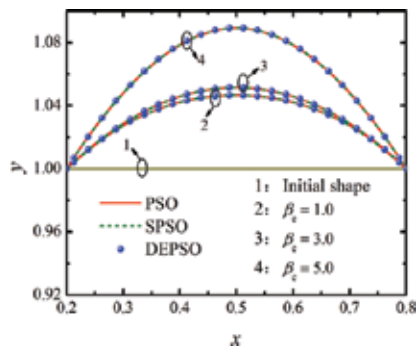
**Figure 9.** (a) Geometry shape of the design surface and (b) dimensionless radiative heat flux distribution on the design surface by means of SPSO algorithm for different numbers of control points.

In order to enhance the computational efficiency of inverse geometry design problems, the effects of corresponding parameters are investigated in this study. For the fact that the design surface is discretized into a series of control points in the inverse design process, the number of control point has a direct impact on the inverse geometry design results. The radiative physical parameters of media are kept as the same as the above typical example and the numbers of control point are set as  $N_d=1, 3, 5$ , and  $7$ , respectively. The SPSO algorithm is adopted as the inverse design method and the initial and optimized geometry shape and dimensionless radiative heat flux on the design surface are shown in **Figure 9**, respectively. The curves show that the special design requirement is satisfied under the conditions that  $N_d=3$ ,

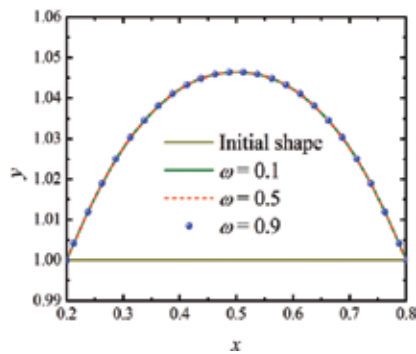
5 and 7, whereas the homogenization degree is relatively poor in the case that  $N_d=1$ . **Table 6** lists the iteration numbers, fitness values and relative errors for different numbers of control point. It can be found that both the iteration number and the relative error are smallest when  $N_d=3$ , which is because too many control points will decrease the sensitivity of radiative heat flux to the shape changing of design surface and one control point cannot provide enough necessary information for the geometry shape of the boundary [2, 9].

Control point numbers	Iteration numbers	Fitness values	Average relative error (%)	Maximum relative error (%)
1	164	$3.89 \times 10^{-8}$	0.2711	0.5597
3	19	$5.36 \times 10^{-8}$	0.0280	0.0796
5	26	$6.72 \times 10^{-8}$	0.0309	0.0875
7	37	$7.63 \times 10^{-8}$	0.0321	0.1024

**Table 6.** Comparison of inverse design results for different number of control points.



**Figure 10.** Geometry shape of the design surface for different extinction coefficients of media.



**Figure 11.** Geometry shape of the design surface for different scattering albedo of media.

The physical properties of the media have an important influence on the energy transfer and then affect the optimization results of radiative enclosures. The effects of the extinction coefficient and the scattering albedo on the inverse design results are studied here. The scattering albedo of media is fixed as  $\omega=0.5$ , the extinction coefficients are set as  $\beta=1.0, 3.0$  and  $5.0$ , respectively. The initial guessed and final optimized geometry shapes of the design surface are shown in **Figure 10**. It can be seen that three PSO algorithms achieve similar boundaries and the geometry shapes are significantly different for different extinction coefficients. The extinction coefficient of media is fixed as  $\beta=3.0$ , the scattering albedos are set as  $\omega=0.1, 0.5$  and  $0.9$ , respectively. The optimized geometry shapes of the design surface by means of SPSO are shown in **Figure 11**. It can be seen that the designed boundaries are very close to each other for different scattering albedos. **Tables 7** and **8** list the detailed inverse design results for different extinction coefficients and scattering albedos, respectively. As shown, both the optimized geometry the retrieval results for different extinction coefficients are significantly different, and both the height of the final designed surface and the dimensionless radiative heat flux decrease with the extinction coefficient increases. However, the retrieval results for different scattering albedo are close. Therefore, the scattering albedo of the media has little influence on the inverse geometry design results when the extinction coefficient is determined.

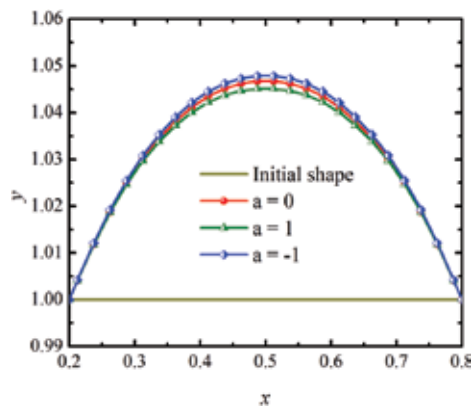
Algorithm	Extinction coefficient	Fitness values	Dimensionless radiative heat flux	Average relative error (%)	Maximum relative error (%)
PSO	$\beta_e=1.0$	$8.22 \times 10^{-8}$	0.2333	0.0329	0.0895
	$\beta_e=3.0$	$7.19 \times 10^{-8}$	0.1137	0.0496	0.0736
	$\beta_e=5.0$	$3.94 \times 10^{-8}$	0.0735	0.0460	0.1297
SPSO	$\beta_e=1.0$	$7.61 \times 10^{-8}$	0.2334	0.0295	0.0816
	$\beta_e=3.0$	$5.63 \times 10^{-8}$	0.1136	0.0398	0.0723
	$\beta_e=5.0$	$2.42 \times 10^{-8}$	0.0734	0.0301	0.0909
DEPSO	$\beta_e=1.0$	$5.07 \times 10^{-8}$	0.2334	0.0230	0.0849
	$\beta_e=3.0$	$4.16 \times 10^{-8}$	0.1136	0.0351	0.0706
	$\beta_e=5.0$	$2.85 \times 10^{-8}$	0.0734	0.0307	0.0889

**Table 7.** Inverse geometry design results for different extinction coefficients.

Algorithm	Scattering albedo	Fitness values	Dimensionless radiative heat flux	Average relative error (%)	Maximum relative error (%)
PSO	$\omega=0.1$	$7.93 \times 10^{-8}$	0.1137	0.0459	0.0867
	$\omega=0.5$	$7.19 \times 10^{-8}$	0.1137	0.0496	0.0736
	$\omega=0.9$	$7.23 \times 10^{-8}$	0.1136	0.0355	0.636
SPSO	$\omega=0.1$	$3.02 \times 10^{-8}$	0.1136	0.0297	0.0804
	$\omega=0.5$	$5.63 \times 10^{-8}$	0.1136	0.0398	0.0723
	$\omega=0.9$	$4.49 \times 10^{-8}$	0.1136	0.0322	0.0665
DEPSO	$\omega=0.1$	$5.07 \times 10^{-8}$	0.1136	0.0352	0.0349
	$\omega=0.5$	$4.16 \times 10^{-8}$	0.1136	0.0351	0.0706
	$\omega=0.9$	$3.49 \times 10^{-8}$	0.1136	0.0358	0.0710

**Table 8.** Inverse geometry design results for different scattering albedo of media.

The scattering of media will affect the original transfer direction of radiative heat or energy, so the scattering characteristic of media also has influence on the inverse geometry design of radiative enclosures. The extinction coefficient and scattering albedo of the media are set as  $\beta=3.0$  and  $\omega=0.5$ , respectively. The initial and final geometry shapes of the design surface by using SPSO algorithm for three kinds of scattering characteristics of media are shown in **Figure 12**. It can be found that the optimal geometry shapes of the design surface are close. **Table 9** plots the inverse geometry design results for different scattering characteristics. As shown, the values of dimensionless radiative heat flux on the design surface and relative errors are significantly different. Therefore, the scattering characteristics of media mainly affect the radiative heat flux on the boundaries and have no obvious effect on the final geometry shape of inverse radiative design problems.



**Figure 12.** Geometry shape of the design surface scattering characteristics of media.

Algorithm	Scattering characteristic	Fitness values	Dimensionless radiative heat flux	Average relative error (%)	Maximum relative error (%)
PSO	$a = 1$	$8.66 \times 10^{-8}$	0.1297	0.0354	0.0828
	$a = 0$	$9.04 \times 10^{-8}$	0.1136	0.0403	0.0776
	$a = -1$	$7.21 \times 10^{-8}$	0.1006	0.0385	0.0941
SPSO	$a = 1$	$7.57 \times 10^{-8}$	0.1297	0.0328	0.0982
	$a = 0$	$8.16 \times 10^{-8}$	0.1136	0.0463	0.0825
	$a = -1$	$1.60 \times 10^{-8}$	0.1007	0.0195	0.0629
DEPSO	$a = 1$	$5.53 \times 10^{-8}$	0.1297	0.0378	0.0797
	$a = 0$	$6.94 \times 10^{-8}$	0.1136	0.0426	0.0735
	$a = -1$	$5.71 \times 10^{-8}$	0.1007	0.0413	0.0882

**Table 9.** Inverse geometry design results for different scattering characteristics of media.

## 5. Conclusions

In this chapter, the basic theoretical principles of PSO algorithm is introduced in detail and three kinds of PSO algorithms—standard PSO, SPSO and DEPSO—are applied for solving the inverse geometry design problem of a two-dimensional radiative enclosure filled with participating media. The design purpose is to produce a uniform distribution of radiative heat flux on the designed surface. The design surface is discretized into a series of control points and the Akima cubic interpolation is used to approximate the geometry shape of the boundary. The radiative heat transfer problem in the irregular enclosures is solved by the DOM with a body-fitted coordinate system. The pre-required radiative heat flux distribution is satisfied by optimizing the positions of control points based on the PSO algorithms. The retrieval results show that PSO algorithms can be successfully applied to solve inverse geometry design problems and SPSO achieves the best performance on computational time. Meanwhile, the scattering albedo and scattering characteristics of media have little effect on the geometry shape of the design surface. To improve the computational efficiency, the number of control points is recommended as  $N_d=3$ .

## Acknowledgements

The supports of this work by the National Natural Science Foundation of China (Nos. 51476043 and 51576053), the Major National Scientific Instruments and Equipment Development Special Foundation of China (No. 51327803), and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 51421063) are gratefully acknowledged. A very special acknowledgement is also made to the editors and referees who make important comments to improve this chapter.

## Author details

Hong Qi\*, Shuang-Cheng Sun, Zhen-Zong He, Shi-Ting Ruan, Li-Ming Ruan and He-Ping Tan

\*Address all correspondence to: [qihong@hit.edu.cn](mailto:qihong@hit.edu.cn)

School of Energy Science and Engineering, Harbin Institute of Technology, Harbin, China

## References

- [1] Daun KJ, Howell JR., Morton DP Geometric optimization of radiative enclosures through nonlinear programming. *Numerical Heat Transfer, Part B*. 2003, 43(3): 203-219.
- [2] Tan JY, Liu LH Inverse geometry design of radiating enclosure filled with participating media using meshless method. *Numerical Heat Transfer Part A*. 2009, 56: 132-152.
- [3] França FHR Ezekoye OA, Howell JR Inverse boundary design combining radiation and convection heat transfer. *Journal of Heat Transfer*. 2001, 123(5): 884-891.
- [4] Rukolaine SA Regularization of inverse boundary design radiative heat transfer problems. *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2007, 104: 171-195.
- [5] Kowsary K, Pooladvand K, Pourshaghagh A Regularized variable metric method versus the conjugate gradient method in solution of radiative boundary design problem. *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2007, 108: 277-294.
- [6] Sarvari SMH, Mansouri SH, Howell JR Inverse boundary design radiation problem in absorbing-emitting media with irregular geometry. *Numerical Heat Transfer, Part A*. 2003, 43:565-584.
- [7] Daun KJ, Morton DP, Howell JR Geometric optimization of radiant enclosures containing specular surfaces. *Journal of Heat Transfer*. 2003, 125(5): 845-851.
- [8] Daun KJ, Howell JR Inverse design methods for radiative transfer systems. *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2005, 93: 43-60.
- [9] Sun SC, Qi Hong H, Zhao FZ, Ruan LM, Li BX. Inverse geometry design of two-dimensional complex radiative enclosures using krill herd optimization algorithm. *Applied Thermal Engineering*. 2016, 98(5): 1104-1115.
- [10] Howell JR, Ezekoye OA, Morales JC Inverse design model for radiative heat transfer. *Journal of Heat Transfer*. 2000, 122(3): 492-502.

- [11] Daun K, França F, Larsen M, et al. Comparison of methods for inverse design of radiant enclosures. *Journal of Heat Transfer*. 2005, 128(3): 269-282.
- [12] Franca FHR, Howell JR Transient inverse design of radiative enclosures for thermal processing of materials. *Inverse Problems in Science and Engineering*. 2006, 14(4): 423-436.
- [13] Tan JY, Zhao JM, Liu L.H Geometric optimization of a radiation-conduction heating device using meshless method. *International Journal of Thermal Sciences*. 2011, 50: 1820-1831.
- [14] Sarvari SMH, Mansouri SH Inverse design for radiative heat source in two-dimensional participating media. *Numerical Heat Transfer, Part B*. 2004, 46(3): 283-300.
- [15] Qi H, Ruan LM, Zhang HC, Wang YM., Tan HP Inverse radiation analysis in a one-dimensional participating slab by the stochastic particle swarm optimizer algorithm. *International Journal of Thermal Sciences*. 2007, 46: 649-661.
- [16] Qi H, Ruan LM, Shi M, An W, Tan HP Application of multi-phase particle swarm optimization technique to inverse radiation problem. *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2008, 109: 476-493.
- [17] Das R, Mishra SC, Ajith M, Uppaluri R An inverse analysis of a transient 2-D conduction-radiation problem using the lattice Boltzmann method and the finite volume method coupled with the genetic algorithm. *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2008, 109(11): 2060-2077.
- [18] Mirsephai A, Mohammadzahari M, Chen L, O'Neill B An artificial intelligence approach to inverse heat transfer modeling of an irradiative dryer. *International Communications in Heat and Mass Transfer*. 2012, 39(1): 40-45.
- [19] Safavinejada A, Mansouria SH, Sakuraib A, Maruyamab S Optimal number and location of heaters in 2-D radiant enclosures composed of specular and diffuse surfaces using micro-genetic algorithm. *Applied Thermal Engineering*. 2009, 29(5-6): 1075-1085.
- [20] Zhang Biao B, Qi Hong H, Ren YT, Sun SC, Ruan LM Application of homogenous continuous ant colony optimization algorithm to inverse problem of one-dimensional coupled radiation and conduction heat transfer. *International Journal of Heat and Mass Transfer*, 2013, 66(4): 507-516.
- [21] Ren YT, Qi H, Huang X, Wang W, Ruan LM, Tan HP Application of improved krill herd algorithms to inverse radiation problems. *International Journal of Thermal Sciences*. 2016. DOI:10.1016/j.ijthermalsci.2015.12.009.
- [22] He ZZ, Qi H, Yao YC, Ruan LM Inverse estimation of the particle size distribution using the fruit fly optimization algorithm. *Applied Thermal Engineering*. 2015, 88: 306-314.

- [23] Moparthi A, Das R, Uppaluri R, Mishra SC Optimization of heat fluxes on the heater and the design surfaces of a radiating-conducting medium. *Numerical Heat Transfer, Part A*. 2009, 56(10): 846-860.
- [24] Amiri H, Mansouri SH, Safavinejad A, Coelho PJ.. The optimal number and location of discrete radiant heaters in enclosures with the participating media using the micro genetic algorithm. *Numerical Heat Transfer, Part A*. 2011, 60(5): 461-483.
- [25] Sarvari SMH Optimal geometry design of radiative enclosures using the genetic algorithm. *Numerical Heat Transfer Part A*. 2007, 52: 127-143.
- [26] Kennedy J, Eberhart RC Particle Swarm Optimization. *IEEE International Conference on Neural Networks*. 1995: 1942-1948.
- [27] Farahmand A, Payan S, Sarvari SMH Geometric optimization of radiative enclosures using PSO algorithm. *International Journal of Therman Sciences*. 2012, 60: 61-69.
- [28] Shi Y, Eberhart RC, A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation*. 1998: 69-73.
- [29] Shi Y, Eberhart RC Empirical study of particle swarm optimization. *IEEE Congress on Evolutionary Computation*. 1999: 1945-1950.
- [30] Zeng JC, Cui ZH A guaranteed global convergence particle swarm optimizer. *Journal of Computer Research and Development*. 2004, 41(8): 1333-1338.
- [31] Cui ZH, Zeng JC, Cai XJ A new stochastic particle swarm optimizer. *IEEE Congress on Evolutionary Computation*. 2004: 316-319.
- [32] Storn R, Price K Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *International Computer Science Institute, Berkeley*. 1995.
- [33] Akima H A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the Association for Computing Machinery*. 1970, 17: 589-602.
- [34] Modest MF Radiative heat transfer. New York: McGraw-Hill; 1993.
- [35] Liu LH, Tan HP Inverse radiation problem in three-dimensional complicated geometric systems with opaque boundaries. *Journal of Quantitative Spectroscopy and Radiative Transfer*. 2001, 68: 559-573.
- [36] Chai JC, Lee HS, Patankar SV Finite volume radiative heat transfer procedure for irregular geometries. *Journal of Thermophysics and Heat Transfer*. 1995, 9(3): 410-415.



---

# Shape Optimization of Busemann-Type Biplane Airfoil for Drag Reduction Under Non-Lifting and Lifting Conditions Using Genetic Algorithms

---

Yi Tian and Ramesh K. Agarwal

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62811>

---

## Abstract

The focus of this chapter is on the shape optimization of the Busemann-type biplane airfoil for drag reduction under both non-lifting and lifting conditions using genetic algorithms. The concept of biplane airfoil was first introduced by Adolf Busemann in 1935. Under design conditions at a specific supersonic flow speed, the Busemann biplane airfoil eliminates all wave drag due to its symmetrical biplane configuration; however, it produces zero lift. Previous research has shown that the original Busemann biplane airfoil shows a poor performance under off-design conditions. In order to address this problem of zero lift and to improve the off-design-condition performance, shape optimization of an asymmetric biplane airfoil is performed. The commercially available computational fluid dynamics (CFD) solver ANSYS FLUENT is employed for computing the inviscid supersonic flow past the biplane airfoil. A single-objective genetic algorithm (SOGA) is employed for shape optimization under the non-lifting condition to minimize the drag, and a multi-objective genetic algorithm (MOGA) is used for shape optimization under the lifting condition to maximize both the lift and the lift-to-drag ratio. The results obtained from both SOGA and MOGA show a significant improvement in the design and off-design-condition performance of the optimized Busemann biplane airfoil compared to the original airfoil.

**Keywords:** genetic algorithm, shape optimization, biplane airfoil, computational fluid dynamics, supersonic flow

## 1. Introduction

For decades, the speed of commercial aircrafts has been bounded by the sound barrier. Even the most successful supersonic transport (SST) plane, the Concorde, could only be deployed in very few routes due to government regulation, low efficiency, and excessive noise generation. Since the retirement of Concorde in 2003, the desire for developing a replacement for Concorde space still remains. In order to accomplish that, one of the biggest design challenges is to eliminate, or at least greatly reduce, the strong bow shock wave generated during supersonic flight, which can cause high wave drag and substantial noise. At supersonic speed, a bow shock is generated ahead of the airplane. This shock wave generates a substantially high wave drag which needs to be overcome by the engine by providing a much higher thrust compared to that for a conventional subsonic/transonic airplane, which results in higher fuel consumption and low propulsive efficiency. In order to address the high-wave-drag problem at supersonic speed, a biplane concept was proposed by Adolf Busemann in 1935 [1] which can potentially avoid the formation of bow shock and thus does not create sonic boom.

During the period from 1935 to 1960, substantial research was conducted on the Busemann biplane concept. Moeckel [2] and Licher [3] performed the theoretical analysis of the optimized lifting supersonic biplanes. Tan [4] took a further step and derived analytical expressions for the drag and lift of a three-dimensional supersonic biplane with a finite span of rectangular plan form. Some experimental results were obtained by Ferri [5] using a wind tunnel, and comparisons were made between the experimental and analytical results. During the past 10 years, considerable interest has been shown in supersonic biplane airfoils. Igra and Arad [6] tested and analyzed the effects of different parameters on the drag coefficient of the Busemann airfoil under various flow conditions. Kusunose et al. [7] proposed a concept for the next-generation supersonic transport using the Busemann biplane design. A series of studies using both computational fluid dynamics (CFD) and wind-tunnel experimental methods have been completed by Kusunose's research group [7–16].

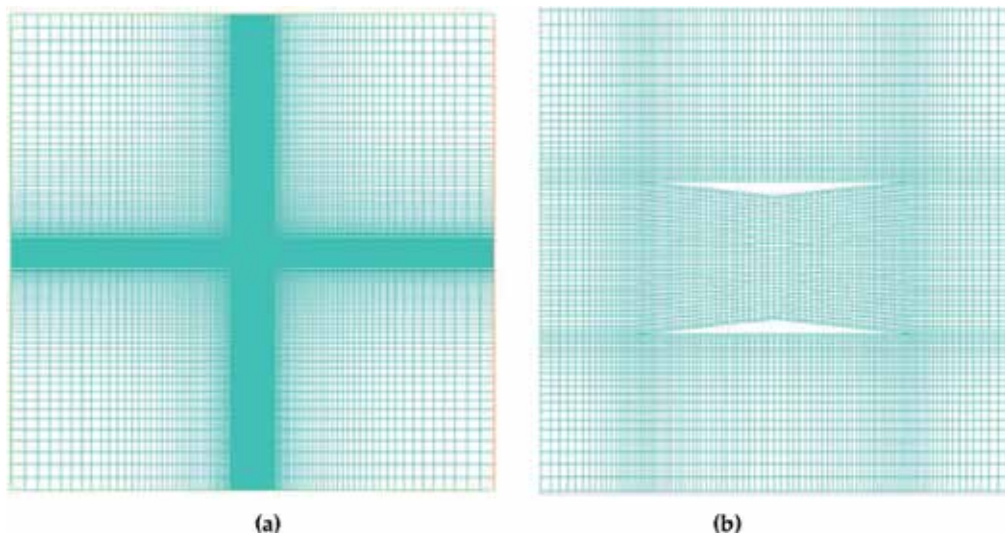
Recently, Hu et al. [17] employed a multi-point adjoint-based aerodynamic design and optimization method to improve the baseline Busemann biplane airfoil's off-design performance and alleviate the flow-hysteresis problem. They also addressed the problem of minimizing its drag by shape optimization. This chapter addresses the aerodynamic shape optimization of two-dimensional symmetric Busemann biplane airfoil under both design and off-design conditions for reducing its wave drag, and the alleviation of the flow-hysteresis and choked-flow effects using a single-objective genetic algorithm. The symmetric Busemann biplane airfoil generates zero lift; this problem is addressed by introducing asymmetry in the shape of two profiles of the biplane. The asymmetric configuration is shape optimized for maximum lift and minimum drag by employing a multi-objective genetic algorithm. The flow field is computed using the commercial CFD software ANSYS FLUENT. Body-fitted H-grids around the airfoils are generated using the ICEM software. Random airfoil shapes with constraints in a given generation of the genetic algorithm are obtained using the Bezier curves (third-order polynomials).

## 2. Flow field simulation of the standard diamond-shaped airfoil and the baseline Busemann biplane airfoil under design and off-design conditions

The flow fields of both the standard diamond-shaped airfoil and the Busemann biplane airfoil are computed at zero angle of attack (the zero-lift condition). For meaningful comparison, the total thickness of the two airfoils is set to be equal. For this specific case under consideration, the thickness-to-chord ratio of the diamond-shaped airfoil is  $t/c = 0.1$ , while the thickness-to-chord ratio of the Busemann airfoil is  $t/c = 0.05$  for both its upper and lower components. The distance between the upper and lower components of the Busemann airfoil is set to be one half of the chord length to obtain the theoretical minimum drag under the design condition at Mach number  $M = 1.7$ . The angle of attack is set to be 0 as we first consider the non-lifting design and off-design conditions by varying the Mach number below 1.7 and greater than 1.7.

### 2.1. Mesh generation

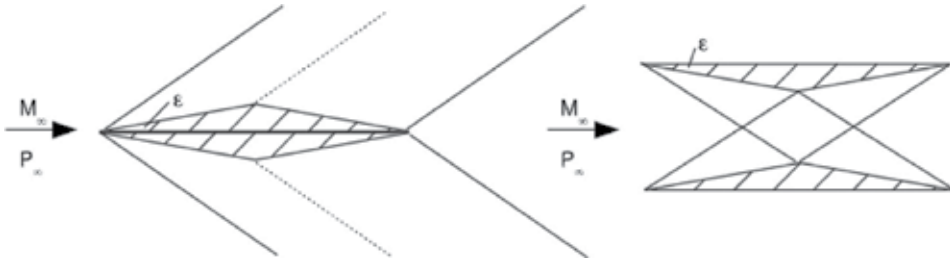
The commercial meshing software ANSYS ICEM CFD is used to generate the mesh for computing the flow field. The far-field boundary is set at 21 chord length by a 20.5-chord-length rectangle from the center of the airfoil. **Figure 1** shows the H-mesh configuration generated for the simulations. There are 64 nodes around the two components both horizontally and vertically. In between the two components, the grid has a dimension of 32 nodes (vertically)  $\times$  64 nodes (horizontally). An ICEM replay script file is created to automatically generate the mesh for the flow past different airfoil shapes once it is called.



**Figure 1.** (a) H-mesh generated in the computational domain and (b) zoomed-in view of the mesh near the biplane airfoil.

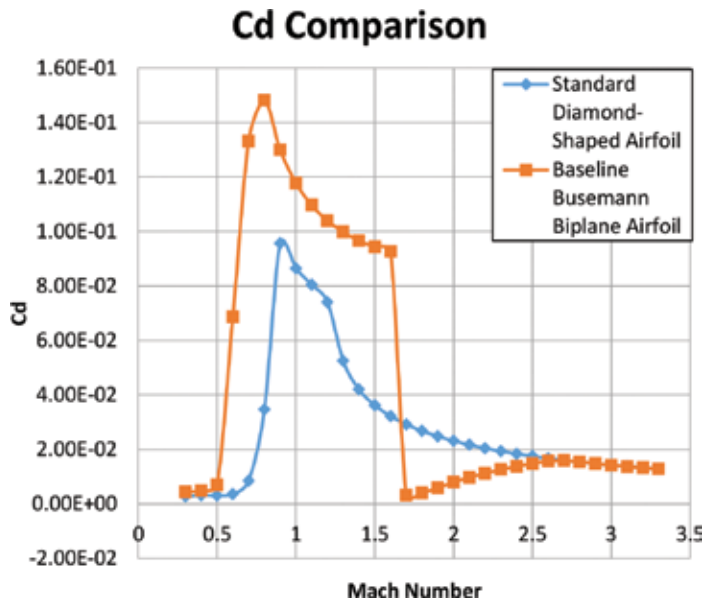
## 2.2. Steady-state flow field simulation at $M < 1.7$

As has been mentioned above, the baseline Busemann biplane airfoil has a much lower drag under its design condition at Mach number  $M = 1.7$ . However, it shows a poor aerodynamic performance under off-design conditions, which may cause much higher drag compared to the standard diamond-shaped airfoil. The configurations of both the standard diamond-shaped airfoil and the baseline Busemann biplane airfoil are shown in **Figure 2**.



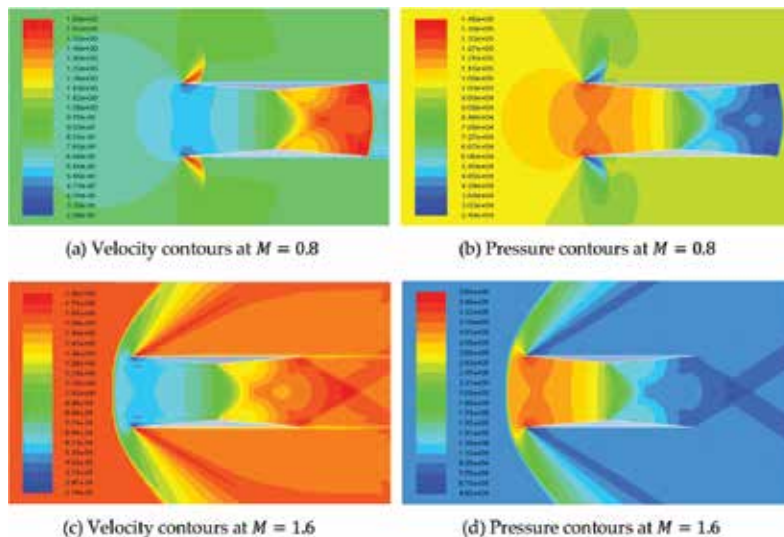
**Figure 2.** Configurations of standard diamond-shaped airfoil (left) and baseline Busemann biplane airfoil (right).

**Figure 3** shows the drag coefficient  $c_d$  of both the standard diamond-shaped airfoil and the baseline Busemann biplane airfoil under the zero-lift condition over a range of Mach numbers ( $0.3 \leq M \leq 3.3$ ). The simulations are performed using ANSYS FLUENT; the flow field is initialized with an impulsive uniform flow.



**Figure 3.** Comparison of  $C_d$  for two different airfoils under non-lifting condition.

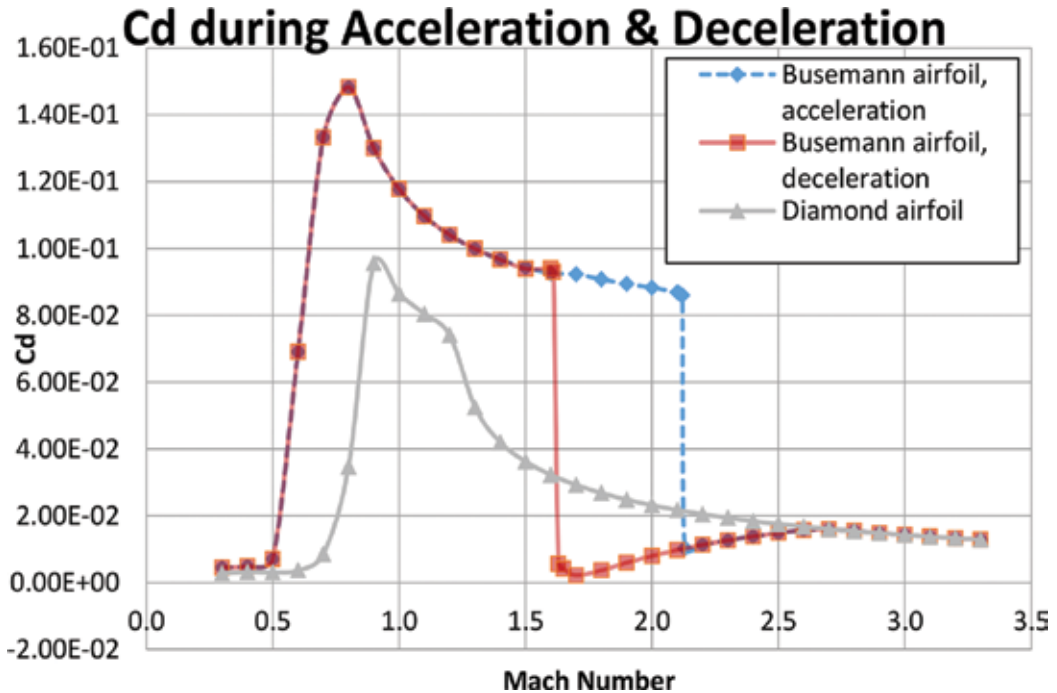
As shown in **Figure 3**, the baseline Busemann biplane airfoil has a higher drag compared to the standard diamond-shaped airfoil when the Mach number is low ( $0.3 \leq M \leq 1.6$ ). In the range of  $1.7 \leq M \leq 2.7$ , however, the drag generated by the baseline Busemann biplane airfoil is smaller than that generated by the standard diamond-shaped airfoil, especially at  $M = 1.7$  which is the design condition for the Busemann airfoil. **Figure 3** demonstrates the advantage of the Busemann biplane airfoil as it produces much lower drag near its design condition ( $M = 1.7$ ) due to the wave-reduction and wave-cancellation effects. The simulations shown above demonstrate that the baseline Busemann biplane airfoil shows a very good performance under its design conditions ( $M = 1.7$ ), while it has a much higher drag under off-design conditions due to the choked-flow phenomenon at lower Mach numbers ( $M < 1.7$ ). **Figure 4(a)** through **4(d)** shows the flow field around the baseline Busemann biplane airfoil under two different off-design conditions, both of which result in high drag but due to different reasons. At  $M = 0.8$ , as shown in **Figure 4(a)**, the flow between the two airfoil components reaches Mach 1 at the mid-chord of the airfoil, and then increases further downstream to supersonic flow. After reaching the trailing edge of the airfoil, however, the flow speed drops to subsonic and hence forms a vertical shock wave. The supersonic flow in the rear part of the airfoil (after the mid-point) creates a low-pressure region, as shown in **Figure 4(b)**, which leads to a higher wave drag ( $c_d = 0.148$ ) at this off-design condition of  $M = 0.8$ . At  $M = 1.6$ , however, as shown in **Figure 4(c)**, the flow field is very different from that at  $M = 0.8$  under subsonic conditions. A bow shock is formed in front of the leading edge of the airfoil. The flow speed drops to subsonic and a high-pressure region, as shown in **Figure 4(d)**, is created which again generates a high drag ( $c_d = 0.0926$ ).



**Figure 4.** Velocity and pressure contours around the baseline Busemann airfoil: (a) velocity contours at  $M = 0.8$ , (b) pressure contours at  $M = 0.8$ , (c) velocity contours at  $M = 1.6$ , (d) pressure contours at  $M = 1.6$ .

### 2.3. Flow field simulation during acceleration and deceleration

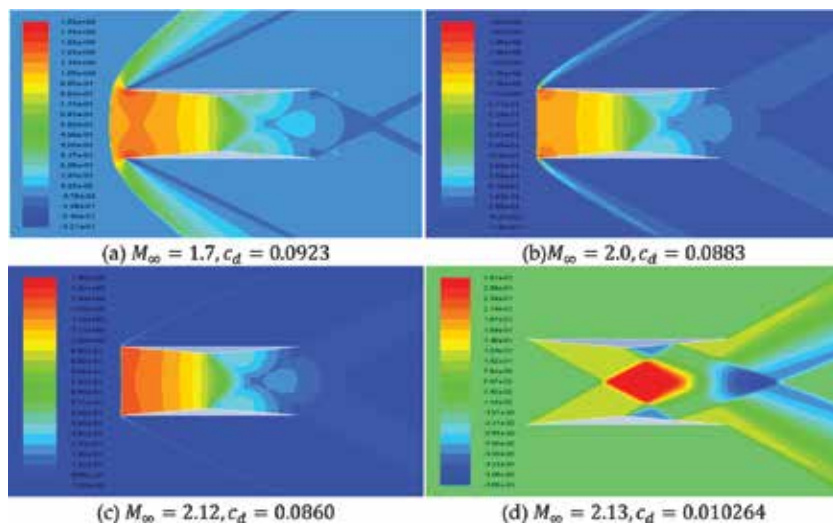
Despite the high drag being generated under off-design conditions, there is an even worse problem caused by the flow-hysteresis phenomenon during acceleration and choked-flow phenomenon during deceleration that need to be addressed under off-design conditions. To demonstrate these phenomena, flow field simulations are conducted under both acceleration and deceleration using the previous simulation results shown in **Figure 4** as the initial condition. **Figure 5** shows two separated  $c_d$  curves during acceleration and deceleration of the biplane airfoil. As shown in **Figure 5**, the change in acceleration and deceleration  $c_d$  curves occurs in the range  $M = 1.6$  to  $M = 2.1$ , where the blue dashed line represents the  $c_d$  of the Busemann biplane airfoil during acceleration and the red solid line represents the  $c_d$  of the Busemann biplane airfoil during deceleration. This separation between the two  $c_d$  curves is caused by the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration. Therefore, in order to minimize the difference in the drag coefficients of the Busemann biplane airfoil during acceleration and deceleration as well as to significantly decrease the drag, shape optimization of the baseline Busemann airfoil is conducted using single-objective genetic algorithm (SOGA) to eliminate the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration; it is discussed in the next section.



**Figure 5.**  $c_d$  plots for diamond-shaped airfoil and baseline Busemann airfoil during acceleration and deceleration under non-lifting conditions.

## 2.4. Flow field of the baseline Busemann biplane airfoil during acceleration

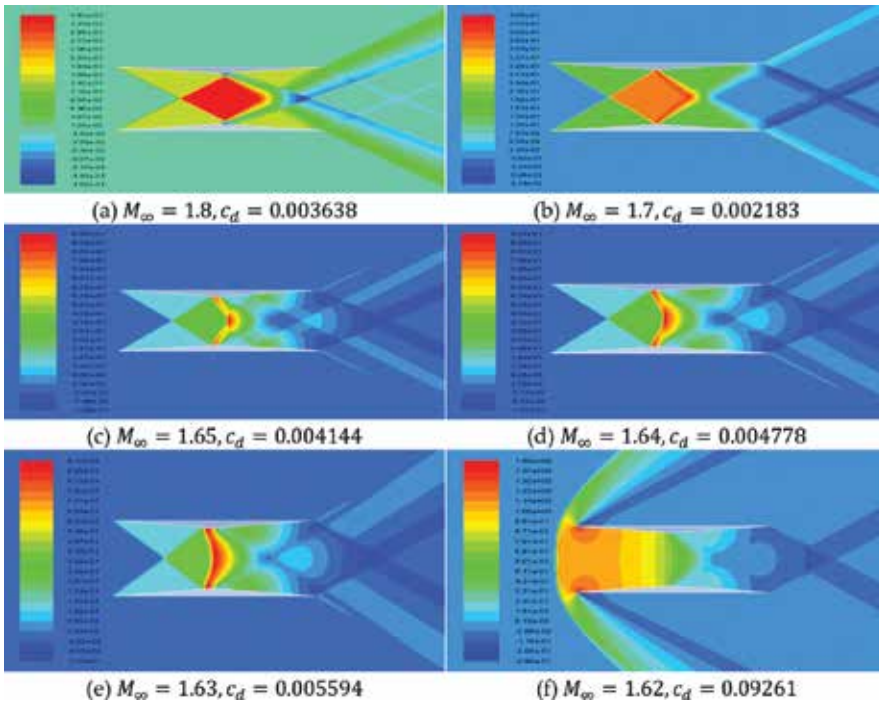
In this section, the flow-hysteresis phenomenon during acceleration is examined for the baseline Busemann biplane airfoil under the non-lifting condition. The pressure coefficient contours for flow past the baseline Busemann biplane airfoil during acceleration are shown in **Figure 6**. To simulate the non-lifting condition, the angle of attack is set to 0. The resulting non-lifting flow field around the baseline Busemann airfoil is shown in **Figure 6** at various supersonic Mach numbers ranging from  $M = 1.7$  to  $M = 2.13$ . As shown in **Figure 6**, the bow shock exists in front of the Busemann airfoil and does not disappear until the Mach number reaches 2.13. It can also be noticed that there is a subsonic region behind the bow shock between the upper and lower components of the airfoil where the pressure coefficients are high. The presence of this bow shock in front of the airfoil results in a substantial increase in the drag compared to that at the design condition. However, when the Mach number increases from 2.12 to 2.13, the bow shock is swallowed into the upper and lower components of the Busemann airfoil and is replaced by two oblique shock waves, and the subsonic region between the two airfoil components finally disappears as shown in **Figure 6(d)**. The drag coefficient of the airfoil also decreases dramatically, and the flow past the airfoil develops into a state similar to that under the design condition. **Figure 6** illustrates the poor performance of the baseline Busemann airfoil under off-design conditions. During acceleration, the design condition cannot be achieved at Mach 1.7. And the drag coefficient at Mach 1.7 is much higher ( $c_d = 0.0923$ ) compared to that of the standard diamond-shaped airfoil ( $c_d = 0.0292$ ). Furthermore, due to the flow-hysteresis phenomenon, the drag coefficients  $c_d$  during acceleration and deceleration are different as shown in **Figure 6**.



**Figure 6.**  $c_p$ -contours around the baseline Busemann biplane airfoil with zero lift during acceleration. (a)  $M_\infty = 1.7$ ,  $c_d = 0.0923$ ; (b)  $M_\infty = 2.0$ ,  $c_d = 0.0883$ ; (c)  $M_\infty = 2.12$ ,  $c_d = 0.0860$ ; (d)  $M_\infty = 2.13$ ,  $c_d = 0.010264$ .

## 2.5. Flow field of the baseline Busemann biplane airfoil during deceleration

In this section, the choked-flow phenomenon of the baseline Busemann biplane airfoil during deceleration is examined. The contours of pressure coefficients for flow past the baseline Busemann airfoil during deceleration are shown in **Figure 7**. To simulate the non-lifting condition, the angle of attack is set to 0 as before. The resulting non-lifting flow field around the baseline Busemann airfoil is shown in **Figure 7** at various supersonic Mach numbers ranging from  $M = 1.8$  to  $M = 1.62$ . As shown in **Figure 7**, a different flow field appears within the small range near the design Mach number ( $M = 1.7$ ); a relatively high drag coefficient still occurs as the Mach number further decreases during deceleration. A strong bow shock is formed in front of the airfoil when the Mach number drops from 1.63 to 1.62, while the drag coefficient increases dramatically from 0.005594 to 0.0926, which is substantially higher than that of the standard diamond-shaped airfoil ( $c_d = 0.03158$ ). The flow between the two components of the airfoil is choked at the location of the maximum thickness of the Busemann airfoil, and a subsonic region is formed. This is also a clear indication of the poor performance of the baseline Busemann biplane airfoil under off-design conditions as the drag coefficient of the baseline Busemann airfoil is much higher than that of the standard diamond-shaped airfoil for  $M \leq 1.62$ .

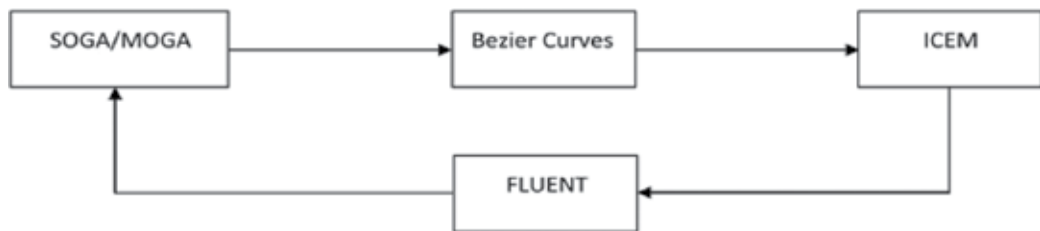


**Figure 7.**  $c_p$ -contours around the baseline Busemann biplane airfoil with zero lift during deceleration: (a)  $M_\infty = 1.8$ ,  $c_d = 0.003638$ ; (b)  $M_\infty = 1.7$ ,  $c_d = 0.002183$ ; (c)  $M_\infty = 1.65$ ,  $c_d = 0.004144$ ; (d)  $M_\infty = 1.64$ ,  $c_d = 0.004778$ ; (e)  $M_\infty = 1.63$ ,  $c_d = 0.005594$ ; (f)  $M_\infty = 1.62$ ,  $c_d = 0.09261$ .

In conclusion, the baseline Busemann biplane airfoil produces a substantially higher drag in the low Mach number range (below the design Mach number of  $M = 1.7$ ). Additionally, it is necessary to accelerate the baseline Busemann biplane airfoil to a much higher Mach number ( $M = 2.13$ ) to reach the shock-wave-swallowing state, while dramatically producing higher drag and largely decreasing its efficiency, and then decelerate to a lower velocity to achieve the design condition at Mach number  $M = 1.7$ . As a result, the baseline Busemann biplane airfoil design needs to be modified and optimized to avoid or at least reduce the high drag coefficient caused by the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration.

### 3. Optimization of Busemann-type biplane airfoil

In this chapter, the shape optimization procedure for the baseline Busemann biplane airfoil using the genetic algorithms (GAs) and the results of the optimized Busemann-type airfoil under both non-lifting and lifting conditions are presented. The optimization process is established by coupling a single-objective genetic algorithm (SOGA)- or a multi-objective genetic algorithm (MOGA)-based optimization method with the mesh generation software ANSYS-ICEM and the CFD solver ANSYS FLUENT as shown in **Figure 8**.



**Figure 8.** Schematic of information flow in the optimization process.

Every individual (airfoil) in each generation of SOGA/MOGA is represented by a set of control points, which randomly generate the airfoil shape by using the Bezier curves. The mesh generation software ICEM is used to generate a two-dimensional structured mesh around the airfoil as an input to the CFD solver FLUENT, which is then used to calculate the supersonic inviscid flow fields for specific flow conditions. Based on the fitness values of all airfoil shapes in a given generation, SOGA/MOGA is applied to create the next generation of airfoils, and this whole process is repeated until the optimal fitness value is obtained. The airfoil shape that corresponds to the optimal fitness value is the final shape of the optimized airfoil [18].

#### 3.1. Overview of genetic algorithm

Genetic algorithms (GAs) are a class of stochastic optimization algorithms inspired by the biological evolution [19]. They efficiently exploit historical information to speculate on new offspring with improved performance [20]. For the Busemann biplane airfoil in particular,

genetic algorithms are used to generate new shapes that produce much lower drag by minimizing the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration. Generally, The GA employs the following steps to complete the optimization process:

1. Initialization: randomly generates a group of individuals.
2. Evaluation: evaluates the fitness of each individual generated.
3. Natural selection: individuals who have the lowest fitness are removed.
4. Reproduction: pairs of individuals are picked to produce the offspring, which is often done by roulette wheel sampling. A crossover function is then used to produce the offspring.
5. Mutation: randomly modifies some small percentage of the population.
6. Check for convergence: if the current generation has converged, the individual having the best fitness will be returned. Otherwise, the process will be repeated starting from Step 2 until the predefined tolerance criteria for acceptable change in fitness between generations are met.

### 3.1.1. Application of single-objective genetic algorithm (SOGA)

For the non-lifting condition, when the lift coefficient  $c_l = 0$ , a single-objective genetic algorithm (SOGA) is employed for the shape optimization of the Busemann biplane airfoil. The single objective to be achieved is to minimize the drag coefficient  $c_d$ . Multiple design points (Mach numbers) are used during the optimization process due to the flow-hysteresis and choked-flow phenomena caused by the baseline Busemann biplane airfoil during acceleration and deceleration, respectively. For the multi-point optimization, the fitness function employed is a weighted average of the drag coefficients  $c_d$ , which can be written in the following form:

$$I = \frac{\sum_{i=1}^n w_i I_i}{\sum_{i=1}^n w_i} \quad (1)$$

where  $I$  is the weighted average drag coefficient, and “ $i$ ” denotes a design point (related to Mach number). For the optimization process under the non-lifting condition, an evenly weighted average of drag coefficients  $c_d$  at different Mach numbers is employed. Therefore, Eq. (1) reduces to

$$I = \frac{\sum_{i=1}^n I_i}{n} \quad (2)$$

As the wave drag of the Busemann biplane airfoil is much lower when the flow is unchoked, it is highly desirable that the strong bow shock wave in front of the airfoil is swallowed into

the area between the upper and lower components of the airfoil before the flow speed approaches the design Mach number ( $M = 1.7$ ). Although a higher weight can be assigned to the most important design Mach number ( $M = 1.7$ ) to produce a lower drag at that design point, previous research conducted by Hu et al. [17] has shown that a slightly higher drag coefficient  $c_d$  is obtained at lower Mach numbers during acceleration and deceleration, as given in **Tables 1** and **2**, respectively, if such an uneven weighting method is used.

Mach number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
Baseline	1050	996	957	928	906	889	873
Optimized (even)	527	473	419	376	332	112	106
Optimized (uneven)	539	486	428	382	336	107	101

**Table 1.** Comparison of  $c_d$  at under zero-lift conditions during acceleration (1 count = 0.0001) [17].

Mach number	1.7	1.6	1.5	1.4	1.3	1.2	1.1
Baseline	873	889	906	928	957	996	1050
Optimized (even)	106	112	127	152	419	473	527
Optimized (uneven)	101	107	122	146	428	486	539

**Table 2.** Comparison of  $c_d$  at under zero-lift conditions during deceleration (1 count = 0.0001) [17].

GA parameters	Description
Generation size	8 individuals per generation
Number of generations	Maximum of 50 generations if convergence not obtained
Number of design variables	14 in total, 7 for acceleration and 7 for deceleration
Selection type	Roulette Wheel Selection
Crossover rate	0.7
Mutation rate	0.1
Error of mutation constant	0.8, which determines how much mutation affects the curves as generations proceed

**Table 3.** GA parameters for shape optimization of Busemann biplane airfoil under non-lifting conditions.

Here, we have a total of seven design points in the Mach numbers ranging from  $M = 1.1$  to  $M = 1.7$ , for both acceleration and deceleration. The reason for picking these seven design points is that this range of Mach numbers,  $M = 1.1$  to  $M = 1.7$ , provides the critical region before the Mach number increases to the design condition  $M = 1.7$ ; it is the region in which we want to maintain the drag coefficient as low as possible. Both acceleration and deceleration scenarios are considered in order to reduce the flow-hysteresis phenomenon and the choked-flow phenomenon, respectively.

A code in the MATLAB package is developed and utilized in the optimization process of the airfoil as well as for ICEM meshing and FLUENT flow field calculations. All SOGA parameters are defined based on the GA methodology, as shown in **Table 3**.

### 3.1.2. Airfoil parameterization

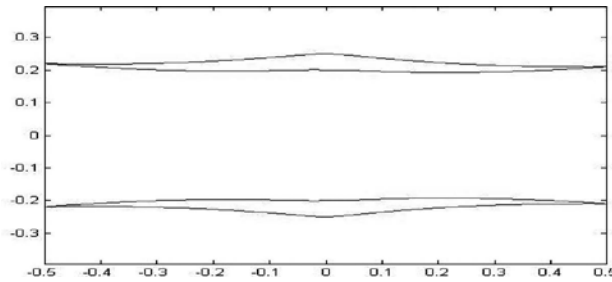
The shape of the airfoil is generated by using the Bezier curves (third-order polynomials). Bezier curves are frequently used in computer graphics to obtain curves that appear to be reasonably smooth at all scales. One of the main reasons that Bezier curves are used in computer graphics is that they can be efficiently constructed; each Bezier curve is simply defined by a set of control points [21].

For the non-lifting case, with  $c_l = 0$ , since the upper and lower components of the airfoil are symmetric with respect to the horizontal axis, only the upper component of the airfoil needs to be defined; the lower component is the mirror image with respect to the horizontal axis. For the upper component of the airfoil, there are two lines that need to be drawn to define its shape. Since the thickness distribution for the entire airfoil remains the same as that of the baseline Busemann biplane airfoil, the  $y$ -coordinates of the lower line are defined by Eqs. (3) and (4). Thus, the upper line (which is a straight horizontal line for the baseline Busemann airfoil) is the only line that needs to be generated by the Bezier curves in order to define the shape of the whole biplane airfoil.

$$y_{low} = y_{up} - 0.05 - 0.1 \times x_{up} \quad (-0.5 \leq x_{up} \leq 0) \quad (3)$$

$$y_{low} = y_{up} - 0.05 + 0.1 \times x_{up} \quad (0 \leq x_{up} \leq 0.5) \quad (4)$$

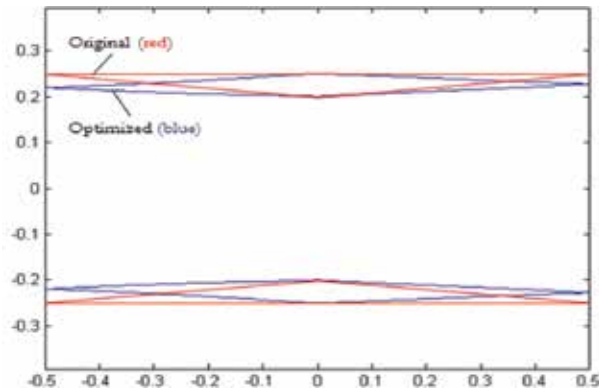
In Eqs. (3) and (4), the subscripts “low” and “up” correspond to the lower and upper lines of the airfoil; the origin is at the center of the upper line. Two Bezier curves are used to generate the shape-defining upper line. Each Bezier curve is defined by a set of four control points. Each control point is constrained by a specified range of  $x$ - and  $y$ -coordinates. **Figure 9** shows a randomly generated Busemann-type biplane airfoil shape using Bezier curves.



**Figure 9.** Randomly generated Busemann-type biplane airfoil shape using Bezier curves.

### 3.1.3. Optimization results

After implementing SOGA for 20 generations with eight individuals in each generation, an optimal shape result for symmetric Busemann-type biplane airfoil under the non-lifting condition with a minimum drag is obtained. **Figure 10** shows the geometry of the original Busemann biplane airfoil (red) and the optimized Busemann biplane airfoil (blue) under the non-lifting condition.



**Figure 10.** Geometry of both the original and optimized Busemann biplane airfoil under non-lifting conditions.

Mach number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
Baseline	1097	1040	999	967	943	926	923
Optimized (GA)	622	506	428	389	132	114	104

**Table 4.** Comparison of  $c_d$  for the original and optimized Busemann biplane airfoil under non-lifting conditions during acceleration (1 count = 0.0001).

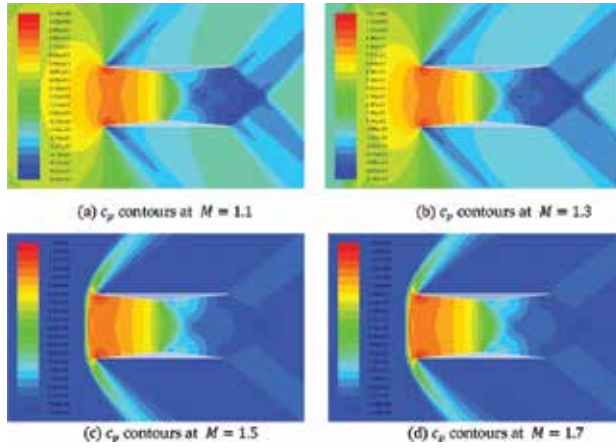
Mach number	1.7	1.6	1.5	1.4	1.3	1.2	1.1
Baseline	32	926	940	967	999	1040	1098
Optimized (GA)	104	114	132	163	428	506	621

**Table 5.** Comparison of  $c_d$  for the original and optimized Busemann biplane airfoil under non-lifting conditions during deceleration (1 count = 0.0001).

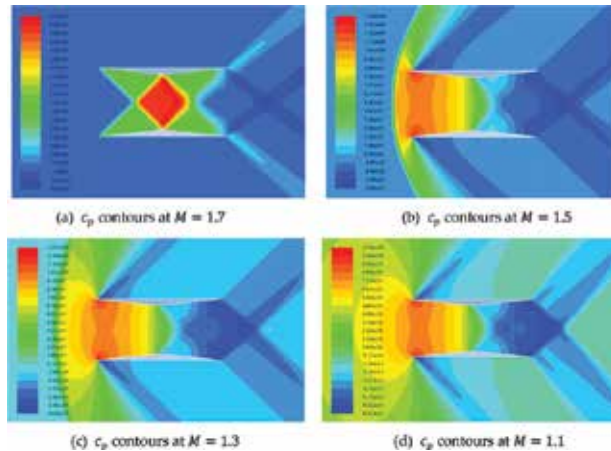
The drag coefficients for the seven design points are compared in **Tables 4** and **5** for both the original and optimized Busemann biplane airfoil under the non-lifting condition during acceleration and deceleration, respectively. As shown in **Tables 4** and **5**, the baseline Busemann biplane airfoil is choked at all Mach numbers within the optimization range, while the optimized Busemann biplane airfoil un-chokes at  $M = 1.5$  during acceleration and chokes at  $M = 1.3$  during deceleration. Even under choked conditions during both acceleration and

deceleration, the optimized Busemann biplane airfoil has significantly lower drag compared to the baseline Busemann biplane airfoil. The only point where the optimized Busemann biplane airfoil has a higher drag compared to the original airfoil is at  $M = 1.7$  during deceleration.

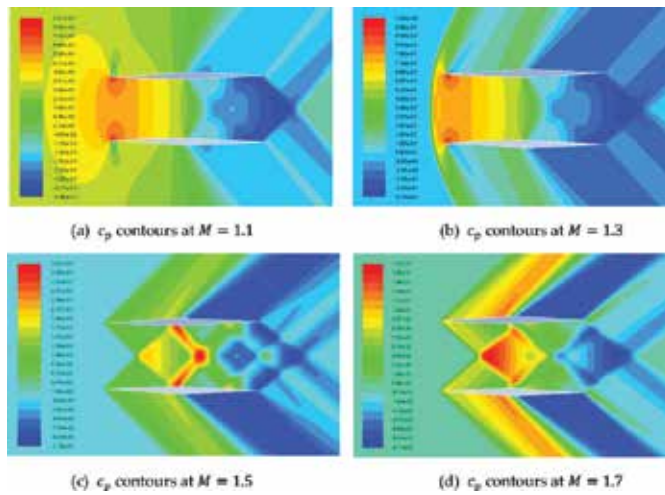
**Figures 11** and **12**, respectively, show the change in the pressure coefficient  $c_p$  around the original Busemann biplane airfoil as the Mach number increases and decreases within the design-point range. The corresponding  $c_p$  contours for the optimized Busemann biplane airfoil are shown in **Figures 13** and **14** to illustrate the wave-cancellation effect.



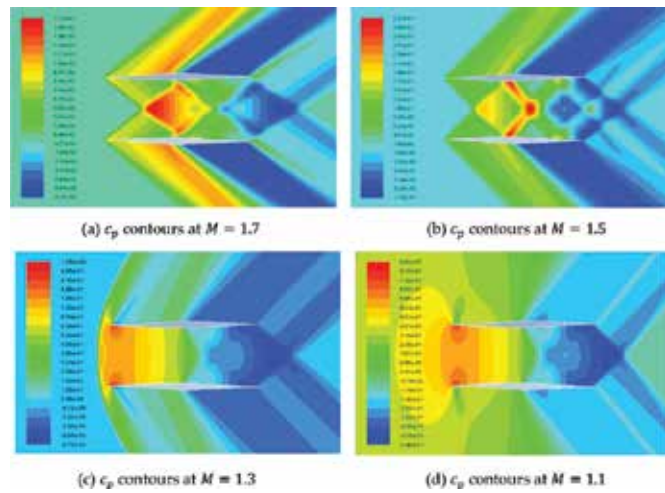
**Figure 11.**  $c_p$  contours around the original Busemann biplane airfoil during acceleration: (a)  $c_p$  contours at  $M = 1.1$ ; (b)  $c_p$  contours at  $M = 1.3$ ; (c)  $c_p$  contours at  $M = 1.5$ ; (d)  $c_p$  contours at  $M = 1.7$ .



**Figure 12.**  $c_p$  contours around the original Busemann biplane airfoil during deceleration: (a)  $c_p$  contours at  $M = 1.7$ ; (b)  $c_p$  contours at  $M = 1.5$ ; (c)  $c_p$  contours at  $M = 1.3$ ; (d)  $c_p$  contours at  $M = 1.1$ .



**Figure 13.**  $c_p$  contours around the optimized Busemann biplane airfoil during acceleration: (a)  $c_p$  contours at  $M = 1.1$ ; (b)  $c_p$  contours at  $M = 1.3$ ; (c)  $c_p$  contours at  $M = 1.5$ ; (d)  $c_p$  contours at  $M = 1.7$ .



**Figure 14.**  $c_p$  contours around the optimized Busemann biplane airfoil during deceleration: (a)  $c_p$  contours at  $M = 1.7$ ; (b)  $c_p$  contours at  $M = 1.5$ ; (c)  $c_p$  contours at  $M = 1.3$ ; (d)  $c_p$  contours at  $M = 1.1$ .

**Figure 15** shows the comparison of the drag coefficients for the standard diamond-shaped airfoil, the baseline Busemann biplane airfoil, and the optimized Busemann biplane airfoil under the non-lifting condition. As shown in this figure, the separation between the acceleration and deceleration curves for  $c_d$  still exists for the optimized Busemann biplane airfoil, which means that the flow-hysteresis and the choked-flow effects are not totally eliminated. However, as clearly shown in **Figure 15**, the flow-hysteresis area has been significantly reduced, and the drag increase during deceleration due to the choked-flow phenomenon is much

smaller than that for the original Busemann biplane airfoil. The drag of the optimized Busemann biplane airfoil in the subsonic region is also smaller than that of the original Busemann biplane airfoil, although it is slightly higher than that of the standard diamond-shaped airfoil with  $0.6 < M < 0.85$ . Under both subsonic and supersonic conditions, the optimized Busemann biplane airfoil has been able to significantly reduce the wave drag compared to the original Busemann biplane airfoil. At the design condition Mach number of 1.7, however, the drag coefficient of the optimized Busemann biplane airfoil ( $c_d = 0.01038$ ) is much higher than that of the original Busemann biplane airfoil  $c_d = 0.002182$ . This is due to the fact that for the shape optimization, our focus has been on reducing the flow-hysteresis and choked-flow effects, and we chose to assign equal weights to all Mach numbers used as the multiple design points. To address this problem, we could have put more weight on the design condition ( $M = 1.7$ ) during the optimization process.

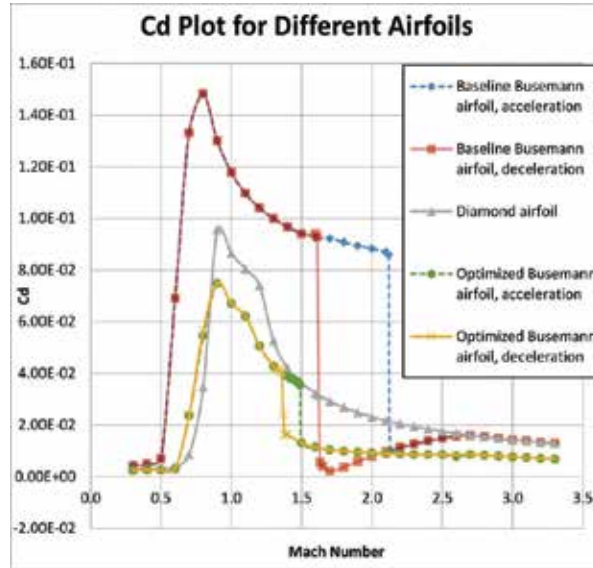
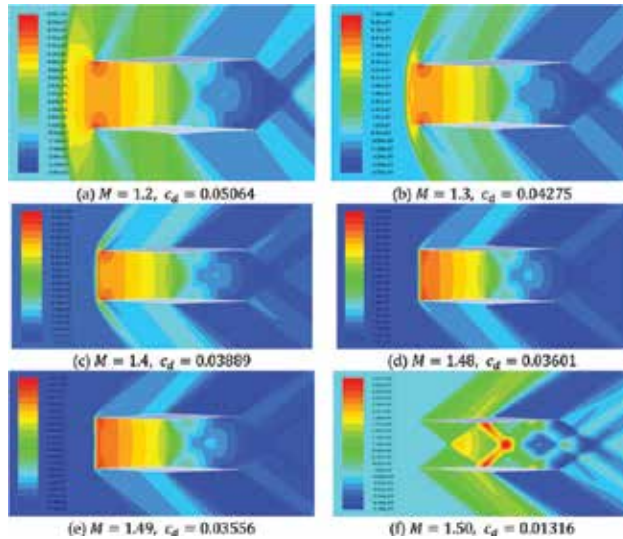


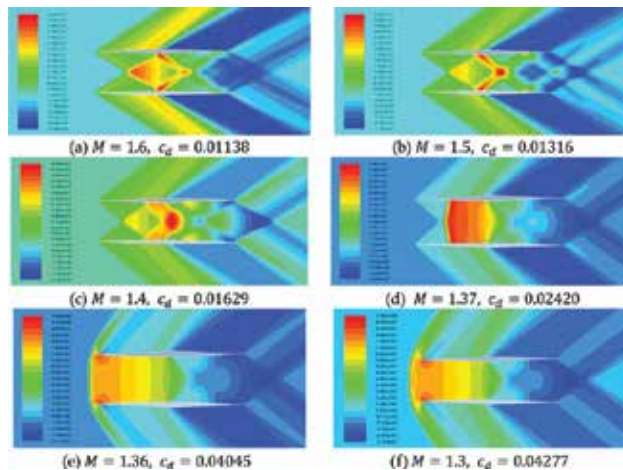
Figure 15.  $c_d$  plot for different airfoils under nonlifting condition.

Next, we examine the details of the flow field during acceleration and deceleration for the optimized Busemann biplane airfoil and compare them with the original baseline Busemann biplane airfoil and the optimization results obtained by Hu et al. [17] using an adjoint-based optimization technique. **Figures 16** and **17** show the pressure coefficient contours of the optimized Busemann biplane airfoil under acceleration and deceleration, respectively. **Figures 18** and **19** show the pressure coefficient contours of the optimized Busemann biplane airfoil using the adjoint-based technique [17] under acceleration and deceleration, respectively. During acceleration, the flow-hysteresis effect still exists, and a bow shock wave is formed in front of the airfoil. The swallowing of the bow shock wave happens when the Mach number increases from 1.49 to 1.50 in our GA optimization as shown in **Figure 16(f)**; it happens when the Mach number increases from 1.52 to 1.53 in the adjoint-based optimization as shown in

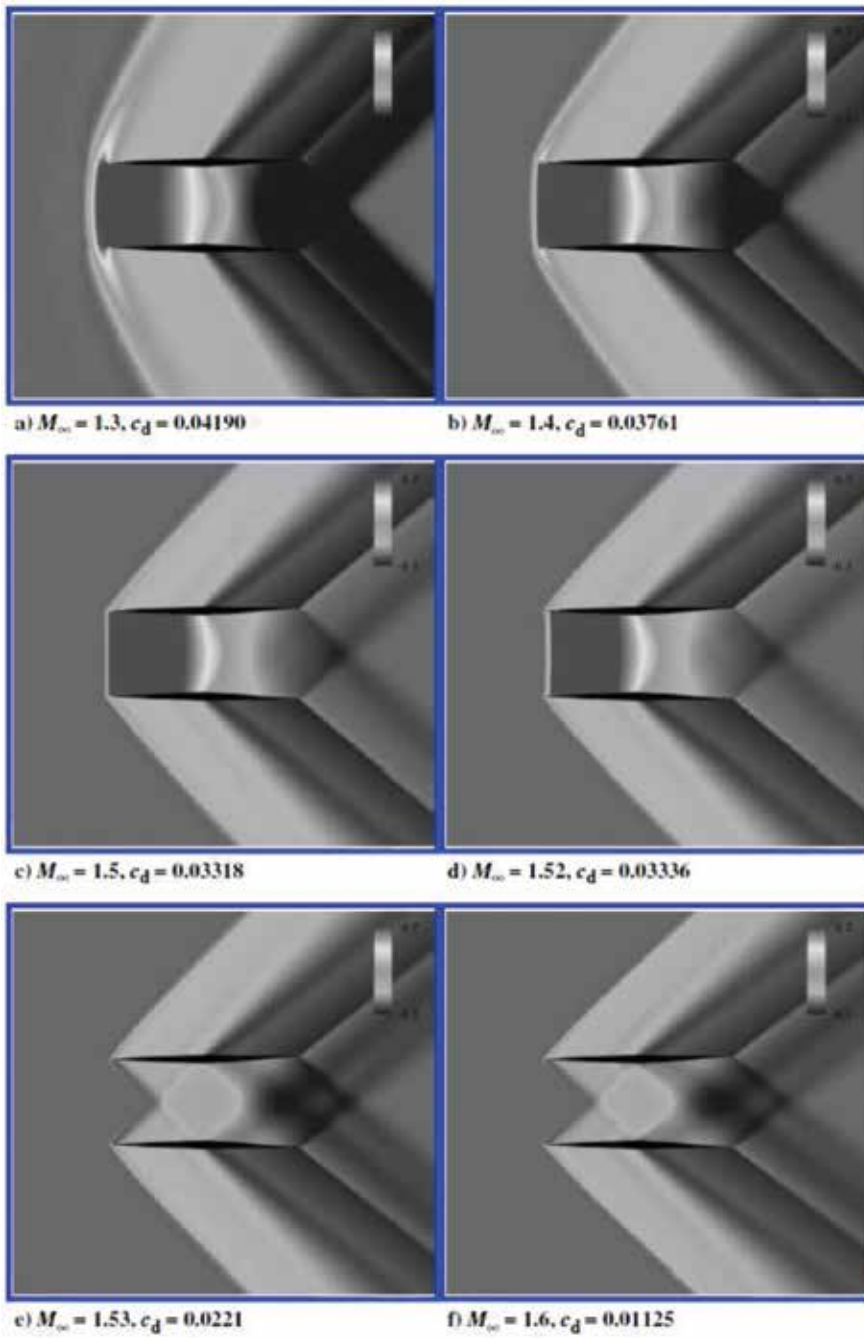
**Figure 18(e)**, and it happens when the Mach number increases from 2.12 to 2.13 for the original Busemann biplane airfoil as shown in **Figure 6(d)**. The drag coefficient decreases from 0.03556 to 0.01316 in the present GA-based optimization and from 0.03336 to 0.01221 in the adjoint-based optimization [17].



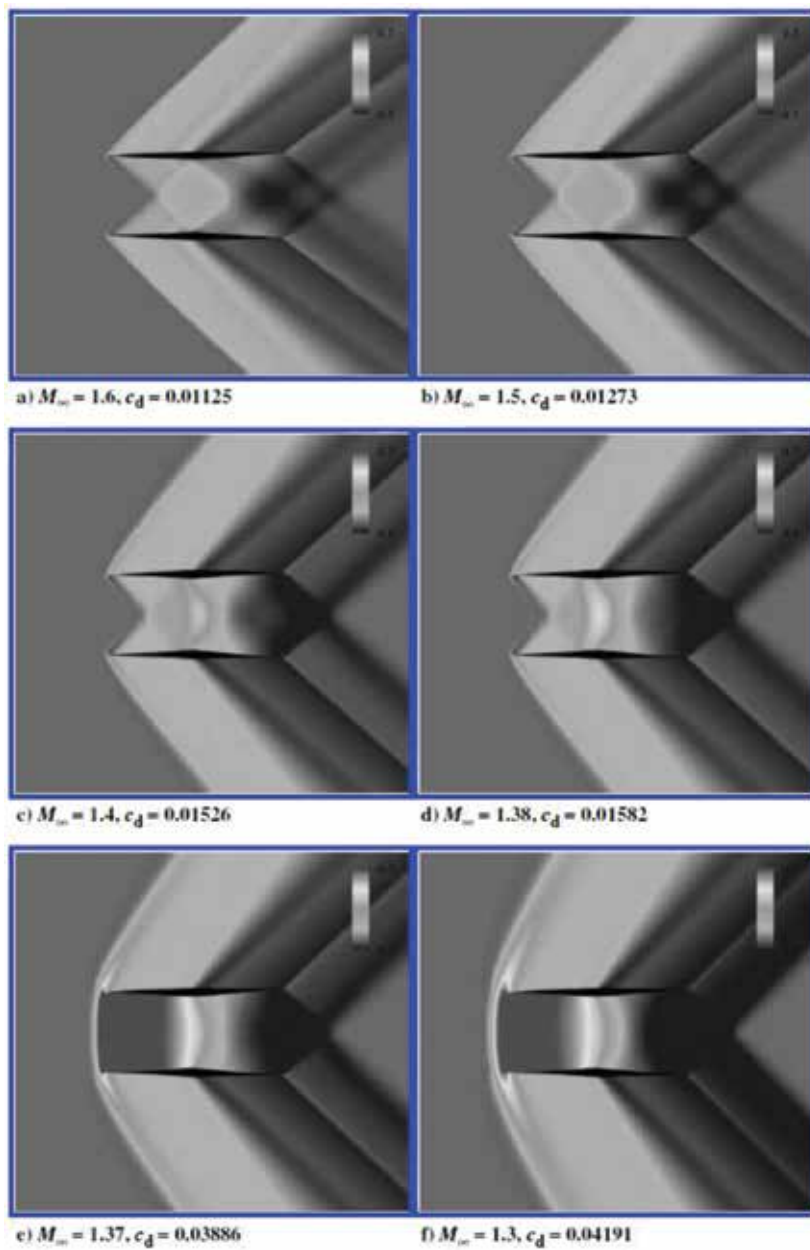
**Figure 16.**  $c_p$  contours of the GA-optimized Busemann biplane airfoil with zero lift during acceleration: (a)  $M = 1.2$ ,  $c_d = 0.05064$ ; (b)  $M = 1.3$ ,  $c_d = 0.04275$ ; (c)  $M = 1.4$ ,  $c_d = 0.03889$ ; (d)  $M = 1.48$ ,  $c_d = 0.03601$ ; (e)  $M = 1.49$ ,  $c_d = 0.03556$ ; (f)  $M = 1.50$ ,  $c_d = 0.01316$ .



**Figure 17.**  $c_p$  contours of the GA-optimized Busemann biplane airfoil with zero lift during deceleration: (a)  $M = 1.6$ ,  $c_d = 0.01138$ ; (b)  $M = 1.5$ ,  $c_d = 0.01316$ ; (c)  $M = 1.4$ ,  $c_d = 0.01629$ ; (d)  $M = 1.37$ ,  $c_d = 0.02420$ ; (e)  $M = 1.36$ ,  $c_d = 0.04045$ ; (f)  $M = 1.3$ ,  $c_d = 0.04277$ .



**Figure 18.**  $c_p$  contours of the adjoint-based-optimized Busemann biplane airfoil with zero lift during acceleration [17]: (a)  $M_\infty = 1.3, c_d = 0.04190$ ; (b)  $M_\infty = 1.4, c_d = 0.03761$ ; (c)  $M_\infty = 1.5, c_d = 0.03318$ ; (d)  $M_\infty = 1.52, c_d = 0.03336$ ; (e)  $M_\infty = 1.53, c_d = 0.0221$ ; (f)  $M_\infty = 1.6, c_d = 0.01125$ .



**Figure 19.**  $c_p$  contours of the adjoint-based-optimized Busemann biplane airfoil with zero lift during deceleration [17]: (a)  $M_{\infty} = 1.6, c_d = 0.01125$ ; (b)  $M_{\infty} = 1.5, c_d = 0.01273$ ; (c)  $M_{\infty} = 1.4, c_d = 0.01526$ ; (d)  $M_{\infty} = 1.38, c_d = 0.01582$ ; (e)  $M_{\infty} = 1.37, c_d = 0.03886$ ; (f)  $M_{\infty} = 1.3, c_d = 0.04191$ .

In conclusion, as shown in **Figure 20**, the drag coefficient of the GA-optimized Busemann biplane airfoil is significantly reduced when compared to the original Busemann biplane airfoil, and it matches with the adjoint-based optimization result obtained by Hu et al. [17].

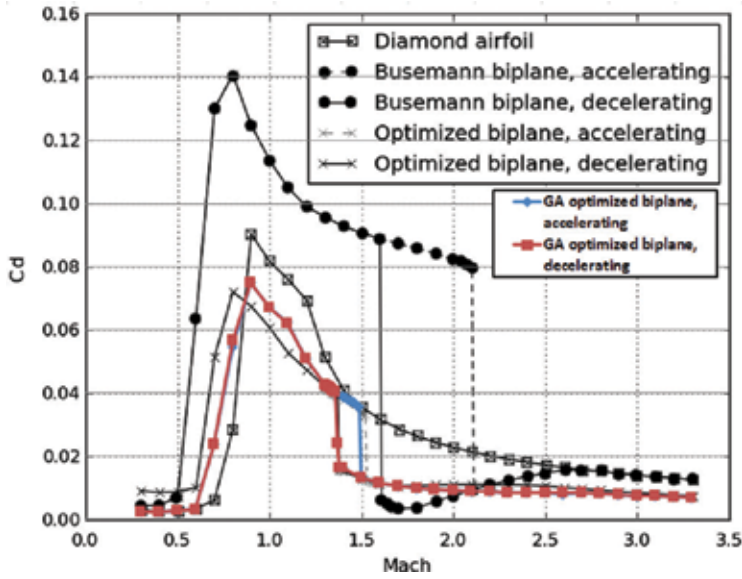


Figure 20.  $c_d$  plot for different airfoils under non-lifting conditions.

### 3.2. Shape optimization of Busemann biplane airfoil under lifting conditions

#### 3.2.1. Application of multi-objective genetic algorithm (MOGA)

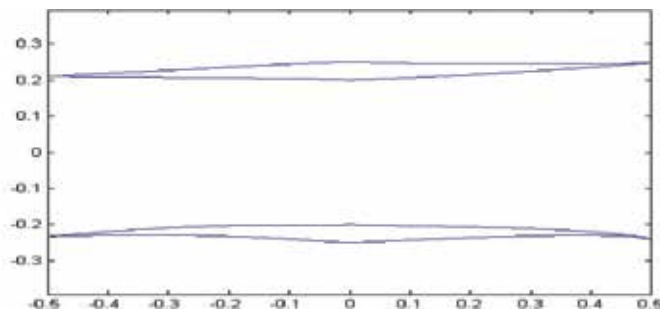
Under the lifting condition with the lift coefficient  $c_l \neq 0$ , a multi-objective genetic algorithm (MOGA) is employed for shape optimization of the original Busemann biplane airfoil. The two objectives to be achieved are to minimize the drag coefficient  $c_d$  while maximizing the lift coefficient  $c_l$ . Similar to that for the non-lifting condition described in Section 3.2, a total of seven design points ranging from  $M = 1.1$  to  $M = 1.7$  are used during the optimization process. For the fitness functions for the lifting case, we use the sum of evenly weighted average of both  $c_d$  and  $c_l$ . The GA parameters used for the lifting case are listed in Table 6.

GA parameters	Description
Generation size	8 individuals per generation
Number of generations	Maximum of 50 generations if convergence not obtained
Number of design variables	28 in total, 14 (7 for $c_d$ & 7 for $c_l$ ) for acceleration and 14 (7 for $c_d$ & 7 for $c_l$ ) for deceleration
Selection type	Roulette Wheel Selection
Crossover rate	0.7
Mutation rate	0.1
Error of mutation constant	0.8, which determines how much mutation affects the curves as generations go on

Table 6. GA parameters for shape optimization of Busemann biplane airfoil under lifting conditions.

### 3.2.2. Airfoil parameterization

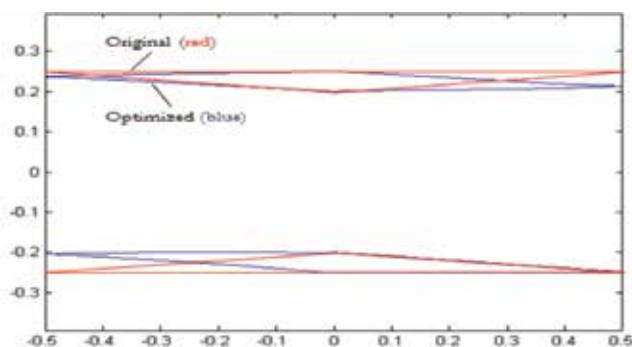
Similar to that for the non-lifting case, the random shapes of the airfoil are generated by using the Bezier curves with control points. For the lifting case with  $c_l \neq 0$ , as the upper and lower components of the biplane airfoil are not symmetric, both the upper and lower components need to be defined separately. The thickness distribution for both the upper and lower components is still maintained the same as for the non-lifting case. Now, a total of four Bezier curves are needed to define the shape of the airfoil. **Figure 21** shows a randomly generated Busemann-type biplane airfoil shape using the Bezier curves for lifting conditions.



**Figure 21.** Randomly generated Busemann-type biplane airfoil shape using the Bezier curves for lifting conditions.

### 3.2.3. Optimization results

After implementing MOGA for 20 generations with eight individuals in each generation, an optimal shape for asymmetric Busemann-type airfoil under lifting conditions with maximum  $c_l$  and minimum  $c_d$  is obtained. **Figure 22** shows the geometry of the original Busemann biplane airfoil (red) and the optimized Busemann biplane airfoil (blue) under lifting conditions.



**Figure 22.** Geometry of both the original and optimized Busemann biplane airfoil under lifting condition.

The drag coefficients for the seven design points are compared in **Tables 7** and **8** for both the original and optimized Busemann biplane airfoil under lifting condition. As shown in

**Tables 7 and 8**, the baseline Busemann biplane airfoil is choked at all Mach numbers within the optimization range, while the optimized Busemann biplane airfoil is unchoked at  $M = 1.6$  during acceleration and is choked at  $M = 1.3$  during deceleration. Even under choked conditions during both acceleration and deceleration, the optimized Busemann biplane airfoil has significantly lower drag compared to the baseline Busemann biplane airfoil. Similar to the non-lifting condition, the only point where the optimized Busemann biplane airfoil has a higher drag compared to the original airfoil is at  $M = 1.7$  during deceleration.

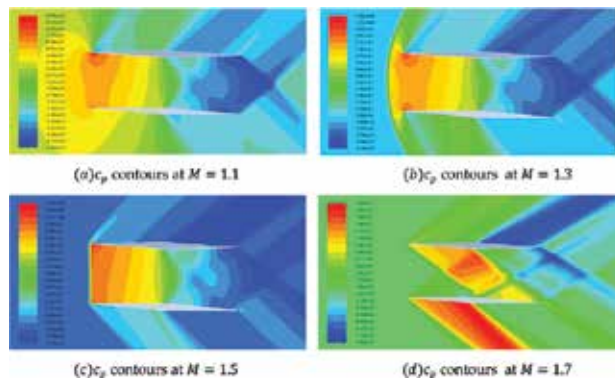
Mach number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
Baseline	1097	1040	999	967	943	926	923
Optimized	683	633	529	461	418	148	136

**Table 7.** Comparison of  $c_d$  for the original and optimized Busemann biplane airfoil under lifting conditions during acceleration (1 count = 0.0001).

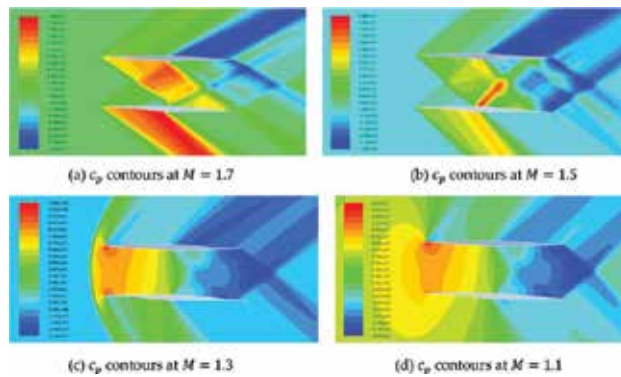
Mach number	1.7	1.6	1.5	1.4	1.3	1.2	1.1
Baseline	32	926	940	967	999	1040	1098
Optimized	136	148	173	224	529	633	682

**Table 8.** Comparison of  $c_d$  for the original and optimized Busemann biplane airfoil under lifting conditions during deceleration (1 count = 0.0001).

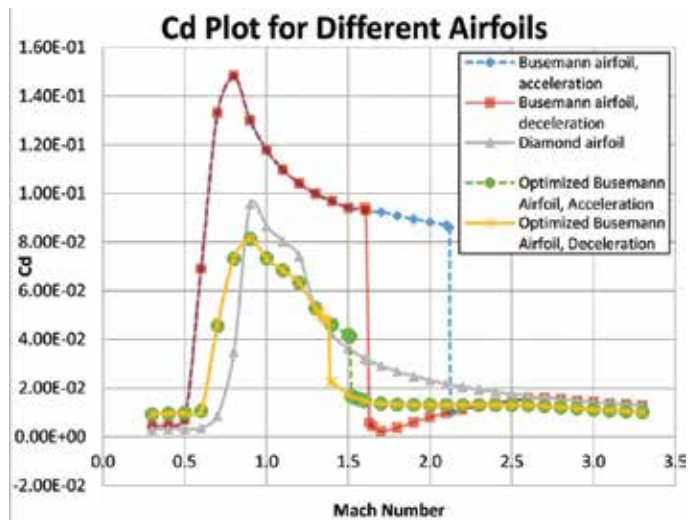
**Figures 23 and 24** show a change in the pressure coefficient  $c_p$  around the optimized Busemann biplane airfoil under lifting condition as the Mach number increases and decreases within the design-point range. As can be seen from these figures, the bow shock wave in front of the airfoil disappears at  $M = 1.6$  during acceleration and is not generated until the flow speed drops down to  $M = 1.3$  during deceleration.



**Figure 23.**  $c_p$  contours for the optimized Busemann biplane airfoil under lifting conditions during acceleration: (a)  $c_p$  contours at  $M = 1.1$ ; (b)  $c_p$  contours at  $M = 1.3$ ; (c)  $c_p$  contours at  $M = 1.5$ ; (d)  $c_p$  contours at  $M = 1.7$ .

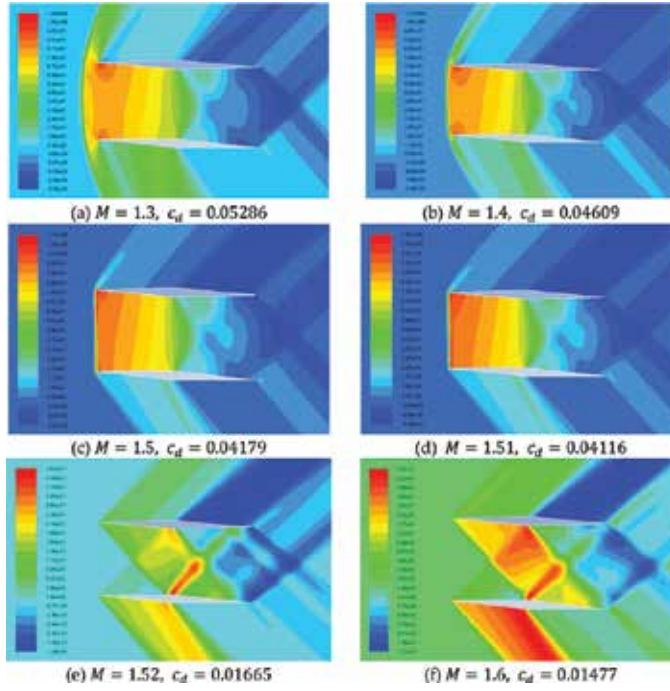


**Figure 24.**  $c_p$  contours for the optimized Busemann biplane airfoil under lifting condition during deceleration: (a)  $c_p$  contours at  $M = 1.7$ ; (b)  $c_p$  contours at  $M = 1.5$ ; (c)  $c_d$  contours at  $M = 1.3$ ; (d)  $c_d$  contours at  $M = 1.1$ .



**Figure 25.**  $c_d$  plot for different airfoils under lifting conditions.

**Figure 25** shows the comparison of the drag coefficients for the standard diamond-shaped airfoil, the baseline Busemann biplane airfoil and the optimized Busemann biplane airfoil under lifting condition. As shown in the figure, the separation between the acceleration and deceleration  $c_d$  lines still exists for the optimized Busemann biplane airfoil, which means that the flow-hysteresis and the choked-flow effects are not totally eliminated. However, as clearly shown in **Figure 25**, the flow-hysteresis area has been significantly reduced, and the drag increase during deceleration due to the choked-flow phenomenon is much smaller than that for the original Busemann biplane airfoil. The drag of the optimized Busemann biplane airfoil in the subsonic region is smaller than that of the original Busemann biplane airfoil, although it is slightly higher than that of the standard diamond-shaped airfoil with  $0.3 < M < 0.85$ . Under



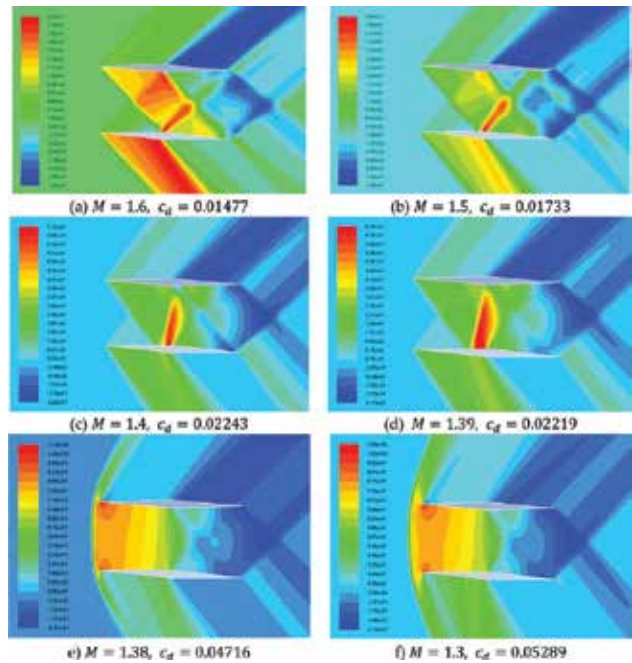
**Figure 26.**  $c_p$  contours of the GA-optimized Busemann biplane airfoil with lift during acceleration: (a)  $M = 1.3$ ,  $c_d = 0.05286$ ; (b)  $M = 1.4$ ,  $c_d = 0.04609$ ; (c)  $M = 1.5$ ,  $c_d = 0.04179$ ; (d)  $M = 1.51$ ,  $c_d = 0.04116$ ; (e)  $M = 1.52$ ,  $c_d = 0.01665$ ; (f)  $M = 1.6$ ,  $c_d = 0.01477$ .

both subsonic and supersonic conditions, the optimized Busemann biplane airfoil has significantly reduced the wave drag compared to the original Busemann biplane airfoil. Similar to the non-lifting condition, the drag coefficient of the optimized Busemann biplane airfoil at  $M = 1.7$  ( $c_d = 0.01362$ ) is higher than that of the original Busemann biplane airfoil ( $c_d = 0.002182$ ).

Next, we examine the details of the flow field during acceleration and deceleration for the optimized Busemann biplane airfoil under lifting conditions and compare them with the flow field of the original Busemann biplane airfoil. **Figures 26** and **27** show the pressure coefficient contours of the optimized Busemann biplane airfoil using GA during acceleration and deceleration, respectively. During acceleration, the flow-hysteresis effect still exists, and a bow shock wave is formed in front of the airfoil. The swallowing of the bow shock wave happens when the Mach number increases from 1.49 to 1.50 as shown in **Figure 16(f)** and from 2.12 to 2.13 for the original Busemann biplane airfoil as shown in **Figure 6(d)**. The corresponding drag coefficient decreases from 0.04116 to 0.01665 for the optimized Busemann biplane airfoil.

During deceleration, the choked-flow effect still exists; however, it is shifted to a lower Mach number of 1.38 compared to 1.6 for the original Busemann biplane airfoil.

In conclusion, the drag coefficient of the optimized Busemann biplane airfoil under lifting conditions is significantly reduced compared to the original Busemann biplane airfoil as shown



**Figure 27.**  $c_p$  contours of the GA-optimized Busemann biplane airfoil with lift during deceleration: (a)  $M = 1.6$ ,  $c_d = 0.01477$ ; (b)  $M = 1.5$ ,  $c_d = 0.01733$ ; (c)  $M = 1.4$ ,  $c_d = 0.02243$ ; (d)  $M = 1.39$ ,  $c_d = 0.02219$ ; (e)  $M = 1.38$ ,  $c_d = 0.04716$ ; (f)  $M = 1.3$ ,  $c_d = 0.05289$ .

in **Figures 25–27**. **Table 9** gives the lift coefficient of the optimized Busemann airfoil under lifting conditions at different design points.

Mach number	1.1	1.2	1.3	1.4	1.5	1.6	1.7
$c_l$ (acceleration)	0.3319	0.3553	0.3061	0.2842	0.2858	0.2525	0.2713
$c_l$ (deceleration)	0.3306	0.3551	0.3064	0.3032	0.2678	0.2525	0.2712

**Table 9.**  $c_l$  for the optimized Busemann biplane airfoil under lifting conditions at different design points.

#### 4. Conclusions

In this chapter, numerical simulations of the flow past the standard diamond-shaped airfoil and the baseline Busemann-type biplane airfoil have been conducted. An impulsive uniform flow, a flow during acceleration, and a flow during deceleration are simulated. The original Busemann biplane airfoil shows a poor performance under off-design conditions due to the flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration. For shape optimization, a single-objective genetic algorithm (SOGA) and a multi-

objective genetic algorithm (MOGA) are employed to optimize the shape of the Busemann-type biplane airfoil under non-lifting and lifting conditions, respectively, to improve its performance under off-design conditions. The commercially available CFD solver FLUENT is employed to calculate the flow field on an unstructured mesh generated using the ICEM grid generation software. A second-order accurate steady-density-based solver in FLUENT is employed to compute the supersonic flow field. The optimization results for the non-lifting case show a significant improvement in reducing the drag coefficient under off-design conditions for the optimized Busemann-type biplane airfoil shape compared to the original shape. The flow-hysteresis phenomenon during acceleration and the choked-flow phenomenon during deceleration are both alleviated significantly for the optimized shape. For the lifting case, the optimized Busemann biplane airfoil is able to significantly reduce the drag coefficient under off-design conditions while generating lift at the same time.

## Author details

Yi Tian and Ramesh K. Agarwal\*

\*Address all correspondence to: rka@wustl.edu

Department of Mechanical Engineering and Materials Science, Washington University in St. Louis, St. Louis, MO, USA

## References

- [1] Busemann, A., *Aerodynamic Lift at Supersonic Speeds*, 12th ed., No. 6, Luftfahrtforschung, 1935, pp. 210–220.
- [2] Moeckel, W. E., *Theoretical Aerodynamic Coefficients of the Two-Dimensional Supersonic Biplane*, NACA Tech. Rept. 1316, 1947.
- [3] Licher, R., *Optimum Two-Dimensional Multiplanes in Supersonic Flow*, Douglas Aircraft Co. Tech Dept. SM-18688, 1955.
- [4] Tan, H. S., *The Aerodynamics of Supersonic Biplanes of Finite Span*, WADC Tech. Rept. 52-276, 1950. Wright Air Development Center. Dayton, OH.
- [5] Ferri, A., *Elements of Aerodynamics of Supersonic Flow*, Macmillan Company, New York, 1949.
- [6] Igra, D., and Arad, E., “A Parametric Study of the Busemann Biplane Phenomena,” *Shock Waves*, 16, 3, 2007, 269–273, doi:10.1007/s00193-006-0070-x.
- [7] Kusunose, K., Matsushima, K., Goto, Y., Yamashita, H., Yonezawa, M., Maruyama, D., and Nakano, T., “A Fundamental Study for the Development of Boomless Supersonic

- Transport Aircraft," AIAA Paper 2006-654, 44rd AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, Reston, VA, 2006.
- [8] Maruyama, D., Matsushima, K., Kusunose, K., and Nakahashi, K., "Aerodynamic Design of Biplane Airfoils for Low Wave Drag Supersonic Flight," AIAA Paper 2006-3323, 24th Applied Aerodynamics Conference, 2006; American Institute of Aeronautics and Astronautics, Reston, VA.
  - [9] Matsushima, K., Maruyama, D., Kusunose, K., and Matsuzawa, T., "Numerical Design and Assessment of a Biplane as Future Supersonic Transport—Revisiting Busemann's Biplane," 25th International Congress of Aeronautical Sciences, ICAS Paper 2006-3.7.1, 2006, pp. 1–10.
  - [10] Maruyama, D., Matsuzawa, T., Kusunose, K., Matsushima, K., and Nakahashi, K., "Consideration at Off-design Conditions of Supersonic Flows around Biplane Airfoils," AIAA Paper 2007-687, 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.
  - [11] Yamashita, H., Yonezawa, M., Obayashi, S., and Kusunose, K., "A Study of Busemann-Type Biplane for Avoiding Choked Flow," AIAA Paper 2007-688, 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.
  - [12] Kuratani, N., Ogawa, T., Yamashita, H., Yonezawa, M., and Obayashi, S., "Experimental and Computational Fluid Dynamics around Supersonic Biplane for Sonic-Boom Reduction," AIAA Paper 2007-3674, 13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference), 2007.
  - [13] Kashitani, M., Yamaguchi, Y., Kai, Y., and Hirata, K., "Preliminary Study on Lift Coefficient of Biplane Airfoil in Smoke Wind Tunnel," AIAA Paper 2008-349, 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008; American Institute of Aeronautics and Astronautics, Reston, VA.
  - [14] Maruyama, D., Matsushima, K., Kusunose, K., and Nakahashi, K., "Three-Dimensional Aerodynamic Design of Low-Wave-Drag Supersonic Biplane Using Inverse Problem Method," *Journal of Aircraft*, 46, 6, 2009, 1906–1918. doi:10.2514/1.40313.
  - [15] Maruyama, D., Kusunose, K., and Matsushima, K., "Aerodynamic Characteristics of a Two-Dimensional Supersonic Biplane Covering its Take-Off to Cruise Conditions," *Shock Waves*, 18, 6, 2009, 437–450.
  - [16] Kusunose, K., Matsushima, K., and Maruyama, D., "Supersonic Biplane," *Progress in Aerospace Sciences*, 47, 1, 2011, 53–87. doi:10.1016/j.paerosci.2010.09.003.
  - [17] Hu, R., Jameson, A., and Wang, Q., "Adjoint-Based Aerodynamic Optimization of Supersonic Biplane Airfoils," *Journal of Aircraft*, 49, 3, 2012, 802–814.
  - [18] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, 6, 2, 2002, 182–197. doi:10.1109/4235.996017.

- [19] He, Y., "Shape Optimization of Airfoils Without and With Ground Effect Using a Multi-Objective Genetic Algorithm," M.S. Thesis, School of Engineering and Applied Science, Washington University in St. Louis, 2014.
- [20] "Some Single- and Multiobjective Optimization Techniques," in *Unsupervised Classification – Similarity Measures, Classical and Metaheuristic Approaches, and Applications*, Chapter 2, 17–21, Springer-Verlag, Berlin, Heidelberg, Bandyopadhyay, S., and Saha, S., 2013. doi: 10.1007/978-3-642-32451-2.
- [21] Available from: <https://www.math.ubc.ca/~cass/gfx/bezier.html> (accessed 9 Dec 2014).

---

# **Performance Analysis of the Differential Evolution and Particle Swarm Optimization Algorithms in Cooperative Wireless Communications**

---

Arif Basgumus, Mustafa Namdar, Gunes Yilmaz and  
Ahmet Altuncu

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62453>

---

## **Abstract**

In this study, we evaluate the performance of differential evolution (DE) and particle swarm optimization (PSO) algorithms in free-space optical (FSO) and mobile radio communications systems. In particular, we obtain the optimal transmission distances for multiple-relay nodes in FSO communication systems and optimal relay locations in mobile radio communications systems for the cooperative-diversity networks, using both algorithms. We investigate the performance comparison of DE and PSO algorithms for the parallel decode-and-forward (DF) relaying. Then, we analyze the cost functions. Furthermore, we present the execution time and the stability of the DE and PSO algorithms.

**Keywords:** free-space optical communications, cooperative-diversity networks, optimal distance, differential evolution algorithm, particle swarm optimization algorithm

---

## **1. Introduction**

The aim of the optimization was to provide the best-suited solution to a problem under the given constraints. The optimization algorithms have recently been much attention and gained significant importance in plenty of engineering problems [1–8]. In this study, we evaluate the performance analysis of differential evolution (DE) and particle swarm optimization (PSO) algorithms both in free-space optical (FSO) and in mobile radio communications systems.

FSO communications have been proposed as a solution for the various applications including fiber backup, and backhaul for wireless communications networks [9]. Despite the fact that the usage of FSO communications is widespread in major applications for wireless communications, the performance limitations for long-range links due to the atmospheric turbulence-induced fading have had profound impacts in FSO communications systems. The method used for relay-assisted FSO transmission links is one of the fading mitigation technique has attracted significant attentions recently in FSO communications networks [9–11]. In [9], the authors consider relay-assisted FSO communications and investigate the outage performance under serial and parallel scheme with amplify-and-forward (AF), and decode-and-forward (DF) relaying models. The authors in [10] consider a cooperative FSO communications via optical AF relay and investigate the bit error probability performance. Bit error rate (BER) analysis of cooperative systems in FSO networks is presented in [11]. The outage performance analysis of FSO communications is presented in both [12] and [13]. Kashani et al. [14] consider the diversity gain analysis and determine the optimal relay locations for both the serial and the parallel relaying schemes. Although cooperative transmissions have greatly been considered in the above manuscripts, to the best of the authors' knowledge, there has not been any notable research for the relay-assisted FSO communications systems using the optimization algorithms. To fill the research gap, in this paper, we analyze the performance of both DE and PSO algorithms in terms of the transmission distances when applied for the parallel DF relaying in FSO systems. Moreover, we investigate the performance comparison of these two algorithms in respect to the execution time, cost, and stability analysis.

In our study, as a second part of this paper, we focus on dual-hop cooperative-diversity network to study the impact of the relay location between the source and the destination. Cooperative-diversity relay networks provide a significant performance increment in the radio frequency power transmission and the spatial diversity. They are also shown to be a promising solution to mitigate the signal fading arising from the multipath propagation in wireless communications [15, 16]. In the cooperative-diversity networks, relay terminals are employed between the transmitter and receiver nodes, over multiple communications routes, in which two main protocols are used as follows: (i) amplify-and-forward and (ii) DF [15–21].

Most of the previous publications have studied the cooperative-diversity performance over different fading channels [15–26]. In [15], the authors analyzed the cooperative-diversity network using AF cooperation protocol, operating over independent, but not necessarily identically distributed Nakagami- $m$  fading channels. The paper in [17] addressed the multi-branch adaptive DF scheme for cooperative-diversity networks. The best relay selection scheme for cooperative-diversity network is studied in [18]. Furthermore, [19] investigated the advantage of the diversity over direct transmission and conventional non-cooperative relaying scheme. In all these papers, analytical framework for performance analysis of BER and the outage probability is provided [15, 17–19]. As far as we know, both DE and PSO algorithms have not been applied for obtaining the optimal location of the relaying terminal over Nakagami- $m$  fading channel.

To fill the research gaps in mobile radio communications using cooperative-diversity relay network, in this paper, we provide an optimization algorithms results, indicating the optimal location of the relaying terminal in the parallel-relaying scheme.

In summary, for the first part of this paper, the key contributions are twofold:

- First, the locations of each individual relay nodes and the transmission distances are optimized for the parallel DF relaying scheme in FSO systems.
- Second, and more importantly, none of the previous studies provide a comparison among optimization algorithms when applied for FSO systems. In this paper, we investigate the performance comparison of DE and PSO algorithms for the parallel DF relaying in respect to the execution time, cost, and stability analysis.

For the second part of this paper, there is a major contribution:

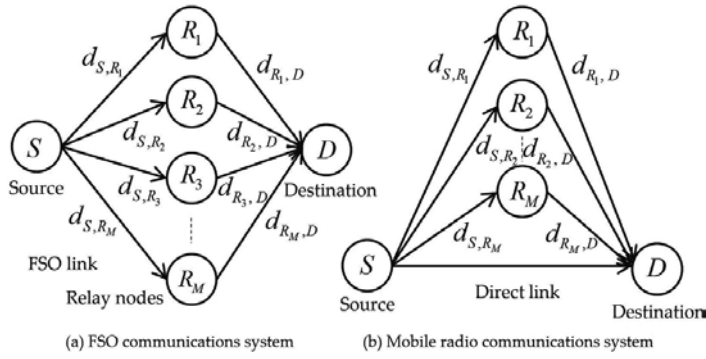
- To fill the research gaps in cooperative-diversity relay network, we provide a rigorous data for the optimal location of the relaying terminal over Nakagami-m fading channel achieving the best error performance using both DE and PSO algorithms in the parallel relaying schemes.

The rest of this paper is organized as follows: The system model and performance analysis are discussed in Section 2 exploiting the DE and PSO algorithms. Section 3 provides the numerical results and simulations. Finally, the concluding remarks are given in Section 4.

## 2. System model and performance analysis using the optimization algorithms

### 2.1. FSO communications systems

This section presents the system model for FSO communications networks with parallel DF cooperative relaying protocol shown in **Figure 1a**. We consider that the FSO links between the source-to-relay ( $S \rightarrow R_j$ ,  $j = 1, 2, \dots, M$ ) and relay-to-destination ( $R_j \rightarrow D$ ) are subjected to atmospheric turbulence-induced log-normal fading [1]. Here, the  $j$  index refers to the number of the relay nodes, where the maximum number of relay nodes ( $R_j$ ) is defined with  $M$  ( $j = 1, 2, \dots, M$ ) as shown in **Figure 1a**. Besides, the normalized path loss can be expressed as  $L(d) = \frac{\ell(d)}{\ell(d_{s,D})} = \left(\frac{d_{s,D}}{d}\right)^2 e^{\sigma(d_{s,D}-d)}$  where  $\ell(d)$  and  $\ell(d_{s,D})$  are defined as the path losses for the distance of  $d$  and for the distance between source and destination ( $d_{s,D}$ ), respectively [13, 14]. Here,  $\sigma$  is the atmospheric attenuation coefficient. In the same figure below,  $d_{s,R_j}$  is the distance between the source and the  $j$ -th decoding relay, and  $d_{R_j,D}$  is the direct link distance between the  $j$ -th relay and the destination where the relay nodes are placed on the straight line connecting the source and the destination.



**Figure 1.** System models for cooperative wireless communications.

In [14], the outage probability for the parallel DF relaying is expressed as follows:

$$P_{out} = \sum_{i=1}^{2^M} \prod_{j \in W(i)} \left( 1 - Q \left( \frac{\ln \left( \frac{L(d_{S,R_j})P}{2M} \right) + 2\mu_\chi(d_{S,R_j})}{2\sigma_\chi(d_{S,R_j})} \right) \right) \times \prod_{j \notin W(i)} Q \left( \frac{\ln \left( \frac{L(d_{S,R_j})P}{2M} \right) + 2\mu_\chi(d_{S,R_j})}{2\sigma_\chi(d_{S,R_j})} \right) Q \left( \frac{\ln \left( \frac{Pe^{\mu_\xi}}{2M} \right)}{\sigma_\xi(\bar{d}_{W(i)})} \right) \quad (1)$$

where  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du$  is the  $Q$  function,  $M$  is the relay number. Here,  $P$  is the power margin and defined by  $P = (P_T/P_{th})$ , where  $P_T$  is the total transmitted power, and  $P_{th}$  is the threshold value for the transmit power in case of no outage is available.  $P_T$  is expressed as  $P_T = P_S + \sum_{j=1}^M P_j$ , where  $P_S$  is the source power, and  $P_j$  is the power of the  $j$ -th relay [9]. In Eq. (1), the variance of the fading log-amplitude,  $\chi$  is defined by  $\sigma_\chi^2(d) = \min \{0.124 k^{(7/6)} C_n^{-2} d^{(11/6)}, 0.5\}$ , where  $k = (2\pi/\lambda)$  is the wave number, and  $C_n^2 = 10^{-14} m^{2/3}$  is the refractive index structure constant [14]. Here,  $\ln(\cdot)$  is the natural logarithm operator and  $\lambda$  is the wavelength [10]. The mean value of the fading log-amplitude is modeled as  $\mu_\chi = -\sigma_\chi^2$  [11].

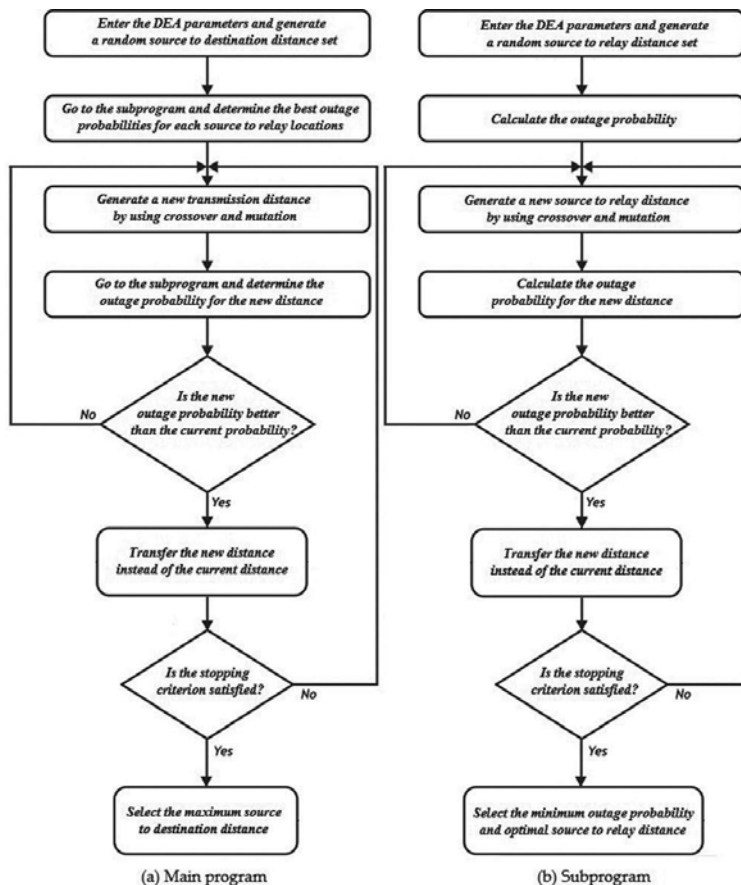
In the outage probability of the parallel DF relaying scheme, there are  $2^M$  possibilities for decoding the signal between  $S$  and  $R_j$ . In Eq. (1) the  $i$  index refers to the number of possible

combinations where the  $i$ -th possible set is defined by  $W(i)$ , and the possible set of distances between the relays and destination is given by  $\bar{d}_{W(i)}$ .  $\mu_\xi$  is the mean value, and  $\sigma_\xi$  is the variance of the log-amplitude factor, also given in [9, 14].

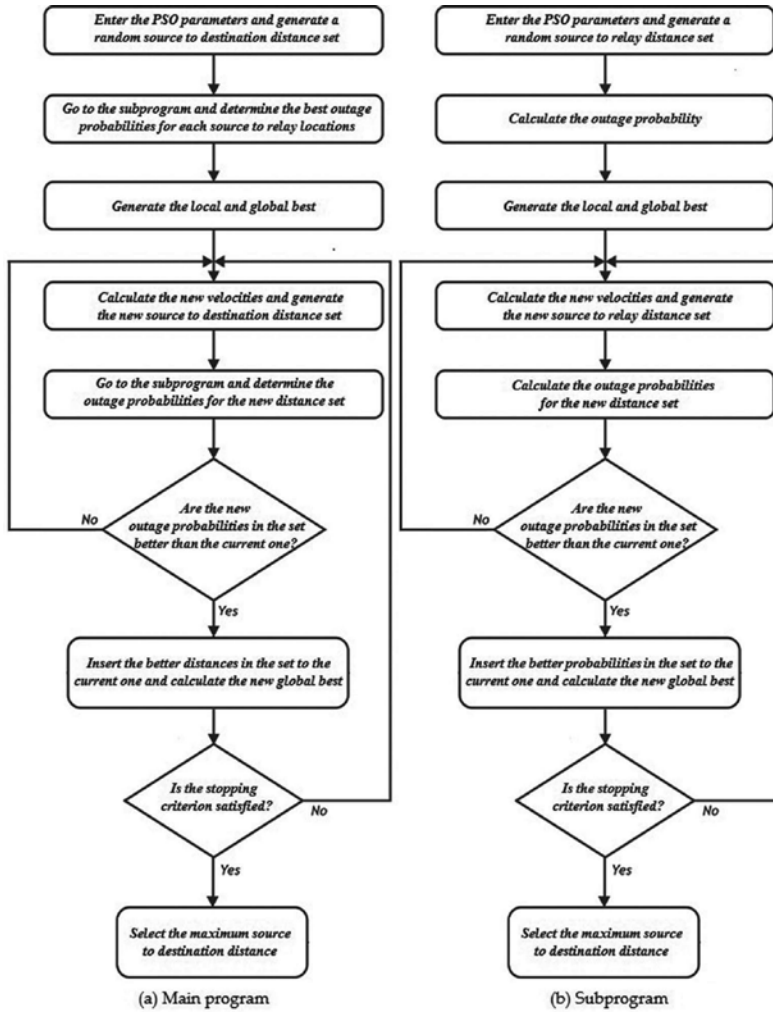
For the optimization problem, a function is employed to minimize the outage probability for the parallel DF relaying, which can be written as  $\min\{P_{out}\} = \min\{f(d_{S,R_1}, d_{S,R_2}, \dots, d_{S,R_M})\}$  where  $0 < d_{S,R_j} < d_{S,D}$  for  $j = 1, 2, \dots, M$ . Optimal transmission distance is maximized by optimizing the locations of the relays, at an outage probability of  $10^{-6}$  as modeled as follows [27]:

$$\max\{d_{S,D}\} = f\left(\min\left\{\left|\min\{P_{out}\} - 10^{-6}\right|\right\}\right) \quad (2)$$

The flowcharts for the optimization of the transmission distance using DE and PSO algorithms are shown in **Figures 2** and **3**, successively.



**Figure 2.** The flowcharts for the optimization of the transmission distance by using DE algorithm.



**Figure 3.** The flowcharts for the optimization of the transmission distance by using PSO algorithm.

## 2.2. Mobile radio communications systems using cooperative-diversity relay networks

A system, consisting of a source terminal ( $S$ ), a cooperative relay terminal ( $R_j$ ,  $j = 1, 2, \dots, M$ ), and a destination terminal ( $D$ ), is considered, as shown in **Figure 1b**. The source and the destination operate in half-duplex mode and both equipped with a single pair of transmit and receive antennas. Source communicates with the destination over both the direct link and relaying terminal links [17, 20]. The cooperation method is based on DF relaying in which the source information is decoded first and then retransmitted to the destination. We consider that the received signals from the source and the relays are combined together using the method of equal gain combining (EGC) [15, 17, 28]. We assume that the diversity paths between the source-to-relay terminals ( $S \rightarrow R_j$ ) and the relay terminals-to-destination ( $R_j \rightarrow D$ ) are

independent non-identically distributed Nakagami- $m$  fading. The direct link between the source and destination ( $S \rightarrow D$ ) has also Nakagami- $m$  distribution. Besides, the additive white Gaussian noise (AWGN) terms of  $S \rightarrow D$  link,  $S \rightarrow R_j$  and  $R_j \rightarrow D$  links have zero mean and equal variance of  $N_0$  [15, 17].

The source signal is transmitted with the energy of  $E_s$ , directly to the destination  $D$ , and the relays. The relays decode and transmit the multiple copies of the original source signal to the destination. Finally, at the destination, the direct and indirect links from the source and the relays are combined together. It should be noted that, at the destination, the total output signal-to-noise ratio (SNR) can be expressed as [15, 17]

$$\gamma_{Total} = \gamma_{SD} + \sum_{i=1}^M \frac{\gamma_{S,R_i} \gamma_{R_i,D}}{\gamma_{S,R_i} + \gamma_{R_i,D} + 1} \quad (3)$$

where  $\gamma_{SD} = \frac{E_s}{N_0} h_{SD}^2$ ,  $\gamma_{S,R_j} = \frac{E_s}{N_0} h_{S,R_j}^2$  and  $\gamma_{R_j,D} = \frac{E_s}{N_0} h_{R_j,D}^2$  are the instantaneous SNRs, in the  $S \rightarrow D$  link, between  $S$  and the  $j$ -th relay, and between the relay,  $R_j$  and the destination, successively [13, 14]. Note that,  $h_{SD}$ ,  $h_{S,R_j}$  and  $h_{R_j,D}$  represent the channel fading coefficients for  $S \rightarrow D$ ,  $S \rightarrow R_j$ , and  $R_j \rightarrow D$  links, respectively [13]. Finally, a closed form for the error performance of the DF scheme for the cooperative-diversity relay network is given in [14] as follows:

$$P(e) \leq \frac{1}{2^{2M+1}} \sum_{i=1}^{M+1} \frac{(2M+1)!}{(M+1-i)!(M+i)!} \times \left[ \left( \frac{3}{2} + \frac{1-\beta^{i-1}}{\pi\beta^{i-1}(1-\beta)} \right) M_{\gamma_{S,D}}(\zeta) \prod_{k=1}^M M_i(\zeta) - \left( \frac{1}{2} + \frac{1-\beta^{i-1}}{\pi\beta^{i-1}(1-\beta)} \right) M_{\gamma_{S,D}}(v) \prod_{k=1}^M M_i(v) \right] \quad (4)$$

where  $\beta = (a/b)$ ,  $\zeta = (b-a)^2/2$ ,  $v = (b+a)^2/2$ . In here,  $a$  and  $b$  are constant values depend on the modulation type. For instance, for differential binary phase shift keying (DBPSK) modulation  $a = 10^{-3}$  and  $b = \sqrt{2}$  [17].  $M_{\gamma_{S,D}}(\zeta) = (1 + \zeta(\bar{\gamma}_{S,D}/m_{S,D}))^{-m_{S,D}}$  is the moment-generating function (MGF) of  $\gamma_{S,D}$ , where  $m_{S,D}$  is the fading parameter between the source and destination. In Eq. (4),  $M_i(\zeta) = A_i + (1-A_i)M_{\gamma_{R_i,D}}(\zeta)$  is the MGF of the  $j$ -th indirect relay link variable  $\zeta$  where  $M_{\gamma_{R_i,D}}(\zeta)$  is the MGF of  $\gamma_{R_i,D}$ , and expressed as  $M_{\gamma_{R_i,D}}(\zeta) = (1 + \zeta(\bar{\gamma}_{R_i,D}/m_{R_i,D}))^{-m_{R_i,D}}$  [15]. Here,  $m_{R_i,D}$  is the fading parameter between the  $j$ -th relay and the destination and  $A_i$  can be given for DBPSK modulation as  $A_i = \frac{1}{2}(m_{S,R_i}/(m_{S,R_i} + \bar{\gamma}_{S,R_i}))^{m_{S,R_i}}$  [17]. The average SNRs between  $S$  and  $D$ ,  $S$  and the  $j$ -th relay, and  $j$ -th relay and the destination are denoted by  $\bar{\gamma}_{S,D} = (E_s/N_0)$ ,  $\bar{\gamma}_{S,R_i} = (d_{S,D}/d_{S,R_i})^\epsilon (E_s/N_0)$ , and  $\bar{\gamma}_{R_i,D} = (d_{S,D}/d_{R_i,D})^\epsilon (E_s/N_0)$ , respectively, where  $\epsilon$  is the path loss exponent. The same approach in [15, 29, 30] is applied with the following model, while evaluating the effect of the path loss on the error performance:  $\bar{h}_{S,R_i}^2 = E(h_{S,R_i}^2) = \left( \frac{d_{S,D}}{d_{S,R_i}} \right)^\epsilon$ ,

$\bar{h}_{R,D}^2 = E(h_{R,D}^2) = \left(\frac{d_{S,D}}{d_{R,D}}\right)^\alpha$ , and  $\bar{h}_{S,D}^2 = \left(\frac{d_{S,D}}{d_{S,D}}\right)^\alpha$ . Here,  $E(\cdot)$  is the statistical average operator,  $d_{x,y}$  is the distance, and  $\bar{h}_{x,y}^2$  is the average channel fading coefficient between the terminals  $x$  and  $y$ .

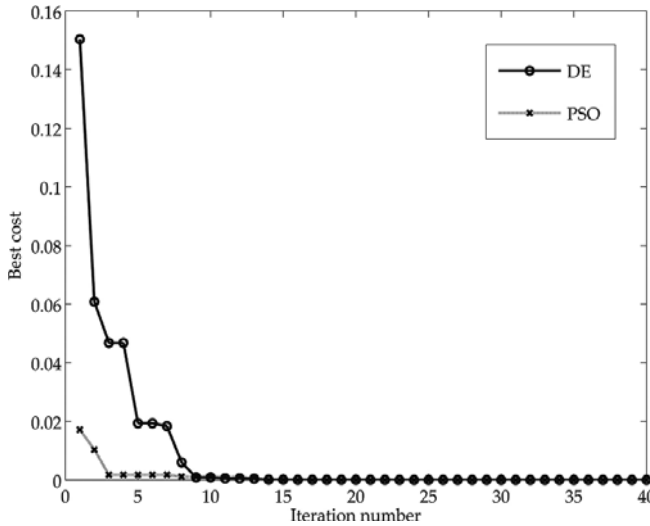
### 3. Numerical results and simulations

In this section, numerical and simulation results are presented for both FSO and mobile radio communication systems.

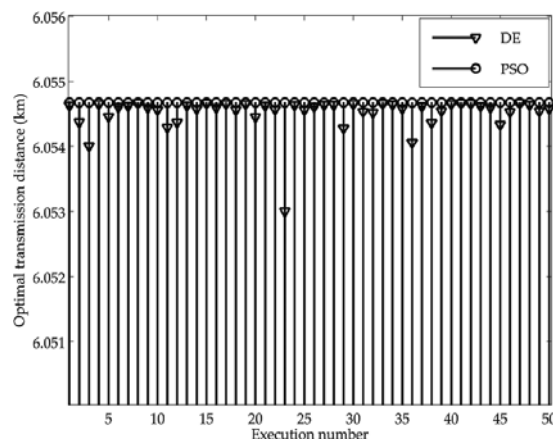
#### 3.1. FSO communications systems

In this section, numerical results are presented. For the optimization algorithms, the parameters  $\lambda = 1550 \text{ nm}$ ,  $C_n^2 = 10^{-14} \text{ m}^{2/3}$ ,  $\sigma = 0.1$ ,  $P = 9 \text{ dB}$  are used, and totally 4 relays are evaluated. The cost function analysis is illustrated in **Figure 4**, for the DE and PSO algorithms in terms of the iteration number. It can be observed from the simulation results that the cost function for the PSO algorithm is minimized for the small number of iterations as compare to the DE algorithm. This result indicates that PSO outperforms DE in terms of the cost function. Here, the iteration number is set to 40, and the execution number is taken as 50.

**Figure 5** shows the optimal  $d_{S,D}$  for the aforementioned algorithms. While 4 relays are used,  $d_{S,D} = 6.0547 \text{ km}$  is calculated. For the each execution, PSO algorithm gives almost the same result. Therefore, it is obvious that PSO is more stable than the DE algorithm under the same setup.



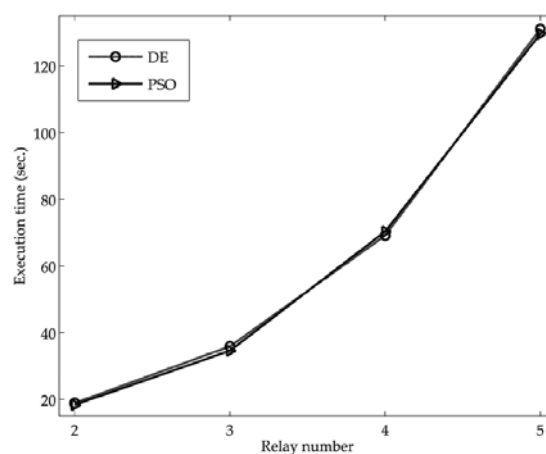
**Figure 4.** Cost function for 40 iterations.



**Figure 5.** Optimal  $d_{s,D}$  vs. execution number for the DE and PSO algorithms.

It can be noticed from **Figure 6** that the execution time for the PSO algorithm closely matches with the execution time of the DE algorithm for different number of relays.

**Figure 7** shows the optimization results for the locations of each individual relay nodes. Accurate relay placements are obtained for  $P = 9$  dB and 4 relays ( $R_j, j = 1, 2, 3, 4$ ). The optimization results indicate that if the places of each individual relay nodes are similar, regardless of the relay number and the  $P$  value, better performance is achieved. **Figure 7** clearly shows that the optimal places of the relay nodes for both algorithms are calculated as  $d_{s,R_j} \approx 0.4305$  using the normalized approach, after the number of iterations, which confirms the accuracy of the employed optimization algorithms. In addition, these results show that both algorithms seem to be the better optimization algorithms in providing the reliable and the rapid results.



**Figure 6.** Execution time vs. relay number for the DE and PSO algorithms.

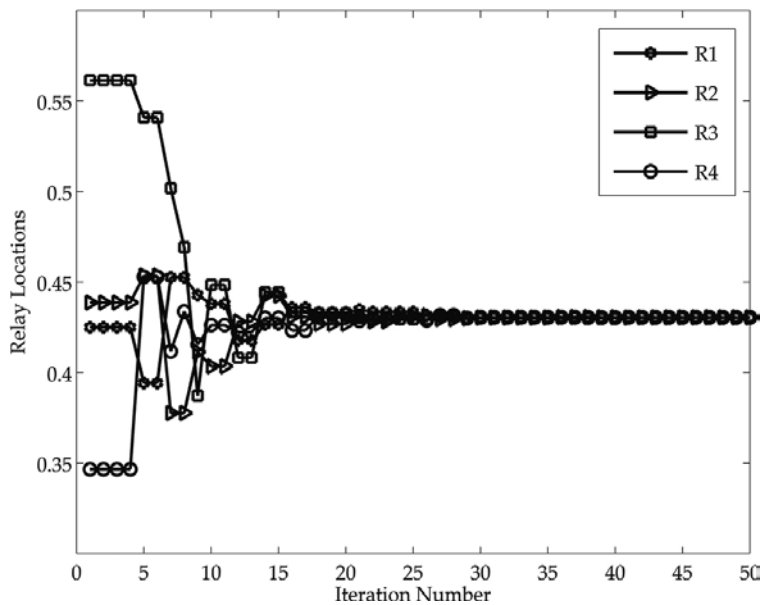


Figure 7. Relay location vs. iteration number for  $R_j, j = 1, \dots, 4$ .

Finally, the impact of the varying  $P$  on the optimal transmission distance for the DE and PSO algorithms is depicted in Figure 8. It can be seen from the below figure that the optimal

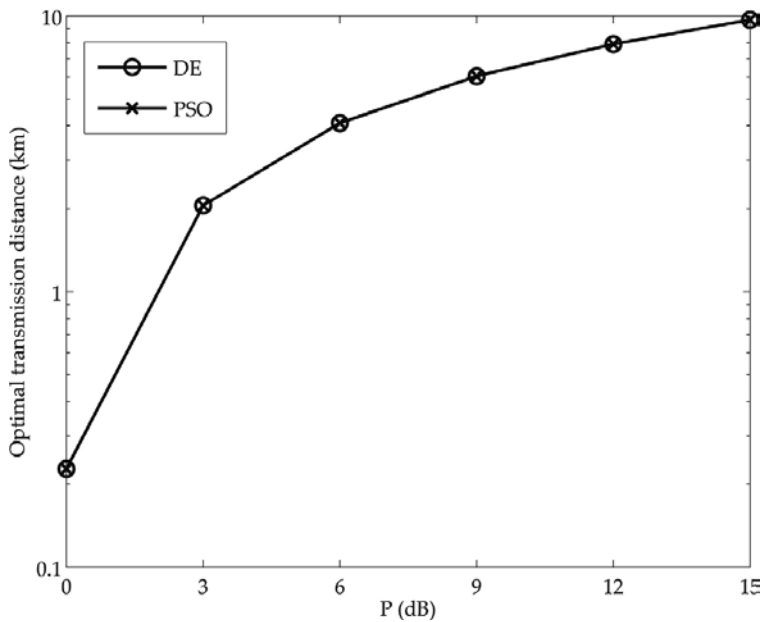


Figure 8. Optimal  $d_{S,D}$  with varying  $P$ .

transmission distance increases with  $P$  and the results for both algorithms closely match with each other for all  $P$  values.

The detailed optimization results with the DE and PSO algorithms for DF parallel relaying scheme are given in **Table 1**. Here, the results for the optimal transmission distances and optimal relay locations are listed for various  $P$  values, where  $d_{s,D}$  is the distance between the source and the destination ( $S \rightarrow D$ ) and  $d_{s,R_j}$  is the distance between the source and the  $j$ -th relay ( $S \rightarrow R_j$ ,  $j = 1, 2, \dots, M$ ). Based on the numerical optimization results provided in **Table 1**, while the number of relays is increased, the relays are obliged to get close up to the source, and the distance between the source and the destination is shortened in low  $P$  region, as expected [31].

$P$ (dB)	2 Relays				3 Relays			
	DE		PSO		DE		PSO	
	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)
0	0.6323	0.2608	0.6323	0.2608	0.4017	0.1464	0.4016	0.1463
3	2.0304	0.8562	2.0304	0.8562	2.0453	0.7813	2.0453	0.7813
6	3.5568	1.5653	3.5568	1.5653	3.8708	1.6044	3.8708	1.6044
9	5.1021	2.3490	5.1021	2.3490	5.6697	2.5207	5.6697	2.5207
12	6.6391	3.1868	6.6392	3.1868	7.4014	3.4801	7.4014	3.4801
15	8.1326	4.0077	8.1326	4.0077	9.0656	4.4476	9.0656	4.4476
$P$ (dB)	4 Relays				5 Relays			
	DE		PSO		DE		PSO	
	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)	Optimal $d_{s,D}$ (km)	Optimal $d_{s,R_j}$ (km)
0	0.2276	0.0756	0.2276	0.0756	0.2276	0.0756	0.2276	0.0756
3	2.0541	0.7280	2.0541	0.7280	2.0541	0.7280	2.0541	0.7280
6	4.0930	1.6114	4.0930	1.6114	4.0930	1.6114	4.0930	1.6114
9	6.0547	2.6065	6.0547	2.6065	6.0547	2.6065	6.0547	2.6065
12	7.9173	3.6665	7.9173	3.6665	7.9173	3.6665	7.9173	3.6665
15	9.6952	4.7477	9.6952	4.7477	9.6952	4.7477	9.6952	4.7477

**Table 1.** Optimization results for the parallel DF relaying with different number of nodes.

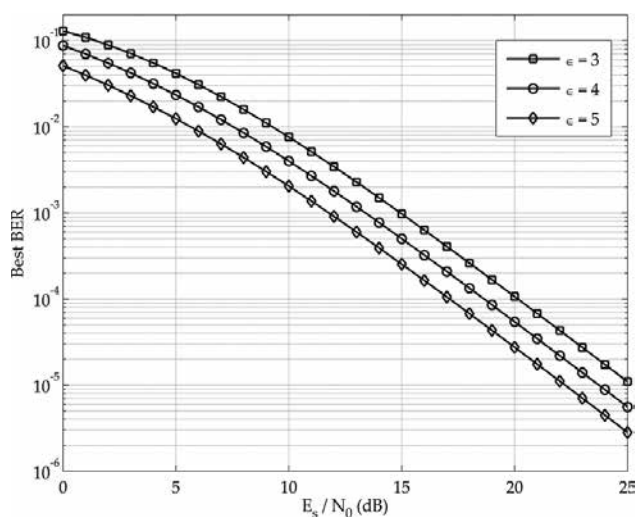
### 3.2. Mobile radio communications systems

The error performance of the DF scheme for the cooperative-diversity relay network is illustrated in **Table 2** with varying path loss exponent for different values of  $E_s/N_0$  when  $m_{s,D}=m_{s,R_j}=m_{R_j,D}=1$ ,  $\bar{h}_{s,D}^2=1$ , and the number of relay is  $M=1$ . The optimal relay placement ( $d_{s,R}$ ) values to minimize the total error rate where the best minimum of  $P(e)$  is achieved for the proposed system are calculated with the help of DE and PSO algorithms.

$\frac{E_s}{N_0}$	$\epsilon = 3$		$\epsilon = 4$		$\epsilon = 5$	
	Best BER	Optimal $d_{s,R}$	Best BER	Optimal $d_{s,R}$	Best BER	Optimal $d_{s,R}$
0	$1.2993 \times 10^{-1}$	0.7518	$8.8041 \times 10^{-2}$	0.6179	$5.1268 \times 10^{-2}$	0.5721
1	$1.0951 \times 10^{-1}$	0.7211	$7.0389 \times 10^{-2}$	0.6083	$3.9879 \times 10^{-2}$	0.5690
2	$8.9635 \times 10^{-2}$	0.6962	$5.5120 \times 10^{-2}$	0.6011	$3.0535 \times 10^{-2}$	0.5667
3	$7.1352 \times 10^{-2}$	0.6768	$4.2307 \times 10^{-2}$	0.5956	$2.3011 \times 10^{-2}$	0.5648
4	$5.5332 \times 10^{-2}$	0.6621	$3.1845 \times 10^{-2}$	0.5914	$1.7065 \times 10^{-2}$	0.5632
5	$4.1869 \times 10^{-2}$	0.6509	$2.3521 \times 10^{-2}$	0.5881	$1.2453 \times 10^{-2}$	0.5620
6	$3.0965 \times 10^{-2}$	0.6424	$1.7057 \times 10^{-2}$	0.5855	$8.9429 \times 10^{-3}$	0.5610
7	$2.2419 \times 10^{-2}$	0.6358	$1.2155 \times 10^{-2}$	0.5834	$6.3230 \times 10^{-3}$	0.5602
8	$1.5917 \times 10^{-2}$	0.6307	$8.5199 \times 10^{-3}$	0.5818	$4.4044 \times 10^{-3}$	0.5595
9	$1.1103 \times 10^{-2}$	0.6267	$5.8820 \times 10^{-3}$	0.5805	$3.0255 \times 10^{-3}$	0.5590
10	$7.6227 \times 10^{-3}$	0.6236	$4.0051 \times 10^{-3}$	0.5795	$2.0518 \times 10^{-3}$	0.5585
11	$5.1610 \times 10^{-3}$	0.6212	$2.6938 \times 10^{-3}$	0.5786	$1.3756 \times 10^{-3}$	0.5582
12	$3.4523 \times 10^{-3}$	0.6193	$1.7924 \times 10^{-3}$	0.5780	$9.1298 \times 10^{-4}$	0.5579
13	$2.2854 \times 10^{-3}$	0.6177	$1.1816 \times 10^{-3}$	0.5774	$6.0063 \times 10^{-4}$	0.5576
14	$1.4998 \times 10^{-3}$	0.6165	$7.7281 \times 10^{-4}$	0.5770	$3.9220 \times 10^{-4}$	0.5575
15	$9.7704 \times 10^{-4}$	0.6156	$5.0210 \times 10^{-4}$	0.5767	$2.5448 \times 10^{-4}$	0.5573
16	$6.3264 \times 10^{-4}$	0.6148	$3.2442 \times 10^{-4}$	0.5764	$1.6426 \times 10^{-4}$	0.5572
17	$4.0759 \times 10^{-4}$	0.6142	$2.0866 \times 10^{-4}$	0.5762	$1.0556 \times 10^{-4}$	0.5571
18	$2.6153 \times 10^{-4}$	0.6138	$1.3370 \times 10^{-4}$	0.5760	$6.7596 \times 10^{-5}$	0.5570
19	$1.6725 \times 10^{-4}$	0.6134	$8.5414 \times 10^{-5}$	0.5759	$4.3161 \times 10^{-5}$	0.5569
20	$1.0668 \times 10^{-4}$	0.6131	$5.4431 \times 10^{-5}$	0.5757	$2.7493 \times 10^{-5}$	0.5569
21	$6.7892 \times 10^{-5}$	0.6128	$3.4618 \times 10^{-5}$	0.5757	$1.7480 \times 10^{-5}$	0.5568
22	$4.3133 \times 10^{-5}$	0.6126	$2.1982 \times 10^{-5}$	0.5756	$1.1097 \times 10^{-5}$	0.5568
23	$2.7365 \times 10^{-5}$	0.6125	$1.3940 \times 10^{-5}$	0.5755	$7.0356 \times 10^{-6}$	0.5568
24	$1.7342 \times 10^{-5}$	0.6124	$8.8312 \times 10^{-6}$	0.5755	$4.4564 \times 10^{-6}$	0.5568
25	$1.0981 \times 10^{-5}$	0.6123	$5.5901 \times 10^{-6}$	0.5755	$2.8205 \times 10^{-6}$	0.5567

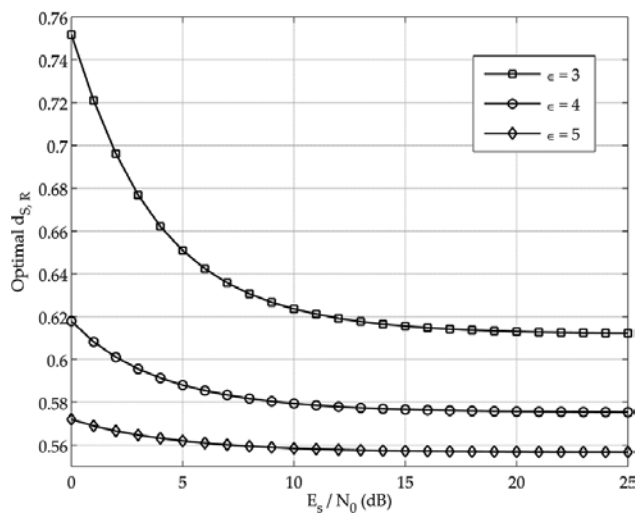
**Table 2.** Optimization results using the normalized approach for the parallel DF relaying with different path loss exponent for  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=1, \bar{h}_{S,D}^2=1$ .

**Figure 9** shows the best BER performance for the considered system versus  $E_s/N_0$  when  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=1, \bar{h}_{S,D}^2=1$ , and  $M = 1$ . The results clearly show that BER significantly decreases with the increase  $E_s/N_0$ . In the same figure, the path loss exponent ( $\epsilon$ ) is varied from 3 to 5.



**Figure 9.** BER performance of the DF scheme for the cooperative-diversity relay network with  $\epsilon = 3, 4, 5$  and  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=1$ .

**Figure 10** demonstrates the effect of  $\epsilon$  on the distance between source and the relay terminal ( $d_{S,R}$ ) with varying  $E_s/N_0$ . This figure is plotted for  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=1$ , and  $\bar{h}_{S,D}^2=1$ .



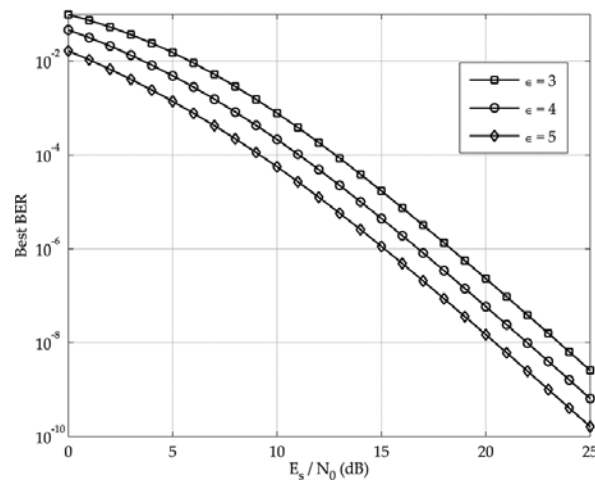
**Figure 10.** Optimal  $d_{S,R}$  against different  $E_s/N_0$  values with  $\epsilon = 3, 4, 5$  and  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=1$ .

**Table 3** shows that, the optimal  $d_{S,R}$  can be calculated where the best minimum of  $P(e)$  is achieved with varying  $E_s/N_0$  for different values of  $\epsilon$  when  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2$ ,  $\bar{h}_{S,D}^2=1$ , and the number of relay is  $M = 1$ .

$\frac{E_s}{N_0}$	$\epsilon = 3$		$\epsilon = 4$		$\epsilon = 5$	
	Best BER	Optimal $d_{S,R}$	Best BER	Optimal $d_{S,R}$	Best BER	Optimal $d_{S,R}$
0	$9.7528 \times 10^{-2}$	0.6525	$4.6132 \times 10^{-2}$	0.5676	$1.6573 \times 10^{-2}$	0.5409
1	$7.4048 \times 10^{-2}$	0.6277	$3.1667 \times 10^{-2}$	0.5610	$1.0673 \times 10^{-2}$	0.5388
2	$5.3487 \times 10^{-2}$	0.6091	$2.0935 \times 10^{-2}$	0.5559	$6.6843 \times 10^{-3}$	0.5370
3	$3.6808 \times 10^{-2}$	0.5955	$1.3343 \times 10^{-2}$	0.5521	$4.0720 \times 10^{-3}$	0.5356
4	$2.4167 \times 10^{-2}$	0.5855	$8.2058 \times 10^{-3}$	0.5491	$2.4121 \times 10^{-3}$	0.5345
5	$1.5160 \times 10^{-2}$	0.5779	$4.8710 \times 10^{-3}$	0.5468	$1.3884 \times 10^{-3}$	0.5335
6	$9.0972 \times 10^{-3}$	0.5722	$2.7915 \times 10^{-3}$	0.5449	$7.7597 \times 10^{-4}$	0.5327
7	$5.2296 \times 10^{-3}$	0.5678	$1.5448 \times 10^{-3}$	0.5433	$4.2078 \times 10^{-4}$	0.5320
8	$2.8842 \times 10^{-3}$	0.5644	$8.2583 \times 10^{-4}$	0.5421	$2.2129 \times 10^{-4}$	0.5314
9	$1.5289 \times 10^{-3}$	0.5617	$4.2681 \times 10^{-4}$	0.5410	$1.1287 \times 10^{-4}$	0.5309
10	$7.8074 \times 10^{-4}$	0.5595	$2.1352 \times 10^{-4}$	0.5402	$5.5873 \times 10^{-5}$	0.5305
11	$3.8507 \times 10^{-4}$	0.5578	$1.0358 \times 10^{-4}$	0.5395	$2.6877 \times 10^{-5}$	0.5301
12	$1.8397 \times 10^{-4}$	0.5564	$4.8835 \times 10^{-5}$	0.5389	$1.2586 \times 10^{-5}$	0.5298
13	$8.5421 \times 10^{-5}$	0.5553	$2.2434 \times 10^{-5}$	0.5385	$5.7510 \times 10^{-6}$	0.5295
14	$3.8672 \times 10^{-5}$	0.5545	$1.0070 \times 10^{-5}$	0.5381	$2.5705 \times 10^{-6}$	0.5293
15	$1.7127 \times 10^{-5}$	0.5538	$4.4297 \times 10^{-6}$	0.5378	$1.1268 \times 10^{-6}$	0.5291
16	$7.4433 \times 10^{-6}$	0.5532	$1.9148 \times 10^{-6}$	0.5375	$4.8576 \times 10^{-7}$	0.5290
17	$3.1836 \times 10^{-6}$	0.5528	$8.1544 \times 10^{-7}$	0.5373	$2.0643 \times 10^{-7}$	0.5289
18	$1.3436 \times 10^{-6}$	0.5524	$3.4296 \times 10^{-7}$	0.5371	$8.6671 \times 10^{-8}$	0.5288
19	$5.6080 \times 10^{-7}$	0.5521	$1.4276 \times 10^{-7}$	0.5370	$3.6027 \times 10^{-8}$	0.5287
20	$2.3194 \times 10^{-7}$	0.5519	$5.8914 \times 10^{-8}$	0.5369	$1.4852 \times 10^{-8}$	0.5286
21	$9.5217 \times 10^{-8}$	0.5517	$2.4143 \times 10^{-8}$	0.5368	$6.0812 \times 10^{-9}$	0.5286
22	$3.8852 \times 10^{-8}$	0.5516	$9.8379 \times 10^{-9}$	0.5368	$2.4762 \times 10^{-9}$	0.5286
23	$1.5775 \times 10^{-8}$	0.5514	$3.9902 \times 10^{-9}$	0.5367	$1.0038 \times 10^{-9}$	0.5285
24	$6.3802 \times 10^{-9}$	0.5513	$1.6124 \times 10^{-9}$	0.5367	$4.0545 \times 10^{-10}$	0.5285
25	$2.5722 \times 10^{-9}$	0.5513	$6.4960 \times 10^{-10}$	0.5366	$1.6329 \times 10^{-10}$	0.5285

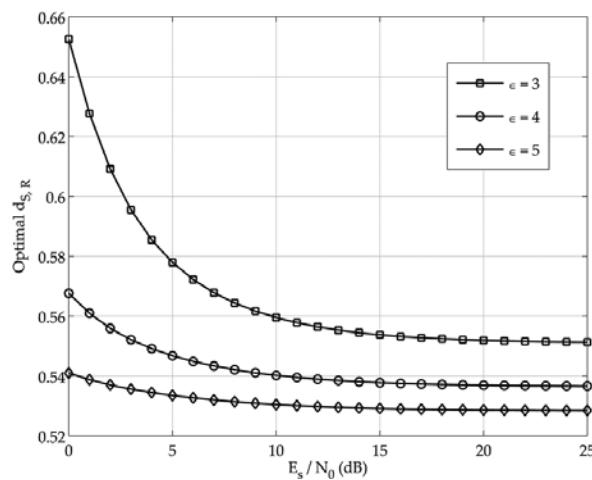
**Table 3.** Optimization results using the normalized approach for the parallel DF relaying with different path loss exponent for  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2, \bar{h}_{S,D}^2=1$ .

The best BER performance for the considered system is depicted in **Figure 11** when  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2$ ,  $\bar{h}_{S,D}^2=1$ , and  $M=1$ . It is seen that, for a fixed  $E_s/N_0$ , the performance of the considered system increases while  $\epsilon$  increases.



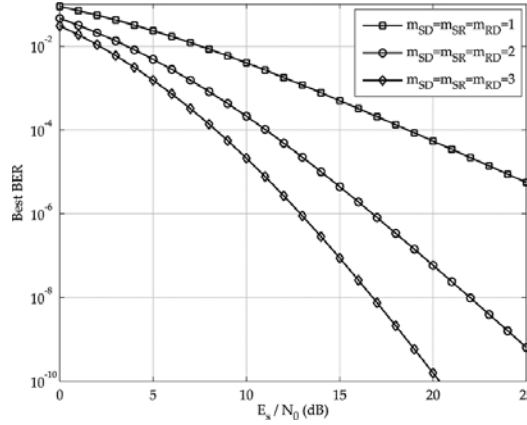
**Figure 11.** BER performance of the DF scheme for the cooperative-diversity relay network with  $\epsilon = 3, 4, 5$  and  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2$ .

The variation of the optimal  $d_{S,R}$  with varying  $E_s/N_0$  for different values of  $\epsilon$  is depicted in **Figure 12**. We assume  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2$ , and  $\bar{h}_{S,D}^2=1$ .



**Figure 12.** Optimal  $d_{S,R}$  against different  $E_s/N_0$  values with  $\epsilon = 3, 4, 5$  and  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2$ .

Finally, the ROC (receiver operating characteristics) curves for  $\epsilon = 4$  are depicted in **Figure 13**. The fading parameters are set to be  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=1$ ,  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=2$ , and  $m_{S,D}=m_{S,R_j}=m_{R_j,D}=3$  successively. It can be observed from the figure that the best BER decreases with the fading parameters for a fixed value of  $E_s/N_0$ .



**Figure 13.** Error probability of DF scheme for the cooperative-diversity relay network using different fading parameter values with  $(m_{S,D}=m_{S,R_j}=m_{R_j,D})=1, 2, 3$ .

## 4. Conclusions

In this paper, we present a comprehensive performance comparison of the DE and PSO algorithms both in FSO and in mobile radio communications systems. For the first part, we investigate the optimal transmission distances for different number of relay nodes and power margin values in the parallel DF relaying scheme. Moreover, we analyze the cost function and the execution time for the DE and PSO algorithms. As a second part of this paper, we consider the cooperative-diversity relay network for the mobile radio communications systems operating over Nakagami-m fading channel. We provide a rigorous data for the optimal locations of the relaying terminal in the parallel DF relaying scheme using DE and PSO algorithms. Then, we analyze the bit error probability with varying  $E_s/N_0$  using different values of path loss exponent,  $\epsilon$  and fading parameters of  $m_{S,D}$ ,  $m_{S,R_j}$  and  $m_{R_j,D}$ .

We demonstrate that the cost functions are suitably minimized proving the accuracy of the employed optimization algorithms. We find out that both algorithms have similar execution time, besides PSO is more stable than the DE algorithm. Furthermore, the PSO algorithm outperforms DE algorithm with regard to the cost function. It should be emphasized that both optimization algorithms are reliable and can be used for the applications both in the FSO and mobile radio communications systems.

## Author details

Arif Basgumus<sup>1\*</sup>, Mustafa Namdar<sup>1</sup>, Gunes Yilmaz<sup>2</sup> and Ahmet Altuncu<sup>1</sup>

\*Address all correspondence to: [arif.basgumus@dpu.edu.tr](mailto:arif.basgumus@dpu.edu.tr)

<sup>1</sup> Dumlupinar University, Department of Electrical and Electronics Engineering, Kutahya, Turkey

<sup>2</sup> Uludag University, Department of Electrical and Electronics Engineering, Bursa, Turkey

## References

- [1] Knievel C, Hoeher PA. On particle swarm optimization for MIMO channel estimation. *J. Electr. Comput. Eng.* 2011;2012:1–10. doi:10.1155/2012/614384.
- [2] Su L, Wang P, Liu F. Particle swarm optimization based resource block allocation algorithm for downlink LTE systems. In: *IEEE Asia-Pacific Conference on Communications (APCC '12)*; 15–17 October 2012; Jeju Island, IEEE; 2012. pp. 970–974.
- [3] Dunand FR, Abrao T. Energy-efficient power allocation for WDM/OCDM networks with particle swarm optimization. *J. Opt. Commun. Netw.* 2013;5:512–523. doi:10.1364/JOCN.5.000512.
- [4] Manickavelu D, Vaidyanathan RU. Particle swarm optimization (PSO)-based node and link lifetime prediction algorithm for route recovery in MANET. *EUROSIP J. Wirel. Commun. Netw.* 2014;2014:1–10. doi:10.1186/1687-1499-2014-107.
- [5] Omidvar A, Mohammadi K. Particle swarm optimization in intelligent routing of delay-tolerant network routing. *EUROSIP J. Wirel. Commun. Netw.* 2014;2014:1–8. doi:10.1186/1687-1499-2014-147.
- [6] Xiang L, Ximing L, Ercan MF, Yi Z. A new hybrid algorithm based on collaborative line search and particle swarm optimization. In: *IEEE International Conference on Autonomous Robots and Agents (ICARA '09)*; 10–12 February 2009; Wellington, IEEE; 2009. pp. 486–489.
- [7] Namdar M, Ilhan H, Durak-Ata L. Dispersed chirp-z transform-based spectrum sensing and utilization in cognitive radio networks. *IET Signal Process.* 2014;8:320–329. doi:10.1049/iet-spr.2013.0127.
- [8] Namdar M, Ilhan H, Durak-Ata L. Partial spectrum utilization for energy detection in cognitive radio networks. In: *IEEE International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT '12)*; 3–5 October 2012; St. Petersburg, IEEE; 2012. pp. 989–994.

- [9] Safari M, Uysal M. Relay-assisted free-space optical communication. *IEEE Trans. Wirel. Commun.* 2008;7:5441–5449. doi: 10.1109/T-WC.2008.071352.
- [10] Karimi M, Nasiri-Kenari M. Free space optical communications via optical amplify-and-forward relaying. *J. Lightwave Technol.* 2011;29:242–248. doi:10.1109/JLT.2010.2102003.
- [11] Karimi M, Nasiri-Kenari M. BER analysis of cooperative systems in free-space optical networks. *J. Lightwave Technol.* 2009;27:5639–5647. doi:10.1109/JLT.2009.2032789.
- [12] Karimi M, Nasiri-Kenari M. Outage analysis of relay-assisted free space optical communications. *IET Commun.* 2010;4:1423–1432. doi:10.1049/iet-com.2009.0335.
- [13] Kashani MA, Uysal M. Outage performance and diversity gain analysis of free-space optical multi-hop parallel relaying. *J. Opt. Commun. Netw.* 2013;5:901–909. doi: 10.1364/JOCN.5.000901.
- [14] Kashani MA, Safari M, Uysal M. Optimal relay placement and diversity analysis of relay-assisted free-space optical communications systems. *J. Opt. Commun. Netw.* 2013;5:37–47. doi:10.1364/JOCN.5.000037.
- [15] Ikki SS, Ahmed MH. Performance of cooperative diversity using equal gain combining (EGC) over Nakagami-m fading channels. *IEEE Trans. Wirel. Commun.* 2009;8:557–562. doi:10.1109/TWC.2009.070966.
- [16] Namdar M, Sahin B, Ilhan H, Durak-Ata L. Chirp-z transform based spectrum sensing via energy detection. In: *IEEE Signal Processing and Communications Applications Conference (SIU '12)*; 18–20 April 2012; Mugla, IEEE; 2012. pp. 1–4
- [17] Ikki SS, Ahmed MH. Performance analysis of decode-and-forward cooperative diversity using differential EGC over Nakagami-m fading channels. In: *IEEE Vehicular Technology Conference (VTC '09)*; 26–29 April 2009; Barcelona, IEEE; 2009. pp. 1–6.
- [18] Ikki SS, Ahmed MH. Performance of multiple-relay cooperative diversity systems with best relay selection over Rayleigh fading channels. *EURASIP J. Adv. Signal Process.* 2008;2008:1–7. doi:10.1155/2008/580368.
- [19] Ikki SS, Ahmed MH. Performance analysis of cooperative diversity using equal gain combining (EGC) technique over Rayleigh fading channels. In: *IEEE International Conference on Communications (ICC '07)*; 24–28 June 2007; Glasgow, IEEE, 2007. pp. 5336–5341.
- [20] Namdar M, Ilhan H, Durak-Ata L. Optimal detection thresholds in spectrum sensing with receiver diversity. *Wirel. Personal Commun.* 2016;87:63–81. doi:10.1007/s11277-015-3026-6.
- [21] Olabiyi O, Annamalai A. Analysis of cooperative relay-based energy detection of unknown deterministic signals in cognitive radio networks. In: *The International Conference on Wireless Networks (ICWN '11)*; 18–21 July 2011; Nevada. pp. 1–6.

- [22] Atapattu S, Tellambura C, Jiang H. Relay based cooperative spectrum sensing in cognitive radio networks. In: IEEE Global Telecommunications Conference (GLOBE-COM '09); 30 November–4 December 2009; Honolulu, IEEE, 2009. pp. 1–5.
- [23] Atapattu S, Tellambura C, Jiang H. Energy detection based cooperative spectrum sensing in cognitive radio networks. *IEEE Trans. Wirel. Commun.* 2011;10:1232–1241. doi:10.1109/TWC.2011.012411.100611.
- [24] Waqar OD, McLernon C, Ghogho M. Performance analysis of non-regenerative opportunistic relaying in Nakagami-m fading. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '09); 13–16 September 2009; Tokyo, IEEE; 2009. pp. 231–235.
- [25] Waqar OD, Mughal MO, Gu J, Kim JM. Detection probability analysis for AF assisted cooperative spectrum sensing in cognitive radio networks. In: IEEE International Conference on ICT Convergence (ICTC '11); 28–30 September 2011, Seoul, IEEE; 2011. pp. 461–464.
- [26] Mughal MO, Razi A, Kim JM. Tight upper bounds on average detection probability in cooperative relay networks with selection combiner. *Trans. Emerg. Telecommun. Technol.* 2013;25:340–345. doi:10.1002/ett.2649.
- [27] Basgumus A, Namdar M, Yilmaz G, Altuncu A. Performance comparison of the differential evolution and particle swarm optimization algorithms for the parallel DF relaying in free-space optical communications systems. *Adv. Electr. Comput. Eng.* 2015;15:17–22. doi:10.4316/AECE.2015.02003.
- [28] Namdar M, Ilhan H, Durak-Ata L. Spectrum sensing for cognitive radio with selection combining receiver antenna diversity. In: IEEE Signal Processing and Communications Applications Conference (SIU '13); 24–26 April 2013; Haspolat, IEEE; 2013. pp. 1–4.
- [29] Laneman JN, Tse DNC, Wornell GW. Cooperative diversity in wireless networks: efficient protocols and outage behaviour. *IEEE Trans. Inform. Theor.* 2004;50:3062–3080. doi:10.1109/TIT.2004.838089.
- [30] Laneman JN, Wornell GW. Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Trans. Inform. Theor.* 2003;49:2415–2525. doi:10.1109/TIT.2003.817829.
- [31] Avestimehr AS, Tse DNC. Outage-optimal relaying in the low SNR regime. In: IEEE International Symposium on Information Theory (ISIT '05); 4–9 September 2005; Adelaide, IEEE; 2005. pp. 941–945.



---

# **Genetic Algorithm-Based Approaches for Solving Inexact Optimization Problems and their Applications for Municipal Solid Waste Management**

---

Weihua Jin, Zhiying Hu and Christine W. Chan

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62475>

---

## **Abstract**

This chapter proposes a genetic algorithm (GA)-based approach as an all-purpose problem-solving method for optimization problems with uncertainty. This chapter explains the GA-based method and presents details on the computation procedures involved for solving the three types of inexact optimization problems, which include the ILP, inexact quadratic programming (IQP) and inexact nonlinear programming (INLP) optimization problems.

In the three-stage GA-based method for solution of ILP problems, also called GAILP, the upper and lower bounds of the inexact numbers of coefficients can be calculated directly without any uncertainty in the coefficients by substituting the initial suboptimal decision variables into the objective function. The GAILP has been extended to solve the IQP problems and the more complicated INLP problems. The implementation of these approaches was performed using the Genetic Algorithm Solver of MATLAB.

The proposed GA-based approaches were applied for management of a set of case scenarios related to municipal solid waste management. A comparison of the results generated by the proposed GA-based optimization approach with those produced by the traditional interactive binary analysis method reveals that the proposed approach has fewer limitations and involves less complex procedures in solving the inexact optimization problems.

**Keywords:** genetic algorithms, inexact optimization problem, linear programming, quadratic programming, nonlinear programming

---

## 1. Introduction

Linear and nonlinear programming are considered powerful optimization tools suitable for modeling and solving complex optimization problems in engineering. To handle uncertainty in real world data, inexact parameters and constraints are combined with various kinds of optimization techniques. Often a detailed solution of an inexact programming optimization problem involves a large number of direct comparisons to interactively identify the uncertain relationships among the objective function and decision variables, whether the problems are medium-sized or larger-scaled. When these methods are applied to complicated and nonlinear problems, the number of direct comparisons can become exponential.

The genetic algorithm (GA) method is a suitable optimization approach especially for solving problems that involve nonsmooth and multimodal search spaces. The GA-based optimization technique is suitable for solving linear and nonlinear programming optimization problems with inexact information; and the fields of application include operations research, industrial engineering and management science.

This chapter is organized as follows. Section 2 presents the background and literature review of this research. Section 3 discusses the proposed GA-based methods for solving inexact linear programming (ILP), inexact quadratic programming (IQP) and inexact nonlinear programming (INLP) problems. Section 4 presents the case study of using GAINLP in the solution of an INLP problem of solid waste disposal planning. Section 5 is the conclusion.

## 2. Background and literature review

Economic optimization in the operation programming of solid waste management was first proposed in the 1960s [1]. Different models of waste management planning have been developed in the following decades. The primary considerations involved include cost control, environmental sustainability and waste reutilization. The techniques employed include linear programming [2–5], mixed integer linear programming [6], multiobjective programming [7–9], nonlinear programming [10, 11], as well as their hybrids, which involve probability, fuzzy set and inexact analysis [12–16]. Due to complexity of the nonlinear programming problems for solid waste management, research works in the area are scant; some exceptions include [17, 18].

The approach of operational programming with inexact analysis often treats the uncertain parameters as intervals with known lower and upper bounds and unclear distributions. In real-life problems, while the available information is often inadequate and the distribution functions are often unknown, it is generally possible to represent the obtained data with inexact numbers that can be readily used in the inexact programming models. For decision makers, it is usually more feasible to represent uncertain information as inexact data than to specify distributions of fuzzy sets or probability functions. Hence, various kinds of inexact programming such as ILP, IQP, inexact integer programming (IIP), inexact dynamic programming

(IDP) and inexact multiobjective programming (IMOP) have been developed and are well discussed [10, 11, 19]. It can be observed from these studies that applications of inexact models to practical solid waste planning systems are effective. These research reports demonstrated substantial effort has been developed to traditional binary analysis for ILP and IQP. However, traditional binary analysis methods for ILP and IQP involve unavoidable simplifications and assumptions, which often increase the chance for error in the problem-solving process and adversely affected the quality of the results. Moreover, a more complex model often increases the chance of error in the solution. It has been observed that more complex models often produce less optimal results, and studies that focus on INLP problems are scarce. For example, in [20], the methodology mainly focused on combining endpoint values of the inexact parameters to form a set of deterministic problems, which will only work for particular monotone functions within a small-scale model. Therefore, a more flexible problem-solving method for the general inexact optimization problems is desired.

Engineering problems that have traditionally been formulated as IQP or INLP problems often involve large and uneven search spaces, for which a global optimal solution is often not required. GA is a suitable optimization tool especially for solving complex and nonlinear problems, which involve nonsmooth and multimodal search spaces. Therefore, we suggest a GA-based method as a more effective problem-solving approach than the traditional inexact programming methods.

For implementation of GA, the Genetic Algorithm Solver of Global Optimization Toolbox (GASGOT), developed by MATLAB (Trademark of MathWord), has been adopted. GASGOT implements simulated evolution in the MATLAB environment using both binary and floating point representations and the ordered base representation. This enables flexible implementation of the genetic operators, selection functions, termination functions and evaluation functions. GASGOT was developed by the Department of Industrial Engineering of North Carolina State University as a toolbox of MATLAB. Hence, it runs in a MATLAB workspace and can be easily invoked by other programs.

In this study, the GA linear program solving engine of GASGOT has been adopted for ILP problems and GA nonlinear program solving engine of GASGOT has been adopted for IQP and INLP problems.

### 3. Methodology

#### 3.1. GA-based method for solving ILP problems (GAILP)

A typical ILP problem can be expressed as follows:

$$\text{Max } f^{\pm} = \sum_{j=1}^n [c_j^{\pm} x_j^{\pm}] \quad (1)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm} x_j^{\pm} \leq b_i^{\pm}, i = 1, 2, \dots, m$$

$$x_j^{\pm} \geq 0, j = 1, 2, \dots, n$$

where  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$  are inexact parameters and  $x_j^{\pm}$  is an inexact variable. It is assumed that an optimal solution exists. For an inexact number  $g^{\pm} \in [g^-, g^+]$ ,  $g^+$  and  $g^-$  are the upper and lower bounds, respectively.

GA has been adopted for solving ILP problem. In this GA approach, the upper and lower bounds of the inexact numbers of coefficients  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$  can be determined by substituting the initial suboptimal decision variables into the objective function.  $f^+$  and  $f^-$  can be calculated directly without any uncertainty in the coefficients. This approach is called the GA-based method for solving ILP problems, or the GAILP method.

GAILP has been designed to include three stages, which are discussed as follows:

The objective of the first stage is to get an initial suboptimal  $x_j^s$  for the following problem, which is transformed from the ILP problem defined in Eq. (1):

$$Max f^{\pm} = \sum_{j=1}^n [c_j^r x_j^s] \quad (2)$$

$$s.t. \sum_{j=1}^n a_{ij}^r x_j^s \leq b_i^r, i = 1, 2, \dots, m$$

$$x_j \geq 0, j = 1, 2, \dots, n$$

Where  $a_{ij}^r$ ,  $b_i^r$ ,  $c_j^r$  are random numbers that satisfy the continuous uniform distribution in the intervals of  $[a_{ij}^-, a_{ij}^+]$ ,  $[b_i^-, b_i^+]$  and  $[c_j^-, c_j^+]$ , respectively. Then, the problem is solved by the GA linear program solving engine of GASGOT, which uses the objective function in Eq. (2) as the positive term of the fitness function and the constraints of Eq. (1) as the negative punishment terms. Thus, a suboptimal solution  $f^s$  can be identified and the corresponding decision variables of  $x_j^s$  are also obtained.

In the second stage, the inexact coefficients of  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$  will be determined. Let the determined coefficients corresponding to  $f^+$  be  $a_{ij}^{+\pm}$ ,  $b_i^{+\pm}$ ,  $c_j^{+\pm}$  and those corresponding to  $f^-$  be  $a_{ij}^{-\pm}$ ,  $b_i^{-\pm}$ ,  $c_j^{-\pm}$ . These two sets of coefficients can be obtained using the following method.

Substituting  $x_j^s$  into the formula of Eq. (1) will convert it into Eq. (3).

$$\text{Max } f^{\pm} = \sum_{j=1}^n [c_j^{\pm} x_j^s] \quad (3)$$

$$\text{s.t. } \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

To identify the coefficients  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$  corresponding to  $f^{\pm}$ , a set of objective functions needs to be constructed and solved. Since  $x_j^s$  are suboptimal variables, which tend to make the objective function closer to  $f^+$ , consider  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$  as variables, then the objective function of Eq. (4) can be constructed so as to find  $c_j^{\pm}$ .

$$\text{Max } f^{\pm} = \sum_{j=1}^n [c_j^{\pm} x_j^s] \quad (4)$$

$$\text{s.t. } \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

The coefficients  $c_j^{\pm}$  are considered to be corresponding to  $f^{\pm}$ .

Meanwhile, the objective function presented in Eq. (5) can be constructed so as to find  $c_j^{\pm}$ .

$$\text{Min } f^{\pm} = \sum_{j=1}^n [c_j^{\pm} x_j^s] \quad (5)$$

$$\text{s.t. } \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

There are two kinds of decision schemes for inexact programming problems, which are the conservative scheme and the optimistic scheme [21]. The former assumes less risk than the latter, so that for a maximization objective function, planning for the lower bound of an objective value represents the conservative scheme and planning for the upper bound of an objective value represents the optimistic scheme. In terms of constraints, the conservative scheme involves more rigorous or stringent constraints and the optimistic scheme adopts more tolerant ones.

Thus, the problem of searching for  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$  of the optimistic scheme and corresponding to the upper bound of the objective value of  $f^+$  can be represented as follows:

$$\text{Max } \sum_{j=1}^n \text{abs}(a_{ij}^{\pm} x_j^s - b_i^{\pm}) \quad (6)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

The problem,

$$Min \sum_{j=1}^n abs(a_{ij}^{\pm} x_j^s - b_i^{\pm}) \quad (7)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

will give  $a_{ij}^{\pm-}$ ,  $b_i^{\pm-}$  of the conservative scheme, corresponding to the lower bound of the objective value of  $f^-$ .

Hence, the values of  $a_{ij}^{\pm+}$ ,  $b_i^{\pm+}$ ,  $c_j^{\pm+}$  and  $a_{ij}^{\pm-}$ ,  $b_i^{\pm-}$ ,  $c_j^{\pm-}$  can be calculated.

In the third stage, the problem represented in Eq. (1) is converted into the following two subproblems:

For  $f^+$ ,

$$Max f^+ = \sum_{j=1}^n [c_j^+ x_j^+] \quad (8)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm+} x_j^{\pm+} \leq b_i^{\pm+}, i = 1, 2, \dots, m$$

$$x_j^{\pm+} \geq 0, j = 1, 2, \dots, n$$

For  $f^-$ ,

$$Max f^- = \sum_{j=1}^n [c_j^{\pm-} x_j^{\pm-}] \quad (9)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm-} x_j^{\pm-} \leq b_i^{\pm-}, i = 1, 2, \dots, m$$

$$x_j^{\pm-} \geq 0, j = 1, 2, \dots, n$$

This step eliminates the inexact parameters in Eq. (1) and generates instead Eq. (8) and Eq. (9) as typical linear programming (LP) problems, which can be solved easily.

Generally speaking, the interactive binary algorithm (IBA) proposed in [19, 22] can be used for solving inexact linear problems reliably and relatively quickly. However, this binary algorithm has some limitations. One of them, for example, is the limitation that the upper and lower bounds of an inexact coefficient cannot have different signs. In contrast, the GAILP does not have this kind of limitation because the GA method does not depend on any assumed distribution of the inexact parameter. Hence, the GAILP method effectively extends the scope of problems solvable using the methods of ILP. It is more adaptable for real world applications of optimization problems with uncertainty.

A sample ILP problem in [22] is as follows,

$$\text{Max } f^{\pm} = c_1 x_1^{\pm} + c_2 x_2^{\pm} \quad (10)$$

$$\text{s.t. } a_{11} x_1^{\pm} + a_{12} x_2^{\pm} \leq b_1$$

$$a_{21} x_1^{\pm} + a_{22} x_2^{\pm} \leq b_2$$

where  $c_1 = [26, 30]$ ,  $c_2 = [-6, -5.5]$ ,  $a_{11} = [8, 10]$ ,  $a_{12} = [-14, -12]$ ,  $b_1 = [3.8, 4.2]$ ,  $a_{21} = [2.4, 2.8]$ ,  $a_{22} = [3.4, 4]$ ,  $b_2 = 6.5$

By using the traditional IBA method [22], two submodels are obtained,

$$\text{Max } f^{+} = 30x_1^{+} - 5.5x_2^{-}$$

$$\text{s.t. } 8x_1^{+} - 14x_2^{-} \leq 4.2$$

$$2.4x_1^{+} + 4x_2^{-} \leq 6.5$$

$$x_1^{+} \geq 0, x_2^{-} \geq 0$$

and

$$\text{Max } f^{-} = 26x_1^{-} - 6.0x_2^{+}$$

$$\text{s.t. } 10x_1^{-} - 12x_2^{+} \leq 3.8$$

$$2.8x_1^- + 3.4x_2^+ \leq 6.5$$

$$x_1^- \geq 0, x_2^+ \geq 0$$

The results were  $f^+ = 45.78$ ,  $x_1 = 1.64$ ,  $x_2 = 0.64$ ;  $f^- = 30.77$ ,  $x_1 = 1.37$ ,  $x_2 = 0.79$ .

By using the GAILP, the results can be calculated with the following objective functions:

$$\text{Max } f^+ = 30x_1^+ - 5.5x_2^+$$

$$\text{s.t. } 8x_1^+ - 14x_2^+ \leq 4.2$$

$$2.4x_1^+ + 3.4x_2^+ \leq 6.5$$

$$x_1^+ \geq 0, x_2^+ \geq 0$$

and

$$\text{Max } f^- = 26x_1^- - 6.0x_2^-$$

$$\text{s.t. } 10x_1^- - 12x_2^- \leq 3.8$$

$$2.8x_1^- + 4x_2^- \leq 6.5$$

$$x_1^- \geq 0, x_2^- \geq 0$$

The results were  $f^+ = 48.15$ ,  $x_1 = 1.73$ ,  $x_2 = 0.69$ ;  $f^- = 29.15$ ,  $x_1 = 1.29$ ,  $x_2 = 0.72$ .

The GAILP method generates a solution, which is different from that obtained using the IBA proposed in [22]. A comparison will be discussed as follows:

For the  $f^+$  optimistic scheme, the GAILP method can generate a result that is guaranteed to be as close as possible to the upper bound of the constraints. Hence, the maximized value of the objective function is greater than that produced by the IBA. For the  $f^-$  conservative scheme, the GAILP method has a higher probability of satisfying the requirements of the constraints as close as possible to the lowest limit. Hence, the maximized objective value is smaller.

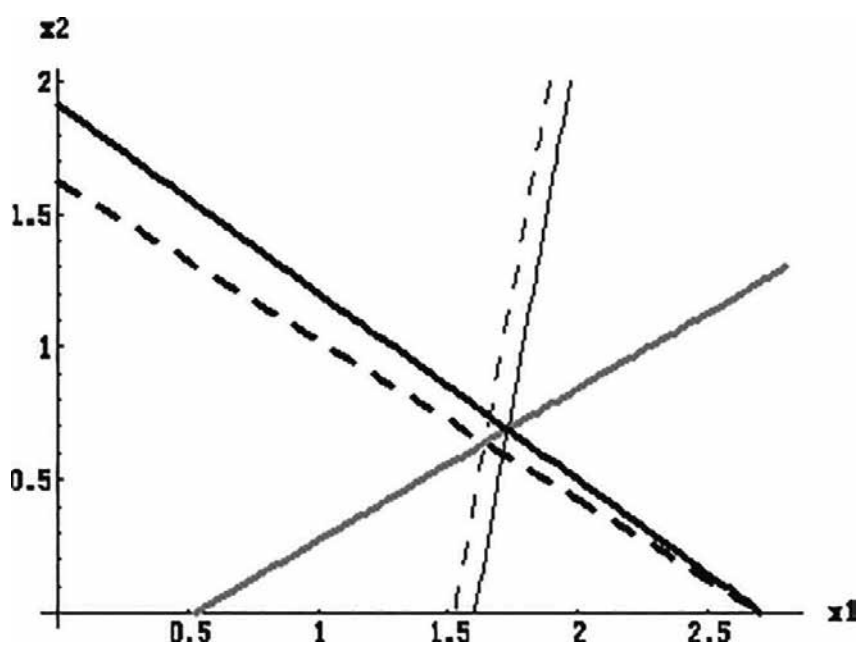


Figure 1. Optimistic scheme,  $f^+$ .

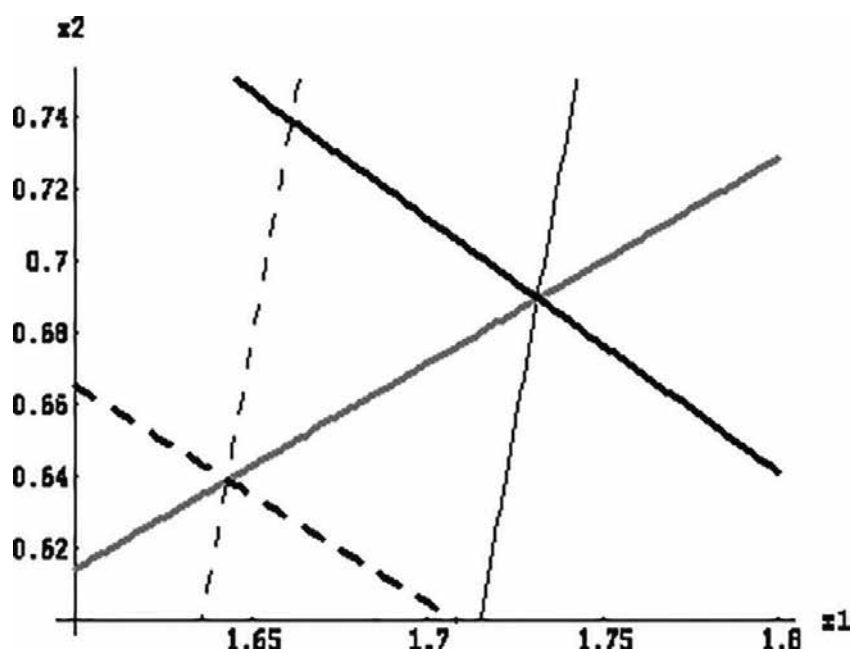
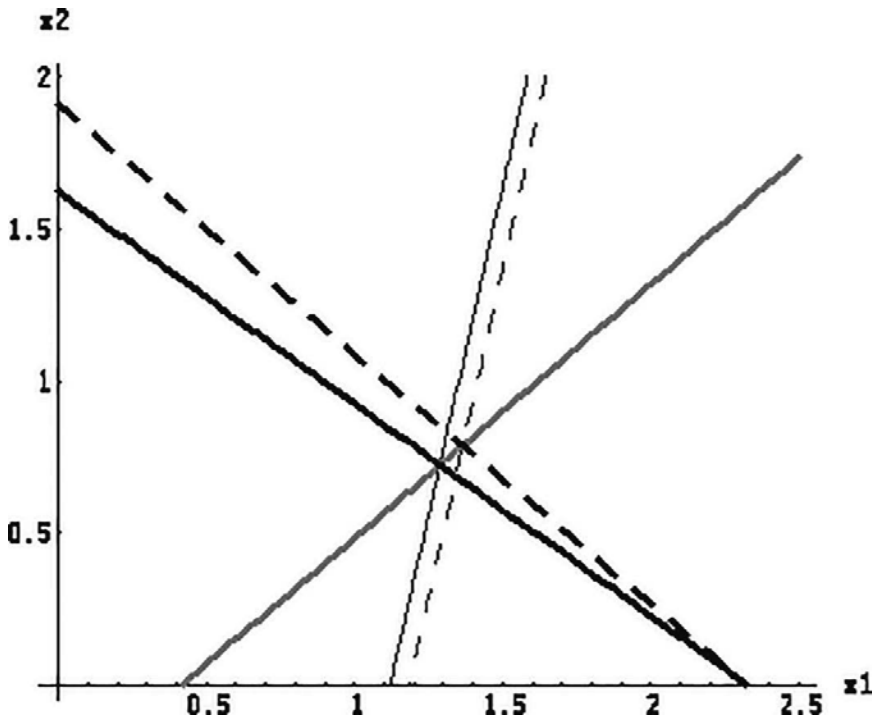


Figure 2. Zoom-in of the optimistic scheme,  $f^+$ .

In **Figures 1 to 4**, the bold lines denote the boundaries of the constraints, which limit the possible values for  $x_1, x_2$  to the left lower area. The constraint  $a_{11}x_1^+ + a_{12}x_2^+ \leq b_1$  is shown in these figures as the grey bold solid lines, which is the same for both the IBA and GAILP methods. The dark bold dotted lines represent the constraint of  $a_{21}x_1^+ + a_{22}x_2^+ \leq b_2$  given by the IBA and the dark bold solid lines represent the same constraint given by the proposed GAILP method.

The boundaries, together with the  $x_1, x_2$  axes, enclose the entire area defined by the constraints. The objective functions  $f^+ = 30x_1^+ - 5.5x_2^+$  or  $f^- = 26x_1^- - 6x_2^-$  are groups of parallel lines, as shown in **Figures 1 to 4** by the thin solid and dotted lines. With different values of  $x_1$  and  $x_2$ , these objective function lines would produce different intercepts on the two axes. These constraints restrict the objective function lines to cross with the constraints area, so that, at some vertex, the objective function would reach its extreme (i.e., maximized or minimized) values.

In **Figures 1 to 4**, the thin dotted lines are given by the IBA and the thin solid lines represent the objective functions given by the proposed GAILP method. The legends for **Figure 1 to 4** are listed in **Table 1**.



**Figure 3.** Conservative scheme,  $f^-$ .

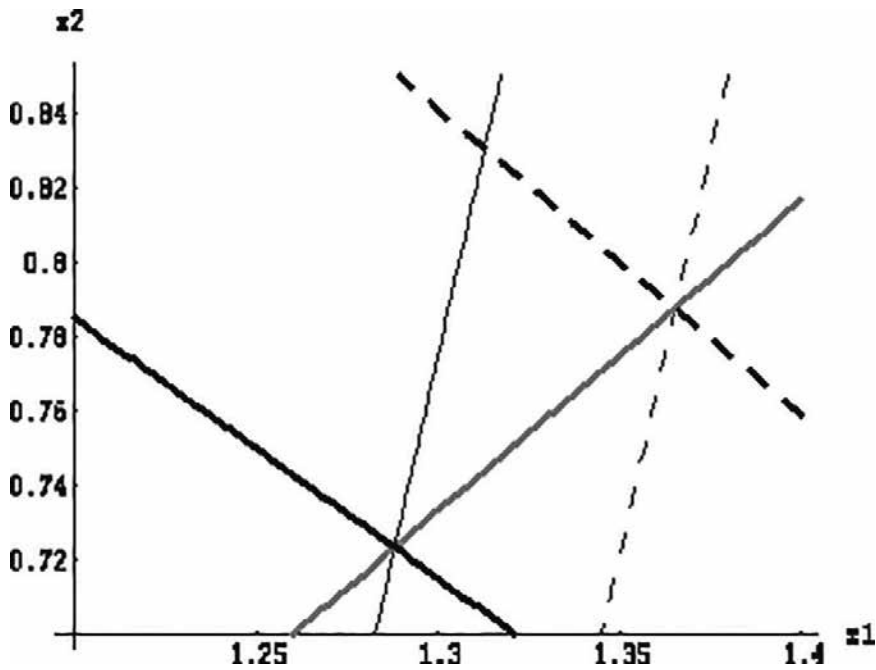


Figure 4. Zoom-in of the conservative scheme,  $f^-$ .




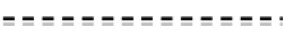

	The constraint $a_{21}x_1^+ + a_{22}x_2^+ \leq b_2$ given by IBA
	The constraint $a_{21}x_1^+ + a_{22}x_2^+ \leq b_2$ given by GAILP
	The constraint $a_{11}x_1^+ + a_{12}x_2^+ \leq b_1$
	Objective function line given by IBA
	Objective function line given by GAILP

Table 1. Legends for Figures 1 to 4.

### 3.2. GA-based method for solving IQP problems (GAIQP)

The GAILP method can be extended to solve the IQP problems or other more complicated INLP problems.

A typical IQP problem is formulated as follows:

$$Max f^+ = \sum_{j=1}^n [c_j^+ x_j^+ + d_j^+ (x_j^+)^2] \quad (11)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm} x_j^{\pm} \leq b_i^{\pm}, i = 1, 2, \dots, m$$

$$x_j^{\pm} \geq 0, j = 1, 2, \dots, n$$

where  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$ ,  $d_j^{\pm}$  are inexact parameters and  $x_j^{\pm}$  is an inexact variable.

In stage one, to obtain an initial suboptimal  $x_j^s$  from a problem transformed from the IQP problem:

$$Max f = \sum_{j=1}^n [c_j^r x_j + d_j^r (x_j)^2] \quad (12)$$

$$s.t. \sum_{j=1}^n a_{ij}^r x_j^r \leq b_i^r, i = 1, 2, \dots, m$$

$$x_j \geq 0, j = 1, 2, \dots, n.$$

where  $a_{ij}^r$ ,  $b_i^r$ ,  $c_j^r$ ,  $d_j^r$  are random numbers that satisfy the continuous uniform distribution in the intervals  $[a_{ij}^-, a_{ij}^+]$ ,  $[b_i^-, b_i^+]$ ,  $[c_j^-, c_j^+]$  and  $[d_j^-, d_j^+]$ . Then, a suboptimal solution  $f^s$  can be identified, and the corresponding decision variables  $x_j^s$  are also obtained.

In the second stage, substituting  $x_j^s$  into the formula in Eq. (11). To determine the coefficients  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$ ,  $d_j^{\pm}$  corresponding to  $f^{\pm}$

$$Max f^{\pm} = \sum_{j=1}^n [c_j^{\pm} x_j^s + d_j^{\pm} (x_j^s)^2] \quad (13)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

and

$$Min f^{\pm} = \sum_{j=1}^n [c_j^{\pm} x_j^s + d_j^{\pm} (x_j^s)^2] \quad (14)$$

$$s.t. \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

To determine  $a_j^{\pm}, b_i^{\pm}$  of the optimistic scheme and corresponding to the upper limit of the objective value of  $f^+$

$$\text{Max} \sum_{j=1}^n \text{abs}(a_{ij}^{\pm} x_j^s - b_i^{\pm}) \quad (15)$$

$$\text{s.t.} \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

To obtain  $a_j^{\pm}, b_i^{\pm}$ ,

$$\text{Min} \sum_{j=1}^n \text{abs}(a_{ij}^{\pm} x_j^s - b_i^{\pm}) \quad (16)$$

$$\text{s.t.} \sum_{j=1}^n a_{ij}^{\pm} x_j^s \leq b_i^{\pm}, i = 1, 2, \dots, m$$

In the third stage, the problem expressed in Eq. (11) has been converted into the following two subproblems:

For  $f^+$ ,

$$\text{Max } f^+ = \sum_{j=1}^n [c_j^{\pm} x_j^{\pm} + d_j^{\pm} (x_j^{\pm})^2] \quad (17)$$

$$\text{s.t.} \sum_{j=1}^n a_{ij}^{\pm} x_j^{\pm} \leq b_i^{\pm}, i = 1, 2, \dots, m$$

$$x_j^{\pm} \geq 0, j = 1, 2, \dots, n$$

For  $f^-$ ,

$$\text{Max } f^- = \sum_{j=1}^n [c_j^{\pm} x_j^{\pm} + d_j^{\pm} (x_j^{\pm})^2] \quad (18)$$

$$\text{s.t.} \sum_{j=1}^n a_{ij}^{\pm} x_j^{\pm} \leq b_i^{\pm}, i = 1, 2, \dots, m$$

$$x_j^{\pm} \geq 0, j = 1, 2, \dots, n$$

The inexact information has been incorporated in these two subproblems. These two subproblems, as typical nonlinear programming problems, can be solved by the GA nonlinear program solving engine of GASGOT.

### 3.3. GA-based method for solving inexact nonlinear problems (GAINLP)

Quadratic programming problems are specific cases of nonlinear programming problems. Due to the lack of generally applicable algorithms for handling the nonlinear structure and the inexact information embedded in the structure, most nonlinear programming problems are difficult to solve. The IBA method proposed in [11, 22] is not intended for dealing with generic nonlinear problems. In contrast, the GA-based method can be used as a general problem solver for this type of problems because there is not much difference for GA between treating the term of  $x_i^2$  in quadratic programming problems and the terms  $x_i x_j$  or  $x_i^{0.28}$  in generic nonlinear programming problems. GAIQP can be modified to solve generic inexact nonlinear programming.

In the following, a computation experiment will be conducted to illustrate how the GAINLP method can handle complicated inexact nonlinear problems. A sample INLP problem is as follows:

$$\text{Max } f^\pm = c_1^\pm x_1^\pm - c_2^\pm (x_1^\pm)^{0.3} - d_1^\pm x_2^\pm + d_2^\pm (x_1^\pm x_2^\pm) \quad (19)$$

$$\text{s.t. } a_{11}^\pm (x_1^\pm)^{0.5} + a_{12}^\pm x_2^\pm \leq b_1^\pm,$$

$$x_1^\pm + a_2^\pm x_2^\pm \leq b_2^\pm,$$

$$x_j^\pm \geq 0, j = 1, 2.$$

where  $a_{ij}^\pm, b_i^\pm, c_j^\pm, d_j^\pm$  are inexact parameters and  $x_j^\pm$  is an inexact variable. In this experiment,

$$\begin{aligned} [c_1^-, c_1^+] &= [16, 18]; [c_2^-, c_2^+] = [12, 14]; [d_1^-, d_1^+] = \\ &[4, 5]; [d_2^-, d_2^+] = [14, 15]; [a_{11}^-, a_{11}^+] = [4.5, 5.5]; [a_{12}^-, a_{12}^+] = \\ &[1.8, 2.2]; [b_1^-, b_1^+] = [1.8, 2.1]; [a_2^-, a_2^+] = [1.8, 2.2]; [b_2^-, b_2^+] = [0.9, 1.1]. \end{aligned}$$

GAINLP has been designed to include the three stages of problem solving.

In stage one, to obtain the initial suboptimal  $x_j^s$ , the random numbers of  $a_{ij}^r, b_i^r, c_j^r, d_j^r$  were selected to transform this INLP problem into a NLP problem, such that  $a_{ij}^r, b_i^r, c_j^r, d_j^r$  satisfy the continuous uniform distribution in the intervals of  $[a_{ij}^-, a_{ij}^+], [b_i^-, b_i^+], [c_j^-, c_j^+]$  and  $[d_j^-, d_j^+]$ .

$$\text{Max } f^s = c_1^r x_1^s - c_2^r (x_1^s)^{0.3} - d_1^r x_2^s + d_2^r (x_1^s x_2^s) \quad (20)$$

$$\text{s.t. } a_{11}^r (x_1^s)^{0.5} + a_{12}^r x_2^s \leq b_1^r,$$

$$x_1^s + a_2^r x_2^s \leq b_2^r,$$

$$x_j^s \geq 0, j = 1, 2.$$

Then, the heuristic search algorithm of the GA nonlinear program solving engine of GASGOT can be used to identify a suboptimal solution  $f^s$ , and the corresponding decision variable  $x_j^s$ . The objective function in Eq. (20) was used as the positive term of the fitness function and the constraints of Eq. (19) adopted as the negative punishment terms. The results are  $x_1^s = 0.346$ ,  $x_2^s = 0.171$ ,  $f^s = -2.296$ .

In stage two, by substituting  $x_1^s$ ,  $x_2^s$  into Eq. (19), the inexact coefficients of  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$ ,  $d_j^{\pm}$  will be determined. The  $x_1^s$ ,  $x_2^s$  obtained in stage one are used to construct two optimization problems in order to determine the coefficients of  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$ ,  $d_j^{\pm}$  and  $a_{ij}^{\pm}$ ,  $b_i^{\pm}$ ,  $c_j^{\pm}$ ,  $d_j^{\pm}$ , respectively. The coefficients from the first group are considered to be corresponding to the optimistic scheme  $f^+$ , while those from the second group correspond to the conservative scheme  $f^-$ . Considering  $c_j^{\pm}$ ,  $d_j^{\pm}$  are variables, the following two objective functions can be constructed:

$$\text{Max } f^+ = c_1^{\pm+} x_1^s - c_2^{\pm+} (x_1^s)^{0.3} - d_1^{\pm+} x_2^s + d_2^{\pm+} (x_1^s x_2^s) \quad (21)$$

and

$$\text{Min } f^- = c_1^{\pm-} x_1^s - c_2^{\pm-} (x_1^s)^{0.3} - d_1^{\pm-} x_2^s + d_2^{\pm-} (x_1^s x_2^s) \quad (22)$$

$$\text{s.t. } c_1^{\pm+}, c_1^{\pm-} \in [16, 18]$$

$$c_2^{\pm+}, c_2^{\pm-} \in [12, 14]$$

$$d_1^{\pm+}, d_1^{\pm-} \in [4, 5]$$

$$d_2^{\pm+}, d_2^{\pm-} \in [14, 15]$$

To determine  $a_{ij}^{\pm+}, b_i^{\pm+}$  of the optimistic scheme corresponding to the upper limit of the objective value  $f^+$ , the objective function can be constructed as follows:

$$\text{Max } \text{abs}(a_{11}^{\pm+}(x_1^s)^{0.5} + a_{12}^{\pm+}x_2^s - b_1^{\pm+}) \quad (23)$$

$$\text{s.t. } a_{11}^{\pm+}(x_1^s)^{0.5} + a_{12}^{\pm+}x_2^s \leq b_1^{\pm+}$$

and

$$\text{Max } \text{abs}(x_1^s + a_2^{\pm+}x_2^s - b_2^{\pm+})$$

$$\text{s.t. } x_1^s + a_2^{\pm+}x_2^s \leq b_2^{\pm+}$$

The objective functions to get  $a_{ij}^{\pm-}, b_i^{\pm-}$  of the conservative scheme are

$$\text{Min } \text{abs}(a_{11}^{\pm-}(x_1^s)^{0.5} + a_{12}^{\pm-}x_2^s - b_1^{\pm-}) \quad (24)$$

$$\text{s.t. } a_{11}^{\pm-}(x_1^s)^{0.5} + a_{12}^{\pm-}x_2^s \leq b_1^{\pm-}$$

and

$$\text{Min } \text{abs}(x_1^s + a_2^{\pm-}x_2^s - b_2^{\pm-})$$

$$\text{s.t. } x_1^s + a_2^{\pm-}x_2^s \leq b_2^{\pm-}$$

By solving Eqs. (21)–(24), the values of all the inexact coefficients are obtained, i.e.,  
 $a_{11}^{\pm+}=4.5, a_{12}^{\pm+}=1.8, b_1^{\pm+}=2.1, a_2^{\pm+}=1.8, b_2^{\pm+}=1.1;$   $a_{11}^{\pm-}=5.5, a_{12}^{\pm-}=2.2, b_1^{\pm-}=1.8, a_2^{\pm-}=2.2, b_2^{\pm-}=0.9;$   
 $c_1^{\pm+}=18, c_2^{\pm+}=12, d_1^{\pm+}=4, d_2^{\pm+}=15; c_1^{\pm-}=16, c_2^{\pm-}=14, d_1^{\pm-}=5, d_2^{\pm-}=14$

In stage three, the objective function presented in Eq. (20) is converted into the following two subproblems:

$$\text{Max } f^+ = 18x_1^\pm - 12(x_1^\pm)^{0.3} - 4x_2^\pm + 15(x_1^\pm x_2^\pm)$$

$$\text{s.t. } 4.5(x_1^\pm)^{0.5} + 1.8x_2^\pm \leq 2.1,$$

$$x_1^\pm + 1.8x_2^\pm \leq 1.1,$$

$$x_1^\pm \geq 0, x_2^\pm \geq 0.$$

and

$$\text{Max } f^- = 16x_1^\pm - 14(x_1^\pm)^{0.3} - 5x_2^\pm + 14(x_1^\pm x_2^\pm)$$

$$\text{s.t. } 5.5(x_1^\pm)^{0.5} + 2.2x_2^\pm \leq 1.8,$$

$$x_1^\pm + 2.2x_2^\pm \leq 0.9,$$

$$x_1^\pm \geq 0, x_2^\pm \geq 0.$$

The inexact parameters in Eq. (20) have been eliminated, and two typical nonlinear optimization problems have been generated instead. The solution of the example (Eq. (19)) is  $f^\pm = [-5.5575, -1.72]$ ,  $x_1^\pm = [0.24727, 0.38496]$ , and  $x_2^\pm = [0.1989, 0.2053]$ .

As demonstrated above, it can be seen that the GAINLP method can generate the optimal result without any simplification or assumption, and it can be adapted for applications of optimization problems with uncertainty. The next section demonstrates application of this method to a real world regional waste management problem.

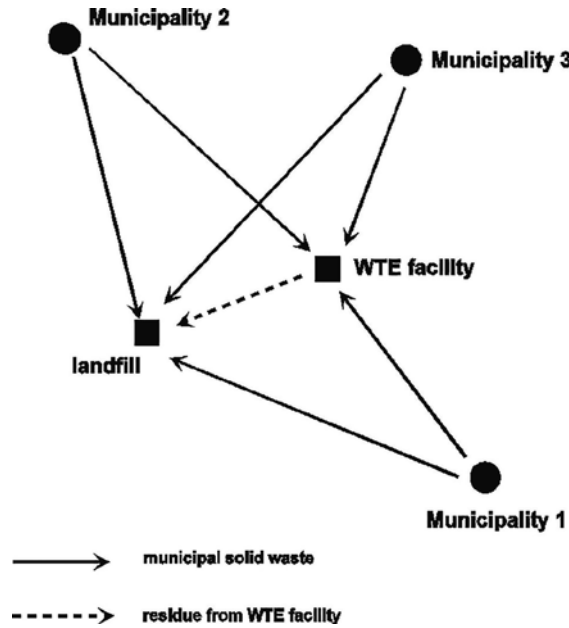
## 4. Case study

Solid waste management is the process of removing waste materials from the surrounding environment, which involves the collection, separation, storage, processing, treatment, transport, recovery and disposal of solid waste. Landfill and incineration are two of the most commonly used solid waste disposal methods. The objective of a solid waste management process is to dispose discarded materials in a timely manner so as to prevent the spread of disease, minimize the likelihood of contamination and reduce their effects on human health and the environment.

The economy of scale (ES) is a microeconomics term, and it refers to the advantages that enterprises obtain due to their size or scale of operation, with the cost per unit of output generally decreasing as the scale increases and fixed costs are distributed over more units of output. In a solid waste management system, ES exists within the transportation process [23] and it can be expressed as a sizing model with a power law [11].

$$C_t = C_{re} (X_t / X_{re})^{1+m} \quad (25)$$

where  $X_t$  (t/d) is a waste flow decision variable;  $X_{re}$  (t/d) is a reference waste flow;  $C_t$  (\$/t) is the transportation unit cost due to the ES of waste flow  $X_t$  (t/d);  $C_{re}$  (\$/t) is a coefficient reflecting the significance of the ES to the unit cost of waste transported for reference waste flow  $X_{re}$  (t/d),  $C_{re} < 0$ ; and  $m$  is an ES exponent which reflects the unit cost decline with respect to the waste flow,  $-1 < m < 0$ .



**Figure 5.** Case study of municipalities and waste management facilities.

The study region includes three municipalities, a waste-to-energy (WTE) facility and a landfill, as shown in **Figure 5**. Three time periods are considered; each has an interval of five years. Over the 15-year planning horizon, an existing landfill and WTE facilities are available to serve the municipal solid waste (MSW) disposal needs in the region. The landfill has an existing capacity of  $[2.05, 2.30] \times 10^6$  t, and the WTE facility has a capacity of  $[500, 600]$  t/d. The WTE facility generates residues of approximately 30% (on a mass basis) of the incoming waste streams, and its revenue from energy sale is  $[15, 25]$  \$/t combusted.

**Table 2** shows the waste generation rates of the three municipalities and the operating costs of the two facilities in the three periods.

Time period	k=1	k=2	k=3
<b>Waste generation <math>WG_{jk}^{\pm}</math> (t/d)</b>			
Municipality 1 ( $j=1$ )	[260, 340]	[310, 390]	[360, 440]
Municipality 2 ( $j=2$ )	[160, 240]	[185, 265]	[210, 290]
Municipality 3 ( $j=3$ )	[260, 340]	[260, 340]	[310, 390]
<b>Operation cost <math>OP_{ik}^{\pm}</math> (\$/t)</b>			
Landfill ( $i=1$ )	[30, 45]	[40, 60]	[50, 80]
WTE facility ( $i=2$ )	[55, 75]	[60, 85]	[65, 95]

**Table 2.** Data for the waste generation and treatment/disposal.

Taking into consideration the effects of the ES, the INLP model can be formulated as follows:

$$\begin{aligned}
 \text{Min } f^{\pm} = & \sum_{i=1}^2 \sum_{j=1}^3 \sum_{k=1}^3 L_k x_{ijk}^{\pm} [A_{re_{ijk}}^{\pm} + C_{re_{ijk}}^{\pm} (x_{ijk}^{\pm} / X_{re_{ijk}}^{\pm})^{1+m}] \\
 & + (OP_{ik}^{\pm}) + \sum_{k=1}^3 L_k (FE * \sum_{j=1}^3 (x_{2jk}^{\pm})) \\
 & \left\{ A_{re_{WTE-LF_k}}^{\pm} + C_{re_{WTE-LF_k}}^{\pm} (FE * \sum_{j=1}^3 x_{2jk}^{\pm} / X_{re_{WTE-LF_k}}^{\pm})^{1+m} + (OP_{1k}^{\pm}) \right\} \\
 & - \sum_{k=1}^3 \sum_{j=1}^3 x_{2jk}^{\pm} RE_k^{\pm} \\
 \text{s.t. } & \sum_{j=1}^3 \sum_{k=1}^3 L_k [x_{1jk}^{\pm} + x_{2jk}^{\pm} FE] \leq TL^{\pm} \\
 & \sum_{j=1}^3 x_{2jk}^{\pm} \leq TE^{\pm}, \forall k \\
 & \sum_{i=1}^2 x_{ijk}^{\pm} = WG_{jk}^{\pm}, \forall j, k \\
 & X_{ijk}^{\pm} \geq 0, \forall i, j, k
 \end{aligned} \tag{26}$$

where  $i$  is the type of waste management facility ( $i=1, 2$ , where  $i=1$  for landfill, 2 for WTE);  $j$  is the city,  $j=1, 2, 3$ ;  $k$  is the time period,  $L_k$  is the length of period  $k$ ,  $L_1=L_2=L_3=365 \times 5$  (day);  $OP_{ik}^\pm$  is the operating cost of facility during period  $k$  (\$/t);  $RE_k^\pm$  is the revenue from WTE during period  $k$  (\$/t),  $RE_1^\pm=RE_2^\pm=RE_3^\pm=[15, 25]$ ;  $TE^\pm$  is the capacity of WTE (t/d);  $TL^\pm$  is the capacity of the landfill (t);  $WG_{jk}^\pm$  is the waste disposal demand in city during period  $k$  (t/d);  $x_{ijk}^\pm$  is the waste flow from city  $j$  to facility  $i$  during period  $k$  (t/d).

In this objective function (Eq. (26)), the first term on the right side reflects the transportation costs in each management period ( $k=1$  to 3) from each city to each waste treatment unit, and the related operation costs. The second term reflects the cost incurred in transporting the products from the WTE facility to the landfill, and the operation cost at the landfill. The third term is the revenue generated from the WTE facility.

The MSW generation rates generally vary between different municipalities and for different periods, and the costs for the waste transportation and treatment also vary temporally and spatially. Furthermore, interactions exist between the waste flows and their transportation costs due to the effects of the ES (Eq. (25)). **Table 3** and **Table 4** show the parameters related to the ES, which include the fixed unit transportation cost  $A_{re}$ , the reference waste flow  $X_{re}$  and the coefficient  $C_{re}$  corresponding to  $X_{re}$ .

Fixed unit transportation cost (\$/t)				Reference waste flow (t/d)			
	$k=1$	$k=2$	$k=3$		$k=1$	$k=2$	$k=3$
City-to-landfill							
$A_{re_{11k}}^\pm$	[14.58, 19.40]	[16.04, 21.34]	[17.64, 23.48]	$X_{re_{11k}}^\pm$	[220, 250]	[240, 280]	[260, 320]
$A_{re_{12k}}^\pm$	[12.65, 16.87]	[13.92, 18.56]	[15.31, 20.41]	$X_{re_{12k}}^\pm$	[160, 200]	[180, 220]	[220, 260]
$A_{re_{13k}}^\pm$	[15.30, 20.49]	[16.83, 22.53]	[18.52, 24.79]	$X_{re_{13k}}^\pm$	[160, 200]	[180, 240]	[200, 240]
City-to-WTE							
$A_{re_{21k}}^\pm$	[11.57, 15.42]	[12.73, 16.97]	[14.00, 18.66]	$X_{re_{21k}}^\pm$	[200, 240]	[240, 280]	[280, 320]
$A_{re_{22k}}^\pm$	[12.17, 16.15]	[13.39, 17.76]	[14.73, 19.54]	$X_{re_{22k}}^\pm$	[120, 170]	[150, 190]	[180, 220]
$A_{re_{23k}}^\pm$	[10.60, 14.10]	[11.67, 15.51]	[12.83, 17.06]	$X_{re_{23k}}^\pm$	[220, 270]	[220, 270]	[240, 270]
WTE-to-landfill							
$A_{re_{WTE-L} F_k}^\pm$	[5.71, 7.62]	[6.28, 8.38]	[6.91, 9.33]	$X_{re_{WTE-L} F_k}^\pm$	[170, 200]	[200, 260]	[240, 270]

**Table 3.** Fixed unit transportation costs and reference waste flows.

	k=1	k=2	k=3		k=1	k=2	k=3
$C_{re_{11k}}^-$	-2.7	-3.4	-3.8	$C_{re_{21k}}^-$	-1.9	-2.6	-3.3
$C_{re_{11k}}^+$	-4.1	-5.0	-6.3	$C_{re_{21k}}^+$	-3.1	-4.0	-5.0
$C_{re_{12k}}^-$	-1.7	-2.1	-2.8	$C_{re_{22k}}^-$	-1.2	-1.7	-2.2
$C_{re_{12k}}^+$	-2.8	-3.4	-4.5	$C_{re_{22k}}^+$	-2.3	-2.8	-3.6
$C_{re_{13k}}^-$	-2.1	-2.5	-3.1	$C_{re_{23k}}^-$	-2.0	-2.2	-2.6
$C_{re_{13k}}^+$	-3.4	-4.5	-5.0	$C_{re_{23k}}^+$	-3.2	-3.5	-3.9
$C_{re_{WTE-L F_k}}^-$	-0.8	-1.1	-1.4	$C_{re_{WTE-L F_k}}^+$	-1.3	-1.8	-2.1

Note: The + and – superscript sign of  $C_{re}$  represents the value of  $C_{re}$  relevant to the upper and lower bound of  $X_{re}$  only.

**Table 4.**  $C_{re}$  (\$/t) The economy of scale coefficient corresponding to reference waste flow  $X_{re}$ .

Hence, it can be observed that the traditional IBA cannot solve this problem without additional assumptions or simplifications. The following discussion will explain how traditional methods solve this problem by simplifying the nonlinear effects of the ES.

- (i) Let  $m = -1$ , the effects of the ES are totally ignored. This converts the INLP problem to an ILP problem, and the GAILP method can solve the problem.
- (ii) Assuming  $-0.2 < m < -0.1$ , it is indicated that the nonlinear relationships in Eq. (26) can be approximated with grey quadratic functions within a predetermined degree of error. Thus, the INLP problem is converted into an IQP problem.

The left two columns of **Table 5** list the solutions for  $m = -1$  and  $-0.2 < m < -0.1$ .

Both of the above simplifications introduce inaccuracy and limitations. When the value of  $m$  deviates away from the predetermined value, this inaccuracy will increase dramatically.

Applying the GAINLP model on the inexact nonlinear programming problem, the optimization problem can be solved directly without additional assumptions for the effects of the ES. Three different scenarios, ( $m = -0.1$ ,  $m = -0.3$ , and  $m = -0.5$ ) have been tested, and the solutions given by the GAINLP model are shown in the right three columns of **Table 5**.

The above three scenarios assume that the ES exponent is universal in the whole region during the entire period. However, this is not always necessarily true for practical engineering problems. More common situations may involve different scale exponents for various combinations of municipalities and facilities in different periods. Thus, **Table 6** illustrates the solutions for the 4th scenario, which involves different scale exponents.

Decision variable(t/d)	ILP solution	IQP solution	Other solutions		
	m=-1	-0.2<m<-0.1	m=-0.1	m=-0.3	m=-0.5
$x_{111}^{\pm}$	[210, 290]	[250, 290]	[203, 292]	[100, 221]	[35, 88]
$x_{112}^{\pm}$	0	[310, 350]	[1, 36]	[1, 44]	[1, 36]
$x_{113}^{\pm}$	[0, 30]	[360, 440]	[1, 44]	[126, 190]	[240, 300]
$x_{121}^{\pm}$	0	[0, 30]	[1, 43]	[60, 141]	[144, 240]
$x_{122}^{\pm}$	[0, 65]	[185, 225]	[1, 73]	[20, 103]	[75, 148]
$x_{123}^{\pm}$	[210, 290]	[50, 80]	[200, 290]	[200, 259]	[197, 260]
$x_{131}^{\pm}$	[0, 30]	0	[1, 37]	[90, 190]	[225, 312]
$x_{132}^{\pm}$	[260, 330]	0	[247, 332]	[189, 270]	[120, 200]
$x_{133}^{\pm}$	[170, 200]	0	[154, 209]	[139, 210]	[143, 192]
$x_{211}^{\pm}$	50	[10, 50]	[35, 58]	[120, 167]	[220, 307]
$x_{212}^{\pm}$	[310, 390]	[0, 40]	[295, 390]	[299, 385]	[295, 390]
$x_{213}^{\pm}$	[360, 410]	0	[329, 426]	[202, 323]	[120, 161]
$x_{221}^{\pm}$	[160, 240]	[160, 210]	[147, 240]	[55, 145]	[1, 30]
$x_{222}^{\pm}$	[185, 200]	[0, 40]	[165, 222]	[142, 200]	[80, 154]
$x_{223}^{\pm}$	0	[160, 210]	[1, 25]	[1, 40]	[1, 43]
$x_{231}^{\pm}$	[260, 310]	[260, 340]	[230, 320]	[122, 164]	[12, 40]
$x_{232}^{\pm}$	[0, 10]	[260, 340]	[1, 28]	[30, 100]	[108, 167]
$x_{233}^{\pm}$	[140, 190]	[310, 390]	[125, 200]	[125, 194]	[120, 214]
$f^{\pm} (\$10^6)$	[220.2, 507.4]	[239.5, 514.1]	[209.8, 522.3]	[200.5, 519.6]	[197.6, 516.8]

**Table 5.** Solutions obtained by ILP model ( $m=-1$ ), IQP model ( $-0.2 < m < -0.1$ ) and  $m=-0.1, 0.3$  and  $0.5$ .

In the 4th scenario, the weight of the transportation cost in the system operation cost varies according to different  $C_t$  values. The effect becomes significant when waste flow becomes lower and the hauling distances are substantial. This effect is a nonlinear function of the waste flow  $x_{ijk}$  in which the reference waste flow  $x_{re}$  and the ES exponent  $m$  are the parameters. This problem is a complicated nonlinear programming problem, and the GAINLP has been shown to be adequate for solving this kind of problems. On the other hand, the traditional IBA

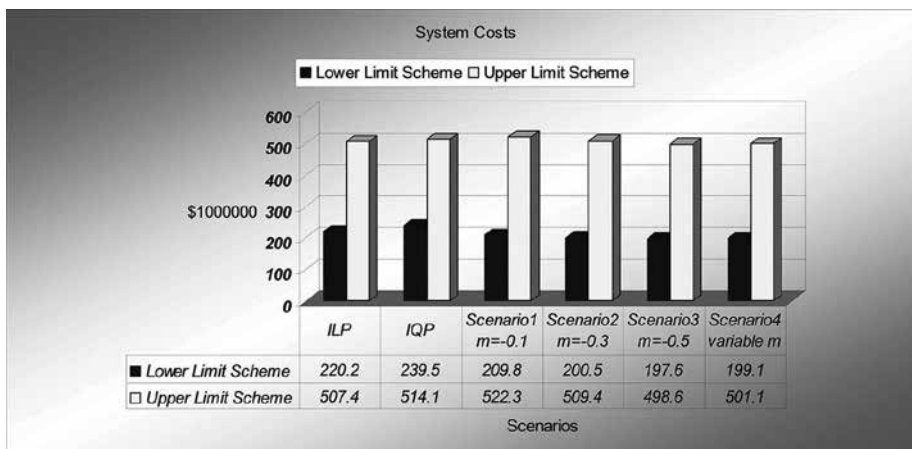
methods will not be able to handle situations like the 4th scenario without additional assumptions and simplification.

Decision variable	Solution (t/d)	$m$	Decision variable	Solution (t/d)	$m$
$x_{111}^{\pm}$	[100, 192]	-0.15	$x_{211}^{\pm}$	[125, 189]	-0.1
$x_{112}^{\pm}$	[0, 40]	-0.2	$x_{212}^{\pm}$	[300, 390]	-0.15
$x_{113}^{\pm}$	[112, 178]	-0.35	$x_{213}^{\pm}$	[205, 312]	-0.15
$x_{121}^{\pm}$	[65, 139]	-0.2	$x_{221}^{\pm}$	[42, 142]	-0.45
$x_{122}^{\pm}$	[40, 78]	-0.3	$x_{222}^{\pm}$	[129, 192]	-0.3
$x_{123}^{\pm}$	[199, 279]	-0.3	$x_{223}^{\pm}$	[0, 40]	-0.1
$x_{131}^{\pm}$	[90, 189]	-0.25	$x_{231}^{\pm}$	[100, 198]	-0.3
$x_{132}^{\pm}$	[135, 280]	-0.25	$x_{232}^{\pm}$	[44, 113]	-0.45
$x_{133}^{\pm}$	[127, 218]	-0.3	$x_{233}^{\pm}$	[99, 219]	-0.4

$$f^{\pm} = [194.7, 500.1] (\$10^6)$$

Note: for transportation from WTE facility to landfill,  $m = -0.5$ .

**Table 6.** Solutions when  $m$  is different for each municipality and each period.



**Figure 6.** System cost comparisons.

The results also show that when the value of the ES exponent  $m$  becomes smaller, from -0.1, -0.3 to -0.5, for both  $f$  positive scheme and  $f$  negative scheme, the value of the minimized

objective function also becomes smaller. At the same time, the range of the intervals of the minimized objective function also decreases. This reflects how the ES exponent affects the overall cost for the entire period. A comparison of the results for the four scenarios is given in Figure 6.

## 5. Conclusions

In this chapter, the GA-based methods have been proposed and applied for identifying an all-purpose optimization solution for the ILP, IQP and INLP problems. These methods are called GAILP, GAIQP and GAINLP. Compared to these GA-based methods, the traditional problem-solving method has limitations due to the complexity involved in selecting the upper or lower bounds of variables and parameters when the subobjective functions are being constructed. The complexity arises due to the extensive computation and necessary associated assumptions and simplification. The solution procedures of the proposed GA-based optimization methods do not involve any such assumption or simplification, and the quality of the result is guaranteed. The GAINLP was applied to a solid waste management optimization problem, and the result analysis illustrates the practicality and flexibility of the proposed GAINLP method for solving more complex INLP problems.

GAILP, GAIQP and GAINLP have been implemented in MATLAB, and can be easily extended to include other nonlinear operation programming software packages so as to enhance the flexibility and efficiency of the problem-solving process. The GA-based heuristic optimization approach is flexible and it can be extended to find solutions for various types of operation programming scenarios that involve nonlinear optimization and inexact information. It can also be used as an all-purpose algorithm for economic optimizations.

## Acknowledgements

The authors gratefully acknowledge the support of the Canada Research Chair program and the Natural Sciences and Engineering Research Council of Canada.

## Author details

Weihua Jin, Zhiying Hu and Christine W. Chan\*

\*Address all correspondence to: christine.chan@uregina.ca

Energy Informatics Laboratory, Faculty of Engineering and Applied Science, University of Regina, Regina Saskatchewan, Canada

## References

- [1] Anderson LE, Nigam A. A mathematical model for the optimization of a waste management system. University of California at Berkeley, Sanitary Engineering Research Laboratory, SERL Report. 1968 (68-1).
- [2] Christensen HL, Haddix GF. A model for sanitary landfill management and design. *Computers & Operations Research*. 1974;1(2):275-81.
- [3] Fuertes LA, Hudson JF, Marks DH. Solid waste management: Equity trade-off models. *Journal of the Urban Planning and Development Division*. 1974;100(2):155-71.
- [4] Jenkins L. Parametric mixed integer programming: An application to solid waste management. *Management Science*. 1982;28(11):1270-84.
- [5] Jacobs TL, Everett JW. Optimal scheduling of consecutive landfill operations with recycling. *Journal of Environmental Engineering*. 1992;118(3):420-9.
- [6] Badran MF, El-Haggag SM. Optimization of municipal solid waste management in Port Said-Egypt. *Waste Management*. 2006;26(5):534-45.
- [7] Sushi A, Vrat P. Waste management policy analysis and growth monitoring: An integrated approach to perspective planning. *International Journal of Systems Science*. 1989;20(6):907-26.
- [8] Chang NB, Wen CG, Chen YL, Yong YC. A grey fuzzy multiobjective programming approach for the optimal planning of a reservoir watershed. Part A: Theoretical development. *Water Research*. 1996;30(10):2329-34.
- [9] Chang NB, Shoemaker CA, Schuler RE. Solid waste management system analysis with air pollution and leachate impact limitations. *Waste Management & Research*. 1996;14(5):463-81.
- [10] Huang GH, Baetz BW, Patry GG. Grey integer programming: An application to waste management planning under uncertainty. *European Journal of Operational Research*. 1995;83(3):594-620.
- [11] Huang GH, Baetz BW, Patry GG. Grey quadratic programming and its application to municipal solid waste management planning under uncertainty. *Engineering Optimization*. 1995;23(3):201-23.
- [12] Li Y, Huang G. Dynamic analysis for solid waste management systems: An inexact multistage integer programming approach. *Journal of the Air & Waste Management Association*. 2009;59(3):279-92.
- [13] Tan Q, Huang GH, Cai Y. A superiority-inferiority-based inexact fuzzy stochastic programming approach for solid waste management under uncertainty. *Environmental Modeling & Assessment*. 2010;15(5):381-96.

- [14] Ekmekçioğlu M, Kaya T, Kahraman C. Fuzzy multicriteria disposal method and site selection for municipal solid waste. *Waste Management*. 2010;30(8):1729–36.
- [15] Pires A, Martinho G, Chang NB. Solid waste management in European countries: A review of systems analysis techniques. *Journal of Environmental Management*. 2011;92(4):1033–50.
- [16] Beliën J, De Boeck L, Van Ackere J. Municipal solid waste collection and management problems: A literature review. *Transportation Science*. 2012;48(1):78–102.
- [17] Or I, Curi K. Improving the efficiency of the solid waste collection system in Izmir, Turkey, through mathematical programming. *Waste Management & Research*. 1993;11(4):297–311.
- [18] Sun W, Huang GH, Lv Y, Li G. Inexact joint-probabilistic chance-constrained programming with left-hand-side randomness: An application to solid waste management. *European Journal of Operational Research*. 2013;228(1):217–25.
- [19] Huang GH, Baetz BW, Patry GG. A grey fuzzy linear programming approach for municipal solid waste management planning under uncertainty. *Civil Engineering Systems*. 1993;10(2):123–46.
- [20] Chang NB, Schuler RE, Shoemaker CA. Environmental and economic optimization of an integrated solid waste management system. *J. Resour. Manage. Technol*. 1993;21(2): 87–98.
- [21] Huang G, Baetz BW, Patry GG. A grey linear programming approach for municipal solid waste management planning under uncertainty. *Civil Engineering Systems*. 1992;9(4):319–35.
- [22] Huang GH, Baetz BW, Patry GG. Grey dynamic programming for waste-management planning under uncertainty. *Journal of Urban Planning and Development*. 1994; 120(3): 132–56.
- [23] Callan SJ, Thomas JM. Economies of scale and scope: A cost analysis of municipal solid waste services. *Land Economics*. 2001;77(4):548–60.

---

## Applications in Various Areas

---



---

# Optimization Algorithms for Chemoinformatics and Material-Informatics

---

Abraham Yosipof and Hanoch Senderowitz

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62483>

---

## Abstract

Modeling complex phenomena in chemoinformatics and material-informatics can often be formulated as single-objective or multi-objective optimization problems (SOOPs or MOOPs). For example, the design of new drugs or new materials is inherently a MOOP since drugs/materials require the simultaneous optimization of multiple parameters.

In this chapter, we present several algorithms based on global stochastic optimization. These algorithms are applicable to multiple tasks in chemoinformatics and material-informatics including the following: (1) representativeness analysis, namely the selection of a representative subset from within a parent data set. (2) Derivation of quantitative structure–activity relationship models. Such models are used in multiple areas to predict activities from structures and to provide insight into factors (e.g., descriptors) governing activities. (3) Outlier removal to clean a parent data set from objects (e.g., compounds) that may demonstrate abnormal behavior.

The performances of the new algorithms were evaluated using different data sets and multiple measures and were found to outperform previously reported methods.

Due to the modular nature of the algorithms, they could be combined into machine-learning workflows. In the final section, we provide an example of one such workflow and apply it to the development of predictive models in pharmaceutical and material sciences.

**Keywords:** chemoinformatics, material-informatics, simulated annealing, QSAR, outlier removal, machine learning, representativeness, *k*NN

## 1. Introduction

Modeling complex phenomena in chemoinformatics and material-informatics can often be formulated as multi-variables/single-objective or multi-variables/multi-objectives optimization problems. Common examples of the former include sampling of complex energy landscapes using conformational search methods [1], docking or molecular simulation techniques [2, 3], derivation of statistical models, namely quantitative structure–activity relationship (QSAR) models [4, 5], and diversity or representativeness analysis [6, 7]. The design of new compounds with pharmaceutical relevance on the other hand is inherently a multi-objective optimization problem (MOOP) since drugs require the simultaneous optimization of many parameters and consequently constitute a compromise between often conflicting requirements. In a similar manner, the design of new materials could also be regarded as a MOOP. For example, the design of new photovoltaic cells requires the simultaneous optimization of the current and the voltage.

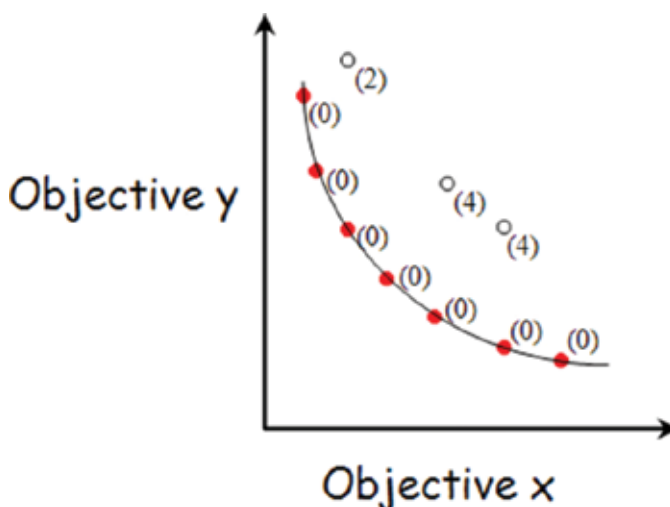
This chapter focuses on optimization algorithms from three different areas: (1) Representativeness analysis, that is, the selection of a representative subset from within a parent data set. Representativeness analysis has multiple applications in chemoinformatics and material-informatics, for example, for rationally selecting subsets of compounds for experimental analysis and for rationally partitioning a parent data set into a modeling set and a test set. (2) Derivation of predictive, nonlinear machine-learning models correlating activities with descriptors while inherently incorporating feature selection to identify the most relevant descriptors. Such the so-called QSAR models find many usages in chemistry, biology, environmental sciences, and material sciences to predict the activities of new compounds/materials and to provide insight into the factors governing these activities [8]. (3) Outlier removal, to clean a parent data set from objects (e.g., compounds) that may demonstrate abnormal behavior. Outlier removal is a mandatory step prior to the derivation of any statistical model. In all cases, the corresponding problems (i.e., how to select a representative subset from within a parent data set, how to remove outliers from within a parent data set, and how to build predictive QSAR models) were formulated as either single-objective optimization problems (SOOPs) or MOOPs. These problems were then solved using Monte Carlo (MC)/Simulated Annealing (SA) or Genetic Algorithm (GA) as the optimization engines.

Finally, while developed independently and with multiple potential applications in mind, all algorithms could be used as components of machine-learning workflows for data mining and the derivation of QSAR models (see Section 5).

This chapter is organized as follows: We begin with a short introduction to SOOPs and MOOPs (Section 2), followed by a description of two of the most common optimization engines, MC/SA (Section 3.1) and GA (Section 3.2). Section 4 provides a description of the different optimization-based algorithms, and Section 5 lists a few representative examples. Finally, Section 6 concludes the chapter.

## 2. Single-objective and multi-objectives optimization

As noted above, multiple problems in chemoinformatics and material-informatics could be formulated as optimization problems. Such a formulation requires the definition of a target (or objective) function(s) ( $f$ ) and a set of variables ( $X_1, X_2, X_3, \dots, X_n$ ), which are related to the scientific problem of interest, and which together define a complex, multi-dimensional surface with an a priori unknown distribution of optima. The task then is to locate the global optimum or preferably, since phenomena in these fields are rarely governed by a single solution, a set of optima.



**Figure 1.** Potential solutions of a two objectives problem represented by the Pareto front. Dominated solutions are shown as empty circle, and the number of solutions which dominate them is written in parentheses.

Optimization problems could be broadly divided into two categories, namely SOOP and MOOP depending on the number of target functions which should be simultaneously optimized. SOOPs search for the best optimum on a surface defined by a single target function and its variables. While this might be a difficult task in particular for complex, multi-dimensional surfaces, a solution exists and a thorough enough search of the space, at least in theory, is bound to locate it. This, however, is not the case for MOOPs, which extend optimization theory by permitting several design objectives to be optimized simultaneously. The principle of MOOP was first formalized by Pareto [9]. In MOOP, a single solution that outperforms all other solutions in all objectives does not necessarily exist. Instead, several equally good (termed non-dominated) solutions exist representing various compromises among the objectives. A solution is said to be non-dominated if it is better than all other solutions in at least one objective. The set of non-dominated solutions represents the Pareto front. This is illustrated in **Figure 1**, where each circle represents a solution to the problem. The curved line represents the Pareto front. Each solution is designated a Pareto rank that is based on the number of solutions which dominate it. The solid circles are non-dominated solutions, which

have Pareto rank of zero and fall on the Pareto front. Dominated solutions are shown as empty circle, and the number of solutions which dominate them is written in parentheses.

Due to the complex nature of many of the chemoinformatics and material-informatics-related target functions (e.g., nonlinearity, non-continuity, non-derivability), non-derivatives-based, stochastic optimization algorithms should be used as the optimization engine. Several global stochastic algorithms including GAs [10], genetic programming [11], particle swarm optimization [12], MC [13]/SA [14], and the iterative stochastic elimination method [15] are reported in the literature. The algorithms presented in this chapter utilize MC/SA- and GA-based optimizers to solve SOOPs and MOOPs.

### 3. Optimization engines

#### 3.1. Monte Carlo/simulated annealing

MC methods and in particular the Metropolis variant [13] use random moves in order to optimize a multi-dimensional space defined by a dependent target function and a set of independent variables.

Metropolis MC starts from an initial random position. At each iteration, a new position ( $position_{i+1}$ ) is randomly generated by a random displacement from the current position ( $position_i$ ). The “energy” (i.e., the value of the target function) of the resulting new position is computed, and  $\Delta E$ , the energetic difference between the current and the new position, is determined (Equation 1).

$$\Delta E = E(position_{i+1}) - E(position_i) \quad (1)$$

The probability that this new position is accepted is based on the Metropolis test as defined in Equations 2 and 3.

$$\text{if } (\Delta E < 0) \text{ then accept step} \quad (2)$$

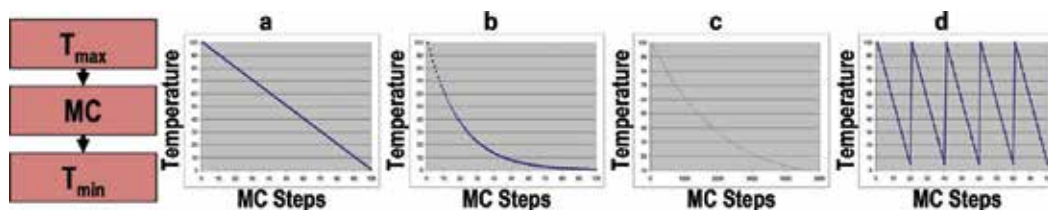
*else* Equation 3

$$\text{if } \left( \text{random}[0,1] \leq e^{-\frac{\Delta E}{kT}} \right) \text{ then accept step} \quad (3)$$

Thus, if the new position has a lower value of the target function (commonly referred to as lower energy), the transition is accepted. Otherwise, a uniformly distributed random number between 0 and 1 is drawn and the new position will only be accepted if the Boltzmann

probability is higher or equal to the random number as defined in Equation 3, where  $kT$  is the effective temperature.

MC simulations are often coupled to a SA [14] procedure and usually called SA optimization. SA gradually decreases the temperature according to a predefined cooling schedule (**Figure 2**). This procedure controls the probability of acceptance or rejection of high energy moves. The SA procedure increases the probability of locating the global minimum.



**Figure 2.** Examples of cooling schedules for SA: (a) smooth linear cooling; (b) smooth exponential cooling (c) stepwise exponential cooling; (d) saw-tooth linear cooling composed of repeating cooling cycles in order to avoid trapping in local minima.

### 3.2. Genetic algorithms

GAs [10] are global optimizers designed from evolutionary principles. Such algorithms evolve a set of chromosomes, each corresponding to a unique solution to the optimization problem, using a set of genetic operators such as selection of the fittest, mutation, and crossover. The selection of individuals to be subjected to the genetic operators is governed by their fitness values (calculated by a fitness function related to the specific scientific problem) so that chromosomes which represent a better solution to the optimization problem are given more chances to “reproduce” than those chromosomes which are poorer solutions. This process iterates multiple times (generations) until no improvement in the fitness function is observed or until the number of predefined generation has been exhausted. By considering, at each generation, multiple rather than single chromosomes, GAs provide multiple solutions to the optimization problem.

## 4. Optimization-based algorithms

### 4.1. Representativeness and diversity analysis

Advances in combinatorial chemistry and high-throughput screening techniques have greatly expanded the number of drug-like compounds that could be synthesized and tested. However, even then, only a small fraction of the accessible chemistry space could be synthesized and screened [16]. This in turn has led to the development of rational approaches for the design of

screening libraries and in particular libraries composed of diverse compounds (i.e., compounds largely differing from one another). Hassan et al. [6] formulated the selection of a diverse subset from within a parent data set as an optimization problem using several diversity functions. In particular, the MaxMin function calculates the square of the minimal distance,  $d_{ij}^2$ , over all  $(i,j)$  pairs comprising the selected subset according to Equation 4.

$$MaxMin = \max \left( \min_{i \neq j} (d_{ij}^2) \right) \quad (4)$$

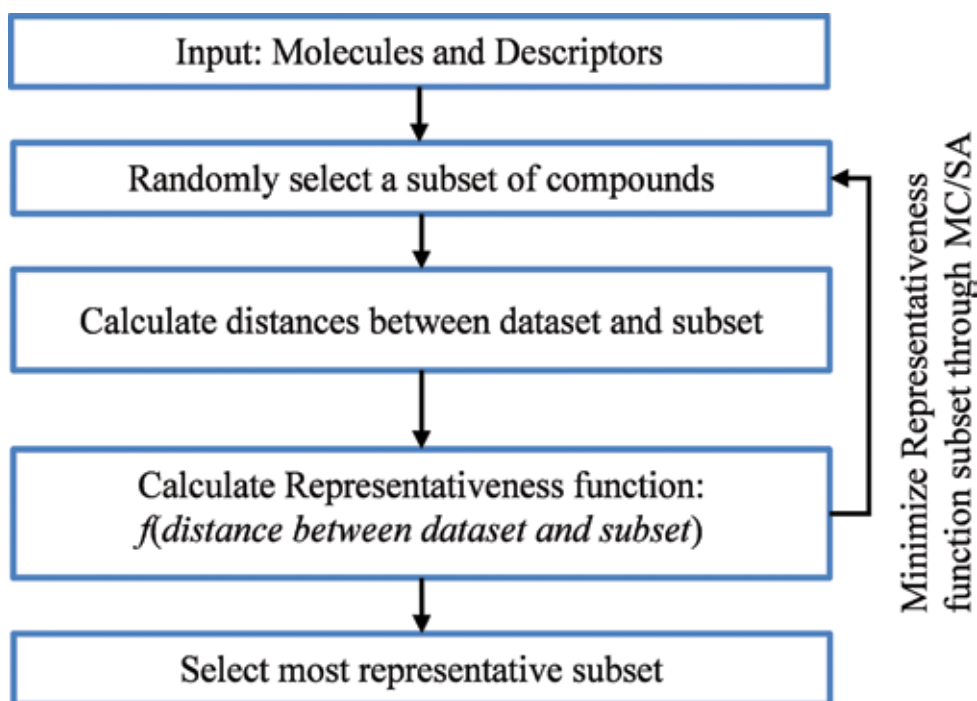
where  $d_{ij}$  is the distance between compounds  $i$  and  $j$ , and the summation runs over all the descriptors (features). The MaxMin function is optimized (maximized) by means of a SA algorithm to produce the subset with the largest value (i.e., the most diverse subset).

Treating the selection of a diverse subset as an optimization problem has the advantage that this objective could be combined with other objectives into a MOOP. For example, it is possible to select a subset that the best balances its internal diversity, pharmacological profile, and price.

However, diverse subsets are often biased toward the inclusion of outliers and as a result do not well represent the parent data set. In drug discovery, focusing on outliers may translate into selecting and testing “extreme” compounds, for example, compounds with too many functional groups or compounds with too high molecular weights. Such compounds are likely to be difficult to optimize into drug candidates [17]. Unless such “extreme” compounds could be easily detected and removed (e.g., if their “extremeness” results from a single property), they are likely to remain in the data set.

Thus, instead of selecting diverse subsets, it might be advisable to select representative subsets, which better mirror the distribution of the parent data set. If properly selected, such subsets will include compounds which are different from one another yet are not “bizarre.” However, despite their potential usefulness, representative subsets have gained much less attention than diverse subsets.

Representative subsets could be selected using clustering algorithms such as hierarchical clustering and k-means clustering. Both partition an input data set into a predefined number of clusters. Hierarchical clustering produces a dendrogram of clusters in which the root node contains all the compounds and the leaf nodes contain each, a single compound. Divisive hierarchical clustering starts at the root node and iteratively divides clusters until the leaf nodes are reached. Agglomerate hierarchical clustering starts with the leaf nodes and iteratively combines closest neighbors clusters until the root node is reached. Extracting from the dendrogram a specific number of clusters and selecting a compound from each cluster leads to a representative/diverse subset. k-means clustering [18] operates by first selecting at random a user defined number of seeds ( $k$ ), then by assigning each compound to its closest seed. This leads to the formation of  $k$  initial clusters. Centroids of all clusters are then calculated, and compounds are re-assigned to their closest centroid. This process is repeated until the clusters are stable, that is, no compounds are re-assigned. Again selecting a compound from each cluster provides a representative/diverse subset.



**Figure 3.** Schematic representation of the representativeness optimization algorithm.

In 2014, Yosipof and Senderowitz [19] introduced an optimization algorithm for the direct selection of a representative subset from within a data set. The algorithm optimizes by means of a MC/SA procedure a representativeness function, based on pairwise distances between subset and data set compounds. The algorithm consists of the following steps (**Figure 3**):

1. Characterize each compound in the data set by a set of descriptors.
2. Normalize descriptors by converting them into Z-scores according to Equation 5 where  $\mu$  is the mean, and  $\sigma$  is the standard deviation:

$$Z = \frac{x_i - \mu}{\sigma} \quad (5)$$

The resulting scores are then converted to [0,1] range by calculating the cumulative probability assuming a normal distribution ( $\mu = 0$ ;  $\sigma^2 = 1$ ).

3. Select a random subset  $s$  of size  $k$  from within the  $l$  compounds comprising the data set.
4. Calculate the Euclidean distance between compound  $i$  from the data set and all  $k$  compounds comprising subset  $s$ .
5. Take the minimum Euclidean distance from step (4) as the score for compound  $i$ :

$$score_i = \min(dist_{i,\{s\}})$$

6. Repeat steps (4) and (5) for all  $(l-k)$  compounds remaining in the data set.
7. Calculate the average score over all  $(l-k)$  compounds. This score characterizes subset  $s$ :

$$score_s = \frac{1}{(l-k)} \sum_{i=1}^{l-k} score_i$$

Minimize  $score_s$  through a MC/SA procedure [14]. At each step replace, at random, a single compound from  $s$  with a candidate compound from the unselected portion of the data set, calculate a new score,  $score'_s$ , and accept it according to the MC/SA algorithm.

Similar to diversity analysis, a major advantage in treating the selection of a representative subset as an optimization problem is the ability to combine it with additional objectives into a MOOP [20, 21]. This was demonstrated by the Pareto-based optimization of the representativeness and MaxMin functions. The Pareto algorithm evaluates MaxMin (Equation 4) and the representativeness function for a selected subset (termed a solution to the MOOP) and assigns to it a Pareto rank based on the number of solutions dominating it. In this case, solution  $i$  dominates solution  $j$  if  $\text{MaxMin}(i) < \text{MaxMin}(j)$  and  $\text{Score}(i) < \text{Score}(j)$  (where  $\text{Score}$  is calculated according to step 7 given above). Under this dominance criterion, the value of MaxMin is minimized rather than maximized in contrast with the original implementation of this function for diversity selection. MaxMin minimization biases the selected subset toward the more populated regions of the database allowing the two functions (MaxMin and representativeness) to work in concert. The Pareto rank is then minimized using Metropolis Monte Carlo, and the solutions with rank = 0 (i.e., non-dominant solutions) are kept to construct the Pareto front. Finally, a solution on the Pareto front is randomly selected. Alternatively, all the solutions could be presented to the user for manual inspection, evaluation, and selection.

Representative subsets could be used under two general scenarios: (1) Results obtained for a representative subset are used to infer on the properties of the parent data set. For example, the biological evaluation of a representative subset could provide information on the activities of the entire parent data set. Thus, testing only a representative subset will provide a similar amount of information as would have been gained by testing the entire data set. (2) A representative subset is selected from within a parent data set, set aside, and used to validate models generated by machine-learning algorithms. In the area of chemoinformatics, such models are known as quantitative structure–activity relationship (QSAR) models (see Section 4.2) [22].

To evaluate the representativeness algorithms under the first scenario, Yosipof and Senderowitz [19] selected a subset of 200 compounds from the Comprehensive Medicinal Chemistry (CMC) database using five representative/diversity algorithms, namely representativeness optimization, Pareto-based optimization, hierarchical clustering, k-means clustering, and MaxMin optimization. The CMC database contains 4,855 pharmaceutical compounds classified into 105 different biological indications. The degree with which each subset is able to represent the parent database (i.e., include a similar distribution of indications) was estimated

using the  $\chi^2$  goodness-of-fit test. In this test, the null hypothesis ( $H_0$ ) states that the distribution of biological indications within the subset and database are similar. In contrast, the  $H_1$  hypothesis states that these distributions are significantly different. The objective is therefore to stay on the null hypothesis. The  $\chi^2$  statistics is defined as follows:

$$\chi^2 = \sum_{i=1}^{n=105} \frac{(O_i - E_i)^2}{E_i} \quad (6)$$

where  $O_i$  and  $E_i$  represent, respectively, the observed and expected frequencies for a biological indication  $i$  in the 200 compounds subset.  $E_i$  is derived from the frequency of indication  $i$  in the parent database. The results of this test demonstrated that the distribution of indications within the subsets selected by the representativeness optimization and the Pareto-based optimization are statistically indistinguishable ( $p$ -value  $> 0.05$ ) from that in the parent database. In contrast, subset selected by k-means clustering, hierarchical clustering, and MaxMin display distributions which are markedly different from the database ( $p$ -value  $< 0.05$ ).

To evaluate the representativeness algorithms under the second scenario (a representative subset is selected, set aside, and used to validate models built on the rest of the data set), it was incorporated into a workflow developed for the derivation of predictive QSAR models using machine-learning algorithms. Following the work of Tropsha [23] and others [22], it is today recognized that the predictive power of such models could only be evaluated from their performances on external test sets. In particular, test sets that uniformly span the chemistry space of the parent data set provide reliable performance estimates for QSAR models operating in this space. It therefore follows that such performance estimates correlate with how well a set of compounds represents the parent data set from which it was selected. Yosipof and Senderowitz [19] used the new representativeness algorithm to rationally select test sets of varying sizes from two data sets of pharmaceutical relevance (logBBB and *Plasmodium falciparum* inhibition) and estimated the performances of models derived with five classification techniques (decision trees, random forests, ANN, SVM, kNN) on these test sets. Similar test sets were also selected from the Pareto front generated by the simultaneous optimization of representativeness and MaxMin as well as by the k-means clustering, hierarchical clustering, and MaxMin for comparison. Model performances were estimated using the corrected classification rate (CCR; Equation 7)

$$CCR = \frac{1}{2} \left( \frac{T_N}{N_N} + \frac{T_P}{N_P} \right) \quad (7)$$

where  $T_N$  and  $T_P$  represent the number of true negative and true positive predictions, respectively, and  $N_N$  and  $N_P$  represent the total number of the two activity classes.

The results (**Table 1**) indicate that the best performances were obtained with the Pareto method followed by the representativeness function and k-means clustering. The other two methods,

namely hierarchical clustering and the MaxMin function led to poorer performances. Thus, representativeness-based methods indeed produce subsets which are more representative of the parent data sets.

Method	CCR
Hierarchical clustering	0.78
k-meansclustering	0.80
MaxMin	0.74
Representativeness	0.80
Pareto-based	0.83

**Table 1.** Average performances of QSAR models on test sets selected with the hierarchical clustering, k-means clustering, MaxMin, representativeness, and Pareto-based methods. Averaging performed over the two data sets, the five model building algorithms, and the different sizes of the selected test sets.

## 4.2. Derivation of predictive QSAR models

QSAR (or QSPR, quantitative structure–property relationship) is a general name for a host of methods that attempt to correlate a specific activity for a set of compounds with their structure-derived descriptors (i.e., features) by means of a mathematical model.

QSAR models take the form  $A_i = f(D_1, D_2, \dots, D_n)$  where  $A_i$  is the dependent variable representing the activity (or any other property of interest) for a set of objects (e.g., compounds or materials), and  $D_1, D_2, \dots, D_n$  are calculated (or experimentally measured) independent variables (i.e., descriptors).  $f$  is an empirical mathematical transformation that should be applied to the descriptors in order to calculate the property values for the objects.

QSAR models are typically built according to a basic workflow which involves the following steps: (1) data collection; (2) preprocessing (data set preparation and curation, descriptors calculation, descriptors filtering); (3) model generation; (4) model validation.

Data collection involves the assembly of a data set of compounds/materials with known activities. Once collected, the data should be carefully curated, errors should be corrected (or if not possible, problematic compounds should be removed), and a set of descriptors should be obtained (calculated or measured). Finally, constant or nearly constant descriptors should be removed and usually, correlated descriptors are removed as well.

QSAR models are built using multiple machine-learning approaches. The modeling process begins with a modeling set and proceeds by performing regression-based or classification-based analysis to construct a model of the activity as a function of the descriptors. Machine-learning techniques are mostly used in this area, because they can deal with very complex relationships between descriptors and activities [8].

There are two types of regression-based methods, namely linear and nonlinear. An example of a linear method is provided by multiple linear regression which is extensively applied in Hansch analysis [24]. Examples of nonlinear methods are the  $k$ -nearest neighbors ( $k$ NN) [5],

and the random forest (RF) [25] methods. Finally, the resulting model should be validated on an external test set. An external set can be obtained by splitting the input and curated data set prior to the model development process or by obtaining additional data [26, 27].

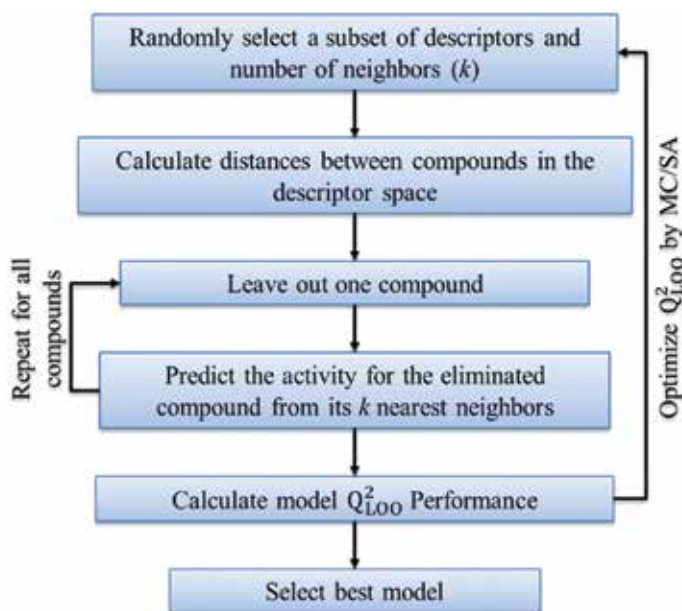
Inherent to the most QSAR methods is feature (i.e., descriptor) selection. The purpose of this process is to select from within a large number of calculated/experimental descriptors those that best correlate with the activity. This stage is required since typically, the number of calculate-able descriptors by far exceeds that of compounds with measured activity data. Having too many descriptors calculated for too few compounds may lead to over fitting and chance correlation [23, 28]. In some cases, features selected for QSAR modeling are more important than the predictive model itself. These selected descriptors contain useful information that can reveal significant information and can help in understanding and rationalization the data and the results. The selection of descriptors subset could be treated as an optimization problem whereby a function related to model performances is optimized in the space of the descriptors.

One algorithm that couples feature selection procedure with a machine-learning algorithm at the model derivation stage is the  $k$ NN algorithm. This algorithm assumes that the activity of a compound could be predicted from the average activities of the  $k$  compounds most similar to it ( $k$ NN). This idea follows directly from the similar property principle [29] which states that similar compounds have similar properties. The similar property principle is well-validated in pharmaceutical sciences and was recently extended to photovoltaic cells [30]. Since chemical similarity between two compounds depends on the molecular descriptors used to characterize them, the algorithm searches the space of available descriptors subsets for that subset in terms of which the similar properties principle is best satisfied. This is done by optimizing the leave-one-out (LOO) cross-validated value ( $Q^2_{LOO}$ , Equation 8) in the space of the descriptors (the space of the descriptors is a multidimensional space where each dimension corresponds to one descriptor. In this space compounds are represented by points and the distance between any two points represents the degree of similarity between the corresponding compounds).  $Q^2_{LOO}$  is given by

$$Q^2_{LOO} = 1 - \frac{\sum_Y (Y_{exp} - Y_{LOO})^2}{\sum_Y (Y_{exp} - \bar{Y}_{exp})^2} \quad (8)$$

where  $Y_{exp}$  is the experimental value  $Y_{LOO}$  is the predicted value and  $\bar{Y}_{exp}$  is the mean of the experimental results.

The  $k$ NN algorithm was first implemented by Zheng and Tropsha [5] using SA as the optimization engine. In this original implementation, only the descriptors space was searched with the SA procedure, while the number of nearest neighbors ( $k$ ) was evaluated exhaustively (between 1 and 5). In 2015, Yosipof and Senderowitz [31] implemented the  $k$ NN algorithm optimizing  $Q^2_{LOO}$  in the space of the descriptors and the number of nearest neighbors. A schematic representation of the  $k$ NN optimization algorithm is provided in **Figure 4**.



**Figure 4.** Schematic representation of the  $k$ NN optimization algorithm.

The  $k$ NN algorithm has been extensively used to build predictive QSAR models in many fields including computer-aided drug design and environmental sciences [8]. Recently, the method has been applied in the field of material-informatics and used to predict the photovoltaic properties of solar cell libraries (see Section 5) [30].

### 4.3. Outlier removal

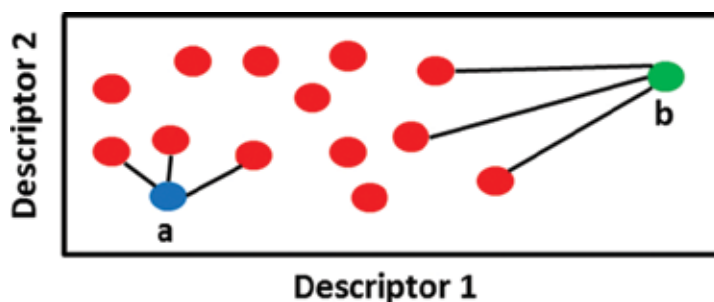
Data sets in general and data sets consisting of molecular compounds in particular often contain objects (e.g., compounds) that are different in some respect from the rest of the data set. Such compounds are called outliers. The presence of outliers in a data set can affect machine-learning-related activities including model derivation, interpretation, and subsequent decision-making. In particular, outliers can compromise the ability of machine-learning algorithms to develop predictive models since many algorithms typically used for this purpose will attempt to fit the outliers at the expense of the bulk. Thus, while outliers may point to an interesting behavior that needs to be investigated separately [32], they should be removed from the data set prior to the model construction.

Several methods for outliers removal have been reported in the literature [33]. Statistical estimators could be used to identify outliers if their values follow a well-defined distribution [34, 35]. These methods are called parametric methods. For example, for a Gaussian distribution, outliers could be defined as compounds with descriptors values deviating from the mean by a certain number of standard deviations. For non-well-defined distributions, nonparametric methods should be used. For example, distance-based methods identify outliers by measuring

Euclidean distances between objects in a predefined descriptors space. In the basic distance-based method, outliers are defined as compounds having at least  $p$  percent (define by the user) of their distances to the other compounds larger than a user defined threshold distance [36]. An improvement to this method was proposed by Ramaswamy et al. [37] and termed the  $K$ -based method. According to this method, compounds are ranked according to their Euclidean distances to their  $k^{\text{th}}$ -nearest neighbors and the  $n$  compounds with the highest rank (largest distances) are considered as outliers. Another method is based on compounds clustering and on subsequent removal of compounds (i.e., outliers) populating small clusters (e.g., singletons) [38]. Another nonparametric method utilizes a variant of the support vector machine (SVM) algorithm, namely one-class SVM, which isolates the outlier's class from the rest of the compounds [39].

The above-described techniques remove outliers in a single step from one predefined descriptors space and are therefore termed one-pass methods. These methods have several disadvantages. First, outliers are descriptor space-dependent with different spaces giving rise to different outliers. Second, if several outliers are present in a data set, they may mask each other so that some will not be recognized [40]. Finally, the removal of one outlier based on a specific descriptor space may affect the distribution of the remaining compounds either in the same space or in different spaces, leading to the potential appearance of new outliers. These challenges could be met by removing outliers in an iterative manner.

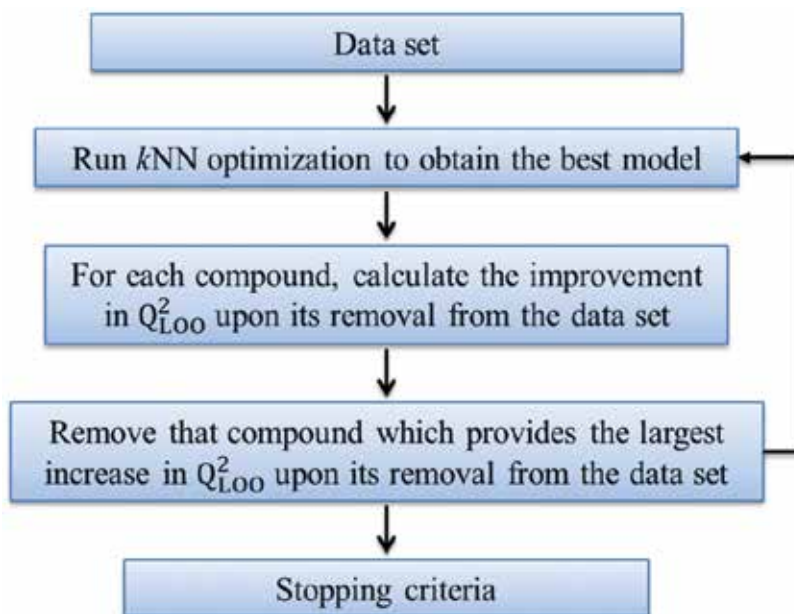
Recently, Yosipof and Senderowitz [31] presented a new method for the iterative identification and subsequent removal of outliers identified in potentially different descriptors spaces based on the  $k$  nearest neighbors optimization algorithm ( $k$ NN-OR algorithm). According to this approach, an outlier is defined as a compound whose distance to its  $k$ NNs is too large for its activity to be reliably predicted (see **Figure 5**). At each iteration, the algorithm builds a  $k$ NN model, evaluates it according to the LOO cross-validation metric ( $Q_{LOO}^2$ ; see Equation 8) and removes from the data set the compound whose elimination results in the largest increase in  $Q_{LOO}^2$ . This procedure is repeated until  $Q_{LOO}^2$  exceeds a pre-defined threshold.



**Figure 5.** Compounds (red spheres) embedded in a two-dimensional descriptors space. Compound *a* (blue sphere) is in close vicinity to at least some of the other compounds leading to short distances to its (three) nearest neighbors and to a likely reliable  $k$ NN-based activity prediction. Compound *b* (green sphere) has large distances to its (three) nearest neighbors and is therefore an outlier. The  $k$ NN-based activity prediction for this compound is likely to be erroneous.

In the example presented in **Figure 5**, the removal of compound *b* from the data set is expected to improve the model and to increase the value of  $kNN-Q_{LOO}^2$ . More generally, for a set of compounds whose activities are predicted via  $kNN$ , outlier(s) removal will lead to an increase in  $Q_{LOO}^2$ . Thus, the following procedure for outlier removal was developed (**Figure 6**):

1. For a set of compounds, run  $kNN$  to obtain the model with the highest  $Q_{LOO}^2$ .
2. For each compound, calculate the improvement in  $Q_{LOO}^2$  upon its removal from the data set.
3. Remove the compound whose elimination from the data set results in the largest increase in  $Q_{LOO}^2$ . When a compound is removed from the data set, it is also removed from the list of nearest neighbors of all other compounds. In such cases, the removed compound will be replaced by the next-in-line nearest neighbor for the purpose of activity prediction.
4. If no compound could be removed from the data set based on the first model (i.e., for all compounds, their removal from the data set does not lead to an improved  $Q_{LOO}^2$ ), repeat steps 2–3 for the second best model (which is built in a different descriptor space).
5. Repeat steps 1–4 above until  $Q_{LOO}^2$  is sufficiently high (stopping criterion).



**Figure 6.** Schematic representation of the  $kNN$  optimization-based outlier removal algorithm.

The above-described  $kNN$ -OR algorithm is "greedy" in nature, removing at each iteration the compound that leads to the largest improvement in  $Q_{LOO}^2$  without considering the possibility

that a sub-optimal improvement at a given iteration may pay off later. Nahum et al. [41] introduced a “look ahead” mechanism into outlier removal by treating it as a multi-objective optimization problem using genetic algorithm (GA-*k*NN). The new method simultaneously minimizes the number of compounds to be removed and maximizes *k*NN-derived  $Q_{LOO}^2$ . The multi-objective optimization is performed using the strength Pareto evolutionary algorithm 2 (SPEA2) [42], which approximates the Pareto front for MOOPs. SPEA2 uses an external set (archive) for storing primarily non-dominated solutions. At each generation, it combines archive solutions with the current population to form the next archive that is then used to produce offspring for the next generation. Each individual *i* in the archive  $A_t$  and the population  $P_t$  is assigned a raw fitness value  $R(i)$ , determined by the number of its dominators in both archive and population.  $R(i)=0$  corresponds to a non-dominated individual, whereas a high  $R(i)$  value means that individual *i* is dominated by many individuals. These raw values are then used to rank the individuals for the purpose of selecting candidates for reproduction. However, the raw fitness value by itself may be insufficient for ranking when the most individuals do not dominate each other. Therefore, additional information, based on the  $k^{th}$  nearest neighbor density of the individuals, is incorporated to remove rank redundancy. The workflow of the SPEA2 algorithm is described below:

---

**Algorithm—SPEA2**

---

Input:	$N$ —Archive size $M$ —Offspring population size $T$ —Maximum number of generations
Output:	$A^*$ —Non-dominated set of solutions to the optimization problem
<ol style="list-style-type: none"> <li>1. Initialization: Generate an initial population <math>P_0</math> and create the empty archive (external set) <math>A_0=\emptyset</math>. Set <math>t=0</math>.</li> <li>2. Fitness assignment: Calculate fitness values of individuals in <math>P_t</math> and <math>A_t</math>.</li> <li>3. Environmental selection: Copy all non-dominated individuals in <math>P_t</math> and <math>A_t</math> to <math>A_{t+1}</math>. If size of <math>A_{t+1}</math> exceeds <math>N</math> then reduce <math>A_{t+1}</math> by means of the truncation operator, otherwise if size of <math>A_{t+1}</math> is less than <math>N</math> then fill <math>A_{t+1}</math> with dominated individuals in <math>P_t</math> and <math>A_t</math>.</li> <li>4. Termination: If <math>t \geq T</math> or another stopping criterion is satisfied then set <math>A^*</math> to the set of decision vectors represented by the non-dominated individuals in <math>A_{t+1}</math>. Stop.</li> <li>5. Mating selection: Perform selection with replacement on <math>A_{t+1}</math> in order to fill the mating pool.</li> <li>6. Variation: Apply recombination and mutation operators to the mating pool and set <math>P_{t+1}</math> to the resulting population. Increment generation counter (<math>t=t+1</math>) and go to Step 2.</li> </ol>	

---

Each solution to the multi-objective optimization problem specifies the set of descriptors, and the number of neighbors used by the *k*NN algorithm as well as the identity of compounds considered as outliers. This information was coded by a three component binary array (i.e., chromosome). The first part of the array encoded the number of neighbors using a binary representation. The second part described the descriptors identity (“1” and “0” representing,

respectively, selected and unselected descriptors for the current solution). The third part listed the compounds considered as outliers using the same representation as for the descriptors.

The resulting chromosomes were subjected to a multi-site crossover operator to produce new chromosomes. These contained a new combination of the number of neighbors, descriptors, and outliers. The new chromosomes were further mutated to increase the diversity of the solutions population and to prevent trapping in local minima. Mutations were performed on the entire chromosome and consequently affected the descriptors, the number of nearest neighbors, and the identity of outliers.

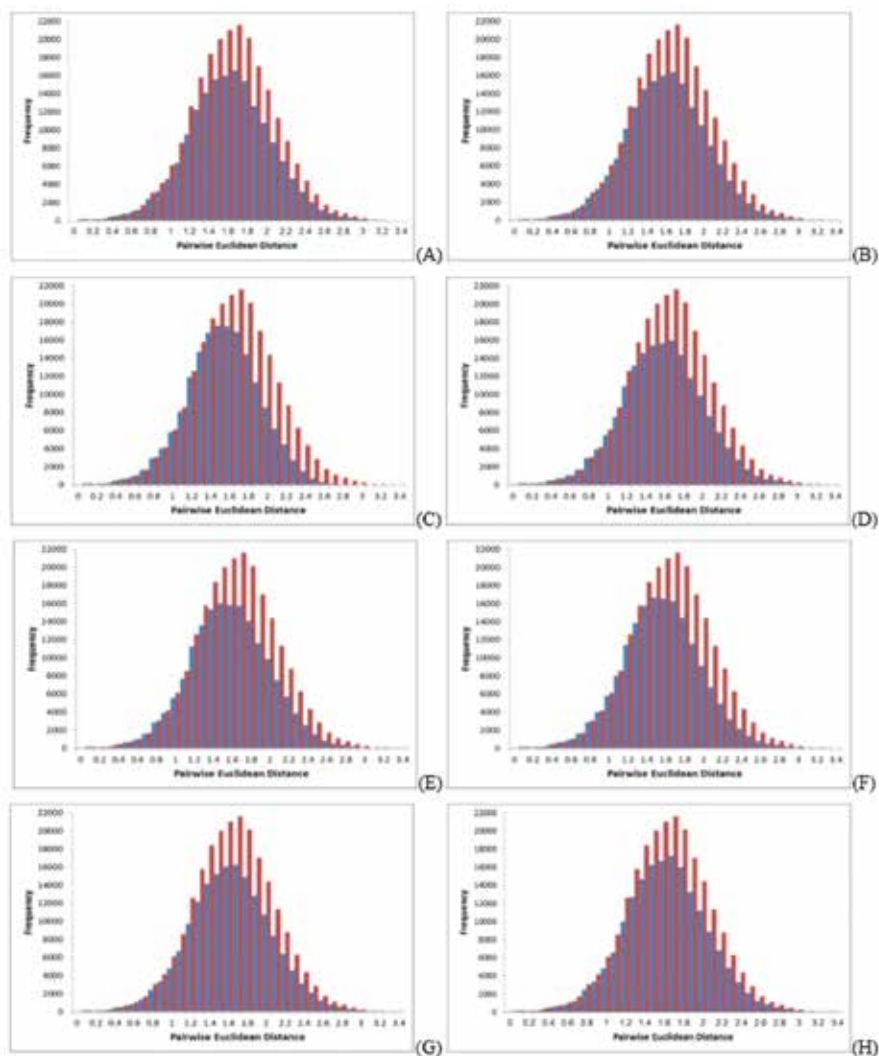
For each new generation, raw fitness values were calculated for each individual based on the information encoded in its chromosome. This calculation was based on  $Q_{LOO}^2$  (which in turn depends on the descriptors selected, the number of neighbors selected, and the identity of outliers removed via the  $k$ NN algorithm) and on the number of outliers removed. The process was repeated until the termination criteria were met.

The performances of the two algorithms ( $k$ NN-OR and GA- $k$ NN) were tested by removing outliers from three data sets of pharmaceutical relevance (logBBB, Factor 7 inhibitors, and dihydrofolate reductase inhibitors [DHFR]) and using the remaining compounds for two purposes: (1) to compare the internal diversities of the filtered data sets with those of the parent sets; (2) to build and validate QSAR models with two machine-learning methods,  $k$ NN, and random forest. The results clearly demonstrated that data sets rationally filtered with the two new outlier removal algorithms were more internally diverse and support QSAR models which provided better prediction statistics than data sets filtered by removing the same number of compounds using five other outlier removal methods (distance based, distance K-based, one-class SVM, statistics and random removal).

As an example, the results for the DHFR data set are presented below. This data set contained 673 compounds with known activity data (dependent variable). Each compound was characterized by 19 descriptors as the independent variables and by its biological activity in the form of an IC50 value as the dependent variable ( $Y$ ). The compounds were subjected to the  $k$ NN-OR and the GA- $k$ NN algorithms using a stopping criterion of  $Q_{LOO}^2 > 0.85$ . For  $k$ NN-OR, this criterion was met after the removal of 87 compounds, leaving a total of 586 compounds for the subsequent diversity analysis and QSAR modeling. For the GA- $k$ NN algorithm, this criterion was met after the removal of only 75 compounds. For comparison, 87 compounds were removed using the GA- $k$ NN algorithm and the other six methods considered in that work. For the GA- $k$ NN model, the removal of 87 compounds led to a model with  $Q_{LOO}^2 > 0.87$ .

The internal diversity of the data set prior to and following compounds removal was evaluated by calculating pairwise Euclidean distances between all compounds in the original descriptors space. The results are presented in **Figure 7** and demonstrate that the removal of either 75 compounds (11.1% of the data set) by the GA- $k$ NN algorithm or 87 compounds (12.9% of the data set) by the GA- $k$ NN and  $k$ NN-OR algorithms did not change the distribution of distances, suggesting that the coverage of chemistry space was largely unaffected by the removal of outliers. This in turn implies that the applicability domain for models derived from the filtered

data set will not be reduced. In contrast, all other methods (except for random removal) demonstrated truncation at long distances implying reduced internal diversity.



**Figure 7.** A comparison between pairwise Euclidean distances distributions before (red lines) and after (blue lines) the removal of outliers from the DHFR data set. (A) GA-*k*NN optimization-based outlier removal (75 compounds removed); (B) GA-*k*NN optimization-based outlier removal (87 compounds removed); (C) *k*NN-OR optimization-based outlier removal (87 compounds removed); (D) distance-based outlier removal (87 compounds removed); (E) distance K-based outlier removal (87 compounds removed); (F) one-class SVM-based outlier removal (87 compounds removed); (G) statistics-based outlier removal (87 compounds removed); (H) random “outlier” removal (87 compounds removed).

Compounds surviving the filtration process were divided into a modeling set (469 compounds) and a test set (117 compounds) and subjected to QSAR modeling using *k*NN and RF. The

resulting models were evaluated by standard parameters. Modeling sets subjected to  $k$ NN were evaluated using LOO cross-validation (Equation 8), while modeling sets subjected to RF (also known as out-of-bag set) were evaluated by the determination coefficient ( $R_{OOB}^2$ ; Equation 9). Test sets (external validation sets) were evaluated by the external explained variance  $Q_{ext}^2$ ; equation Equation 10).

$$R_{OOB}^2 = 1 - \frac{\sum_Y (Y_{exp} - Y_{OOB})^2}{\sum_Y (Y_{exp} - \bar{Y}_{exp})^2} \quad (9)$$

$$Q_{ext}^2 = 1 - \frac{\sum_Y (Y_{exp} - Y_{pre})^2}{\sum_Y (Y_{exp} - \bar{Y}_{exp})^2} \quad (10)$$

where  $Y_{exp}$  is the experimental value  $Y_{OOB}$  and  $Y_{pre}$  are the predicted value and  $\bar{Y}_{exp}$  is the mean of the experimental results over modeling set (training set) compounds.

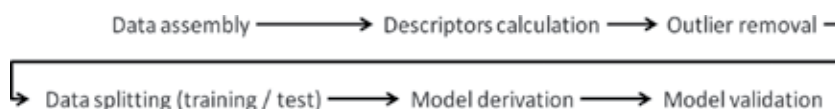
The results are presented in **Table 2** and demonstrate that the rational removal of outliers using the  $k$ NN-OR and the GA- $k$ NN algorithms led to filtered data sets which produced both  $k$ NN-based and RF-based QSAR models with the best prediction statistics.

	$k$ NN		RF	
	$Q_{LOO}^2$	$Q_{ext}^2$	$R_{OOB}^2$	$Q_{ext}^2$
GA- $k$ NN	0.78	0.83	0.73	0.75
$k$ NN-OR	0.79	0.83	0.71	0.77
Distance-based	0.54	0.63	0.55	0.68
Distance K-based	0.64	0.74	0.66	0.73
One-class SVM	0.62	0.65	0.62	0.72
Statistics	0.55	0.54	0.55	0.61
Random consensus	0.59	0.53	0.61	0.62

**Table 2.** Results obtained for the DHFR data set using  $k$ NN and RF. The results given under “random consensus” were averaged over all 10 random models.

## 5. Applications

While developed independently, all algorithms, together with additional tools, were incorporated into a machine-learning workflow for data mining and were used for the derivation of multiple QSAR models. The resulting workflow is depicted in **Figure 8**.



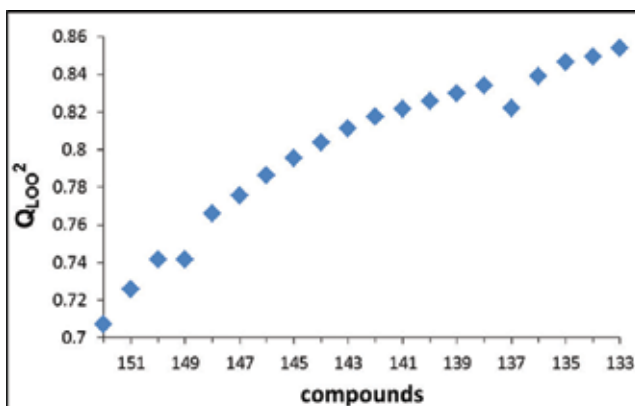
**Figure 8.** A schematic representation of the machine-learning workflow.

Below, we provide two examples taken from the fields of pharmaceutical and material sciences.

### 5.1. Blood–brain barrier permeation (logBBB) model

Blood–brain barrier permeability is an important parameter in drug design. Drugs targeting the central nervous system (CNS) are required to permeate through the blood–brain barrier. Conversely, drugs that do not affect the CNS should not penetrate the barrier due to potential side effects. Blood–brain barrier permeability is typically expressed as logBBB with positive and negative values indicating, respectively, permeating and non-permeating compounds. Since the experimental determination of logBBB values is resources consuming, multiple QSAR models have been developed to predict this property [19, 31, 41].

A data set of 152 compounds with known logBBB values was compiled from the literature, manually curated and characterized by 15 descriptors [19, 31]. This data set was subjected to the *k*NN-OR algorithm (Section 4.3) using a stopping criterion of  $Q_{LOO}^2 > 0.85$ . The application of this criterion led to the removal of 19 compounds, leaving 133 compounds for QSAR modeling. The stepwise  $Q_{LOO}^2$  values upon compounds removal is presented in **Figure 9**.



**Figure 9.**  $Q_{LOO}^2$  values as a function of compound removal for the logBBB data set. Outlier removal began with a set of 152 compounds, and the stopping criterion was met after the removal of 19 compounds.

Compounds surviving the filtration procedures were divided into a modeling set (106 compounds) and a validation test set (27 compounds) using the representativeness function described in Section 4.1. QSAR models were built on the modeling sets using either *k*NN

(Section 4.2) or the random forest (RF) algorithms, and the resulting models were tested on the validation sets. Model performances were evaluated with Equations 8–10 and were found to be  $Q_{LOO}^2 = 0.81$  and  $Q_{ext}^2 = 0.88$  for  $kNN$  and  $R_{LOO}^2 = 0.60$  and  $Q_{ext}^2 = 0.65$  for RF [31]. These values are similar to those obtained by other logBBB QSAR models.

Similar results were obtained when this data set was filtered using the GA- $kNN$  algorithm [41]. In this case, the stopping criterion ( $Q_{LOO}^2 > 0.85$ ) was met after the removal of 13 compounds. As before compounds surviving the filtration procedure were divided into a modeling set (111) and a test set (28) and subjected to QSAR modeling leading to  $Q_{LOO}^2 = 0.80$  and  $Q_{ext}^2 = 0.86$  for  $kNN$  and  $R_{OOb}^2 = 0.67$  and  $Q_{ext}^2 = 0.65$  for RF [41].

## 5.2. Photovoltaic activities of solar cells

The raise in demands for clean energy is likely to increase the importance of solar cells as future energy resources. In particular, cells entirely made of metal oxides have the potential to provide clean and affordable energy if their power conversion efficiencies are improved. Designing solar cells with improved photovoltaic properties could be greatly assisted by the application of optimization algorithms as described in this chapter.

Yosipof et al. [30] used the workflow described in **Figure 8** to build predictive  $kNN$ -based models for a library of 169 solar cells made of a combination of titanium and copper oxides ( $TiO_2|Cu_2O$ ). In this case, the dependent variables which had to be predicted were the open-circuit voltage ( $V_{OC}$ ), the short-circuit current ( $J_{SC}$ ), and the internal quantum efficiency ( $IQE$ ), and the dependent variables consisted of the thicknesses of the copper oxide and the titanium oxide layers, the ratio between the layers thicknesses, the experimentally measured bandgap and the maximum theoretical calculated photocurrent. In this case, subjecting the cells to the  $kNN$ -OR algorithm did not lead to the removal of any outliers and consequently, the entire library could be used for QSAR modeling. The library was divided into a modeling (training) set and a validation test set using the algorithm described in Section 4.1, and the predictive QSAR models were derived with the  $kNN$  algorithm described in Section 4.2. The results are presented in **Table 3**.

End point	$Q_{LOO}^2$	$Q_{ext}^2$	Descriptors selected
$J_{SC}$	0.92	0.92	The thickness of the titanium oxide layer and the thickness of the copper oxide layer
$V_{OC}$	0.78	0.89	The thickness of the titanium oxide layer and the thickness of the copper oxide layer
$IQE$	0.91	0.87	The thickness of the titanium oxide layer and the thickness of the copper oxide layer

**Table 3.** Results obtained with the  $kNN$  algorithm for the  $TiO_2|Cu_2O$  library.

Overall, the resulting models demonstrated good prediction statistics suggesting that they are likely to be useful for the design of new and improved solar cells. In addition, the feature

selection procedure inherent to the  $k$ NN algorithm highlighted the importance of the metal oxide layer thickness in controlling the photovoltaic properties (last column in **Table 3**).

## 6. Conclusions

In this chapter, we introduced several tools based on global stochastic optimization for chemoinformatics and material-informatics applications in three areas. To select representative subsets from within parent data sets, we described a new representativeness function and its optimization either alone or simultaneously with the MaxMin function. The two resulting algorithms were found to outperform previously reported subset selection methods.

For the derivation of predictive, nonlinear machine-learning models we reviewed the  $k$ NN algorithm, which searches the space of available descriptors combinations to identify those that best satisfy the similar properties principle. Descriptors spaces compatible with this principle give rise to models with good prediction statistics.

Finally, we introduced two new algorithms for the identification and subsequent removal of outliers based on the  $k$ NN method. The  $k$ NN-OR algorithm iteratively removes from the parent data set compounds (outliers) which the best improve model performances. The GA- $k$ NN algorithm simultaneously optimizes model performances together with the number of outliers. This algorithm is usually able to remove a smaller number of outliers while still maintaining good model performances. Retaining in the data set submitted to QSAR modeling as many compounds as possible is likely to increase the applicability domain of the resulting model. The new algorithms were found to outperform other outlier removal methods when tested on three data sets.

Finally, the new algorithms, together with additional tools, were combined into a machine-learning workflow and used for the derivation of predictive QSAR models.

The algorithms presented in this chapter are likely to be useful for multiple applications in the fields of chemoinformatics and material-informatics.

## Author details

Abraham Yosipof<sup>1\*</sup> and Hanoch Senderowitz<sup>2</sup>

\*Address all correspondence to: [avi.yosipof@gmail.com](mailto:avi.yosipof@gmail.com)

1 Department of Business Administration, Peres Academic Center, Rehovot, Israel

2 Department of Chemistry, Bar-Ilan University, Ramat-Gan, Israel

## References

- [1] Leach A.R. Molecular modelling: principles and applications. Pearson Education: Harlow, England; 2001.
- [2] Liu M., Wang S. MCDOCK: a Monte Carlo simulation approach to the molecular docking problem. *Journal of Computer-Aided Molecular Design*. 1999;13(5):435–451.
- [3] Jorgensen W.L. Efficient drug lead discovery and optimization. *Accounts of Chemical Research*. 2009;42(6):724–733.
- [4] Agrafiotis D.K., Cedeño W. Feature selection for structure–activity correlation using binary particle swarms. *Journal of Medicinal Chemistry*. 2002;45(5):1098–1107.
- [5] Zheng W., Tropsha A. Novel variable selection quantitative structure–property relationship approach based on the k-nearest-neighbor principle. *Journal of Chemical Information and Computer Sciences*. 1999;40(1):185–194.
- [6] Hassan M., Bielawski J., Hempel J., Waldman M. Optimization and visualization of molecular diversity of combinatorial libraries. *Molecular Diversity*. 1996;2(1):64–74.
- [7] Agrafiotis D.K. Stochastic algorithms for maximizing molecular diversity. *Journal of Chemical Information and Computer Sciences*. 1997;37(5):841–851.
- [8] Cherkasov A., Muratov E.N., Fourches D., Varnek A., Baskin I.I., Cronin M., et al. QSAR modeling: where have you been? Where are you going to? *Journal of Medicinal Chemistry*. 2013;57(12):4977–5010.
- [9] Pareto V. *Manual of political economy*. Milan: Milan, Italy 1906.
- [10] Mitchell M. *An introduction to genetic algorithms*. London: MIT Press; 1998.
- [11] Banzhaf W., Nordin P., Keller R.E., Francone F.D. *Genetic programming: an introduction*. San Francisco, USA: Morgan Kaufmann; 1998.
- [12] Kennedy J., Eberhart R. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*; Perth, WA. IEEE; 1995. pp 1942–1948.
- [13] Metropolis N., Ulam S. The Monte Carlo method. *Journal of the American Statistical Association*. 1949;44(247):335–341.
- [14] Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing. *Science*. 1983;220(4598):671–680.
- [15] Glick M., Rayan A., Goldblum A. A stochastic algorithm for global optimization and for best populations: a test case of side chains in proteins. *Proceedings of the National Academy of Sciences*. 2002;99(2):703–708.

- [16] Drew K.L.M., Baiman H., Khwaounjoo P., Yu B., Reynisson J. Size estimation of chemical space: how big is it? *Journal of Pharmacy and Pharmacology*. 2012;64(4):490–495.
- [17] Clark R.D. OptiSim: an extended dissimilarity selection method for finding diverse representative subsets. *Journal of Chemical Information and Computer Sciences*. 1997;37(6):1181–1188.
- [18] MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Fifth Berkeley Symposium on Mathematical Statistics and Probability*; Berkeley, California. University of California Press; 1967. pp 281–297.
- [19] Yosipof A., Senderowitz H. Optimization of molecular representativeness. *Journal of Chemical Information and Modeling*. 2014;54(6):1567–1577.
- [20] Gillet V.J., Khatib W., Willett P., Fleming P.J., Green D.V.S. Combinatorial library design using a multiobjective genetic algorithm. *Journal of Chemical Information and Computer Sciences*. 2002;42(2):375–385.
- [21] Gillet V.J., Willett P., Fleming P.J., Green D.V.S. Designing focused libraries using MoSELECT. *Journal of Molecular Graphics and Modelling*. 2002;20(6):491–498.
- [22] Eriksson L., Jaworska J., Worth A.P., Cronin M.T., McDowell R.M., Gramatica P. Methods for reliability and uncertainty assessment and for applicability evaluations of classification- and regression-based QSARs. *Environmental Health Perspectives*. 2003;111(10):1361–1375.
- [23] Tropsha A. Best practices for QSAR model development, validation, and exploitation. *Molecular Informatics*. 2010;29(6–7):476–488.
- [24] Hansch C., Fujita T. p-sd-pi analysis. A method for the correlation of biological activity and chemical structure. *Journal of the American Chemical Society*. 1964;86(8):1616–1626.
- [25] Polishchuk P.G., Muratov E.N., Artemenko A.G., Kolumbin O.G., Muratov N.N., Kuz'min V.E. Application of random forest approach to QSAR prediction of aquatic toxicity. *Journal of Chemical Information and Modeling*. 2009;49(11):2481–2488.
- [26] Gramatica P. Principles of QSAR models validation: internal and external. *QSAR & Combinatorial Science*. 2007;26(5):694–701.
- [27] Tropsha A., Gramatica P., Gombar V.K. The importance of being earnest: validation is the absolute essential for successful application and interpretation of QSPR models. *QSAR & Combinatorial Science*. 2003;22(1):69–77.
- [28] Whitley D.C., Ford M.G., Livingstone D.J. Unsupervised forward selection: a method for eliminating redundant variables. *Journal of Chemical Information and Computer Sciences*. 2000;40(5):1160–1168.

- [29] Johnson M.A., Maggiora G.M. Concepts and applications of molecular similarity. New York: John Wiley & Sons; 1990.
- [30] Yosipof A., Nahum O.E., Anderson A.Y., Barad H.N., Zaban A., Senderowitz H. Data mining and machine learning tools for combinatorial material science of all-oxide photovoltaic cells. *Molecular Informatics*. 2015;34(6–7):367–379.
- [31] Yosipof A., Senderowitz H. k-Nearest neighbors optimization-based outlier removal. *Journal of Computational Chemistry*. 2015;36(8):493–506.
- [32] Kim K. Outliers in SAR and QSAR: is unusual binding mode a possible source of outliers? *Journal of Computer-Aided Molecular Design*. 2007;21(1–3):63–86.
- [33] Ben-Gal I. Outlier detection. In: Maimon O., Rokach L., editors. *Data Mining and Knowledge Discovery Handbook*. New York: Springer US; 2005. pp 131–146.
- [34] Barnett V., Lewis T. Outliers in statistical data. New York: Wiley; 1994.
- [35] Hawkins D.M. Identification of outliers. London: Chapman and Hall; 1980.
- [36] Knorr E., Ng R. Algorithms for mining distance-based outliers in large datasets. In: the 24th International Conference on Very Large Data Bases, VLDB; New York, USA: Morgan Kaufmann Publishers Inc.; 1998.
- [37] Ramaswamy S., Rastogi R., Shim K. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*. 2000;29(2):427–438.
- [38] Kaufman L., Rousseeuw P.J. Finding groups in data: an introduction to cluster analysis. New York: John Wiley & Sons; 2009.
- [39] Schölkopf B., Smola A.J., Williamson R.C., Bartlett P. L. New support vector algorithms. *Neural Computation*. 2000;12(5):1207–1245.
- [40] Cao D.S., Liang Y.Z., Xu Q.S., Li H.D., Chen X. A new strategy of outlier detection for QSAR/QSPR. *Journal of Computational Chemistry*. 2010;31(3):592–602.
- [41] Nahum O.E., Yosipof A., Senderowitz H. A multi-objective genetic algorithm for outlier removal. *Journal of Chemical Information and Modeling*. 2015;55(12):2507–2518.
- [42] Zitzler E., Laumanns M., Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) 2001.

---

# Optimization Algorithms in Project Scheduling

---

Amer M. Fahmy

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63108>

---

## Abstract

Scheduling, or planning in a general perspective, is the backbone of project management; thus, the successful implementation of project scheduling is a key factor to projects' success. Due to its complexity and challenging nature, scheduling has become one of the most famous research topics within the operational research context, and it has been widely researched in practical applications within various industries, especially manufacturing, construction, and computer engineering. Accordingly, the literature is rich with many implementations of different optimization algorithms and their extensions within the project scheduling problem (PSP) analysis field. This study is intended to exhibit the general modelling of the PSP, and to survey the implementations of various optimization algorithms adopted for solving the different types of the PSP.

**Keywords:** project scheduling, project schedules optimization, resource-constrained scheduling, scheduling models, optimization algorithms

---

## 1. Introduction

The project scheduling problem (PSP) is one of the most challenging problems in the operations research (OR) field; thus, it has attracted large number of researchers within its modeling, solution methodologies, and optimization algorithms. The OR literature is intensely rich with researches focusing on different PSP problem types, from which the most famous and heavily researched problem type is the resource-constrained project scheduling problem (RCPSp).

PSP, especially RCPSp, has been considered as NP-Hard in the strong sense [1], and accordingly most researches within the last two decades concentrated on heuristics and meta-heuristics for solving different PSP types.

---

This study will start in Section 2 with reviewing the PSP modelling for various problem types. Then, PSPs solution approaches and architectures presented in literature will be demonstrated in Section 3. And finally, the last section of this study will survey the schedule optimization algorithms, and how each optimization technique was adopted and implemented within the project scheduling field; with a large focus on meta-heuristic optimization techniques, which are the base for most practical approaches for solving real-life/practical PSPs.

## 2. PSP modelling

According to the detailed surveys of the problem's developments and extensions [2–9], there is almost a common agreement on the concepts of how the problem should be mathematically modelled; however, the main differences were in the various problems classifications and the different mathematical notations used.

### 2.1. The general PSP model

In literature, the PSP was initially introduced within the manufacturing industry for job-shop scheduling, with a main concern of optimal allocation over time of the shop floor's scarce resources. Consequently, the PSP modelling has developed through the development of the basic modelling of the RCPSP.

Several problem notations and models were presented for the RCPSP, as surveyed by Brucker [5], from which the basic and most widely adopted notation can be summarized as follows: A project is represented by a set of activities  $V = \{1, \dots, n\}$ , where 1 and  $n$  are two dummy activities resembling the project network's start and end nodes. The characteristics defining each activity, that is,  $V$ : processing duration  $d_i$ , resource requirements  $r_{ik}$  (for each resource  $k$ ) along the activity's processing time, and a set of predecessors  $P_i$  logically tied to activity  $i$  with finish-to-start (FS) logic relations. The availability of each resource  $k$  is constrained throughout the project by the available resource units  $a_k$ . And finally, the problem's objective is to minimize the time span  $T$  of the project's schedule  $S$  without violating the resources constraints; where  $S$  is represented by a set of activities' start times  $S_1$  to  $S_n$ ; and  $T = S_n$ .

### 2.2. PSP model extensions

The basic RCPSP model has several short falls with respect to over-simplifying the characteristics of real-life scheduling problems. Consequently, several studies in the PSP context were focused on improving the problem's modelling by developing extensions which add missing practical aspects to the basic PSP model. The most important extensions can be categorized based on the short falls in the basic model which they addressed as follows:

- The basic PSP (and RCPSP) model considers only FS logic relations, and assumes all schedule relations without lag. Demeulemeester and Herroelen [10], Klein and Scholl [11], Chassiakos and Sakellariopoulos [12], and Vanhoucke [13] have researched this short fall and presented few model extensions to capture lag periods and other relation types available

in real applications. They accommodated these issues within the model by transforming all relations into start-to-start (SS) relations ( $S_{ij}$ ), and combined all sets of activities' predecessors into a single set of time lags  $l_{ij}$  corresponding to the schedule's logic; where the values of lag can be set to zero to represent SS relation,  $d_i$  to represent FS relation,  $d_j$  to represent FF relation, or any other necessary time lag. These model extensions are known in the scheduling literature as '*Generalized Precedence Relations*' or '*Minimum/Maximum Time Lags*'.

- Activities in reality are not necessarily processed in a single continuous run as assumed in the original PSP model. The need for presenting activities' splitting, or '*Pre-emption*', was introduced by allowing each schedule activity  $i$  to be split, or pre-empted, into  $j$  sections corresponding to the activity's duration  $d_i$ . For further details on pre-emption extensions, the reader is referred to [14–16]. Another alternative *pre-emption* modelling was presented and developed by Franck et al. [17], Lamothe et al. [18], and Kreter et al. [19], to resemble the non-linear availability of non-renewable resources through break-calendars.
- The '*Multi-Mode Scheduling*' is another practical extension in the project scheduling context, where activities can be executed in one of several alternative ways/methods. This has been modelled by Elmaghraby [20], Kolisch and Drexel [21], Hartmann [22], and Alcaraz et al. [23] through allowing each activity to have a set of execution modes  $M_i$ , where each activity mode is having a different duration and/or resource requirements.
- '*Cumulative Resources*' or '*Inventory Constraints*' is another form of resources irregularity. It was introduced by Neumann and Zimmermann [24] as a resource which is produced by some activities, cumulated in a storage area, and then used by other activities (e.g. precast elements in construction projects).

By combining most of these extensions to the basic model, the problem formulation can be summarized as follows:

$$\text{Objective:} \quad \text{Minimize } S_{n,0} \quad (1)$$

$$\text{Subject to:} \quad S_{1,1} = 0, \quad (2)$$

$$S_{i,0} + l_{ij} \leq S_{j,0} \quad \forall (i,j) \in L; i, j \in V \quad (3)$$

$$S_{i,j-1} + 1 \leq S_{i,j}, i = 1, 2, \dots, n; j = 1, 2, \dots, d_{im}; m \in M_i \quad (4)$$

$$\sum_{i \in S_{tjkm}} r_{ijk} \leq a_k \quad \forall k \in K; j = 1, 2, \dots, d_{im}; m \in M_i; t = 1, 2, \dots, T \quad (5)$$

where  $V$  is the activities set;  $T$  is the schedule's time span;  $S_{ij}$  is the start time for section  $j$  of activity  $i$ ;  $d_{im}$  is the processing time of activity  $i$  under the execution mode  $m$ ;  $L$  is the set of schedule's logic (or precedence), including time lags represented in the form of activity pairs, and  $l_{ij}$  is the time lag between activities  $i$  and  $j$ ;  $S_t$  is a set which encapsulates the progress of all activities within time interval  $[t-1, t]$ ;  $K$  is the renewable resources set;  $r_{ijkm}$  is the resource requirement from resource  $k$  in the execution mode  $m$  for section  $j$  of activity  $i$ ; and finally,  $a_k$  represents the total available units for resource  $k$ .

### 2.3. PSP objectives extensions

The objective of the basic PSP model is minimizing time, while several practical applications involve other cost- or resource-related objectives. All above mentioned model extensions were also related to minimizing the project's make-span; however, other practical objectives were also introduced in literature as special cases of the PSP. Extensions to PSP objectives can be classified under two main categories: cost related and quality related.

Cost-related objectives and/or constraints were introduced to overcome the over-simplification of the basic PSP models with respect to financial side effects of schedule changes. Various models were generated to present cost aspects of PSP, the most famous of which is the *trade-off problem*, while quality-related objectives were introduced to enhance the robustness of optimized schedules.

These model extensions (i.e. with different objective functions) were presented in as different PSP problem types, the most popular of which are:

- *The trade-off problems*: The 'Time/Resource Trade-off Problem (TRTP)' [25–27] and the 'Time-Cost Trade-off Problem (TCTP)' [28, 29] are the most famous objective-related extensions of the PSP model. In these problem types, shorter schedule time is exchanged with the increase in resources (or vice versa); and project cost can be either modelled as a constraint or as a non-renewable resource. In scheduling literature, *trade-off* problems are generally being used in one of two forms:

*Case 1*: Minimize make-span, and constraint total costs with a predefined budget:

*Objective Function*: Minimize  $F_n$

$$\text{New Constraint : } \sum_{i \in V'} c_i \leq \bar{C} \quad (6)$$

where,  $c_i$  = Cost of activity  $i$ , and  $\bar{C}$  = Project maximum allowed cost (or project budget)

*Case 2*: Minimize total costs, and constraint total project make-span with a predefined target date or deadline date:

$$\text{Objective Function : Minimize } \sum_{i \in V'} c_i \quad (7)$$

$$\text{New Constraint : } F_n \leq \bar{d} \quad (8)$$

where:  $c_i$  = Cost of executing activity  $i$ , and  $\bar{d}$  = Deadline for the project's completion

- *Liquidity constraints:* In practical applications, project's liquidity is usually constrained with a predefined value of cash allotted to the project; and accordingly, this value should not be exceeded by the negative side of the project's cash flow. To enable amending the scheduling model with cash flow constraints, several financial aspects will have to be taken into consideration such as selling price, advance payment, retention, and payment lag (the time lag between invoices applications and the actual receipt of payments); then a new constraint is to be added as follows:

$$\text{Cash flow constraint: } CF_t \geq -|NCF_{\max}| \forall t = 1, 2, \dots, T \quad (9)$$

Where:  $CF_t$  = Cash flow at period  $t$ , and  $NCF_{\max}$  = Predefined maximum liquidity

- For self-financed projects, the scheduling literature contains few special case problems such as RCPSP with discounted cash-flow (RCPSP-DC) and maximum net present value (Max-NPV) where the time cost of financing money is being considered in the project's overall cost (refer to [30–32]). Representing this mathematically involves introducing a per period discounting rate ( $\alpha$ ), and adding it exponentially to represent its effect on successor periods; thus, the discounted cash flow for each activity  $i$  will be as shown in equation 10. And finally, the objective function will have to be modified to include the net present value for the per period cash flow values:

$$CF_i = \sum_{t=1}^{d_i} c_{it} e^{\alpha(d_i-t)} \quad (10)$$

$$\sum_{i \in V} q_t CF_i \quad (11)$$

$$\text{Where : } q_t = \exp(-\alpha t) \quad (12)$$

- *Quality indices:* Schedules quality measures were introduced in literature for the purpose of improving schedules robustness [33]. These indices were mainly researched within robust proactive scheduling approaches which focus on building predictive schedules for satisfying performance requirements predictably in a dynamic environment [34–36].

## 2.4. Multi-objective PSP (MOPSP) modelling

The above review of PSP problem types shows clearly that practical scheduling application involves multi-objective nature; which attracted few early scheduling researches (such as [37, 38]) to try to define how multi-objective concepts can be introduced to the scheduling field. Nevertheless, to date, there are only few researches in the scheduling context which adopted multi-objective approaches [39].

In general, the approaches presented for MOPSP modelling (as defined and/or surveyed by Hwang [37], Slowinski [38] and Ballestín and Branco [39]) can be summarized under the following main categories:

1. Non-interactive approach: Where the optimality decision is left to the optimization algorithm after feeding it with either a predefined '*Weighted Objectives Function*', or an '*Objectives Priority List*' through which objectives are being ordered and optimized based on predefined priorities.
2. Semi-interactive approach: The optimization algorithm defines the solutions '*Pareto Front*', or a group of optimal/near-optimal solutions for single/multiple objectives; then the decision for selecting an optimum solution is left to the decision maker.
3. Interactive approach: In this approach, the algorithm interacts with the decision maker throughout the optimization steps. In each step, good quality solutions are proposed to the decision maker to select the most effective solutions, which will then be used by the algorithm in the next iteration to generate further improved solutions.

For further details and applications of the MOPSP, the reader is referred to the research of Tung et al. [40], Hsu et al. [41], Kacem et al. [42, 43], Loukil et al. [44], Xia & Wu [45], and Fahmy et al. [9].

### 3. PSP solution approach and architecture

#### 3.1. Solution approaches

The approach to be used for solving the PSP is dependent on the problem's conditions and the application's environment. In literature, the solution approaches can be classified under two main categories: *static scheduling and dynamic scheduling*.

*Static scheduling* (or *predictive scheduling*) is the process of identifying how and when each activity in schedule should be executed. It involves the generation of a good quality optimized initial (or baseline) schedule. This can be in a *deterministic* approach (like the vast majority of researches within the scheduling context [4]), where the durations of all schedule activities (or activity modes) are initially available; or can be in a *stochastic* project scheduling approach, which aimed to close gaps within the initial available information to enable scheduling under uncertainties. The stochastic activity durations method is a probabilistic modelling approach for scheduling project activities with uncertain durations [46]. The duration of activities is defined in the problem model by random vectors of durations, which are distributed according to a deterministic probability distribution [5].

In practical scheduling, real-time events extremely disrupt schedules integrity, and causes optimized schedules to become neither optimized nor realistic. The *dynamic scheduling* concept was introduced to enable, dynamically, the mitigation of the impacts of real-time events on project schedules. A *dynamic scheduling* solution involves the selection of a scheduling approach, a rescheduling strategy, and a rescheduling policy; or in simple terms, the process

of how to generate the original baseline and the strategy of how to respond to real-time events. For further details on dynamic scheduling, the reader is referred to the surveys of Herroelen and Leus [46], Aytug et al. [47] and Ouelhadj and Petrovic [48].

### 3.2. Solution architectures

The *dynamic scheduling* solution architecture involves several cycles of *static scheduling* to be executed based on certain timing and/or criteria (based on a predefined rescheduling policy), and to involve partial or full schedule optimization (based on a predefined rescheduling strategy). The solution architecture can also involve one optimization level (*single-agent architecture*) or several optimization levels (*multi-level autonomous or mediator architecture*).

Although approaches and architectures of scheduling solutions seem to contain various analysis concepts, the basic outlines of the underlying analysis engine (or the optimization algorithm) remain the same. Whether the optimization process is performed once or at different stages, or the process is executed on single or multiple levels, the core algorithm will be focusing within each optimization cycle on a static schedule snapshot (full or partial). And accordingly, the final section of this study will review optimization algorithms from a generic perspective, regardless of the approach or architecture of the scheduling solution to be integrated with.

## 4. PSP optimization algorithms

### 4.1. Heuristic algorithms

Scheduling is one of the most researched topics within the field of operations research; thus, it will be very difficult to present a survey which covers all heuristic approaches presented for schedules optimization. But in general terms, scheduling heuristic algorithms, or the majority of them, are having a common procedural approach which can be generalized as follows: (1) Initialize an ordered activity list, (2) generate schedule, and (3) improve the schedule quality. And accordingly, these heuristics consist of three main components:

1. A predefined criteria for initial ordering of activities [or a *priority rule*],
2. A process for creating the schedule from the ordered activity list [or a schedule generation scheme (SGS)], and optionally
3. The application of one or more processes to improve the generated schedule's quality.

#### 4.1.1. Priority rules

The basic function of *priority rules (PRs)* is to define an initial arrangement for the activities list in a logical way which will produce a solution with good quality. Kelley [49] introduced the concepts and the first set of PRs; then, several other researches followed Kelly's merits by introducing additional PRs and comparing their results (as surveyed by Kolisch [50]). PRs

provide simple and speedy way to obtain solutions, and that is why they are widely used by commercial scheduling software [51]. **Table 1** lists the most commonly adopted *PRs* within scheduling literature, and their criteria for activities ordering.

Basis	Priority rule	Ordering criteria
Time	Earliest Start Time (EST)	Ascending based on activities earliest start
	Earliest Finish Time (EFT)	Ascending based on activities earliest finish
	Latest Start Time (LST)	Ascending based on activities latest start
	Latest Finish Time (LFT)	Ascending based on activities latest finish
Duration	Shortest Processing Time (SPT)	Ascending based on activities shortest processing mode duration
	Longest Processing Time (LPT)	Ascending based on activities largest processing mode duration
	Greatest Rank Positional Weight (GRPW)	Descending based on the total duration of the activity and all its direct successors
Float	Minimum Slack (MSLK)	Ascending based on activities slack
Resources	Greatest Resource Work Content (GRWC)	Descending based on the total resource requests of the activity
	Greatest Cumulative Resource Work Content (GCRWC)	Descending based on the total resource requests of the activity and all its direct successors
Logic relations	Most Immediate Successors (MIS)	Descending based on the number of their direct successors
	Most Total Successors (MTS)	Descending based on the number of their direct and indirect successors
	Least Non-Related Jobs (LNRJ)	Activities are sorted in ascending order based on the number of activities which are not directly or indirectly inter-related

**Table 1.** Priority rules and related ordering criteria.

#### 4.1.2. Schedule generation scheme (SGS)

Ordered activity lists are then passed to a SGS to produce the output schedules. As per the survey presented by Kolisch [52], the first versions of the *serial* (SSGS) and the parallel (PSGS) were presented Kelley [49]; then later Bedworth and Bailey [53] introduced another approach for the PSGS which they titled as the "*Brooks Algorithm*".

PSGS has been verified that it can only generate non-delay schedules, and the set of non-delay schedules is just a subset of all schedules, hence the SSGS is suggested for RCPSP [54].

#### 4.1.3. Forward-backward scheduling (FBS)

The FBS is one of the most famous schedule improvement techniques. It was proposed by Li and Willis [55], and was found to significantly improve the results. Its procedures involve applying SGS in a forward direction and performing another cycle in reverse order and backward scheduling (reversed precedence network).

#### 4.1.4. Justification schemes

Valls et al. [56] introduced another process for improving schedules quality using a technique they called *justification scheme*. This process involves manipulating the activities positions in the project's time frame without violating the resource constraints, through two cycles, one forward (to the right) and another backward (to the left); which eventually guarantees an overall project duration either shorter or at least the same. Later, Fahmy et al. [57] presented the *stacking justification*, a variation to the original technique; within which the activities selection criteria in each justification cycle was modified in a way to minimize the gaps within resource usage profiles.

### 4.2. Meta-heuristic algorithms

The most common use of Meta-heuristics in PSP solving involves the generation of activities order list which can produce better solutions based on experience gained in previous generation cycles.

Heuristics can solve scheduling problems in short time, but because these procedures cannot adapt dynamically to the problems constraints, so the resulting solutions cannot be guaranteed to be neither optimum nor of good quality.

Due to the overwhelming complexity of scheduling problems within real-life applications, meta-heuristics techniques have been implemented in the development of most practical applications presented in literature during the last two decades.

Various meta-heuristic techniques were adopted in the PSP field, the following sections of this study will exhibit a non-exhaustive review for how each of the commonly adopted meta-heuristic optimization techniques was conceptually implemented within scheduling applications, whilst the details/variations of each of the optimization techniques were considered beyond the scope of this study, and accordingly will only be reviewed to the level needed to support the scheduling algorithms survey perspective.

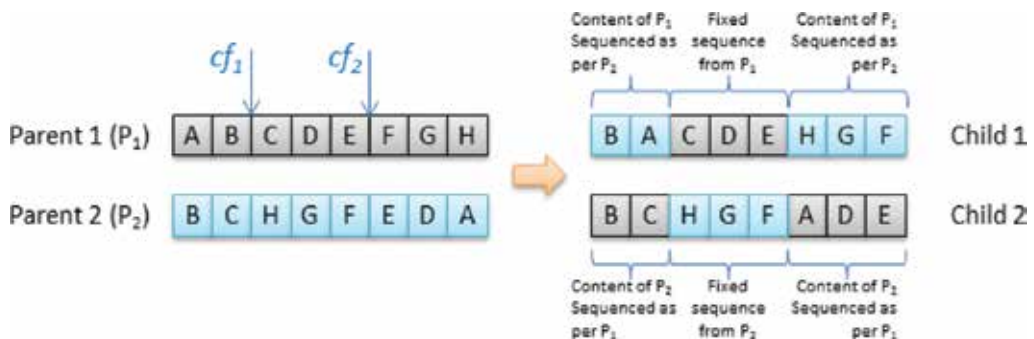
### 4.3. Genetic algorithms

Genetic algorithms (GA) technique is one of the most popular meta-heuristics in the field of scheduling. Holland [58] developed this method as a simplification of evolutionary processes occurring in nature. GA is basically an iterative evolutionary method through which the overall quality of solutions (or *genomes*) population is improved from one generation to the next through three nature resembled mechanisms: *selection*, *crossover*, and *mutation* [59].

In scheduling, the solutions population is selected either randomly or based on a predefined priority rule. The quality of schedules is calculated using an objective function based on the optimization goals, either for a single objective (minimizing time, cost, resource levelling ..., etc.), or combined objectives. Then iteratively, the solutions with higher quality are *selected* for *mating*; and then, using a *crossover* process, a new group of individuals are generated, and added to the best solutions reached previously to form the new generation. Finally, a *mutation* mechanism is applied in each generation to ensure further exploration of solutions-space.

There are several approaches adopted in scheduling literature for the presentation of schedules for GA implementation. The most common approach involves setting the *genomes* as a presentation of the activities priority/sequence list (*S*); where the sequence of activities within the *genome* will be used by the *SGS* to generate the solution's schedule.

For this presentation, the optimization process starts with population generation (randomly or using *PRs*). The *crossover* mechanism will then involve *mating* high fitness solutions, where each pair of *parent* solutions will generate a pair of *child* solutions using a predefined number of *crossing-points* (*cp*) (such as the two-point approach adopted by Shadrokh and Kianfar [60] as shown in **Figure 1**). And finally, a *mutation* mechanism is applied using a predefined probability  $P_{mut}$ ; where a random activity *i* is exchanged in the sequence list (*S*) with another random activity *j* ( $i \neq j$ ), and the same can be performed for the mutation of activity modes list if a multi-mode schedules was used [27].



**Figure 1.** Crossover mechanism for activities priority lists.

Several other priority-based presentations were introduced in literature, most of which were surveyed by Cheng et al. [61, 62]. Few other conceptually different presentations were also available in literature, such as the presentation of Wall [63], where he presented the schedule's *chromosome* as a binary string corresponding to each activity's lag from its normal scheduled position (i.e. a solution scheduled normally based on its precedence will have all its *chromosome* digits set to '0').

According to the detailed performance survey of Kolisch and Hartmann [64], the GA (and its variants/combinations with other techniques) is ranked as the best performing algorithm (as well as most of the top 10 performances) for most problem sizes of the single-mode RCPSP

(SRCPSP). For further review of GA implementations within RCPSP, refer to the publications of Hartmann [65, 66], Alcaraz and Maroto [67], Alcaraz et al. [68] and Valls et al. [56, 69].

#### 4.4. Particle swarm optimization (PSO)

Particle swarm optimization, in comparison with other commonly used optimization techniques, is one of the most recent optimization meta-heuristics. PSO was introduced by Kennedy and Eberhart [70] as a mathematical presentation for the swarming behaviour of flocking birds. PSO is an evolutionary algorithm, where a population of candidate solutions, resembled by particles, and the optimization process occurs by iteratively adjusting the particles' position and velocity within the search space through assessing solutions' quality through predefined measuring criteria.

With a simple mathematical presentation, PSO operates with the following two formulae, where each solution is presented as particle  $i$  with  $n$  number of components,  $V_{ij}^t$  is the velocity of component  $j$  of particle  $i$  in iterations  $t$  (and similarly for  $V_{ij}^{t-1}$  in iteration  $t-1$ );  $X_{ij}^t$  is the positions of component  $j$  of particle  $i$  in iterations  $t$ ; the positions vectors of the best solutions found up to iteration  $t-1$  locally for particle  $i$  and globally in the swarm are stored in  $L_{ij}^{t-1}$  and  $G_j^{t-1}$ , respectively;  $r_1$  and  $r_2$  are two random numbers (from 0 to 1);  $c_1$  and  $c_2$  are two learning coefficients ( $c_1$  defines the influence of the local best solution on the new velocities, while  $c_2$  applies a similar approach for the global best solution).

$$V_{ij}^t = wV_{ij}^{t-1} + r_1c_1(L_{ij}^{t-1} - X_{ij}^{t-1}) + r_2c_2(G_j^{t-1} - X_{ij}^{t-1}) \quad (13)$$

$$X_{ij}^t = X_{ij}^{t-1} + V_{ij}^t \quad (14)$$

Variations of PSO is beyond the scope of this study, but in general most PSO variations focused on improving PSO main drawbacks such as early convergence, parameter dependency, and loss of diversity. The most popular PSO variations are:

- Shi and Eberhart [71] introduced an inertia weight ( $w$ ) to enable the control of iterations velocity influence on succeeding iterations.

$$V_{ij}^t = w \times V_{ij}^{t-1} + c_1 \times r_1 \times (L_{ij}^{t-1} - X_{ij}^{t-1}) + c_2 \times r_2 \times (G_j^{t-1} - X_{ij}^{t-1}) \quad (15)$$

- Bratton and Kennedy [72] presented the 'Standard PSO' in which they introduced the *constriction factor* ( $\gamma$ ) as a multiplier to the equation of velocity.

$$V_{ij}^t = \gamma \times (V_{ij}^{t-1} + c_1 \times r_1 \times (L_{ij}^{t-1} - X_{ij}^{t-1}) + c_2 \times r_2 \times (G_j^{t-1} - X_{ij}^{t-1})) \quad (16)$$

In the scheduling field, PSO is conceptually implemented by resembling each schedule (or solution) as a particle, with the activities' priorities as the components of this particle [73, 74]. The quality of schedules (or solutions' fitness) is calculated via a predefined objective function. Initially, activities priorities can be initialized using a single priority rule, or a combined rule [57]. Then, iteratively, positions and velocities vectors of particles/components are adjusted, then the quality rating of each solution is assessed, the global best solution and the local best solution for each particle are logged. The stopping position can be set to a maximum analysis duration, a certain number of schedules to be generated, or a specific quality to be reached (Figure 2).

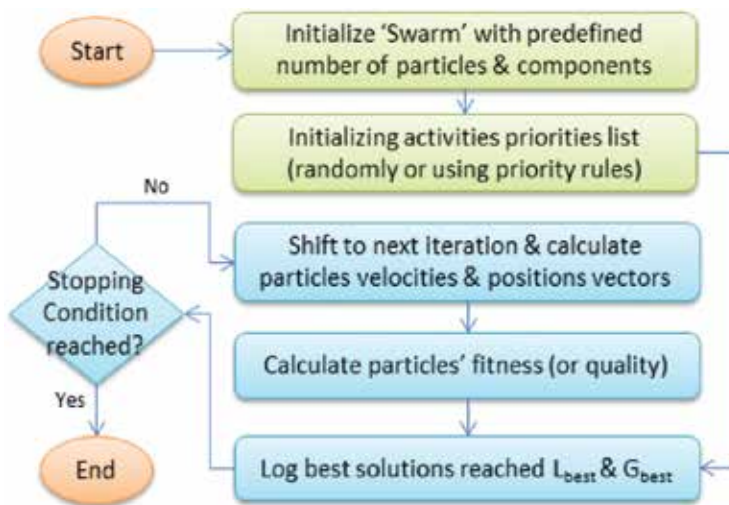


Figure 2. General PSO flow chart for scheduling.

Albeit that the number of application of PSO in the scheduling context is not large as GA, the results of its application, especially for SRCPSP, is highly ranked in general comparisons with all techniques (*c.f.* [54, 57]).

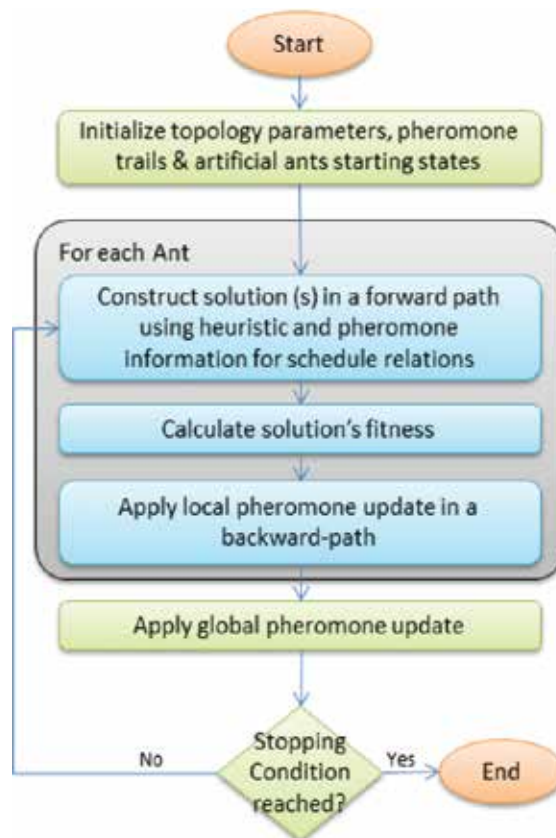
#### 4.5. Ant colony optimization (ACO)

ACO is a population based, multi-agent, meta-heuristic technique within the broader family of *swarm intelligence* optimization methods; initially proposed by Dorigo [75]. ACO uses the concepts of food seeking in ant colonies as a basis for an optimization algorithm for seeking optimal solutions within solutions space.

In nature, ants use *pheromone* as the means of communication when searching for food. Upon finding food, after random wandering in the colony's neighbourhood, ants lay *pheromone* in their way back to the colony; thus, other ants can use these traces to guide their movements from colony to food sources. These *pheromone* traces decays/evaporates with time, which leads to traces for shorter paths to be higher in strength than others, and accordingly guides ants to shorter paths to food.

Application of ACO in scheduling started with a proposed ant system for job-shop scheduling by Colormi et al. [76]; followed by several other applications in various scheduling problems, such as flow-shop scheduling [77–79], flexible job-shop scheduling [80], resource-constrained scheduling [81, 82], and total tardiness problems [83, 84].

The most common presentation of ACO in the scheduling literature was outlined by *Stützle and Dorigo* [85] as follows: set of network nodes ( $C = \{c_1, c_2, c_3, \dots, c_N\}$ ), set of problem states (sequence or relationships) of element  $C$  ( $x = \{c_i, c_j, c_k, \dots\}$ ), set of constraints ( $\Omega$ ), set of all states ( $X$ ), set of all feasible states ( $\bar{X} \subseteq X$ ), objective function  $f(s)$  (where  $s$  is a candidate solution,  $s \in S$ ), set of all feasible solutions ( $S^* \subseteq \bar{X}$ ), a *pheromone* trail ( $\tau_{ij}$ ) representing the desirability of relation ( $r_{ij}$ ), and finally heuristic problem-specific information can be defined within ( $\eta_{ij}$ ).



**Figure 3.** Simplified ACO flow chart for scheduling.

Then, the problem topology and the simulation behaviour can be simplified as shown in the flow chart in **Figure 3**. The algorithm starts with initializing the problem's topology (or the

network's details), the pheromone trails ( $\tau_0$ ) either randomly or using a priority rule, and the artificial ants starting states.

Then, within each iteration cycle, each ant moves a two-way path seeking a food source (or a feasible solution). In the forward path, the ant's movements (or selected network relations) is defined using heuristic information ( $\eta$ ) and current pheromone information ( $\tau$ ). After solution ( $s$ ) is constructed, the solution's fitness is calculated using the objective function ( $f$ ). The ant then returns in a backward path applying local pheromone based on the quality achieved for solution ( $s$ ). Finally, the global pheromone is updated before the stopping condition is checked.

Although, as per the detailed performance surveys mentioned earlier, ACO did not demonstrate a competitive performance to other meta-heuristics (GA, PSO, and TS), except for large size problems; this can be positively inferred that ACO's performance increases with the increase in problem's size due to its high exploration capabilities.

#### 4.6. Other methods

Due to the complicated and challenging nature of scheduling problems, the scheduling field has been usually one of the first testing fields for any meta-heuristic optimization technique being introduced within the *operational research* literature. It would be a very exhaustive task to summarize all meta-heuristics adopted for solving different scheduling problem types; so, beside the techniques mentioned in the previous sections, the following paragraphs briefly summarize other meta-heuristics widely adopted in the scheduling research context.

Tabu search (TS) is one of the techniques with high performance results within the scheduling literature. The TS is a local search technique, initially proposed by Glover [86]. It involves exploring the search space through searching neighbourhoods of potential solution(s). The following are some examples of TS applications in scheduling: resource-constrained scheduling [87–89], flow-shop scheduling [90, 91], and flexible job-shop scheduling [92].

The simulated annealing (SA) was presented by Kirkpatrick et al. [93], as an optimization method which mimics the behaviour of a system in thermodynamic equilibrium at certain temperature. It is a probabilistic meta-heuristic which focus on finding a quick approximated global optimum of a search space. SA showed moderate performance with scheduling applications, such as Rutenbar [94], Bouleimen and Lecocq [95], and Dai et al. [96].

Other meta-heuristics adopted in scheduling include, Neural networks [97–99], Scatter Search [100], Electromagnetism-like method [101, 102], Sampling method [103, 104], and Bees Algorithm [105, 106].

In addition, scheduling literature contains vast amount of meta-heuristics with combinations of the various optimization techniques, such as Kochetov and Stolyar [107] for GA and TS, Niu et al. [108] for PSO and GA, Chen and Shahandashti [109] for GA and SA, Moslehi and Mahnam [110] for PSO and LS, and Deane [111] for GA and NN.

## 5. Conclusion

The design and implementation of a robust scheduling system are essential for the successful use of planning and scheduling practices within projects. A scheduling system involves modelling the problem, selecting a solution approach to be used in a static and/or dynamic analysis for optimizing schedules, and finally the selection of an optimization technique which suits most the characteristics and conditions of the project type under analysis.

This study reviewed the concepts and researches presented for these three factors of building a scheduling system, with a more detailed focus on meta-heuristic optimization algorithms adopted in the project-scheduling context.

## Author details

Amer M. Fahmy

Address all correspondence to: [amer.fahmy@dynamicscheduling.net](mailto:amer.fahmy@dynamicscheduling.net)

CCC/TAV JV, Development of Muscat International Airport Project, Muscat, Oman

## References

- [1] Blazewicz, J., Lenstra, J.K., Rinnooy Kan, A. H. G. (1983). *"Scheduling subject to resource constraints: Classification and complexity"*. Discrete Applied Mathematics 5, 11-24.
- [2] Icmeli, O., Erenguc, S.S., & Zappe, C.J. (1993). *"Project scheduling problems: A survey"*. International Journal of Operations & Production Management 13 (11), 80-91.
- [3] Elmaghraby, S.E. (1995). *"Activity nets: A guided tour through some recent developments"*. European Journal of Operational Research 82, 383-408.
- [4] Herroelen, W., Reyck B.D., & Demeulemeester E. (1998). *"Resource-constrained project scheduling: A survey of recent developments"*. Computers and Operations Research 25(4), 279-302.
- [5] Brucker, P., Drexl, A., Mohring, R., Neumann, K., & Pesch, E. (1999). *"Resource-constrained project scheduling: Notation, classification, models, and methods"*, European Journal of Operational Research 112(1), 3-41.
- [6] Kolisch, R., & Padman, R. (2001). *"An integrated survey of deterministic project scheduling"*. Omega, the International Journal of Management Science 29, 249-272.
- [7] Neumann, K., Schwindt[p3], C., & Zimmermann, J. (2003). *"Project Scheduling with Time Windows & Scarce Resources"*. Springer.

- [8] Hartmann, S., & Briskorn, D. (2010). "A survey of variants and extensions of the resource-constrained project scheduling problem". *European Journal of Operational Research* 207(1), 1-14.
- [9] Fahmy, A.M., Hassan, T.M., & Bassioni, H. (2014). "A Dynamic Scheduling Model for Construction Enterprises". PhD thesis, School of Civil & Building Engineering, Loughborough University, UK.
- [10] Demeulemeester, E., & Herroelen, W. (1996). "Modelling setup times, process batches and transfer batches using activity network logic". *European Journal of Operational Research* 89, 355-365.
- [11] Klein, R., & Scholl, A. (1999). "Progress: Optimally solving the generalized resource constrained project scheduling problem". *Mathematical Methods of Operations Research* 52(3), 467-488.
- [12] Chassiakos, A., Sakellariopoulos, S. (2005). "Time-cost optimization of construction projects with generalized activity constraints". *Journal of Construction Engineering and Management* 131, 1115-1124.
- [13] Vanhoucke, M. (2006). "Work continuity constraints in project scheduling". *Journal of Construction Engineering and Management* 132, 14-25.
- [14] Demeulemeester, E., & Herroelen, W. (1996). "An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem". *European Journal of Operational Research* 90(2), 334-348.
- [15] Brucker, P., Knust, S. (2001). "Resource-constrained project scheduling and timetabling". *Lecture Notes in Computer Science* 2079, 277-293.
- [16] Ballestín, F., Valls, V., & Quintanilla, S. (2008). "Pre-emption in resource-constrained project scheduling". *European Journal of Operational Research* 189(3), 1136-1152.
- [17] Franck, B., Neumann, K., & Schwindt, C. (2001). "Project scheduling with calendars". *OR Spectrum*, 23(3), 325-334.
- [18] Lamothe, J., Marmier, F., Dupuy, M., Gaborit, P., & Dupont, L. (2016). "Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints". *Computers & Operations Research* 39 (6), 1236-1244.
- [19] Kreter, S., Rieck, J., & Zimmermann, J. (2016). "Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars". *European Journal of Operational Research* 251 (2), 387-403.
- [20] Elmaghraby, S. (1977). "Activity networks: Project Planning and Control by Network Models". Wiley, New York.
- [21] Kolisch, R., & Drexel, A. (1997). "Local search for non-preemptive multi-mode resource constrained project scheduling". *IIE Transactions* 29, 987-999.

- [22] Hartmann, S. (2001). *"Project scheduling with multiple modes: A genetic algorithm"*. Annals of Operations Research 102, 111-135.
- [23] Alcaraz, J., Maroto, C., & Ruiz, R. (2003). *"Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms"*. Journal of the Operational Research Society 54, 614-626.
- [24] Neumann, K., & Zimmermann, J. (2002). *"Project scheduling with inventory constraints"*. Mathematical Methods of Operations Research 56, 513-533.
- [25] Deckro, R., Hebert, J., Verdini, W., Grimsrud, P., & Venkateshwar, S. (1995). *"Nonlinear time/cost tradeoff models in project management"*. Computers and Industrial Engineering 28 (2), 219-229.
- [26] Demeulemeester, E., de Reyck, B., & Herroelen, W. (2000). *"The discrete time/resource trade-off problem in project networks – A branch-and-bound approach"*. IIE Transactions 32, 1059-1069.
- [27] Ranjbar, M., & Kianfar, F. (2007). *"Solving the discrete time/resource trade-off problem in project scheduling with genetic algorithms"*. Applied Mathematics and Computation 191(2), 2007 451-456.
- [28] Demeulemeester, E., de Reyck, B., Foubert, B., Herroelen, W., & Vanhoucke, M. (1998). *"New computational results on the discrete time/cost trade-off problem in project networks"*. Journal of the Operational Research Society 49, 614-626.
- [29] Ranjbar, M., Kianfar, F., & Shadrokh, S. (2008). *"Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm"*. Applied Mathematics and Computation 196(2), 879-888.
- [30] Kimms, A. (2001). *"Maximizing the net present value of a project under resource constraints using a lagrangian relaxation based heuristic with tight upper bounds"*. Annals of Operations Research 102 (1-4), 221-236.
- [31] Mika, M., Waligóra, G., & Weglvarz, J. (2005). *"Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models"*, European Journal of Operational Research 164 (3), 639-668.
- [32] Padman, R., & Zhu, D. (2006). *"Knowledge integration using problem spaces: A study in resource-constrained project scheduling"*. Journal of Scheduling 9 (2), 133-152.
- [33] Icmeli-Tukel, O., & Rom, W.O. (1997). *"Ensuring quality in resource constrained project scheduling"*. European Journal of Operational Research 103 (3), 483-496.
- [34] Mehta, S. V., & Uzsoy, R. (1999). *"Predictable scheduling of a single machine subject to breakdowns"*. International Journal of Computer Integrated Manufacturing 12(1), 15-38.
- [35] Vieira, G.E., Hermann, J.W., & Lin, E. (2003). *"Rescheduling manufacturing systems: a framework of strategies, policies and methods"*. Journal of Scheduling 6(1), 36-92.

- [36] Frederickson, G. N., & Solis-Oba, R. (2006). *"Efficient algorithms for robustness in resource allocation and scheduling problems"*. Theoretical Computer Science 352(1-3), 250-265.
- [37] Hwang, C., & Masud[p4], A. (1979). *"Multi-objective decision making, methods and applications. A state of the art survey"*. Lecture notes in economics and mathematical systems, Springer-Verlag.
- [38] Slowinski, R. (1981). *"Multi-objective network scheduling with efficient use of renewable and nonrenewable resources"*. European Journal of Operational Research 7, 265-273.
- [39] Ballestín, F., & Blanco, R. (2011). *"Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems"*. Computers & Operations Research 38, 51-62.
- [40] Tung, L., Li, L., & Nagi, R. (1999). *"Multi-objective scheduling for the hierarchical control of flexible manufacturing systems"*. The International Journal of Flexible Manufacturing Systems, 11, 379-409.
- [41] Hsu, T., Dupas, R., & Jolly, D., Goncalves, G. (2002). *"Evaluation of mutation heuristics for the solving of multiobjective flexible job shop by an evolutionary algorithm"*. In: Proceedings of the 2002 IEEE international conference on systems, man and cybernetics (pp. 6-9).
- [42] Kacem, I., Hammadi, S., & Borne, P. (2002). *"Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems"*. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 32(1), 1-13.
- [43] Kacem, I., Hammadi, S., & Borne, P.,(2002). *"Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic"*. Mathematics and Computers in Simulation 60, 245-276.
- [44] Loukil, T., Teghem, J., & Tuytens, D. (2005). *"Solving multi-objective production scheduling problems using metaheuristics"*. European Journal of Operational Research 161, 42-61.
- [45] Xia, W., & Wu, Z. (2005). *"An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems"*. Computers & Industrial Engineering 48, 409-425.
- [46] Herroelen, W., & Leus, R. (2005). *"Project scheduling under uncertainty: Survey and research potentials"*. European Journal of Operational Research 165(2), 289-306.
- [47] Aytug, H., Lawley, M.A., McKay, K., Mohan, S. & Uzsoy, R. (2005). *"Executing production schedules in the face of uncertainties: A review and some future directions"*. European Journal of Operational Research, 161(1), 86-110.
- [48] Ouelhadj, D., & Petrovic, S. (2009). *"A survey of dynamic scheduling in manufacturing systems"*. Journal of Scheduling 12, 417-431.

- [49] Kelley, J.E., Jr. (1963). *"The critical-path method: Resources planning and scheduling"*. In: J.F. Muth and G.L. Thompson (Eds.). Industrial Scheduling, Prentice-Hall, Englewood Cliffs. NJ, 347-365.
- [50] Kolisch, R. (1996). *"Efficient priority rules for the resource-constrained project scheduling problem"*. Journal of Operations Management 14 (3), 179-192.
- [51] Herroelen, W.S. (2005). *"Project scheduling-theory and practice"*. Production and Operations Management 14(4), 413-432.
- [52] Kolisch, R. (1996). *"Serial and parallel resource-constrained project scheduling methods revisited: theory and computation"*. European Journal of Operational Research 90 (2), 320-333.
- [53] Bedworth, D.D., & Bailey, J.E. (1982), *"Integrated Production Control Systems - Management, Analysis, Design"*. Wiley, New York.
- [54] Chen, R.M. (2011). *"Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem"*. Expert Systems with Applications 38, 7102-7111.
- [55] Li, K. Y., & Willis, R. J. (1992). *"An iterative scheduling technique for resource constrained project scheduling"*. European Journal of Operational Research 56, 370-379.
- [56] Valls, V., Ballestin, F., & Quintanilla, M.S. (2005). *"Justification and RCPSP: A technique that pays"*. European Journal of Operational Research 165, 375-386.
- [57] Fahmy, A.M., Hassan, T.M., & Bassioni, H. (2014). *"Improving RCPSP solutions quality with Stacking Justification – Application with particle swarm optimization"*. Expert Systems with Applications 41(1), 5870-5881.
- [58] Holland, H.J. (1975). *"Adaptation in Natural and Artificial Systems"*. The University of Michigan Press, Ann Arbor.
- [59] Goldberg, David (1989). *"Genetic Algorithms in Search, Optimization and Machine Learning"*. Addison-Wesley Professional, Boston, MA.
- [60] Shadrokh, S., & Kianfar, F. (2007). *"A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty"*. European Journal of Operational Research 181, 86-101.
- [61] Cheng, R., Gen, M., & Tsujimura Y. (1996). *"A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies"*. Computers & Industrial Engineering 30 (4), 983-997.
- [62] Cheng, R., Gen, M., & Tsujimura Y. (1999). *"A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies"*. Computers & Industrial Engineering 36 (2), 343-364.

- [63] Wall, M. B. (1996). "A Genetic Algorithm for Resource-Constrained Scheduling". PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, USA.
- [64] Kolisch, R., & Hartmann, S. (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update". *European Journal of Operational Research* 174, 23-37.
- [65] Hartmann, S. (1998). "A competitive genetic algorithm for resource-constrained project scheduling". *Naval Research Logistics* 45, 733-750.
- [66] Hartmann, S. (2002). "A self-adapting genetic algorithm for project scheduling under resource constraints". *Naval research Logistics* 49, 433-448.
- [67] Alcaraz, J., & Maroto, C. (2001). "A robust genetic algorithm for resource allocation in project scheduling". *Annals of Operations Research* 102, 83-109.
- [68] Alcaraz, J., Maroto, C., & Ruiz, R. (2004). "Improving the performance of genetic algorithms for the RCPS problem". In: *Proceedings of the Ninth International Workshop on Project Management and Scheduling*, Nancy, 40-43.
- [69] Valls, V., Ballestin, F., & Quintanilla, M.S. (2008). "A hybrid genetic algorithm for the resource-constrained project scheduling problem". *European Journal of Operational Research* 185(2), 495-508.
- [70] Kennedy, J., & Eberhart, R. (1995). "Particle swarm optimization". *Proceedings of the 1995 IEEE international conference on neural networks*, 4, 1942-1948.
- [71] Shi, Y., & Eberhart, R.C. (1998). "A modified particle swarm optimizer". *Proceedings of IEEE Congress on Evolutionary Computation* 1998, 69-73.
- [72] Bratton, D., & Kennedy, J. (2007). "Defining a standard for particle swarm optimization". *Proceedings of IEEE swarm intelligence symposium, SIS 2007*, 120-127.
- [73] Zhang, C., Sun, J., Zhu, X., & Yang, Q. (2008). "An improved particle swarm optimization algorithm for flowshop scheduling problem". *Information Processing Letters* 108(4), 204-209.
- [74] Zhang, H., Li, H., & Tam, C. M. (2005). "Particle swarm optimization-based schemes for resource-constrained project scheduling". *Automation in Construction* 14, 393-404.
- [75] Dorigo, M. (1992). "Optimization, Learning and Natural Algorithms". PhD thesis, Politecnico di Milano, Italy.
- [76] Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). "Ant system for job-shop scheduling". *Belgian Journal of Operations Research, Statistics and Computer Science* 34 (1), 39-53.
- [77] Stützle, T., & Hoos, H.H. (1997). "The max-min ant system and local search for the traveling salesman problem". *Proceedings of ICEC&97*, 309-314.

- [78] Stützle, T., & Hoos, H.H. (2000). "MAX MIN Ant System". *Future Generation Computer Systems* 16, 889-914.
- [79] Shyu, S.J., Lin, B.M.T., & Hsiao, T.S. (2004). "Ant colony optimization algorithm for the cell assignment problem in PCS networks". *Computers & Operations Research* 33 (6), 1713-1740.
- [80] Xing, L.N., Chen, Y.W., Wang, P., Zhao, Q.S., & Xiong, J. (2010). "A knowledge-based ant colony optimization for flexible job shop scheduling problems". *Applied soft Computing*, 10 (3), 888-896.
- [81] Merkle, D., Middendorf, M., & Schneck, H. (2002). "Ant colony optimization for resource-constrained project scheduling". *IEEE Transactions on Evolutionary Computation* 6, 333-346.
- [82] Lo, S. T., Chen, R. M., Huang, Y. M., & Wu, C. L. (2008). "Multiprocessor system scheduling with precedence and resource constraints using an enhanced ant colony system". *Expert Systems with Applications* 34(3), 2071-2081.
- [83] Bauer, A., Bullnhermer, B., Hartl, R.F., & Strauss, C. (1999). "An ant colony optimization approach for the single machine total tardiness problem". *Proceedings of the 1999 Congress on Evolutionary Computation*. Washington, DC, USA, 1445-1450.
- [84] Merkle, D., & Middendorf, M. (2003). "An ant algorithm with a new pheromone evaluation rule for total tardiness problems". *Applied Intelligence* 18 (1), 105-111.
- [85] Stützle, T., & Dorigo, M. (2002). "The Ant Colony Optimization metaheuristic: algorithms, applications, and advances". In: F. Glover, G. Kochenberger (Eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, Norwell, MA.
- [86] Glover, F. (1986). "Future Paths for Integer Programming and Links to Artificial Intelligence". *Computers and Operations Research*, 13 (5), 533-549.
- [87] Thomas, P. R., & Salhi, S. (1998). "A tabu search approach for the resource constrained project scheduling problem". *Journal of Heuristics*, 4(2), 123-139.
- [88] Baar, T., Brucker, P., & Knust, S. (1998). "Tabu-search algorithms and lower bounds for the resource-constrained project scheduling problem". In: S. Voss, S. Martello, I. Osman, C. Roucairol (Eds.), *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Dordrecht, 1-8.
- [89] Nonobe, K., & Ibaraki, T. (2002). "Formulation and Tabu search algorithm for the resource constrained project scheduling problem (RCPSP)". In: Ribeiro, C.C., Hansen, P. (Eds.), *Essays and Surveys in Metaheuristics. Operations Research/Computer Science Interfaces Series*, Kluwer Academic Publishers 15, 557-588
- [90] Ben-Daya, M., & Al-Fawzan, M. (1998). "A tabu search approach for the flow shop scheduling problem". *European Journal of Operational Research*, 106 (2-3), 226-253.

- [91] Wojciech, B., Pempera, J., & Smutnicki, C. (2013). *"Parallel tabu search algorithm for the hybrid flow shop problem"*. Computers & Industrial Engineering 65 (3), 466-474.
- [92] Jia, S., & Hu, Z.H. (2014). *"Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem"*. Computers & Operations Research 47, 11-26.
- [93] Kirkpatrick, S, Gelatt, C. D, & Vecchi, M. P. (1983). *"Optimization by Simulated Annealing"*. Science. 220(4598), 671-680.
- [94] Rutenbar, R. A. (1989). *"Simulated annealing algorithms: An overview"*. Circuits and Devices Magazine, IEEE 5, 19-26.
- [95] Bouleimen, K., & Lecocq, H. (2003). *"A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode versions"*. European Journal of Operational Research 140(2), 268-281.
- [96] Dai, M., Tang, D., Giret, A., Salido, M.A., & Li, W.D. (2013). *"Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm"*. Robotics and Computer-Integrated Manufacturing 29 (5), 418-429.
- [97] Foo, S. Y, & Takefuji, Y. (1988). *"Stochastic neural networks for solving job-shop scheduling: Part 1"*. In: IEEE International Conference on Neural Networks 1988, San Diego, CA, USA, 275-282.
- [98] Cedimoglu, I. H. (1993) *"Neural networks in shop floor scheduling"*. Ph.D. Thesis, School of Industrial and Manufacturing Science, Cranfield University, UK.
- [99] Sim, S. K, Yeo, K. T, & Lee, W. H. (1994). *"An expert neural network system for dynamic job-shop scheduling"*. International Journal of Production Research 32(8), 1759-1773.
- [100] Debels, D., De Reyck, B., Leus, R., & Vanhoucke, M. (2006). *"A hybrid scatter search/ electromagnetism meta-heuristic for project scheduling"*. European Journal of Operational Research 169, 638-653.
- [101] Chang, P.C., Chen, S.H., & Fan, C.Y. (2009). *"A hybrid electromagnetism-like algorithm for single machine scheduling problem"*. Expert Systems with Applications 36 (2-1), 1259-1267.
- [102] Khalili, M., & Tavakkoli-Moghaddam, R. (2012). *"A multi-objective electromagnetism algorithm for a bi-objective flowshop scheduling problem"*. Journal of Manufacturing Systems 31 (2), 232-239.
- [103] Tormos, P., & Lova, A. (2001). *"A competitive heuristic solution technique for resource-constrained project scheduling"*. Annals of Operations Research 102, 65-81.
- [104] Tormos, P., & Lova, A. (2003). *"An efficient multi-pass heuristic for project scheduling with constrained resources"*. International Journal of Production Research 41(5), 1071-1086.
- [105] Low, C.S., Sivakumar, M.A, & Gay, K.L. (2007). *"Using a Bee Colony Algorithm for Neighborhood Search in Job Shop Scheduling Problems"*. In: 21st European Conference on Modelling and Simulation ECMS.

- [106] Wong, L.P., Puan, C.Y., Low, M. Y., & Chong C. S. (2008). *"Bee Colony Optimization algorithm with Big Valley landscape exploitation for Job Shop Scheduling problems"*. In: Simulation Conference, 2008, WSC.
- [107] Kochetov, Y., & Stolyar, A. (2003). *"Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem"*. In: Proceedings of the 3rd International Workshop of Computer Science and Information Technologies, Russia.
- [108] Niu, Q., Jiao, B., & Gu, X. (2008). *"Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time"*. Applied Mathematics and Computation 205 (1), 148-158.
- [109] Chen, P.H., & Shahandashti, S. M. (2009). *"Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints"*. Automation in Construction 18 (4), 434-443.
- [110] Moslehi, G., & Mahnam, M. (2011). *"A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search"*. International Journal of Production Economics, 129 (1), 14-22.
- [111] Deane, J. (2012). *"Hybrid genetic algorithm and augmented neural network application for solving the online advertisement scheduling problem with contextual targeting"*. Expert Systems with Applications 39 (5), 5168-5177.



---

# Survey of Meta-Heuristic Algorithms for Deep Learning Training

---

Zhonghuan Tian and Simon Fong

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63785>

---

## Abstract

Deep learning (DL) is a type of machine learning that mimics the thinking patterns of a human brain to learn the new abstract features automatically by deep and hierarchical layers. DL is implemented by deep neural network (DNN) which has multi-hidden layers. DNN is developed from traditional artificial neural network (ANN). However, in the training process of DL, it has certain inefficiency due to very long training time required. Meta-heuristic aims to find good or near-optimal solutions at a reasonable computational cost. In this article, meta-heuristic algorithms are reviewed, such as genetic algorithm (GA) and particle swarm optimization (PSO), for traditional neural network's training and parameter optimization. Thereafter the possibilities of applying meta-heuristic algorithms on DL training and parameter optimization are discussed.

**Keywords:** deep learning, meta-heuristic algorithm, neural network training, nature-inspired computing algorithms, algorithm design

---

## 1. Introduction

Deep learning (DL) is a branch of machine learning. Based on a set of algorithms, DL attempts to model high-level abstractions in data by using multiple processing layers with complex structures. Developed by Professor Hinton [1] in 2006, DL now is becoming the most prevalent research area in machine learning. DL is a collective concept of a series of algorithms and models including but not limited to convolutional neural networks (CNNs), deep belief networks (DBNs), restricted Boltzmann machines (RBMs), deep Boltzmann machines (DBM), recursive auto encoders and deep representation respectively, regarding the problem to be

solved. The most famous models for machine learning are RBM and DBN and CNN for image classification.

DL is the so-called “Deep” towards those widely used “shallow learning” algorithms such as support vector machine (SVM), boosting and maximum entropy method. Those shallow learning extracts feature mostly by artificial sampling or empirical sampling, thus the model or network will learn a non layer structure feature. On the contrary, DL learns raw data layer by layer by transforming data from raw feature space to transformed feature space. Additionally, deep structure can learn and approach nonlinear function. All these advantages are beneficial to classification and feature visualization [2, 25].



**Figure 1.** Error rate of Image Net competition.

To process big data and large scale dataset, DL has been widely used in many research areas and industrial areas including pattern recognition, image classification and natural language processing (NLP). For instance, in Image Net large-scale visual recognition challenge (ILSVRC) [3], DL and convolutional neural network are first implemented in image classification and made a great success. As shown in **Figure 1**, the error rate declines from 25 to 15% when DL and CNN are implemented. From then on, “DL + big data + GPU” is integrated into traditional image classification application, and companies such as Google and Baidu have successfully updated their image searching production by DL’s implementation [4].

Though DL outperforms those shallow learning methods and it has been widely adopted by the R&D industry, its application and implementation still have some shortcomings. A large deep neural network (DNN) may have millions of parameters and is mostly trained by contrastive divergence (CD) learning algorithm which is iterative and known to be time consuming [5]. The most significant problem is that when facing very large scaled data the DNN will take several days or even months to learn even though the greedy search strategy is proposed. In this case, many companies and researchers are trying to improve hardware capabilities and applying high-processing facilities such as GPU and parallel computing. Others are focusing on using other training algorithms to substitute traditional CD to speed up the training process.

Meta-heuristic algorithms aim to find global or near-optimal solutions at a reasonable computational cost. In terms of meta-heuristic, the words of “meta” and “heuristic” are Greek, where “meta” is “higher level” or “beyond” and heuristics means “to find”, “to know”, “to guide an investigation” or “to discover”. Heuristics are methods to find good (near-) optimal solutions in a reasonable computational cost without guaranteeing feasibility or optimality. In other words, meta-heuristics are a set of intelligent strategies to enhance the efficiency of heuristic procedures [6].

Meta-heuristic algorithms usually rely on different agents such as chromosome (genetic algorithm [GA]), particles (particle swarm optimization [PSO]), and firefly (firefly algorithm), searching iteratively to find the global optimum or near-global optimum. Many strategies such as evolutionary strategy, social behavior and information transition are implemented to guarantee the whole population will move towards global optimum iteratively and protect from falling in local optimum.

Traditional artificial neural network (ANN) used to have multilayer feed-forward neural network structure and use back-propagation (BP) to modify the weight. The mainly used strategy is gradient descent (GD). Meta-heuristic has been successfully implemented on traditional neural network to speed up the training process by substituting the GD strategy with iterative evolutionary strategy or swarm intelligence strategy [5, 7–10].

Gudise and Venayagamoorthy [5] compared feed forward with PSO and feed forward with BP, and the results show that feed forward with PSO is better than BP in terms of nonlinear function.

Leung et al. [10] presents the tuning of the structure and parameters of a neural network using an improved GA. It will also be shown that the improved GA performs better than the standard GA based on some benchmark test functions.

Juang [8] proposed a new evolutionary learning algorithm based on a hybrid of GA and PSO called HGAPSO. In each epoch, the upper-half of the GA population are defined as elites and the rest will be discarded. Enhanced by PSO and GA, the elites will form the next generation of GA. The hybrid method outperforms PSO and GA in recurrent or fuzzy neural network.

Meissner et al. [9] used optimized particle swarm optimization (OPSO) to accelerate the training process of neural network. The main idea of OPSO is to optimize the free parameters of the PSO by having swarms within a swarm. Applying the OPSO method to neural network training with the aim to build a quantitative model, OPSO approach yields parameter combinations improving overall optimization performance.

Zhang et al. [7] proposed a hybrid algorithm of PSO plus BP for neural network training. By utilizing the advantage of PSO’s global searching ability as well as BP’s deep search ability, the hybrid algorithm has a very good performance in both convergent speed and convergent accuracy.

Structures of DL model are similar to the traditional ANN, some modifications are implemented for better learning ability. For instance, the CNN is a traditional ANN modified with pooling procession and the structure of RBM is an undirected graph or a bidirectional neural

network. DL model shares similar model with neural network and may take different training algorithm instead of GD strategy.

In this survey, the relevant work of applying meta-heuristic algorithms for ANN’s training is reviewed. In addition, the structure and working process of RBM are analyzed and as the basic structure of DL model, RBM’s training process is introduced. We survey the possibility of implementing meta-heuristic algorithms on RBM’s parameter training process.

The rest of the sections is organized as follows: meta-heuristic algorithms including GA and PSO are introduced in Section 2. ANN is introduced in Section 3. The implementation of meta-heuristic on ANN is introduced in Section 4. DL and RBM are introduced in Section 5. Some conclusion and discussion are drawn in Section 6.

2. Meta-heuristic algorithm

Meta-heuristic is a collective concept of a series of algorithms including evolutionary algorithm, the most famous one is GA [11], naturally inspired algorithm, the most prevalent one is PSO [12], trajectory algorithm, such as Tabu search [13], and so on. The classification of meta-heuristic is shown in **Figure 2**. In this paper, I mainly focus on surveying GA and PSO because these two are the most prevalent as well as most widely implemented. GA and its working process are introduced in Section 2.1 and PSO is introduced in Section 2.2, respectively.

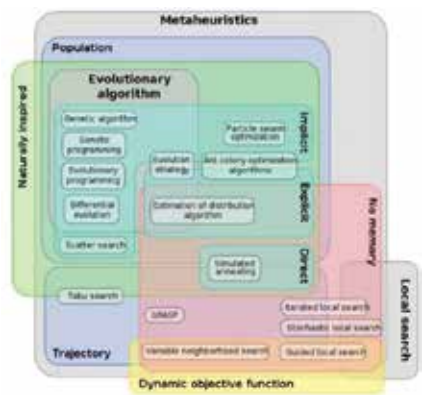


Figure 2. Classification of meta-heuristic algorithm.

2.1. Genetic algorithm

Founded in 1975 by Professor Holland, GA sets up an evolution model that simulates Darwinian genetic selection and the natural elimination process [11]. Chromosomes carry information and the operation of crossover and mutation on chromosome guarantees the whole algorithm could find the global optimum or near-global optimum iteratively. Crossover operation keeps the whole population moving towards the global optimum through giving better chromosome higher chance for propagation. Mutation operation keeps the whole

population's diversity and avoids population falling in local optimum easily. The main working process could be summarized as follows:



**Figure 3.** Chromosome image in biology.

Step 1. Chromosomes which carry information are randomly generated.

Step 2. All chromosomes are evaluated with fitness functions to calculate fitness values of each as  $fitness_i$ .

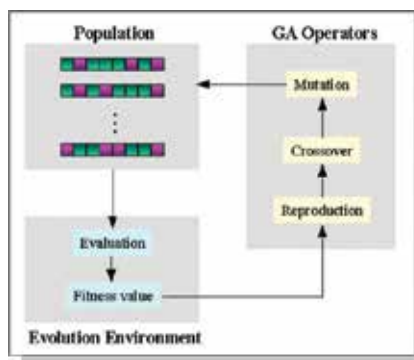
Step 3. Based on Russian roulette strategy, chromosomes are randomly chosen as parents for next generation.

Step 4. Parents do crossover operations to get next generations.

Step 5. Do mutation operation on each chromosome.

Step 6. Back to Step 2 until meeting the stop criteria or exceeding pre-set iteration number.

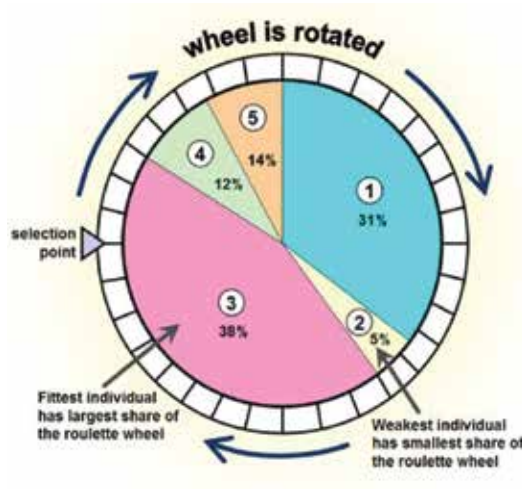
A biological picture of chromosome is shown in **Figure 3**. The working flow of GA is shown in **Figure 4**.



**Figure 4.** GA working flow.

### 2.1.1. Russian roulette strategy

Russian roulette strategy is designed for choosing the parents of the next generation, as shown in **Figure 5**. Each chromosome has a possibility  $p_i$  of being chosen as parents.



**Figure 5.** Russian roulette strategy.

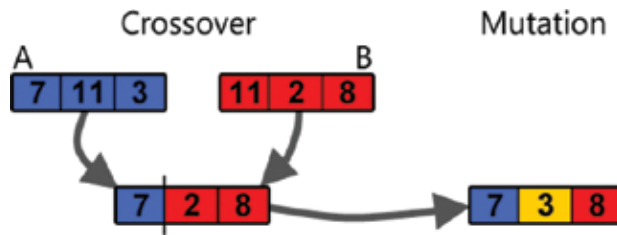
$p_i$  is calculated as below for each chromosome  $i$ :

$$p_i = \frac{fitness_i}{\sum_{i=1}^n fitness_i}$$

This strategy keeps the idea that the chromosome with higher fitness value will have higher possibility of being chosen thus make sure better chromosome's information will not be missed or has few possibilities of missing in the next generation. In the meantime, it also allows those chromosomes without better fitness value to transmit their information to the next generation [11].

### 2.1.2. Crossover and mutation

**Figure 6** shows us a typical crossover operation and mutation operation. The most common way of applying crossover operation is selecting a site, cutting up each chromosome and reconstructing both to get next generation. If their fitness value is better than their parents, submit as next generation otherwise keep their parents as next generation. Furthermore, regarding different real problem to be solved, different crossover operation such as multisite crossover would be used and these will be discussed later.



**Figure 6.** Crossover and mutation.

**Figure 6** also shows us the typical mutation operation in GA. The most common way of applying mutation is to randomly select some site of chromosome and randomly change it with any possible value regardless of the fitness value will be better or not. Of course the mutation operation may make one chromosome “worse” while mutation operation can keep the population’s diversity and avoid falling in local optimum.

### 2.1.3. Fitness function and genetic coding strategy

GA is first used to optimize complex continuous functions which are not able to calculate its mathematical global optimal location. Thus the design of chromosome itself, fitness function and crossover operation are very easy. However, when trying implementing GA on real problems, many types of problems such as discrete problem, time series problem and multi objective problem are faced. In this way, how to design the chromosome itself to represent the real problem better and how to calculate its fitness value when given chromosome need reconsidering. To solve the above troubles, genetic coding strategy is put forward which encodes real problem to chromosome and calculates its fitness value. When whole optimizing is finished, the best chromosome will be decoded to real problem’s expression and then analyzed and discussed why it is global best.

## 2.2. Particle swarm optimization

PSO is developed by Professor James Kennedy and Russell Eberhart [12]. The main difference of PSO is that it takes “social behaviors” and “memory” to guarantee whole population’s moving towards global optimum iteratively instead of evolutionary strategy.

PSO optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formula over the particle’s position  $x_i$  and velocity  $v_i$ . Each particle has memory and its movement is influenced by its local best known position  $pbest_i$  but is also guided toward the best known positions in the search-space, which are updated as better positions found by all particles called  $gbest$ .

The iterative formulas of PSO are

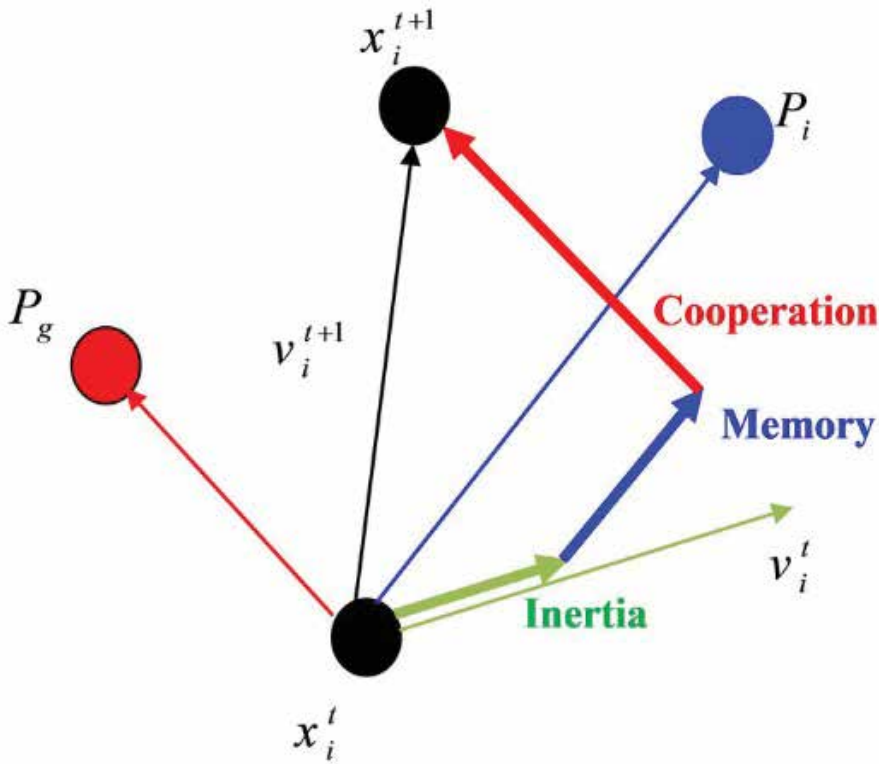
$$v_{i+1} = v_i * w + c_1 * rand * (pbest_i - x_i) + c_2 * rand * (gbest - x_i)$$

Velocity constraint ()

$$x_{i+1} = x_i + v_{i+1}$$

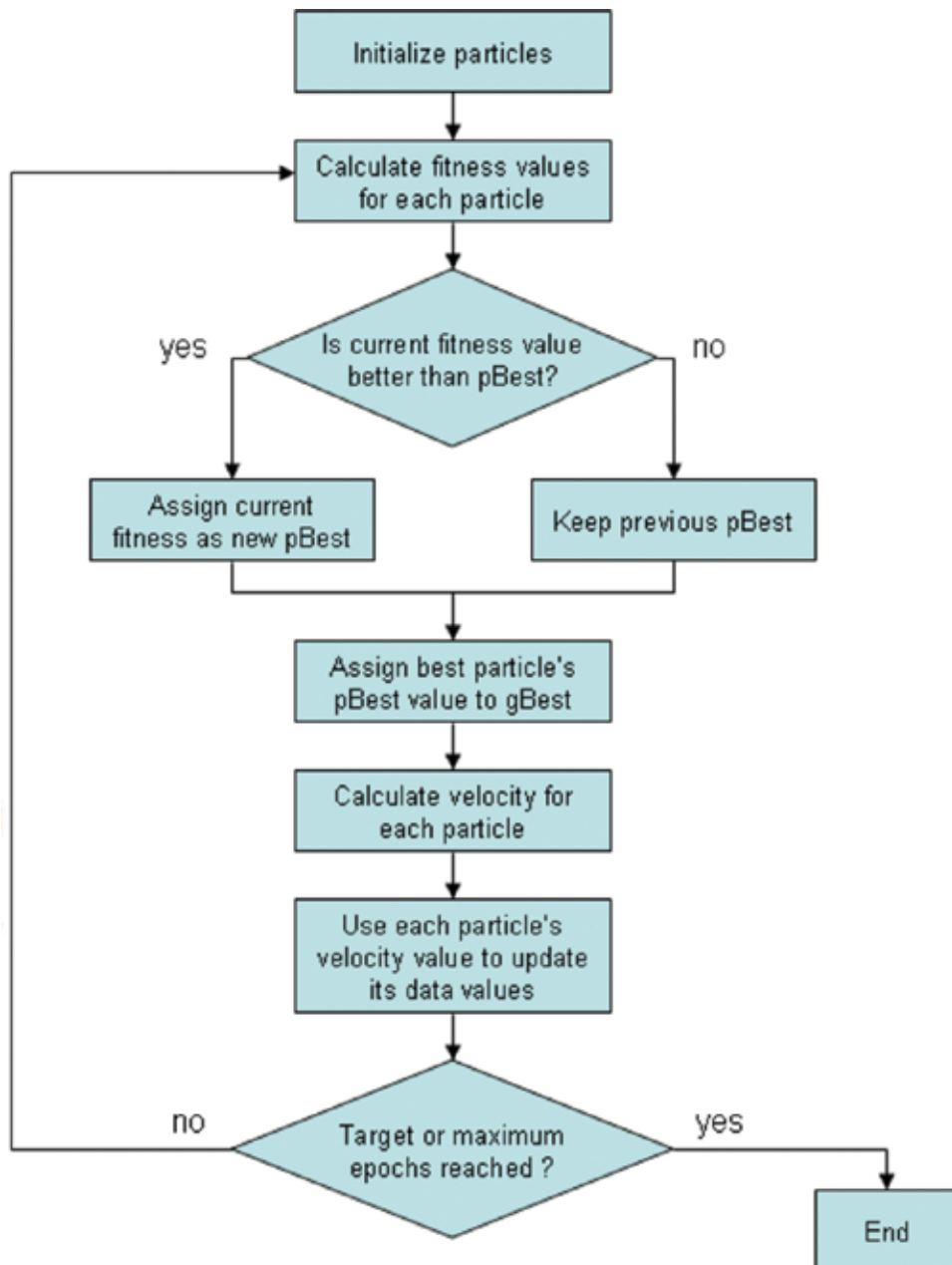
Search range constraint ()

$w$  is called the inertia weight that controls the exploration and exploitation of the search space because it dynamically adjusts velocity and is usually set as 0.8.  $c_1$  and  $c_2$  are called learning factors and usually set as 2.



**Figure 7.** Schematic of PSO.

In the iterative formula of PSO,  $v_i * w$  refers to the particle keeps its own search direction.  $c_1 * rand * (pbest_i - x_i)$  refers to the particle's movement is influenced by its own personal best record.  $c_2 * rand * (gbest - x_i)$  refers to the particle's movement is influenced by whole population's best record. To keep whole population will not premature or overfit, when implementing iteration process, a limitation of  $v_{max}$  and  $v_{min}$  is predefined to constraint particles' movement. Additionally, the search space is also predefined thus for each particle's location  $x_i$ , there are upper bound  $x_{max}$  and lower bound  $x_{min}$  to make sure each particle will not be out of range. A schematic diagram is shown in **Figure 7** and the working flow of PSO is shown in **Figure 8**.



**Figure 8.** Working flow of PSO.

PSO is prevalent because of its concise evolutionary strategy as well as its concise operations. Unlike GA's crossover and mutation operation and complex genetic coding strategy, PSO only needs randomly generated location  $x$  and velocity  $v$ . However, naïve PSO is designed for math function optimization which is a continuous problem. For application, researchers still need

to design own coding strategy to make sure PSO's iterative formulas are meaningful, especially for discrete problems. One example is applying PSO on graph optimization to solve some NP-hard problem such as travelling salesman problem (TSP) problem [14]. TSP asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? It is an NP-hard problem in combinatorial optimization, important in operations research and theoretical computer science.

With the development of meta-heuristic, researchers are beginning to try implementing meta-heuristic on TSP in case to get a "good enough" result in a reasonable computation time. To solve and adapt TSP to PSO, researchers innovate a concept of "swap operator" and "swap sequence" which gives the "+" operator a new meaning [14, 15].

### 3. Neural network

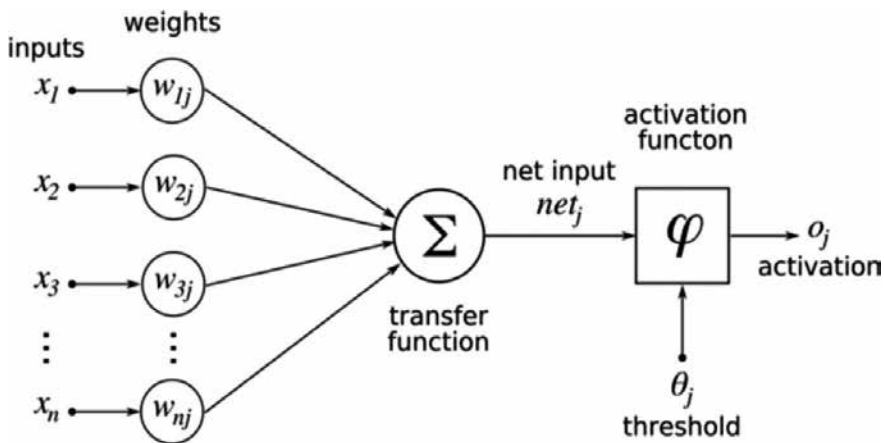
In machine learning and cognitive science, ANNs are a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. ANNs are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning [16]. The ANN was developed in the 1950s, there are almost three generations of neural networks: perceptron [17], feed-forward BP neural network [18] and spiking neural network (SNN) [19].

#### 3.1. Perceptron

Founded in the 1950s by Professor Rosenblatt, the basic perceptron is put forward as a probabilistic model to simulate human brain's working process when receiving information. **Figure 9** shows us a typical structure of perceptron that only has one layer, with multi-input neurons as  $(t_1, t_2, \dots, t_n)$  in which  $k$  is the total number of neurons. Their corresponding inputs are  $(x_1, x_2, \dots, x_n)$ . Each neuron has its own weight  $(w_{1j}, w_{2j}, \dots, w_{nj})$  in which  $j$  refers to the output layer. The net input  $net_j = \sum_{i=1}^n x_i * w_{ij}$  is processed by the activation  $\varphi$ . If  $\varphi(net_j) > \text{threshold } \theta_j$ , then the output neuron is activated and output  $o_j = 1$ , otherwise output neuron is not activated and output  $o_j = 0$ . The training process of perceptron is that suppose the desired output is  $o_d$  and actual output is  $o_a$ , then

$$\Delta w_{ij} = (o_d - o_a) * x_i$$

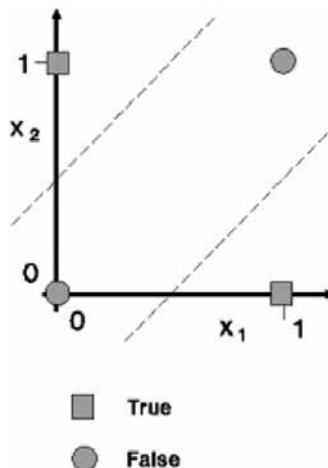
$$w_{ij}^{f+1} = w_{ij}^f + \Delta w_{ij}$$



**Figure 9.** Structure of perceptron.

By using these training strategies, when  $|o_d - o_a| < eps$ , in which *eps* is a predefined accuracy, i.e.,  $10^{-5}$ , the perceptron could be seen as mature. This training strategy is the ancestor of the widely used BP algorithm.

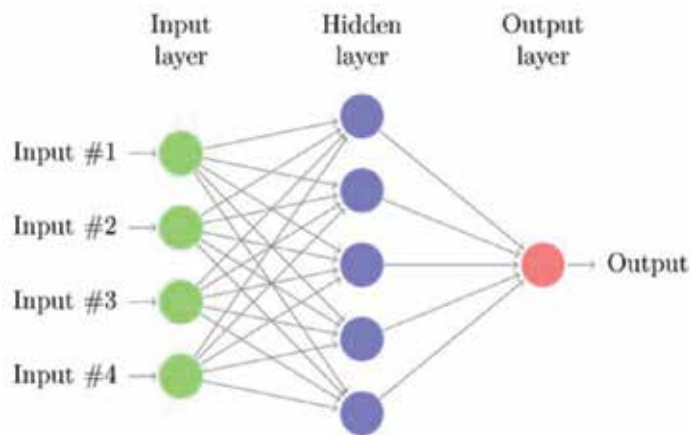
Geometrically, trained perceptron could be seen as a linear function. It can correctly classify the linear separable problem but not linear inseparable problem such as XOR problem (**Figure 10**). XOR problem contains two classes A and B while the diagonal belongs to the same class, i.e., data (0, 0) and data (1, 1) belong to class A while data (1, 0) and (0, 1) belong to class B. This will cause no lines or linear functions can separate these two categories correctly unless using curves or reflect the data to high-dimensional space [18].



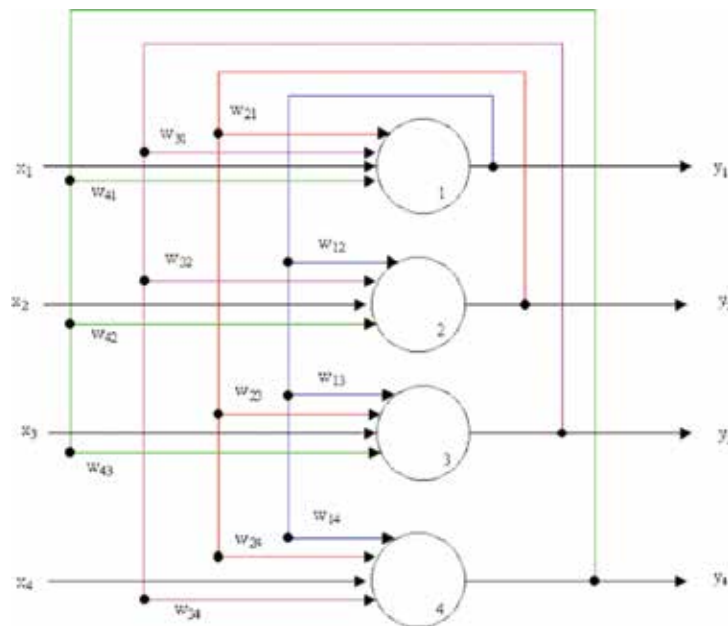
**Figure 10.** XOR problem.

### 3.2. Feed-forward neural network

**Figure 11** shows us a typical multilayer unidirectional feed-forward neural network and **Figure 12** shows us a typical recurrent neural network, Hopfield network. The most prevalent structure of three-layer neural network with input layer, hidden layer and output layer, every two neurons are connected with weight  $w$ .



**Figure 11.** Feed-forward neural network.



**Figure 12.** Hopfield neural network.

In a typical feed-forward neural network, the information is transmitted forward from input layer to hidden layer and then from hidden layer to output layer. It works similar to perceptron above, for each neuron  $j$  in hidden layer, calculate

$$y_j = \sum_{i=1}^n x_i * w_{ij}$$

Then for each neuron  $h$  in output layer, calculate

$$o_h = \sum_{i=1}^n y_i * w_{jh}$$

The output layer could have one or more neurons to represent the final result, different output layer structure may lead to different training time and prediction accuracy.

BP algorithm is the most prevalent supervised learning algorithm for neural network. BP's main idea is using GD algorithm to find the gradient-descent-most way to modify the weight in neural network. The modification starts from the output layer, then to hidden layer and input layer. Because the GD is based on the difference or error between the desired output and actual output, BP is also called error-BP method. Suppose desired output is  $o_d$  and actual output is  $o_a$ , the error function is defined as:

$$E = \frac{1}{2}(o_d - o_a)^2$$

Suppose the layers are input layer  $i$ , hidden layer  $j$  and output layer  $h$ , respectively, using partial derivative,

$$\Delta w_{ij} = \eta \frac{\partial E}{\partial w_{ij}}, \Delta w_{jh} = \eta \frac{\partial E}{\partial w_{jh}}$$

in which  $\eta$  is called learning rate, usually set as 0.01 or 0.1 according to the set of weight of the network. Weight  $w$  is usually set as a random number in  $[0, 1]$  in case the learning process is too long. Then, update the weight between each neurons,

$$w_{ih} = w_{ih} + \Delta w_{ih}, w_{hj} = w_{hj} + \Delta w_{hj}$$

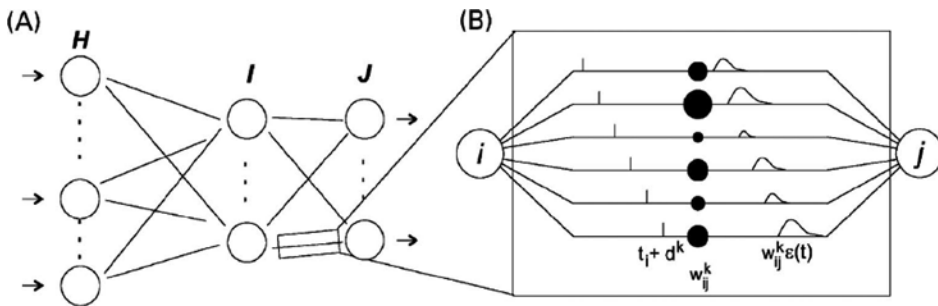
One epoch of training process is finished. GD will modify weight iteratively using the formula above till the whole network is trained mature.

Unlike perceptron, multilayer network can represent complex nonlinear function with hidden layers, thus can process XOR problems well. To some extent, multilayer network can represent any type of nonlinear function by using hidden layers. However, when implementing BP, if the depth of network is over 5, the error transmitted back to input layer or first hidden layer will be decayed significantly to almost 0 thus making the modification useless, so in real application, the depth of network is smaller than 5 [16].

### 3.3. Spiking neural network

SNN is recognized as the third generation of neural network. Maass [19] proved the spiking neuron and SNN has a powerful learning ability and information processing ability than traditional neural network. Put forward by Gerstner [20] in 1997, the SNN gives us a concise biological neuron model processed by temporal coding called spiking neuron model (SRM).

**Figure 13(A)** and **(B)** shows us the basic structure of SNN, the network structure and information transition of neural network, SNN is similar to using feed-forward multilayer neural network. However, the transition process is a little different with feed-forward neural network. Between neurons  $i$  and  $j$ , there are multiple delays ( $d_1, d_2, \dots, d_k$ ) and every link has its own weight ( $w_{ij}^1, w_{ij}^2, \dots, w_{ij}^k$ ) corresponding to each delay. Compared to traditional NN, SNN has more powerful learning ability and information processing ability.



**Figure 13.** (A) Feed-forward SNN and (B) connection of multiple delay.

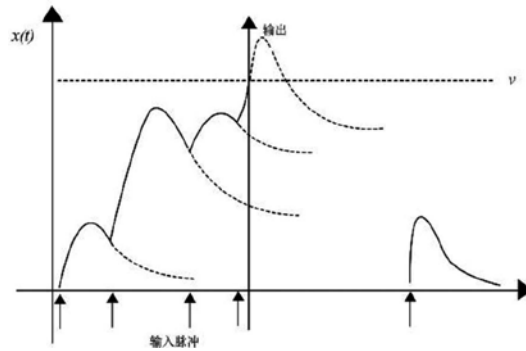
For each spiking neuron  $j$ , its internal state  $x_j(t)$  is

$$x_j(t) = \sum_{i \in \Gamma_j} \sum_k w_{ij}^k \varepsilon(t - t_i - d_{ij}^k)$$

Neuron  $i$  is its predecessor and weight  $w_{ij}^k$  and delay  $d_{ij}^k$  are the synaptic connection between neuron  $i$  and  $j$ .  $t_i$  is the  $i$ th input from presynaptic neuron. If  $x_i(t) \geq v$ ,  $v$  is pre-set threshold, the neuron is spiked and could transmit spike to other neurons.  $\varepsilon(t)$  is called spiking response function,

$$\varepsilon(t) = \begin{cases} \frac{t}{\tau} e^{1-\frac{t}{\tau}}, & t > 0 \\ 0, & t \leq 0 \end{cases}$$

$\tau$  is the membrane potential decay time constant that determines the rise and decay time of the postsynaptic potential (PSP). **Figure 14** shows us how one spiking neuron begins to receive input pulse step by step and become spiked. Each neuron could begin to transmit information to its posterity only when this neuron is spiked. SNN implements temporal coding strategy which means the whole network is only based on one variable  $t$  which is initially set as 0. Then  $t$  increment 0.01 epoch by epoch until the output layer is spiked. The final output is the output layer spiking time  $t$ .



**Figure 14.** Process of neuron spike when receiving input spike.

Owing to the multiple delay connection, one SNN can achieve times of computation scale and ability than ANN in the same multilayer structure. With temporal coding, the whole network has only one variable  $t$ , thus the iterative process is simpler than ANN.

For supervised learning algorithm of SNN, Bohte et al. [21] first time give us a proved error-propagation supervised learning algorithm called spikeprop algorithm which is also based on GD. Adeli [22] revised the spikeprop algorithm and put forward another supervised learning algorithm quickprop.

### 3.4. Neural network encoding strategy

When dealing with different types of data for classification or prediction, encoding strategy similar to genetic coding needs being implemented to transform real problem to neural network format. In addition, because of the longtime of neural network training, some preprocessing method needs to be taken for extracting significant information or features thus accelerate neural network learning.

For instance, dealing with image classification, feature extraction method need being taken as preprocessing for neural network learning and the famous one is CNN. Furthermore, when

using neural network for speech recognition, feature extraction is also a need for transforming continuous speech data into discrete neural network input.

## 4. Applying meta-heuristic algorithm on neural network training

The main idea of applying meta-heuristic on neural network training is using meta-heuristic algorithm instead of GD algorithm modifying weight training. That is from

$$jw_{t+1} = w_t + GD \Rightarrow w_{t+1} = w_t + Meta$$

By using meta-heuristic algorithm's global optimum searching ability, researchers are aiming to train neural network faster than traditional GD algorithm. In the following part, Section 1 reviewed four researchers' work on implementing meta-heuristic on neural network training including GA on NN, PSO on NN and hybrid GA and PSO on NN.

### 4.1. GA on neural network

Leung et al. [10] first tried implementing GA on neural network training. Their work was published in IEEE Transactions on Neural Network, 2003.

An improved GA is put forward by Leung. Crossover operations, mutation operations and fitness function of GA are all redefined by Professor Leung. Firstly, when chromosomes  $p_1$  and  $p_2$  do crossover operation, four possible offsprings will be generated and one with the biggest fitness value will be chosen as offspring. The four possible crossover offsprings  $os_1$  to  $os_4$  are generated as regulations listed below:

$$os_c^1 = [os_1^1, os_2^1, \dots, os_n^1] = \frac{p_1 + p_2}{2}$$

$$os_c^2 = [os_1^2, os_2^2, \dots, os_n^2] = p_{\max}(1-w) + \max(p_1, p_2)w$$

$$os_c^3 = [os_1^3, os_2^3, \dots, os_n^3] = p_{\min}(1-w) + \min(p_1, p_2)w$$

$$os_c^4 = [os_1^4, os_2^4, \dots, os_n^4] = \frac{(p_{\min} + p_{\max})(1-w) + (p_1 + p_2)w}{2}$$

$p_{\max} = [para_{\max}^1, para_{\max}^1, \dots, para_{\max}^n]$ ,  $p_{\min} = [para_{\min}^1, para_{\min}^1, \dots, para_{\min}^n]$  are calculated respectively. For instance,  $\text{Max}([1, -1, 4], [-3, 3, 2]) = [1, 3, 4]$  and  $\text{Min}([1, -1, 4], [-3, 3, 2]) = [-3, -1, 2]$ .

Secondly, mutation operations are redefined. The regulations are given below

$$os = \max(\text{fitness}(os_c^1), \text{fitness}(os_c^2), \text{fitness}(os_c^3), \text{fitness}(os_c^4))$$

$$os' = os + [b_1 \Delta nos_1, b_2 \Delta nos_2, \dots, b_n \Delta nos_n]$$

$os$  is the chromosome with biggest fitness value in all four possible offsprings.  $b_i$  random equals to 0 or 1 and  $\Delta nos_i$  is a random number making sure  $para_{\min}^i \leq os_i + b_i \Delta nos_i \leq para_{\max}^i$ .  $os'$  is the final generation after crossover operation and mutation operation.

Thirdly, the fitness value is defined. By adding parameters in the neural network mathematical expression, the actual output of GA-optimized neural network  $y_k$  equals to:

$$y_k = \sum_{j=1}^{n_h} \delta(s_{jk}^2) w_{jk} \text{logsig} \left[ \sum_{i=1}^{n_i} \delta(s_{ij}^1) w_{ij} x_i - \delta(s_j^1 b_j^1) \right] - \delta(s_k^2) \text{logsig}(b_k^2)$$

in which  $k=1, 2, \dots, n_{out}$ ,  $s_{ij}$  denotes link from  $i$ th neuron in input layer to  $j$ th neuron in hidden layer,  $s_{jk}$  denotes link from  $j$ th neuron in hidden layer to  $k$ th neuron in output layer,  $w_{jk}$  denotes weight between each neuron,  $b_k^1$  and  $b_k^2$  denote bias in input layer and hidden layer respectively,  $n_{in}$ ,  $n_h$  and  $n_{out}$  denote the number of neurons of input layer, hidden layer and output layer, respectively.

The error of the whole network is defined as mean of all chromosomes:

$$err = \sum_{k=1}^{n_{out}} \frac{|y_k - y_k^d|}{n_d}$$

in which  $n_d$  denotes the number of chromosomes used in the experiment,  $y_k^d$  denotes the desired output of output neuron  $k$ . Given the error of the network, GA is implemented to optimize the network, thus minimizing the error. The fitness function is defined as

$$\text{fitness} = \frac{1}{1 + err}$$

the smaller the error and the bigger the fitness value. GA is implemented to find the global optimum of the fitness function, thus the parameter combinations of weight  $w$  are the trained weight for the network.

#### 4.2. PSO on neural network

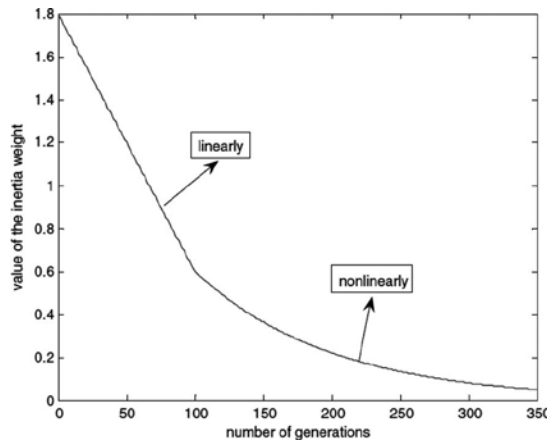
Gudise and Venayagamoorthy [5] implemented PSO on neural network training in 2003.

The fitness value of each particle (member) of the swarm is the value of the error function evaluated at the current position of the particle and position vector of the particle corresponds to the weight matrix of the network.

Zhang et al. [7] developed a hybrid algorithm of BP and PSO that could balance training speed and accuracy.

The PSO algorithm was showed to converge rapidly during the initial stages of a global search, but around global optimum, the search process will become very slow. On the contrary, the gradient descending method can achieve faster convergent speed around global optimum, and at the same time, the convergent accuracy can be higher.

When the iteration process is approaching end and current best solution is near-global optimum, if the change of the weight in PSO is big, the result will vibrate severely. Under this condition, Zhang supposed with the increase of iteration time, the weight in PSO should decline with the iteration time's increasing to narrow the search range thus paying more attention to local search for global best. He suggests the weight decline linearly first, then decline nonlinearly as shown in **Figure 15**.



**Figure 15.** Change of weight in PSO with number of generations.

The concrete working process is summarized below:

For all particle  $p_i$ , it has a global best location  $p_{globalbest}$ . If the  $p_{globalbest}$  keep unchanged for over 10 generations, that may infer the PSO pays too much time on global search thus BP is implemented for  $p_{globalbest}$  to deep search for a better solution.

Similar to GA's implementation in neural network, the fitness function defined is also based on whole network's error and to minimize the error as the optimization of PSO.

The learning rate  $\eta$  of neural network is also controlled in the algorithm, as

$$\eta = k \times e^{-\eta_0 \cdot epoch}$$

where  $\eta$  is the learning rate,  $k$  and  $\eta_0$  are constants,  $epoch$  is a variable that represents iterative times, through adjusting  $k$  and  $\eta_0$ , the reducing speed of learning rate is controlled.

By implementing the strategy that BP focusing on deep searching and PSO focusing on global searching, the hybrid algorithm has a very good performance.

#### 4.3. Hybrid GA and PSO on neural network

Juang [6] hybrids GA and PSO thus optimize recurrent network's training. The work was published in IEEE Transactions on Systems, Man, and Cybernetics, 2004.

The hybrid algorithm called HGAPSO is put forward because the learning performance of GA may be unsatisfactory for complex problems. In addition, for the learning of recurrent network weights, many possible solutions exist. Two individuals with high fitness values are likely to have dissimilar set of weights, and the recombination may result in offspring with poor performance.

Juang put forward a conception of "elite" of the first half to enhance the next generation's performance. In each generation, after the fitness values of all the individuals in the same population are calculated, the top-half best-performing ones are marked. These individuals are regarded as elites.

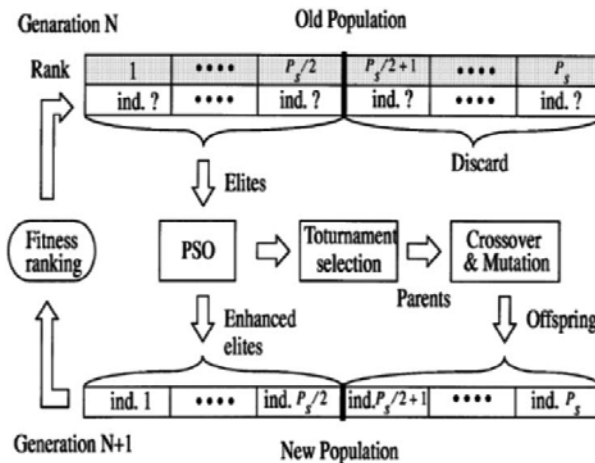
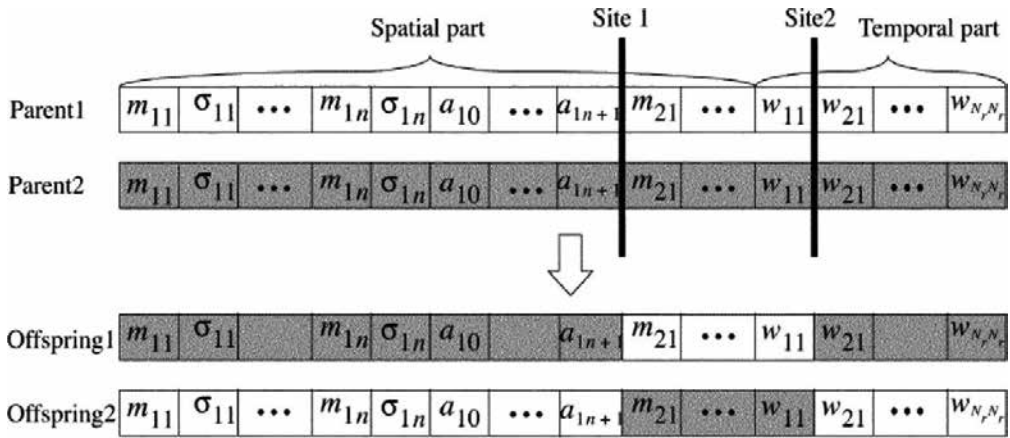


Figure 16. Working flow of HGAPSO.

In every epoch, the worse half of the chromosome is discarded. The better half is chosen for reproduction through PSO's enhancement. All elite chromosomes are regarded as particles in

PSO. By performing PSO on the elites, we may avoid the premature convergence in elite GAs and increase the search ability. Half of the population in the next generation is occupied by the enhanced individuals, the others by crossover operation. The working flow of algorithm is shown in **Figure 16**.

The crossover operation of HGAPSO is similar to normal GA, randomly selecting site on chromosome and exchange the sited piece of chromosome to finish the crossover operation. The crossover schematic diagram is shown in **Figure 17**. In HGAPSO, uniform mutation is adopted, that is, the mutated gene is drawn randomly, uniformly from the corresponding search interval.

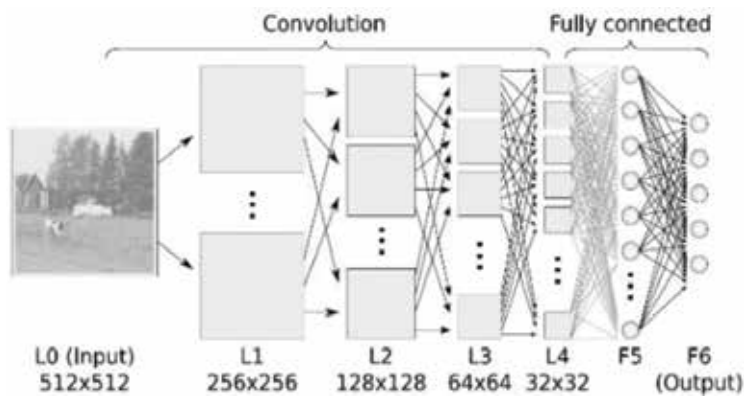


**Figure 17.** Schematic diagram of crossover operation.

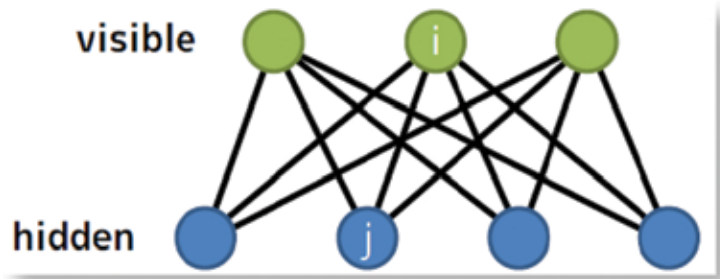
## 5. DL and RBM

When dealing with image classification or other problems, traditional method is using preprocessing transforming data as input values for neural network learning while using DL method for classification, raw data (pixel values) are used as input values. This will keep to the maximum extent protecting all information regardless of useful or not from being destroyed by extraction methods. The most advantage lies that all the extraction methods are based on expert knowledge and expert choice and thus are not extensible to other problems, while DL algorithm can overcome these limitations by using all data with its powerful processing ability. A CNN is shown in **Figure 18**.

The Boltzmann machine only has two layers (as **Figure 19**), the first layer is the input layer and the second layer is the output layer, although the structure is very simple and only contains two layers, its mathematical function in it is not simple. Herein we need to introduce the following probability equation to know the RBM.



**Figure 18.** Convolutional neural network.



**Figure 19.** Restricted Boltzmann machine.

From the equation we know there are three items in it, they are the energy of the visible layer neural, the energy of the hidden layer neural, and the energy is consisted of the two layers. From the energy function of the RBM, we can see there are three kinds of parameters lie in the RBM, different from the neural network, each neuron has a parameter too, but for the neural network only the connection of two layers has the parameters. Also, the neural network does not carry any energy function. They use the exponential function to express the potential function. There are also the probabilities existing in the RBM. They work exactly in the same way as the two kind of the probabilities such as  $p(v|h)$  and  $p(h|v)$ . It has a similarity with probability graph model in details, examples are Bayesian network and Markov network. It looks like a Bayesian network because of the conditional probability. On the other hand it does not look like a Bayesian because of the two direction probabilities. These probabilities are lying on the two variables that have only one direction.

Compared with the Markov network, the RBM seems to be having a little bit relation with it, because the RBM has the energy function just as the Markov network. But it is not so much alike because the RBM's variable has parameter, and the Markov network does not have any parameter. Furthermore, the Markov network does not have conditional probability because it has no direction but just the interaction. From the graph's perspective, variables in the

Markov network use cliques or clusters to represent the relations of close and communicated variables. It uses the production of the potentials of the clique to express the joint probability instead of conditional probability just like the RBM. Its input data are the kind of the Boolean data, within the range between 0 and 1.

The training way of the RBM is to maximize the probability of the visible layer  $p(v)$ , and to generate the distribution of the input data. RBM is a kind of special Markov random function and a special kind of Boltzmann machine. Its graphical model is corresponding to the factor product analysis.

Different from the probability graphical model, the RBM's joint distribution directly uses the energy function of both visible layer  $v$  and hidden layer  $h$  to define instead of potential of it given as

$$E(v, h) = -a^T v - b^T h - v^T W h$$

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)}$$

Later we will know that  $Z$  is the partition function defined as the sum of  $e^{-E(V, h)}$  over all the possible configurations. In other words, it is just a constant normalizing the sum over all the possible hidden layer configurations.

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v, h)}$$

The hidden unit activations are mutually independent given the activations. That is, for  $m$  visible and  $n$  hidden units, the conditional probability of a configuration of the visible unit  $v$ , given a configuration

$$P(v|h) = \prod_{i=1}^m P(v_i|h)$$

Conversely, the conditional probability of  $h$  given  $v$  is  $P(h|v) = \prod_{j=1}^n P(h_j|v)$ . Our goal is to infer the weights that maximize the marginal of the visible, in details we can step through the following equation to infer and learn the RBM.

$$\arg \max_w E \left[ \sum_{v \in V'} \log P(v) \right]$$

As for the training algorithm, the main idea is also applied GD idea into RBM. Hinton put forward CD [23] as a faster learning algorithm. Firstly, the derivative of the log probability of a training vector with respect to a weight is computed as

$$\frac{\partial \log P(v)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$

where the angle brackets are used to denote expectations under the distribution specified by the subscript that follows. This leads to a very simple learning rule for performing stochastic steepest ascent in the log probability of the training data:

$$\Delta w_{ij} = \varepsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$$

where  $\varepsilon$  is a learning rate.

Because there are no direct connections between hidden units in an RBM, it is very easy to get an unbiased sample of  $v_j h_{jdata}$ . Given a randomly selected training image,  $v$ , the binary state,  $h_j$ , of each hidden unit,  $j$ , is set to 1 with probability

$$p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij})$$

where  $b_j$  is the current state of hidden neuron  $j$ ,  $\sigma(x)$  is the logistic sigmoid function  $\sigma(x) = 1 / (1 + \exp(-x))$ .  $v_j h_j$  is then an unbiased sample. The CD is used to calculate the latter part  $v_j h_{jmodel}$ . Details can be found in their respective publications.

Considering the complicated computation of implementing CD, the training process of RBM is not easy. Under this condition, implementing meta-heuristic on RBM training to substitute CD is of high possibility.

## 6. Conclusion and discussion

Meta-heuristic has successfully implemented in neural network training. The algorithm used includes GA, PSO, their hybrid and many other meta-heuristic algorithms. Moreover, feed-forward BP neural network and SNN [24] are all trained with tests on famous classification problems.

The basic structure of DL is similar to traditional neural network. CNN is a special neural network with different weight computation regulations and RBM is a weighted bi-dimensional neural network or bi-dimensional graph. Their training processes are also mainly

executed through iterative formula on error which is similar to traditional neural network's training.

Having seen the above two concluding remarks, it can be assumed that it is of high possibility of applying meta-heuristic in DL to speed up training without declining performance. However, relevant publications along this direction are still rare.

Lastly, there still exists a question that goes under today's computation ability, especially the GPU whose computation ability is several times stronger than CPU that has been widely used in industrial area. One may ask, although elegant, is meta-heuristic still necessary? This question is not easy to answer. However, one can be certain that when traversing all the possible solutions is certainly highly time-consuming. Search for near-optimal results by meta-heuristic is still useful that can provide us a reasonable searched result (instead of non optimal or far from optimal) near a global optimum value at an acceptable computational cost.

## Author details

Zhonghuan Tian and Simon Fong\*

\*Address all correspondence to: ccfong@umac.mo

Department of Computer and Information Science, University of Macau, Macau SAR, China

## References

- [1] Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*. 2006;18(7):1527-1554.
- [2] Yu Kai, Jia Lei, Chen Yuqiang, and Xu Wei. Deep learning: yesterday, today, and tomorrow. *Journal of Computer Research and Development*. 2013;50(9):1799-1804.
- [3] Olga Russakovsky\*, Jia Deng\*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (\* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.
- [4] Izadinia, Hamid, Bryan C. Russell, Ali Farhadi, Matthew D. Hoffman, and Aaron Hertzmann. "Deep classifiers from image tags in the wild." In *Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*, pp. 13-18. ACM, 2015.
- [5] Gudise, V. G. and Venayagamoorthy, G. K. Comparison of particle swarm optimization and back propagation as training algorithms for neural networks. In: *Proceedings of In Swarm Intelligence Symposium SIS'03; 2006*. p. 110-117.

- [6] Beheshti, Z. and Shamsuddin, S. M. H. A review of population-based meta-heuristic algorithms. *International Journal of Advances in Soft Computing & Its Applications* 2013;5(1):1-35.
- [7] Zhang, J. R., Zhang, J., Lok, T. M., and Lyu, M. R. A hybrid particle swarm optimization-back-propagation algorithm for feed forward neural network training. *Applied Mathematics and Computation*. 2007;185(2):1026-1037.
- [8] Juang, C. F. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions* 2004;34(2):997-1006.
- [9] Meissner, M., Schmuker, M., and Schneider, G. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics*. 2006;7(1):125.
- [10] Leung, F. H., Lam, H. K., Ling, S. H., and Tam, P. K. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*. 2003;14(1):79-88.
- [11] Goldberg, D. E. and Holland, J. H. Genetic algorithms and machine learning. *Machine Learning*. 1988;3(2):95-99.
- [12] Kennedy, J. *Particle Swarm Optimization*; Springer, USA; 2010. p. 760-766
- [13] Glover, F. Tabu search-part I. *ORSA Journal on Computing*. 1989;1(3):190-206.
- [14] Wang, K. P., Huang, L., Zhou, C. G., and Pang, W. Particle swarm optimization for traveling salesman problem. In: *International Conference on Machine Learning and Cybernetics*; IEEE; 2003. p. 1583-1585.
- [15] Clerc, M. Discrete particle swarm optimization, illustrated by the traveling salesman problem. In: *New Optimization Techniques in Engineering*; Springer, Berlin Heidelberg; 2004. p. 219-239.
- [16] Cochocki, A. and Unbehauen, R. *Neural Networks for Optimization and Signal Processing*; John Wiley & Sons, Inc., New York, NY, USA; 1993.
- [17] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. 1958;65(6):386.
- [18] Nitta, T. Solving the XOR problem and the detection of symmetry using a single complex-valued neuron. *Neural Networks*. 2003;16(8):1101-1105.
- [19] Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*. 1997;10(9):1659-1671.
- [20] Kistler, W. M., Gerstner, W., and Hemmen, J. Reduction of the Hodgkin-Huxley equations to a single-variable threshold model. *Neural Computation*. 1997;9(5):1015-1045.

- [21] Bohte, S. M., Kok, J. N., and La Poutre, H. Error-back propagation in temporally encoded networks of spiking neurons. *Neurocomputing*. 2002;48(1):17-37.
- [22] Ghosh-Dastidar, S. and Adeli, H. Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering*. 2007;14(3):187-212.
- [23] Hinton, G. A practical guide to training restricted Boltzmann machines. *Momentum*. 2010;9(1):926.
- [24] Pavlidis, N. G., Tasoulis, D. K., Plagianakos, V. P., Nikiforidis, G., and Vrahatis, M. N. Spiking neural network training using evolutionary algorithms. In: *Proceedings of IEEE International Joint Conference on Neural Networks IJCNN'05*; July 31–August 2005; IEEE; 2005. p. 2190-2194.
- [25] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks* 2015;61:85-117.

# Design and Characterization of EUV and X-ray Multilayers

Hui Jiang

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62385>

## Abstract

Multilayers, which consist of periodic/apperiodic nanometer-scale stacks of two or more alternating materials, fill a gap between visible light optics and natural crystal by realizing high near-normal incidence reflectivity in extreme ultraviolet and soft X-ray regions and diffraction-limited focusing in hard X-ray region. Before fabricating a multilayer, it is essential to design a structure that realizes the required optical features. The optimization process uses merit functions that are defined by the design targets. In this chapter, the designs of two typical aperiodic multilayer structure, X-ray supermirror and EUV beam splitter, are introduced. Precision characterization of multilayer structures is the key process in multilayer sciences as well in order to improve fabricating process and determine optical properties in use. Searching a most suitable structure model to approaching real one by comparing experimental and simulated results is essentially an optimization problem. In this chapter, by fitting the X-ray grazing incidence reflectivity and diffuse scattering curves, the realistic multilayer structures are determined accurately.

**Keywords:** multilayer, merit function, downhill simplex, particle swarm optimization, reflectivity

## 1. Introduction

As we all know, mirror and lens are capable of realizing high reflectivity and focusing in visible light regime (wavelength  $\lambda \sim 400\text{--}760\text{ nm}$ ). But the situation is very different in X-ray and extreme ultraviolet (EUV) regimes. In soft X-ray ( $\lambda \sim 1\text{--}10\text{ nm}$ ) and EUV ( $\lambda \sim 10\text{--}100\text{ nm}$ ) regimes, the absorption for all materials is strong enough so that conventional type of mirror is unable to realize high reflectivity near normal incidence, meanwhile the natural crystal does not work as

well due to small lattice constant. In hard X-ray regime, because the refractive indices for most materials are close to unify, it is impossible to use lens to focus X-rays.

Multilayers, consisting of periodic/apperiodic nanometer-scale stacks of two or more alternating materials deposited on a substructure, fill the gap between mirror and crystal by realizing high near-normal incidence reflectivity in EUV [1, 2] and soft X-ray [3, 4] regimes, and also challenge diffraction-limited focusing in hard X-ray regime by multilayer Kirkpatrick-Baez [5] or multilayer Laue lens [6] systems.

The history of multilayers is from the 1940s. Initial attempts to fabricate Ag/Cu periodic structures failed due to their serious interdiffusions [7]. Twenty years later, stable Fe/Mg periodic structures were first made [8]. In 1972, Spiller [3] found that a mirror consisting of two alternate materials with different refractive indices can increase the near-normal incidence reflectivity in the EUV and soft X-rays regimes. In the 1980s, Vinogradov [9] and Barbee [10] developed relative theories of multilayers and, in 1992 Yamamoto [11] developed an intuitive and effective design method. Since the 1980s, as a result, the ultra-precise manufacture and technologies of thin films were rapidly improved to promote the development of multilayers. Typical bilayer periodic pairs for different energy regions are Mo/Si [2], W/B<sub>4</sub>C [12], Cr/C [13], Mo/Y [14], and Mg/SiC [15], etc. Aperiodic multilayers, developed more recently, can provide tailored spectral requirements, such as broadband or broad-angle high reflectivities [16, 17], high integral throughput [18], broadband polarizers [19, 20] and chirped mirrors [21, 22]. Such multilayers have wider applicability than periodic multilayers and natural crystals.

Nowadays, multilayers have presented wide applications in many important fields. In semiconductor industry, multilayers were used in mask illumination and replication [1] for next-generation extreme ultraviolet lithography. In synchrotron field, multilayers were always considered as key components for reflection [4], polarization [23], focusing [24] and monochromatization [25]. Biology imaging was often used at water window regime ( $\lambda=2.3\text{--}4.4\text{ nm}$ ) based on excellent contrast by using multilayer near-normal reflection [26] or required higher energy resolution to avoid spot blurs [27]. In addition, multilayers were also recorded in significant applications like space telescope [28], plasma diagnosis [29], neutron science [30], etc.

In this chapter, design and characterization of EUV and X-ray multilayers are presented. Optimization algorithms play important roles in these works. They help us to find optimal structures to satisfy any required spectral performances and based on experimental curves to retrieve real structure information.

## 2. Multilayer model

The optical behavior in a periodic multilayer can be described by a corrected Bragg equation with constant periodic thickness  $D$ , fractional thicknesses of absorber (scattering) layers  $d_a$  and spacer layers  $d_s$  and complex refractive indices  $n_a=1-\delta_a+i\beta_a$  and  $n_s=1-\delta_s+i\beta_s$ ,

$$m\lambda = 2D\sin\theta \left[ 1 - \frac{2(d_a\delta_a + d_s\delta_s)}{\sin^2\theta} \right]^{1/2} \quad (1)$$

where  $m$  is the reflection order,  $\lambda$  is the wavelength and  $\theta$  is the grating incidence angle. For a multilayer with two materials of similar absorption, the highest reflectivity (in the first reflection order) requires the thickness of each layer to be close to a quarter of the wavelength.

For a periodic/apperiodic multilayer with a substrate of finite thickness, the reflectance and transmittance coefficients at each interface can be calculated using the Fresnel equations which, for the  $j^{\text{th}}$  interface, can be written as

$$\begin{aligned} r'_{p,j} &= \frac{n_{j-1}\cos\theta_j - n_j\cos\theta_{j-1}}{n_{j-1}\cos\theta_j + n_j\cos\theta_{j-1}} \\ r'_{s,j} &= \frac{n_{j-1}\cos\theta_{j-1} - n_j\cos\theta_j}{n_{j-1}\cos\theta_{j-1} + n_j\cos\theta_j} \\ t'_{p,j} &= \frac{2n_{j-1}\cos\theta_{j-1}}{n_{j-1}\cos\theta_j + n_j\cos\theta_{j-1}} \\ t'_{s,j} &= \frac{2n_{j-1}\cos\theta_{j-1}}{n_{j-1}\cos\theta_{j-1} + n_j\cos\theta_j} \end{aligned} \quad (2)$$

where  $n_j$  is the refractive index of the  $j^{\text{th}}$  layer,  $\theta_j$  is the incidence angle in that layer and s and p refer to the two polarizations. The s- and p-component amplitude reflections and transmissions also can be expressed using the recurrence formula [31]

$$\begin{aligned} r_j &= \frac{r'_{j+1}E_{j+1} + r_{j+1}\exp(-2i\delta_{j+1})}{1 + r'_{j+1}E_{j+1}r_{j+1}\exp(-2i\delta_{j+1})} \\ t_j &= \frac{t'_jt_{j-1}\exp(-i\delta_j)}{1 + r'_jE_jr_j\exp(-2i\delta_j)} \end{aligned} \quad (3)$$

where  $\delta_j$  is the phase factor of the  $j^{\text{th}}$  layer and  $E_j$  is the roughness factor of the  $j^{\text{th}}$  interface, normally the Debye-Waller factor  $\exp(q_j^2\sigma_j^2/2)$  or the Nevot-Croce factor  $\exp(q_jq_{j+1}\sigma_j^2/2)$  [32], where  $q$  is the wave vector and  $\sigma$  is the root mean square (RMS) interfacial roughness. For smooth interfaces, the Nevot-Croce factor was considered to be better than the Debye-Waller factor. In applying equation 4, the thickness of the substrate is usually considered to be infinite, and so the initial conditions are  $r_{n+2}=0$  and  $t_0=1$  as shown in **Figure 1**. The recurrence formula can then be used to calculate  $r_0$  and  $t_{n+2}$ . The reflectance  $R$  and transmittance  $T$  in an  $n$ -layer multilayer system can be expressed as

$$R_{p/s} = \frac{R_{0,p/s}}{T_{0,p/s}} = r_{p/s,0} r_{p/s,0}^*$$

$$T_{p/s} = \frac{T_{n+2,p/s}}{T_{0,p/s}} = t_{p/s,n+2} t_{p/s,n+2}^*$$
(4)

The scattering signal is around the specular direction. The scattering potential is divided into a non-disturbed part and a disturbance. Interferences of reflected and transmitted waves have four types of interaction based on the dynamic scattering process [33]. The whole diffuse scattering signal is represented as

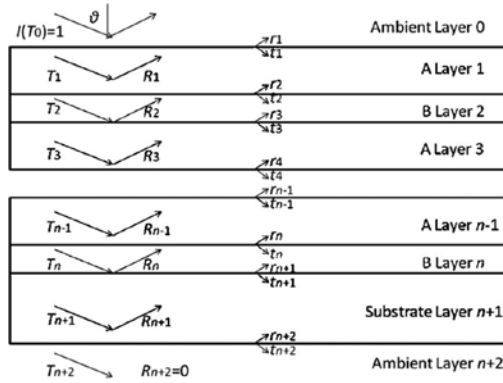
$$I_{diff} = I_0 \Delta\Omega / 2 \cdot A_s / A_b \sum_{j,k=1}^N |n_j^2 - n_{j+1}^2|^2$$

$$\cdot \sum_{m,n=0}^3 S_{j,k}^{mn}(q_x) \tilde{G}_j^m(\tilde{G}_k^n) \exp \left\{ -\frac{1}{2} \left[ (q_{z,j}^m \sigma_j)^2 + ((q_{z,k}^n)^* \sigma_k)^2 \right] \right\}$$
(5)

where  $\Delta\Omega$  is the detector acceptance angle,  $A_s/A_b$  is the area ratio of the radiation on the sample to beam spot,  $G_j^m$  is the four mutual products of  $T_i$ (or  $R_i$ ) and  $T_s$ (or  $R_s$ ) and  $S(q_x)$  is the structure factor.

$$S_{j,k} = \int_0^\infty C_{j,k}(x) \cos(q_x x) dx \cdot \exp(-|Z_j - Z_k| / \xi_\perp)$$
(6)

where  $\xi_\perp$  is the vertical correlation length and  $C(x)$  is the lateral correlation function which is based on the self-affine characteristic of rough interface [34] and  $\xi_{//}$  is the lateral correlation length and  $h$  is the fractal exponent.



**Figure 1.** Reflectance and transmittance coefficients at each interface of a multilayer structure.

### 3. Optimization algorithm

#### 3.1. Introduction

Normally, before using a multilayer, it included three important steps: design, fabrication and characterization. It is essential to design a structure that realizes the required optical features before the multilayer fabrication. Fabrication technologies, such as sputtering and evaporation, are typical random processes. Their technology parameters need to be attempted and optimized repeatedly. The attempts are based on accurate structural characterization. The most effective structural characterization methods are hard X-ray grazing incidence specular reflectance (XRR) and diffuse scattering (XDS). The experimental curve can be compared to the simulated curve calculated from a guessed multilayer model until reach a satisfying agreement. It is clear whether design or characterization, the core is optimization algorithm. Suitable optimization algorithms enable us to search optimal multilayer structure in design and determine the most realistic multilayer structure in measurement.

In general, optimization algorithms include local and global optimizations. The former refers to situations in which an approximate range of optimal values is known prior to the optimization. If a group of candidate structures can be defined at the outset, it is straightforward to find optimal structures in a short time. Common local algorithms are mainly based on the least gradient, including quasi-Newton method [35], steepest-descent method [36], Levenberg-Marquardt algorithm [37, 38], downhill simplex algorithm [39], etc. In the design of aperiodic multilayers, several techniques have been developed to search for initial candidate structures, including Kozhevnikov's method [17] and the search for suitable positions and numbers of layers such as needle optimization [40, 41], in order to make these local algorithms converge to better results.

The global algorithm has a larger search space and so always takes more time to search for the optimal structure than a local algorithm does, but prevents local results that miss the global optimum. Global algorithms play a more significant role than local algorithms because of their wider search ranges and stronger search capabilities. Global algorithms, often based on natural phenomena and processes, include random search (RS) [42], genetic algorithms [43] (GA), simulated annealing (SA) [44], differential evolution [45], hybrid multistate (MS), topographical optimization [46], particle swarm optimization (PSO) [47] and the ant colony optimization (ACO) algorithm [48].

Many optimization algorithms have been used in multilayer design. According to different requirements, different optimization targets are chosen as the functions to estimate the pros and cons of the algorithm. This kind of function is defined as merit functions. The balance between calculation time and search accuracy needs to be considered when choosing a suitable optimization algorithm.

#### 3.2. Downhill simplex algorithm

The downhill simplex algorithm (Nelder-Mead method) [49] is a typical local optimization. The method only requires function evaluation and not derivatives. It is based on a movable

simplex with  $N+1$  vertices in  $N$  dimensions ( $N$  variables). In two dimensions, the simplex is a triangle.

The optimization is started with the  $N+1$  vertices defining an initial simplex. If one of these vertices is defined as the starting point  $\mathbf{P}_0$ , the other  $N$  vertices can be expressed as  $\mathbf{P}_i = \mathbf{P}_0 + \tau \mathbf{e}_i$ , where  $\mathbf{e}_i$  is a unit vector and  $\tau$  is a constant defining the characteristic length scale. The coordinates of the vertices define the current multilayer structure, and the merit function of each vertex can be calculated. Most optimization steps move the vertex of the simplex with the best merit function through the opposite face of the simplex to a lower vertex. These basic steps are called reflections. The center of the reflection is the weighted center of the simplex based on the merit function of each vertex. If the merit function of the reflected vertex is better than that of the best vertex in the previous step, this reflection direction can be regarded as the correct one and the reflection can be doubled, which is called expansion. In contrast, if the reflected vertex is worse than the previous worst vertex, the simplex should move along the transverse reflection direction – this is called inward contraction. If the reflected vertex is just better than the worst previous vertex, the reflection rate should be halved, which is called contraction. Based on these steps, the simplex moves toward a valley in the solution. When the merit functions of all the vertices are sufficiently close, the optimal multilayer structure can be considered to have been found.

### 3.3. Particle swarm optimization

The PSO algorithm, one of the most important swarm intelligence methods, is a parallel evolutionary computation technique developed by Kennedy and Eberhart in 1995 [47]. The motivation of PSO was to understand the behavior of birds or fish searching for food. The process is similar to GA, but only has a few operators, including selection, crossover and mutation. Any particle moves with its own velocity, which is updated according to its own experience and that of the other particles. Some studies [50] have shown that in PSO the inertia weight (defined following equation 8) influences the trade-off velocities between global and local exploration abilities, and in GA crossover and mutation have different effects at the beginning and end of the process. The one-way information transfer means that PSO convergences to best solution faster than GA, in which the whole swarm moves uniformly. PSO also has stronger ergodicity than GA and is the only evolutionary algorithm that does not incorporate survival of the fittest.

In an  $N$ -dimension objective space ( $N$  variables for optimization), there are  $M$  particles to search for the optimal solutions. The location and velocity of the  $i^{\text{th}}$  particle are defined by groups of candidate solutions

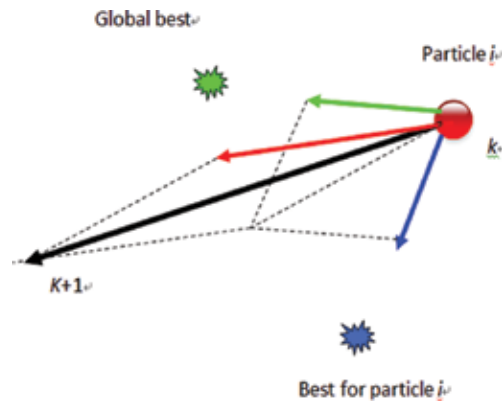
$$\begin{cases} X_i = (X_{i1}, X_{i2}, \dots, X_{iN}) \\ V_i = (V_{i1}, V_{i2}, \dots, V_{iN}). \end{cases} \quad (7)$$

For the design of multilayers, the location variables are the layer thicknesses and the working angle. For the characterization, the variables can be the layer thicknesses, densities, roughness,

material interdiffusions, etc. The velocity expresses the search direction of any particle, which updates the new location and velocity at the next iteration to be

$$\begin{cases} V_i(k+1) = wV_i(k) + c_1r_1(p_i(k) - X_i(k)) + c_2r_2(g(k) - X_i(k)) \\ X_i(k+1) = X_i(k) + V_i(k+1). \end{cases} \quad (8)$$

The new velocity includes inertia, cognitive and social learning terms. The inertia term forces a particle to tend to follow its old velocity, including an inertia weight  $w$  which influences the trade-off between global and local exploration abilities, changing from  $w_{\max} = 0.9$  at the beginning to  $w_{\min}$  at the maximum iteration. In the cognitive and social learning terms,  $c_1$  and  $c_2$  are acceleration constants and  $r_1$  and  $r_2$  are random numbers uniformly distributed in the interval (0,1). The functions  $p_i(k)$  and  $g(k)$  are, respectively, the previous best location for the  $i^{\text{th}}$  particle and the previous best locations for all particles at the  $k^{\text{th}}$  iteration. These two terms guarantee that every particle can extract its own experience and share information from the society, making faster converge at the beginning of a run. "The behavior of any particle can be sketched in **Figure 2**."



**Figure 2.** The searching process of one particle.

The basic process of the algorithm is as follows: Step 1 is the random initialization of locations and velocities. The ranges of possible locations are selected using a priori knowledge. The maximum velocities should be about 20% of the location range in any step. If the velocities are too large, the particles in the next iteration can fly beyond the location boundary. If they are too small, the convergence rate is very low. The merit function of each particle is then determined and the optimum one is sought. Step 2 is the update of locations and velocities to the next iteration according to equation 8. If any location is beyond the boundary, it is replaced by the boundary value and the velocity direction is reversed, analogous to reflection from a wall. The merit functions of the particles are then calculated and the best location of each particle and the best global location are updated. Step 3 is the inspection of the defined convergence

condition. If this is satisfied or the set maximum number of iterations have taken place, the program terminates. Otherwise step 2 is repeated.

If the other parameters are fixed, a smaller  $w_{\min}$  can increase the speed of convergence, but the optimization is more likely to fall into a local optimum solution. Following an analysis of the relationship between parameter selection and convergence [51], large  $w_{\min}$  close to  $w_{\max}$  and relatively small acceleration constants of about 0.3–0.7 were chosen to make the whole optimization converge effectively. For a larger number of variables (e.g., more than two materials), more particles and more iterations have to be used.

## 4. Design of multilayers

### 4.1. Introduction

The usual design targets are reflectivity, transmittivity and phase performance. Simple periodic multilayer structures are suitable for single wavelengths or working angles, but more complicated aperiodic structures are needed to tailor spectral or phase responses. In general, the optimization process uses merit functions that are defined by the design targets. The selection of suitable multilayer materials, based on optical constants and material stability, is the starting point of the design process. The optimization of particular designs is complicated and can only be enabled using various computer algorithms. An important technological constraint is that the optimal structure should be as simple as possible, avoiding ultra-thin layers and drastically variable layer thicknesses.

In this section, the designs of two typical aperiodic multilayers, respectively, X-ray supermirror and EUV intensity beam splitter, based on my PhD studies [52], are presented to show the important role of optimization algorithm on multilayer design.

### 4.2. X-ray supermirror

Supermirrors are aperiodic multilayers often used in X-ray [53] or neutron [54] imaging systems. They are designed to provide particular characteristics such as increased reflectivity or flat optical response over broader angular or energy ranges than possible with conventional periodic multilayer mirrors. Based on these characteristics, they have been chosen for use in X-ray telescope [55] and applications of synchrotron radiation [56].

In this study, a broad-angle supermirror will provide relatively high peak reflectivity over a wide angular range and several times the integrated reflectivity.

The principle of supermirror design is also based on the Bragg diffraction equation. Compared to a periodic multilayer mirror with fixed layer thicknesses and thickness ratios, a broad-angle supermirror with variable periodic thickness and thickness ratio provides the interference effect of the standing waves over a range of working angles. The target of the design is to obtain high average reflectivity and small reflectivity fluctuations over a defined angular range. The merit function for such designs is

$$F = \min \sum_{i=1}^j (R_j - R_{\text{target}})^2 \quad (9)$$

where  $j$  is the number of working angles at which the reflectivity is calculated and  $R_{\text{target}}$  is the target for the average reflectivity.

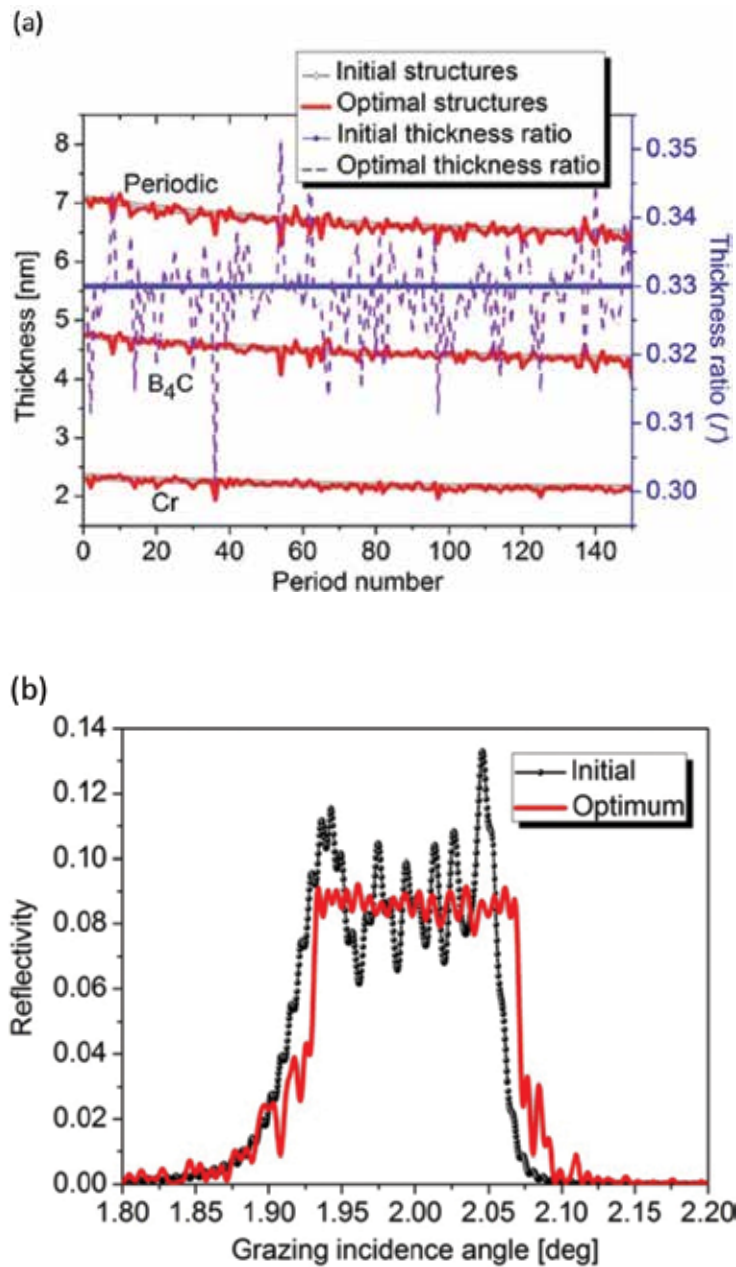
There are two methods to define the initial multilayer structures in the optimization. The first is to determine the periodic structure related to the center of the working angular range. The second is based on Kozhevnikov's study [17], which is an analytic method that gives an approximate initial depth-graded structure to enable a faster approach to the optimum solution. This method has been applied to various broad-angle [57] and broadband [58] designs of multilayer mirrors. The expression for the reflectivity of a depth-graded multilayer mirror is

$$R(\lambda) = |r(\lambda)|^2 = \frac{2\eta(\lambda)\sqrt{|q'(z)|}}{|\eta^2(\lambda) + |q'(z)||} \exp(-4\kappa_2(\lambda)z) \quad (10)$$

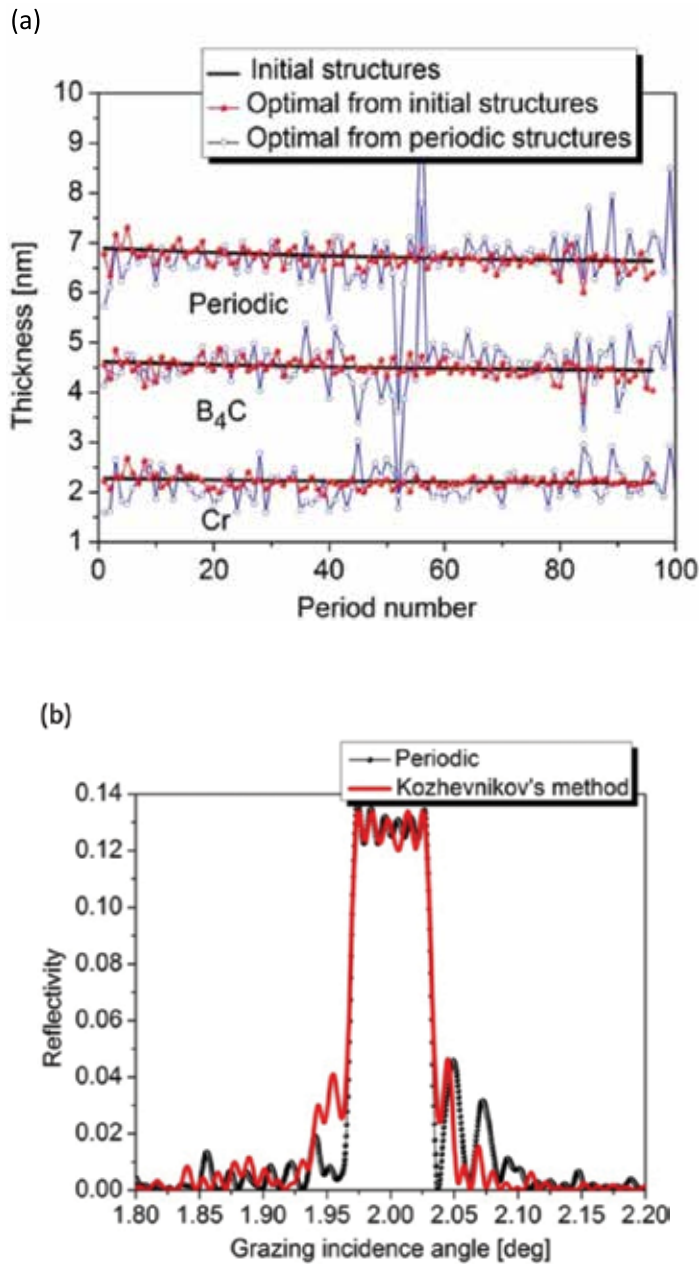
$$\eta(\lambda) = \frac{(\varepsilon_1(\lambda) - \varepsilon_2(\lambda))\sin(\pi L)}{2\lambda\sqrt{\mu(\lambda) - \cos^2\theta}}.$$

where  $q(z)$  (which describes the multilayer structure) is positive, continuous and differentiable,  $L$  is the total multilayer thickness, and  $\varepsilon_1$  and  $\varepsilon_2$  are the dielectric permittivities of the two materials.  $\mu$  is the mean dielectric permittivity,  $\kappa = k(\mu - \cos^2\theta)^{1/2}$  where  $k$  is the wave vector.  $\kappa_2$  is the imaginary part of  $\kappa$ . Alternatively, for a given target reflectivity, the multilayer structure  $q(z)$  can be determined by solving the inverse problem.

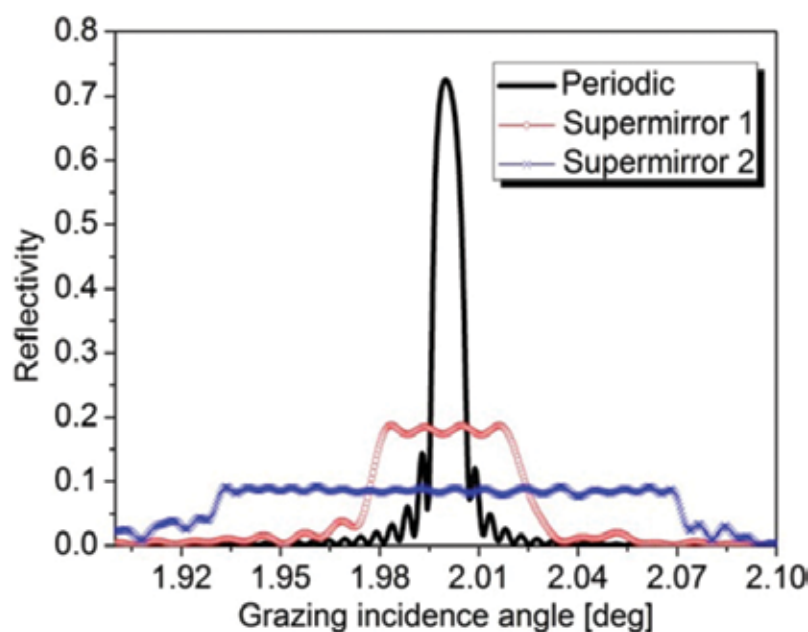
A Cr/B<sub>4</sub>C supermirror was designed for chromium K <sub>$\alpha$</sub>  radiation and incidence angles in a range around 2°, the latter to fit within the structure of the X-ray microprobe source. **Figure 3(a)** shows the design results for an angular width of 0.14°. Using downhill simplex algorithm, the design was close to the initial depth-graded structure and easily fabricated. Mostly, the thickness ratios of the layer pairs are between 0.32 and 0.34. **Figure 3(b)** shows the change from the initial reflectivity to the optimal value. **Figure 4(a)** and **(b)** show the influences of using different initial structures. Although the reflectivity curves are similar, the optimal structures from the initial periodic structures have larger fluctuations that lead to more difficulties in fabrication, since rapidly varying thicknesses make it hard to ensuring correct deposition rates. **Figure 5** and **Table 1** show designs for different angular ranges. Supermirrors can increase the angular width by several times and, although the maximum reflectivity decreases substantially, the overall reflectivity for all angles used in the optimization increases.



**Figure 3.** (a) The initial and optimized layer thicknesses for a Cr/B<sub>4</sub>C aperiodic multilayer mirror with an angular width of 0.14°. (b) Comparison of the reflectivity curves for the initial and optimum structures.



**Figure 4.** (a) The thicknesses of Cr/B<sub>4</sub>C aperiodic multilayer mirrors for different initial structures. (b) Comparison of the optimum reflectivity curves for the two initial structures.



**Figure 5.** Different angular width requirements lead to different optimum results reflectivity profiles.

Sample	FWHM [deg]	Peak Reflectivity [%]	Overall Reflectivity
Periodic	0.01	71.4	0.008
Supermirror 1	0.05	18.0	0.010
Supermirror 2	0.15	8.4	0.014

**Table 1.** Comparison of FWHM, peak reflectivity and integrated reflectivity for two aperiodic supermirrors and a periodic multilayer.

4.3. EUV intensity beam splitter

A large amount of work has been done on multilayers for mask illumination and replication [1, 59] in extreme ultraviolet lithography (EUVL), but so less thought has been given to inspection of the masks and produced components. Defects in EUVL masks can occur in both the absorber patterns and the multilayer-coated mask blanks, and inspection is essential as the number of defects has to be minimized. However, conventional full-field or scanning imaging is not appropriate as the masks and components are too absorbing to the radiation. An alternative is to use interferometry [60] like Schwarzschild optics [61]. A multilayer beam splitter divides an incident beam into two coherent parts by reflecting part and transmitting part, respectively, reflected from a sample multilayer and a reference multilayer. Interference fringes are produced when they are recombined.

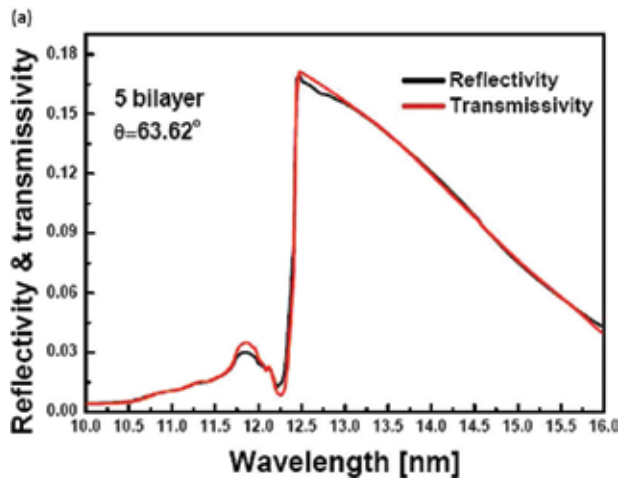
There is a trend to utilize broadband spectrum from the source based on tin or xenon plasmas instead of single wavelength around 13nm to irradiate wafers, in order to decrease exposure times [62]. Hence, broadband beam splitters optimized to the output of tin or xenon plasmas are required.

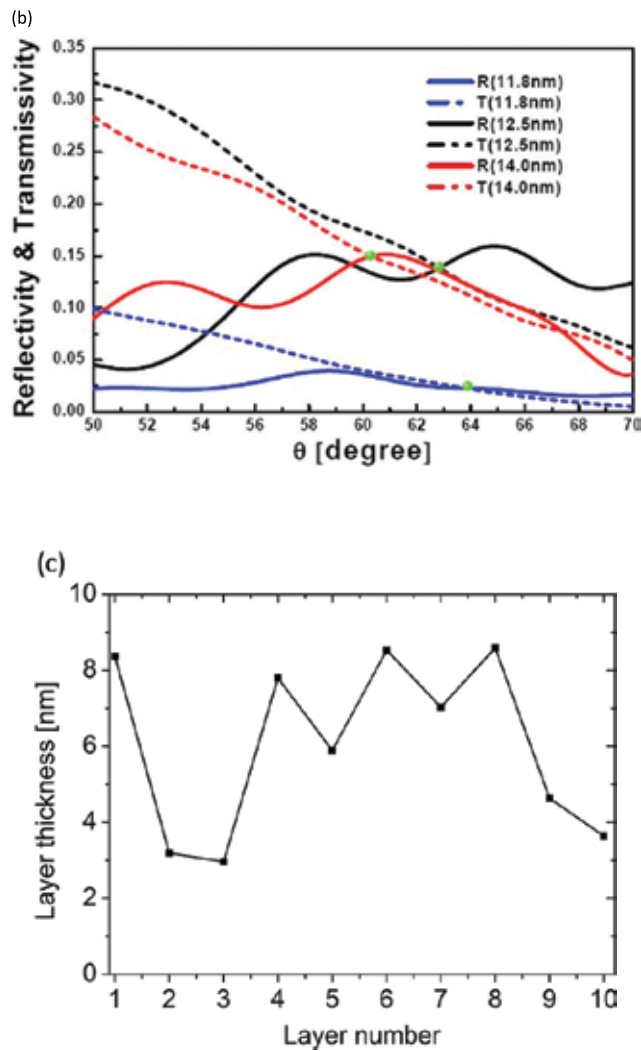
The merit function must satisfy two requirements for an intensity beam splitter: (1) high reflectance and transmittance throughputs and (2) excellent agreement between the reflected and the transmitted beam intensities over the range of wavelengths. The merit function chosen was

$$F = \sum_{i=1}^m \frac{[R(\lambda_i) - T(\lambda_i)]^2}{R(\lambda_i) + T(\lambda_i)} \quad (11)$$

where  $m$  is the number of wavelength calculation points. The squared numerator ensures good agreement between the intensities and the denominator prevents the intensities from becoming too small.

The principle of material choice for beam splitters is a little different from that for high-reflectivity multilayers. The differences in the optical constants are more important for high reflectivity, while the absorption coefficients have more influence for beam splitters. Since silicon nitride is used as the substrate, due to its high strength, stability and transmittance, the transmitted beam is affected by the silicon absorption edge at 12.4 nm. Ruthenium and molybdenum are suitable for the scattering layers and beryllium, yttrium and silicon can be chosen for the spacer layers. Among their pairs, Mo/Si [63] and Mo/Be [64] have been widely studied for use at 13.5 and 11.4 nm. In this design, Mo/Si was chosen as the material pair.





**Figure 6.** Reflectivity and transmittivity for a 5-bilayer Mo/Si multilayer as functions of (a) wavelength and (b) incidence angle at three different wavelengths; the green points mark the optimum angles where reflectivity and transmittivity are equal; (c) layer thickness of design.

The thicknesses of each layer and the working angle are considered as the variables of optimization. By designing with different numbers of bilayers, it was found that the intensities of reflected and transmitted beams are in best agreement for five bilayers in the wavelength range 10–16 nm, see **Figure 6(a)**. The highest intensity, which is at a wavelength slightly larger than silicon absorption edge, is about 17% and the optimal incidence angle is 63.62°. The average thicknesses of the silicon and molybdenum layers are 7.38 nm and 4.31 nm, respectively. As shown in **Figure 6(b)**, as the wavelength increases, the optimal incidence angle decreases; 63.62° is the average value. **Figure 6(c)** presents the layer thicknesses of design. Further robustness considerations [65] can be found in order to design better structures with

more stable optical performance when layer thickness errors are present. If a constraint condition is established before calculating the merit function, the process is stopped if the constraint becomes unsatisfied or feedback is provided to the merit function, then the optimal structure may be more robust than in normal optimization.

## 5. Characterization of multilayers

### 5.1. Introduction

Characterization of multilayers plays an important role in estimating multilayer quality and improving fabrication technology. In order to determine ultra-small layer thicknesses in multilayers, most direct and effective methods are hard X-ray grazing incidence reflection and diffuse scattering. Due to excellent penetration of hard X-rays, reflected and/or scattered X-rays produce interference fringes. If a structure is known, by using reflectivity (equation 4) or scattering (equation 5) model, the reflectivity or rocking scattering curves can be simulated. However, the inverse problems are always difficulties, especially for determining the structural parameters from diffuse scattering curves.

Reflectivity curve includes the information of layer thickness, density and interfacial width. Comparing experimental curve with theoretical curve by optimization algorithm, one can obtain these parameters. As a fitting method, in order to obtain believable results, reliable initial structure and parameter constraints are necessary. X-ray reflectivity is unable to distinguish between interfacial roughness and interdiffusion, because both deteriorate the reflectivity in a similar way. X-ray diffuse scattering is one of the most direct techniques for determining interfacial roughnesses and can also afford the information of thin film growth such as Hurst exponent and lateral/vertical correlation lengths.

In this section, curve fittings for reflectivity and diffuse scattering data are presented based on my previous studies on multilayers. These studies determined the accurate structural parameters of various material-pair multilayers and study the surface and interface performances.

### 5.2. Structure determination from reflectivity data

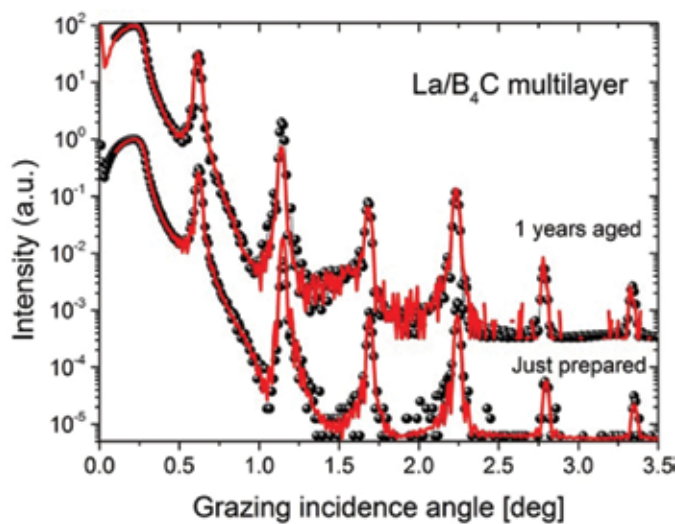
As a consumable optics, multilayer structures are not always stable. Aging effects related to the inherent characteristic of the materials deserve to be researched. Three  $B_4C$ -based multilayers [66] were deposited by using magnetron sputtering including  $W/B_4C$ ,  $Mo/B_4C$  and  $La/B_4C$  multilayers. They were measured by using hard X-ray grazing incidence reflectivity while just prepared and after long-time storage in a dry atmosphere environment.

Because long-time storage may make multilayer surface oxidized and/or contaminated, in the fitting process the surface  $B_4C$  layer was regarded to have independent structural parameters from  $B_4C$  layers in interior periodic structure. Thus, there are 10 parameters need to be fitted, viz. 10 variables in the optimization, including the layer thickness, density and interface width for metal layer,  $B_4C$  layer in periodic structure and surface  $B_4C$  layer and the intensity coefficient. The fitting was based on the PSO algorithm.

As can be seen in **Table 2**, the periodic thickness increased 0.39% for La/B<sub>4</sub>C multilayer and 0.24% for Mo/B<sub>4</sub>C multilayer after 1 year, but decreased 0.05% for W/B<sub>4</sub>C multilayer after 2 years. The changes may result from interdiffusion mechanism and stress release. Surface oxidation increased the stress and the changes of the layer thickness balanced the whole stress of multilayer. The outmost layer thickness increases apparently over 10% for La/B<sub>4</sub>C and Mo/B<sub>4</sub>C multilayers because lanthanum and molybdenum have stronger oxidation capability than tungsten so that oxygen atoms penetrate through B<sub>4</sub>C layer gradually and react with metal atoms. The surface B<sub>4</sub>C layer was also determined to absorb oxygen [67]. **Figure 7** shows the reflectivity and their fitting curves of La/B<sub>4</sub>C multilayer while just prepared and after 1-year storage.

Sample	Status	Metal			B <sub>4</sub> C			Surface B <sub>4</sub> C		
		<i>d</i>	<i>σ</i>	<i>ρ</i>	<i>d</i>	<i>σ</i>	<i>ρ</i>	<i>d</i>	<i>σ</i>	<i>ρ</i>
		[nm]	[nm]	[% bulk]	[nm]	[nm]	[% bulk]	[nm]	[nm]	[% bulk]
W/B <sub>4</sub> C <i>N</i> = 20	Just prepared	1.94	0.32	90.42	2.23	0.31	82.91	2.66	0.20	114.7
	2 year aged	1.93	0.32	93.70	2.23	0.31	82.62	2.61	0.21	117.8
Mo/B <sub>4</sub> C <i>N</i> = 20	Just prepared	2.30	0.51	91.30	2.65	0.20	75.23	3.10	0.44	89.21
	1 year aged	2.30	0.49	98.70	2.66	0.18	76.90	3.33	0.58	85.87
La/B <sub>4</sub> C <i>N</i> = 15	Just prepared	5.00	0.47	83.81	2.93	0.33	107.02	4.53	0.20	109.76
	1 year aged	5.01	0.38	82.83	2.95	0.33	105.62	5.08	0.21	90.67

**Table 2.** The change of the structure parameters after time aging for 3 metal/B<sub>4</sub>C multilayers.



**Figure 7.** The grazing incidence reflectivity and their fitting curves for La/B<sub>4</sub>C multilayer while just prepared and after 1-year storage.

The other previous work of structural determination is to study the stability of Ru/C multilayer monochromator with different periodic thicknesses after cryogenic cooling treatment [68]. The results show that the structural parameters keep stable after cryogenic cooling and present sufficient experimental evidences for using cryogenic-cooled multilayer monochromators in the high-thermal-load undulator beamline.

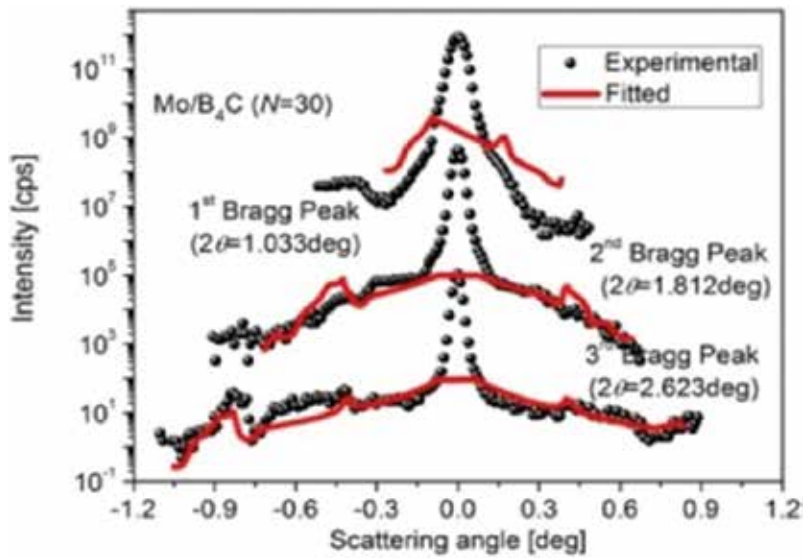
### 5.3. Interface investigation from diffuse scattering data

The diffuse scattering signals are distributed around the reflection direction. Common scan methods include rocking curve scans ( $\omega$  scan), offset scans, detector  $2\theta$  scans and full reciprocal space scans. Rocking curve scan is by fixing the detector and scanning the incidence angle. This method is very sensitive to determine roughness information. The method of scattering curve fitting is also based on the global optimization to approach the real multilayer structure.

Sample	$d$ [nm]	$\rho$ [% bulk]	$\sigma_r$ [nm]	$\sigma_d$ [nm]	$\xi_{//}$ [nm]	$h$	$\xi_{\perp}$ [nm]
W/B <sub>4</sub> C	0.83	82.10	0.29	0.10	17.19	0.10	286.64
$N = 20$	1.53	93.17	0.17	0.15	10.19	0.24	
Mo/B <sub>4</sub> C	4.42	96.67	0.30	0.32	439.34	0.01	125.25
$N = 30$	6.06	90.07	0.14	0.17	65.86	0.07	
La/B <sub>4</sub> C	5.01	82.81	0.35	0.32	440.56	0.01	28.32
$N = 15$	2.95	105.63	0.11	0.31	28.92	0.30	

**Table 3.** The characterization results of different metal/B<sub>4</sub>C multilayers by using X-ray diffuse scattering technique.

Three metal/B<sub>4</sub>C multilayers [68] were chosen to measure X-ray diffuse scattering. In order to improve the analysis precision, rocking scan curves near different Bragg maximums were fitted simultaneously. According to the fitted interfacial roughness  $\sigma_r$  and the interfacial width  $\sigma$  obtained by X-ray reflectivity measurement, the interdiffusion  $\sigma_d$  can be calculated by the equation  $\sigma_d^2 = \sigma^2 - \sigma_r^2$ . As can be seen in **Table 3**, the RMS roughnesses for three multilayers are almost same, but the Mo/B<sub>4</sub>C and La/B<sub>4</sub>C multilayers have larger interdiffusions. Comparing the lateral correlation length  $\xi_{//}$  and fractal exponent  $h$ , the Mo/B<sub>4</sub>C and La/B<sub>4</sub>C multilayers have stronger lateral correlation and more apparent island growth feature than the W/B<sub>4</sub>C multilayer. Due to the weak interdiffusion and small layer thickness, the vertical correlation length in W/B<sub>4</sub>C multilayer is over 100 times the periodic thickness. In contrast, the replication capability is very weak for La/B<sub>4</sub>C multilayers. The vertical correlation length is only about four times the periodic thickness. **Figure 8** presents the diffuse scattering curves of Mo/B<sub>4</sub>C multilayers and their fitting curves.



**Figure 8.** Rocking scan curves near three Bragg maximums and their fitted curves for Mo/B<sub>4</sub>C multilayer ( $N = 30$ ).

## 6. Conclusion

The chapter describes the effective uses of optimization algorithms in design and characterization of X-ray and EUV multilayers. Based on a suitable initial gradient structure, downhill algorithm was used to design X-ray supermirrors. The results show that supermirror can produce 15 times reflection angular range compared to periodic multilayer and increase 70% integral reflective intensity. Particle swarm optimization was used to design EUV intensity beam splitters. This kind of optics realizes equal intensities of reflected and transmitted beams in a broad spectrum around 13 nm so that exposure time decreases in EUV lithography. In the characterization of multilayers, particle swarm algorithm was successfully used to determine the slight changes of structural parameters of B<sub>4</sub>C-based multilayers by fitting the hard X-ray grazing incidence reflectivity and diffuse scattering experimental data. This work compares the deposition technology and layer quality of different B<sub>4</sub>C-based multilayers and helps us to know about the evolution of interfacial defects and oxidation during the aging process.

## Acknowledgements

These works were supported by the National Natural Science Foundation of China (Grant No. 11304339), the Knowledge Innovation Program of Chinese Academy of Sciences and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

## Author details

Hui Jiang

Address all correspondence to: [jianghui@sinap.ac.cn](mailto:jianghui@sinap.ac.cn)

Shanghai Synchrotron Radiation Facility, Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai, People's Republic of China

## References

- [1] Pelizzo MG, Suman M, Monaco G, Nicolosi P, Windt DL. Mint: High performance EUV multilayer structures insensitive to capping layer optical parameters. *Optical Express*. 2008;16(19):15228–15236. DOI: 10.1364/OE.16.015228
- [2] Mirkarimi PB, Bajt S, Wall MA. Mint: Mo/Si and Mo/Be multilayer thin films on Zerodur substrates for extreme-ultraviolet lithography. *Applied Optics*. 2000;39(10):1617–1625. DOI: 10.1364/AO.39.001617
- [3] Spiller E. Mint: Low-loss reflection coatings using absorbing materials. *Applied Physics Letter*. 1972;20(9):365–367. DOI: 10.1063/1.1654189
- [4] Windt DL, Gullikson EM, Walton CC. Mint: Normal-incidence reflectance of optimized W/B<sub>4</sub>C x-ray multilayers in the range 1.0nm< $\lambda$ <2.4nm. *Optics Letters*. 2002;27(24):2212–2214. DOI: 10.1364/OL.27.002212
- [5] Liu WJ, Ice GE, Tischler JZ, Khounsary A, Liu C, Assoufid L, Macrander AT. Mint: Short focal length Kirkpatrick-Baez mirrors for a hard x-ray nanoprobe. *Review of Scientific Instruments*. 2005;76:113701. DOI: 10.1063/1.2125730
- [6] Kang HC, Yan H, Winarski RP, Holt MV, Maser J, Liu C, Conley R, Vogt S, Macrander AT, Stephenson GB. Mint: Focusing of hard x-rays to 16 nanometers with a multilayer Laue lens. *Applied Physics Letters*. 2008;92(22):221114. DOI: 10.1063/1.2912503
- [7] DuMond JWM, Youtz JP. Mint: Sensitive X-ray diffraction from artificially stratified metal films deposited by evaporation. *Physical Reviews*. 1935;48(8):703–703. DOI: 10.1103/PhysRev.48.703
- [8] Shinjo T. Mint: Interface magnetism. *Surface Science Reports*. 1991;12(2):51–98. DOI: 10.1016/0167-5729(91)90010-U
- [9] Vinogradov AV, Sagitov SI. Mint: New types of mirrors for the soft x-ray range (review). *Soviet Journal of Quantum Electronics*. 1983;13(11):1439. DOI: 10.1070/QE1983v013n11ABEH004956

- [10] Barbee T, Underwood JH. Mint: Solid Fabry-Perot etalons for X-rays. *Optics Communications*. 1983;48(3):161–166. DOI: 10.1016/0030-4018(83)90077-9
- [11] Yamamoto M, Namioka T. Mint: Layer-by-layer design method for soft-X-ray multilayers. *Applied Optics*. 1992;31(10):1622–1630 DOI: 10.1364/AO.31.001622
- [12] Seely JF, Gutman G, Wood J, Herman GS, Kowalski MP, Rife JC, Hunter WR. Mint: Normal-incidence reflectance of W/B<sub>4</sub>C multilayer mirrors in the 34–50-Å wavelength region. *Applied Optics*. 1993;32(19):3541–3547 DOI: 10.1364/AO.32.003541
- [13] Yang S, Teer DG. Mint: Investigation of sputtered carbon and carbon/chromium multilayered coatings. *Surface and Coatings Technology*. 2000;131(1–3):412–416 DOI: 10.1016/S0257-8972(00)00859-8
- [14] Kjornrattanawanich B, Bajt S, Seely JF. Mint: Multilayer-coated photodiodes with polarization sensitivity at EUV wavelength. *Proceedings of SPIE*. 2004;5168:31–34 DOI: 10.1117/12.507115
- [15] Takenaka H, Ichimaru S, Ohchi T, Gullikson EM. Mint: Soft-X-ray reflectivity and heat resistance of SiC/Mg multilayer. *Journal of Electron Spectroscopy and Related Phenomena*. 2005;144–147:1047–1049. DOI: 10.1016/j.elspec.2005.01.227
- [16] Kuhlmann AT, Yulin SA, Kaiser N, Bernitzki H, Lauth H. Mint: Design and fabrication of broadband EUV multilayers. *Proceedings of SPIE*. 2002;4688:509–515. DOI: 10.1117/12.472327
- [17] Kozhevnikov I. Mint: Design of X-ray supermirrors. *Nuclear Instruments and Methods in Physics Research A*. 2001;460:424–443. DOI:10.1016/S0168-9002(00)01079-2
- [18] Wang Z, Michette AG. Mint: Broadband multilayer mirrors for optimum use of soft x-ray source output. *Journal of Optics A Pure & Applied Optics*. 2000;2(5):452–457 DOI: 10.1088/1464-4258/2/5/317
- [19] Wang H, Zhu J, Wang Z, Zhang Z, Zhang S, Wu W, Chen L, Michette AG, Powell AK, Pfauntsch SJ, Schafer F, Gaupp A. Mint: Broadband Mo/Si multilayer analyzers for the 15–17 nm wavelength range. *Thin Solid Films*. 2006;515:2523–2526. DOI: 10.1016/j.tsf.2006.04.039
- [20] Wang Z, Wang H, Zhu J, Xu Y, Zhang S, Li C, Wang F, Zhang Z, Wu Y, Cheng X, Chen L, Michette AG, Pfauntsch SJ, Powell AK, Schafer F, Gaupp A, MacDonald M. Mint: Extreme ultraviolet broadband Mo/Y multilayer analyzers. *Applied Physics Letters*. 2006;89(24):241120. DOI: 10.1063/1.2405874
- [21] Morlens A-S, Balcou P, Zeitoun P, Valentin C, Laude V, Kazamias S. Mint: Compression of attosecond harmonic pulses by extreme-ultraviolet chirped mirrors. *Optics Letters*. 2005;30(12):1554–1556 DOI: 10.1364/OL.30.001554
- [22] Kleineberg U, Hachmann W, Heinzmann U, Hendel S, Kabachnik N, Krausz F, Neuhausler U, Uiberacker M, Uphues T, Wonisch A, Yakovlev V. Mint: Chirped

- multilayer soft X-ray mirrors for attosecond soft X-ray pulses. *Frontiers in Optics*. San Jose, California: Springer; 2007. P409–415. DOI: 10.1007/978-1-4020-6018-2
- [23] Wang H, Dhesi SS, Maccherozzi F, Cavill S, Shepherd E, Yuan F, Deshmukh R, Scott S, van der Laan G, Sawhney KJ. Mint: High-precision soft x-ray polarimeter at Diamond Light Source. *Review of Scientific Instruments*. 2011;82:123301. DOI: 10.1063/1.3665928
- [24] Mimura H, Handa S, Kimura T, Yumoto H, Yamakawa D, Yokoyama H, Matsuyama S, Inagaki K, Yamamura K, Sano Y, Tamasaku K, Nishino Y, Yabashi M, Ishikawa T, Yamauchi K. Mint: Breaking the 10 nm barrier in hard-X-ray focusing. *Nature Physics*. 2010; 6(146):122–125. DOI: 10.1038/nphys1501
- [25] Rack A, Weitkamp T, Riotte M, Grigoriev D, Rack T, Helfen L, Baumbach T, Dietsch R, Holz T, Kramer M, Siewert F, Meduna M, Cloetens P, Ziegler E. Mint: Comparative study of multilayers used in monochromators for synchrotron-based coherent hard X-ray imaging. *Journal of Synchrotron Radiation*. 2010;17: 496–510. DOI: 10.1107/S0909049510011623
- [26] Eriksson F, Johansson GA, Hertz HM, Gullikson EM, Kreissig U, Birch J. Mint: 14.5% near-normal incidence reflectance of Cr/Sc x-ray multilayer mirrors for the water window. *Optics Letters*. 2003;28(24):2494–2496. DOI: 10.1364/OL.28.002494
- [27] Jiang H, Michette AG, Pfauntsch SJ, Hart D, Shand M. Design of narrowband multilayer for Cr K $\alpha$  X-rays. In: *Progress In Electromagnetics Research Symposium Proceedings*; 22–26 March 2010; Xi'an, China; 2010, pp. 61–65
- [28] Underwood JH, Bruner ME, Haisch BM, Brown WA, Acton LW. Mint: X-ray photographs of a solar active region with a multilayer telescope at normal incidence. *Science*. 1987;238(4823): 61–64. DOI: 10.1126/science.238.4823.61
- [29] Levashov VE, Mednikov KN, Pirzhkov AS, Ragozin EN. Mint: Aperiodic X-ray multilayer and their application in plasma spectroscopy. *Radiation Physics & Chemistry*. 2006;75(11):1819–1823. DOI:10.1016/j.radphyschem.2005.07.036
- [30] Mishra D, Benitez M, Petravic O, Badini Confalonieri GA, Szary P, Brüssing F, Theis-Brohl K, Devishvili A, Vorobiev A, Konovalov O, Paulus M, Sternemann C, Toperverg BP, Zabei H. Mint: Self-assembled iron oxide nanoparticle multilayer: x-ray and polarized neutron reflectivity. *Nanotechnology*. 2012;23:055707. DOI: 10.1088/0957-4484/23/5/055707
- [31] Parratt LG. Mint: Surface studies of solids by total reflection of x-rays. *Physical Review*. 1954;95(2):359–369. DOI: 10.1103/PhysRev.95.359
- [32] Croce P and Nevot L. Mint: Characterization of surfaces by grazing X-Ray reflection - application to study of polishing of some silicate-glasses. *Rev Phys Appl.* 1980;503(43): 761–779. DOI: 10.1051/rphysap:01980001503076100
- [33] Stepanov SA, Kondrashkina EA, Schmidbauer M, Kohler R, Pfeiffer J-U, Jach T, Souvorov AY. Mint: Diffuse scattering from interface roughness in grazing-incidence

- X-ray diffraction. *Physical Review B*. 1996;54:8150–8162. DOI: /10.1103/PhysRevB.54.8150
- [34] Sinha SK, Sirota EB, Garoff S, Stanley HB. Mint: X-ray and neutron scattering from rough surfaces. *Physical Review B*. 1988;38:2297–2311. DOI: 10.1103/PhysRevB.38.2297
- [35] Chen L, Deng N, Zhang J. Mint: A modified quasi-Newton method for structured optimization with partial information on the Hessian. *Computational Optimization and Applications*. 2006;35:5–18. DOI: 10.1007/s10589-006-6440-6
- [36] Petrova S and Solovev A. Mint: The origin of the method of steepest descent. *Historia Mathematica* 1997;24(4):361–375. DOI: 10.1006/hmat.1996.2146
- [37] Levenberg K. Mint: A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*. 1944;2:164–168. DOI: 10.1515/cppm-2013-0011
- [38] Marquardt DW. Mint: An algorithm for Least-Squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*. 1963;11(2):431–441. DOI:10.1137/0111030
- [39] Nelder JA, Mead R. Mint: A Simplex method for function minimization. *The Computer Journal*. 1965;7:308–313. DOI: 10.1093/comjnl/7.4.308
- [40] Tikhonravov AV, Trubetskov MK, DeBell GW. Mint: Application of the needle optimization technique to the design of optical coatings. *Applied Optics*. 1996;35(28): 5493–5508. DOI: 10.1364/AO.35.005493
- [41] Tikhonravov AV, Trubetskov MK, DeBell GW. Mint: Optical coating design approaches based on the needle optimization technique. *Applied Optics*. 2007;46(5): 704–710. DOI: 10.1364/AO.46.000704
- [42] Ali MM, Storey C. Mint: Modified controlled random search algorithms. *International Journal of Computer Mathematics*. 1994;53:229–235. DOI: 10.1080/00207169408804329
- [43] Martin S, Rivory J, Schoenauer M. Mint: Synthesis of optical multilayer systems using genetic algorithms. *Applied Optics*. 1995;34(13):2247–2254. DOI: 10.1364/AO.34.002247
- [44] Dekkers A, Aarts E. Mint: Global optimization and simulated annealing. *Mathematical Programming*. 1991;50:367–393. DOI: 10.1007/BF01594945
- [45] Wormington M, Panaccione C, Matney K M, Bowen D K. Mint: Characterization of structures from X-ray scattering data using genetic algorithms. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*. 1999;357(1761):2827–2848. DOI:10.1098/rsta.1999.0469
- [46] Törn A, Viitanen S. Mint: Topographical global optimization using pre-sampled points. *Journal of Global Optimization*. 1994;5(3):267–276. DOI: 10.1007/BF01096456

- [47] Kennedy J, Eberhart RC. Mint: Particle swarm optimization. IEEE International Conference on Neural Network. 1995;4(8):1942–1948 (1995). DOI: 10.1109/ICNN.1995.488968
- [48] Dorigo M, Blum C. Mint: Ant colony optimization theory: a survey. Theoretical Computer Science. 2005;344(2–3):243–378. DOI:10.1016/j.tcs.2005.05.020
- [49] Huang M, Aine CJ, Supek S, Best E, Ranken D, Flynn ER. Mint: Multi-start downhill simplex method for spatio-temporal source localization in magnetoencephalography. Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section. 1998;108(1):32–44. DOI:10.1016/S0168-5597(97)00091-9
- [50] Eberhart RC, Shi Y. Comparison between genetic algorithm and particle swarm optimization. In: Porto VW, Saravanan N, Waagen D, Eiben AE, editors. Evolutionary Programming VII. Berlin Heidelberg: Springer; 1998, pp. 611–616. DOI: 10.1007/BFb0040812
- [51] Trelea IC. Mint: The particle swan optimization algorithm: convergence analysis and parameter selection. Information Processing Letters. 2003;85:317–325. DOI: 10.1016/S0020-0190(02)00447-7
- [52] Jiang H. Characteristics of multilayer mirrors in the X-ray and extreme ultraviolet radiation ranges [thesis]. London: King's College London; 2012
- [53] HΦghΦj P, Ziegler E, Susini J, Freund AK, Joensen KD, Gorenstein P, Wood JL. Mint: Focusing of hard X-rays with a W/Si supermirror. Nuclear Instruments and Methods in Physics Research B. 1997;132:528–533. DOI: 10.1016/S0168-583X(97)00437-0
- [54] Boni P. Mint: Supermirror-based beam devices. Physica B: Condensed Matter. 1997;234–236:1038–1043. DOI:10.1016/S0921-4526(96)01255-0
- [55] Tamura K, Yamashita K, Tawara Y, Ogasaka Y, Satake Y, Nomoto K, Okajima T. Mint: Development of supermirror telescope up to 80 keV. Proceedings of SPIE. 2003;4851:1048–1058. DOI: 10.1117/12.461318
- [56] Erko A, Schafers F, Vidal B, Yakshin A, Pietsch U, Mahler W. Mint: X-ray supermirrors for BESSY II. Review of Scientific Instruments. 1995;66(10):4845–4846. DOI: 10.1063/1.1146162
- [57] Wang Z, Wang H, Zhu J, Wang F, Gu Z, Chen L, Michette AG, Powell AK, Pfauntsch SJ, Schafers F. Mint: Broad angular multilayer analyzer for soft X-rays. Optics Express. 2006;14(6):2533–2538. DOI: 10.1364/OE.14.002533
- [58] Wang Z, Wang H, Zhu J, Wang F, Gu Z, Chen L, Michette AG, Powell AK, Pfauntsch SJ, Schafers F. Mint: Broadband multilayer polarizers for the extreme ultraviolet. Journal of Applied Physics. 2006;99:056108. DOI: 10.1063/1.2179152

- [59] Benoit N, Schroder S, Yulin S, Feigl T, Duparre A, Kaiser N, Tunnermann A. Mint: Extreme-ultraviolet-induced oxidation of Mo/Si multilayers. *Applied Optics*. 2008;47(19):3455–3462. DOI: doi: 10.1364/AO.47.003455
- [60] Haga T, Takenaka H, Fukuda M. Mint: At-wavelength extreme ultraviolet lithography mask inspection using a Mirau interferometric microscopy. *Journal of Vacuum Science & Technology B*. 2000;18(6):2916–2920. DOI: 10.1116/1.1319702
- [61] Budano A, Flora F, Mezi L. Mint: Analytical design method for a modified Schwarzschild optics. *Applied Optics*. 2006;45(18):4254–4262. DOI: 10.1364/AO.45.004254
- [62] Suman M, Pelizzo MG, Nicolosi P, Windt DL. Mint: Aperiodic multilayers with enhanced reflectivity for extreme ultraviolet lithography. *Applied Optics*. 2008;47(16):2906–2914. DOI: 10.1354/AO.47.002906
- [63] Hiruma K, Miyagaki S, Yamanashi H, Tanaka Y, Nishiyama I. Mint: Performance and quality analysis of Mo–Si multilayers formed by ion-beam and magnetron sputtering for extreme ultraviolet lithography. *Thin Solid Films*. 2008;516(8):2050–2057. DOI: 10.1016/j.tsf.2007.07.182
- [64] Skulina KM, Alford CS, Bionta Rm, Makowiecki DM, Gullikson EM, Soufli R, Kortright JB, Underwood JH. Mint: Molybdenum/beryllium multilayer mirrors for normal incidence in the extreme ultraviolet. *Applied Optics*. 1995;34(19):3727–3730. DOI: 10.1364/AO.34.003727
- [65] Jiang H, Michette AG. Mint: Robust design of broadband EUV multilayer beam splitters based on particle swarm optimization. *Nuclear Instruments and Methods in Physics Research A*. 2013;703:22–25. DOI: 10.1016/j.nima.2012.11.038
- [66] Jiang H, Wang Z, Zhu J. Mint: Interface characterization of B<sub>4</sub>C-based multilayers by X-ray grazing-incidence reflectivity and diffuse scattering. *Journal of Synchrotron Radiation*. 2013;20:449–454. DOI: 10.1107/S0909049513004329
- [67] Jiang H, Zhu J, Huang Q, Xu J, Wang X, Wang Z, Pfauntsch SJ, Michette AG. Mint: The influence of residual gas on boron carbide thin films prepared by magnetron sputtering. *Applied Surface Science*. 2011;257(23):9946–9952. DOI:10.1016/j.apsusc.2011.06.113
- [68] Jiang H, He Y, He Y, Li A, Wang H, Zheng Y, Dong Z. Mint: Structural characterization and low-temperature properties of Ru/C multilayer monochromators with different periodic thicknesses. *Journal of Synchrotron Radiation*. 2015;22:1379–1385. DOI: 10.1107/S1600577515017828

# A Clustering Approach Based on Charged Particles

Yugal Kumar, Sumit Gupta,  
Dharmender Kumar and Gadadhar Sahoo

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63081>

## Abstract

In pattern recognition, clustering is a powerful technique that can be used to find the identical group of objects from a given dataset. It has proven its importance in various domains such as bioinformatics, machine learning, pattern recognition, document clustering, and so on. But, in clustering, it is difficult to determine the optimal cluster centers in a given set of data. So, in this paper, a new method called magnetic charged system search (MCSS) is applied to determine the optimal cluster centers. This method is based on the behavior of charged particles. The proposed method employs the electric force and magnetic force to initiate the local search while Newton second law of motion is employed for global search. The performance of the proposed algorithm is tested on several datasets which are taken from UCI repository and compared with the other existing methods like K-Means, GA, PSO, ACO, and CSS. The experimental results prove the applicability of the proposed method in clustering domain.

**Keywords:** clustering, charged particles, electric force, magnetic force, Newton Law

## 1. Introduction

Clustering is an unsupervised technique which can be applied to understand the organization of data. The basic principle of clustering is to partition a set of objects into a set of clusters such that the objects within a cluster share more similar characteristics in comparison to the other clusters. A pre-specified criterion has been used to measure the similarity between the objects. In clustering, there is no need to train the data, it only deals with the internal structure of data and used a similarity criterion to group the objects into different clusters. Due to this, it is also known as unsupervised classification technique. It becomes a NP hard problem when number of clusters is greater than three. Consider a set  $S = [A_1, A_2, A_3, \dots, A_N]$  such that  $A_i$  S, consist

of  $N$  number of data objects and another set  $P = [B_1, B_2 \dots B_K]$  consist of  $K$  cluster centers. The objective of clustering is to arrange each data object from the set  $S$  with one of the cluster center  $j$  of set  $P$  such that the value of objective function is minimized. The objective function is defined as sum of squared Euclidean distance between the data object  $A_i$  and cluster center  $B_j$  and it can be described as follows:

$$D = \sum_{i=1}^N \min_j \|A_i - B_j\|^2, j = 1, 2, 3, \dots, K$$

where  $A_i$  denotes the  $i^{\text{th}}$  data objects,  $B_j$  denotes the  $j^{\text{th}}$  cluster center, and  $D$  denotes the distance between  $i^{\text{th}}$  data objects from the  $j^{\text{th}}$  cluster center. It is also noted that:

- Each cluster at least consists of one data object.  
 $B_j \neq \emptyset$ , for all  $j \in \{1, 2, 3 \dots K\}$ , where  $B_j$  represents the  $j^{\text{th}}$  cluster and  $K$  denotes the total number of clusters.
- Each data object is allotted to only one cluster.  
 $B_i \cap B_j = \emptyset$  for all  $i \neq j$  and  $i, j \in \{1, 2, 3 \dots K\}$  such that  $i^{\text{th}}$  and  $j^{\text{th}}$  clusters do not consists same data objects.
- Each data objects should be allocated to a cluster.  
 $\bigcup_{j=1}^K B_j = S$  where  $S$  represents the set of data objects.

Hence, the aim of the partition based clustering algorithm is to determine the  $K$  number of cluster centers in a given dataset. Here, the MCSS algorithm is applied for determining the optimal cluster centers in a dataset.

Clustering has proven its importance in many applications successfully. Some of these are pattern recognition [1, 2], image processing [3–6], process monitoring [7], machine learning [8, 9], quantitative structure activity relationship [10], document retrieval [11], bioinformatics [12], image segmentation [13], construction management [14], marketing [15, 16] and healthcare [17, 18]. Broadly, clustering algorithms can be divided into two categories - partition based clustering algorithms and hierarchal clustering algorithms. In partition based clustering algorithms, partition a dataset into  $k$  clusters on the basis of some fitness functions [19]. While in hierarchical clustering algorithms, clustering of data occurs in the form of tree representation and this representation is known as dendrogram. Hierarchical clustering algorithms do not require any prerequisite knowledge about number of clusters in a dataset but its only drawback is lacking of dynamism as the objects are tightly bound with respective clusters [20–23]. However, our research is focused on partition clustering, which decomposes the data into several disjoint clusters that are optimal in terms of some predefined criteria. From the literature, it is found that K-means algorithm is the oldest, most popular, and extensively used partition based algorithm for data clustering. It is easy, fast, simple structure, and having linear time complexity [24, 25]. In K-means algorithm, a dataset is decomposed into a predefined number of clusters and the data into distinct clusters based on the euclidean distance [25].

Nowadays, heuristic approaches gain wide popularity to solve the clustering problem and become more successful. Numerous researchers have been applying heuristic approaches in the field of clustering. Some of these are summarized as simulated annealing [26], tabu search [27, 28], genetic algorithm [29–32], particle swarm optimization [33, 34], ant colony optimization [35], artificial bee colony algorithm [36, 37, 56], charged system search algorithm [38, 39], cat swarm optimization [40–42, 57], teacher learning based optimization method [43, 44], gravitational search algorithm [45, 46] and binary search based clustering algorithm [47].

## 2. Magnetic charge system search (MCSS) algorithm

The magnetic charged system search (MCSS) algorithm is a recent meta-heuristic algorithm based on electromagnetic theory [48]. According to electromagnetic theory, moving charged particles produce an electric field as well as a magnetic field. Movement of the charged particles in a magnetic field enforces a magnetic force on the other charged particles and the resultant force is proportional to the charge (mass) and speed of the charged particles. The magnitude and direction of the resultant force depend on the two factors: first, velocity of the charged particles, and secondly, magnitude and direction of the magnetic field. Thus, the MCSS algorithm is further advancement in the charge system search (CSS) algorithm using the concept of electromagnetic theory. The difference between the CSS and MCSS is that the CSS algorithm considers only the electric force to determine the movement of CPs while the MCSS utilizes both the forces (electric and magnetic) to determine the same. Along this, MCSS can be either attractive or repulsive in nature. This nature of MCSS algorithm generates more promising solutions in random space. On the other hand, CSS algorithm is attractive by nature. Thus, the performance of the algorithm can be affected with small number of CPs. So, the addition of the magnetic force to the already existing electric force, results in enhancement of both the exploration and exploitation capabilities of CSS and this makes the algorithm more realistic one. Hence, the inclusion of magnetic force in the charge system search (CSS) algorithm results in the formation of a new algorithm known as magnetic charge system search (MCSS). The main steps of the MCSS algorithm are as follows.

### Step 1: Initialization

Algorithm starts by identifying the initial positions of charged particles (CPs) in d-dimensional space in random order and set the initial velocities of CPs is zero. To determine the initial positions of CPs, equation 1 is used. A variable charge memory (CM) is used to store the best results.

$$C_k = X_{j_{\min}} + r_j * ((X_{j_{\max}} - X_{j_{\min}}) / K), \text{ where } j = 1, 2, \dots, d \text{ and } k = 1, 2, \dots, K \quad (1)$$

where,  $C_k$  denotes the  $k^{\text{th}}$  cluster center for a given dataset,  $r_j$  is a random number in the range of 0 and 1,  $X_{j_{\min}}$  and  $X_{j_{\max}}$  denote the minimum and maximum value of the  $j^{\text{th}}$  attribute of the dataset, and  $K$  represents the total number of clusters in a dataset.

**Step 2: Compute the total force ( $F_{\text{total}}$ ) acts on CPs.**

The total force is the combination of the electric force and magnetic force, and this force influences the movement of CPs in d-dimensional space. It can be computed as follows:

- Determine the electric force – when CPs move in d-dimensional space, an electric field is produced surrounding it, and exerted an electric field on the other CPs. This electric force is directly proportional to the magnitude of its charge and the distance between CPs. The magnitude of an electric force generated by a charge particle is enforced on another charge particle and it can be measured using equation 2.

$$E_{ik} = q_k \sum_{i,i \neq k} \left( \frac{q_i}{R^3} * w_1 + \frac{q_i}{r_{ik}^2} * w_2 \right) * p_{ik} * (X_i - C_k), \begin{cases} k = 1, 2, 3, \dots, K \\ w_1 = 1, w_2 = 0 \leftrightarrow r_{ik} < R \\ w_1 = 0, w_2 = 1 \leftrightarrow r_{ik} \geq R \end{cases} \quad (2)$$

In equation 2,  $q_i$  and  $q_k$  represents the fitness values of  $i^{\text{th}}$  and  $k^{\text{th}}$  CP,  $r_{ik}$  denotes the separation distance between  $i^{\text{th}}$  and  $k^{\text{th}}$  CPs,  $w_1$  and  $w_2$  are the two variables whose values are either 0 or 1,  $R$  represents the radius of CPs which is set to unity and it is assumed that each CPs has uniform volume charge density but changes in every iteration, and  $p_{ik}$  denotes the moving probability of each CPs.

- Determine the magnetic force – The movement of CPs also produce magnetic field along with the electric field. As a result of this, a magnetic force is imposed on the other CPs and equation 3 is utilized to compute the magnitude of magnetic force exerted by a CP on other CPs. It can be either positive or negative depending on the value of average electric current of the previous iteration.

$$M_{ik} = q_k \sum_{i,i \neq k} \left( \frac{I_i}{R^2} * r_{ik} * w_1 + \frac{I_i}{r_{ik}} * w_2 \right) * PM_{ik} * (X_i - C_k), \begin{cases} k = 1, 2, 3, \dots, K \\ w_1 = 1, w_2 = 0 \leftrightarrow r_{ik} < R \\ w_1 = 0, w_2 = 1 \leftrightarrow r_{ik} \geq R \end{cases} \quad (3)$$

In equation 3,  $q_k$  represents the fitness values of the  $k^{\text{th}}$  CP,  $I_i$  is the average electric current,  $r_{ik}$  denotes the separation distance between  $i^{\text{th}}$  data instance and  $k^{\text{th}}$  CPs,  $w_1$  and  $w_2$  are the two variables whose values are either 0 or 1,  $R$  represents the radius of CPs which is set to unity, and  $PM_{ik}$  denotes the probability of magnetic influence between  $i^{\text{th}}$  data instance and  $k^{\text{th}}$  CP. In other words, it can be summarized that the magnetic force can be either attractive or repulsive in nature. As a result of this, more promising solutions can be generated during the search. Whereas, the electric force is always attractive in nature. Therefore, this nature of electric force may influence the performance of the algorithm. Hence, to overcome the repulsive nature, a probability function is added with the electric force and finally, the total force acting on other CPs can be computed using equation 4.

$$F_{\text{total}} = p_r * E_{ik} + M_{ik} \quad (4)$$

Where,  $p_r$  denotes a probability value to determine either the electric force ( $E_{ik}$ ) repelling or attracting,  $E_{ik}$  and  $M_{ik}$  present the electric and magnetic forces exerted by the  $k^{\text{th}}$  CP to  $i^{\text{th}}$  data instance.

### Step 3: Determine the new positions and velocities of CPs.

Newton second law of motion is applied to determine the movement of CPs. The magnitude of the total force with Newtonian laws is used to produce the next positions and velocities of CPs. The new positions and velocities of CPs can be computed using equation 5 and 6.

$$C_{k \text{ new}} = \text{rand}_1 * Z_a * \frac{F_{\text{total}}}{m_k} * \Delta t^2 + \text{rand}_2 * Z_v * V_{k \text{ old}} * \Delta t + C_{k \text{ old}} \quad (5)$$

Where,  $\text{rand}_1$  and  $\text{rand}_2$  are the two random variable in the range of 0 and 1,  $Z_a$  and  $Z_v$  act as control parameters to control the influence of total force ( $F_{\text{total}}$ ), and  $V_{k \text{ old}}$  denotes the velocity of  $k^{\text{th}}$  CPs,  $m_k$  is the mass of  $k^{\text{th}}$  CPs which is equal to the  $q_k$ ,  $\Delta t$  represents the time step which is set to 1, and  $C_{k \text{ old}}$  denotes the position of  $k^{\text{th}}$  current CP.

$$V_{k \text{ new}} = \frac{C_{k \text{ new}} - C_{k \text{ old}}}{\Delta t} \quad (6)$$

where  $V_{k \text{ old}}$  denotes the new velocity of  $k^{\text{th}}$  CP,  $C_{k \text{ old}}$  and  $C_{k \text{ new}}$  represents the old and new position of  $k^{\text{th}}$  CP, and  $\Delta t$  represents the time stamp.

### Step 4: Update charge memory (CM)

CPs with better objective function values replace the worst CPs from the CM and store the positions of new CPs in CM.

### Step 5: Termination condition

If the maximum iterations is reached and condition is satisfied, then stop the algorithm and obtain the optimal cluster centers. Otherwise repeat steps 2–4.

## 2.1. Pseudo code of MCSS algorithm for clustering

This section summarizes the pseudo code of the MCSS algorithm for clustering tasks.

Step 1: Load the dataset and initialize the parameters of MCCS algorithm.

Step 2: Initialize the initial positions and velocities of Charged Particles (CPs).

Step 3: Compute the value of objective function using equation 7 and arrange the data instances to the clusters using minimum value of objective function.

$$d_{ik} = \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^d \sqrt{X_{ji} - C_{jk}}^2 \quad (7)$$

where,  $X_{ji}$  denotes the  $j^{\text{th}}$  attribute of the  $i^{\text{th}}$  data instance,  $C_{jk}$  represents the  $j^{\text{th}}$  attribute of the  $k^{\text{th}}$  CP, and  $d_{ik}$  denotes Euclidean distance between  $i^{\text{th}}$  data instance from the  $k^{\text{th}}$  CP.

Step 4: Compute the mass of initial positioned CPs.

Step 5: Store the positions of initial CPs ( $C_k$ ) into a variable, called charge memory (CM).

Step 6: While the termination conditions are not met, compute the value of Electric Force ( $E_{ik}$ ) for each CPs as follows:

Step 6.1: Calculate the value of moving probability ( $P_{ik}$ ) for each charged particle  $C_k$ .

Step 6.2: Compute the fitness of each instance  $q_i$ .

Step 6.3: Compute the separation distance ( $r_{ik}$ ) of CPs.

Step 6.4: Compute the value of  $(X_i - C_k)$ .

Step 6.5: Compute the value of Electric Force ( $E_{ik}$ ) for each CPs

Step 7: Determine the value of Magnetic Force ( $M_{ik}$ ) for each CPs.

Step 7.1: Compute the value of average electric current ( $I_i$ ).

Step 7.2: Compute the probability of magnetic influence ( $PM_{ik}$ ).

Step 7.3: Compute the value of Magnetic Force ( $M_{ik}$ ) for each CPs.

Step 8: Compute the total force ( $F_{\text{total}}$ ) act on each CPs.

Step 9: Calculate the new positions and velocities of charged particles using equation 5 and 6.

Step 10: Recalculate the value of objective function using new positions of charge particles.

Step 11: Compare the newly generated charge particles to the charge particles reside in CM.

Step 12: Memorize the best solution achieved so far and  $\text{Iteration} = \text{Iteration} + 1$ ;

Step 13: Output the best solution obtained.

### 3. Experimental results

This section deals with the experimental setup of our study. It includes the performance measures, parameters settings, datasets to be used, experiment results, and statistical analysis. To prove the effectiveness of the MCSS algorithm, 10 datasets are applied in which two datasets are artificial ones and the rest are taken from UCI repository. The proposed algorithm is

implemented in MATLAB 2010a using a computer with window operating, corei3 processor, 3.4 GHz and 4 GB RAM. Experimental outcomes of MCSS algorithm are compared with other clustering algorithms like K-means, GA [30], PSO [49], ACO [35], and CSS [38].

### 3.1. Performance measures

The performance of MCSS algorithm is examined over the sum of intra cluster distance and F-measure parameters. The sum of intra cluster distance can be measured in terms of best case, average case, and worst case solutions including standard deviation parameter which shows the dispersion of the data. F-measure parameter is used to measure the accuracy of proposed method. Performance measures are described as follows:

#### *Intra cluster distances*

Intra cluster distance can be used to measure the quality of clustering [35, 36]. It indicates the distance between the data objects within a cluster and its cluster center. This parameter also highlights the quality of clustering i.e. minimum is the intra cluster distance, better will be the quality of the solution. The results are measured in terms of best, average, and worst solutions.

#### *Standard Deviation (Std.)*

Standard deviation gives the information about the scattering of data within a cluster [47, 49]. Lower value of standard deviation indicates that the data objects are scattered near its center, while high value indicates that the data is dispersed away from its center point.

#### *F-Measure*

This parameter is measured in terms of recall and precision of an information retrieval system [50, 51]. It is also described in terms of weighted harmonic mean of recall and precision. Recall and precision of an information retrieval system is computed using equation 8 which can be described as follows:

$$\text{Recall}(r(i, j)) = \frac{n_{i,j}}{n_i} \text{ and Precision}(p(i, j)) = \frac{n_{i,j}}{n_j} \quad (8)$$

The value of F-measure ( $F(i, j)$ ) can be computed using equation 9.

$$F(i, j) = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (9)$$

Finally, the value of F-measure for a given clustering algorithm which consists of n number of data instances is calculated using equation 10.

$$F(i,j)=\sum_{i=1}^n\frac{n_i}{n}*\max_i*F(i,j)$$

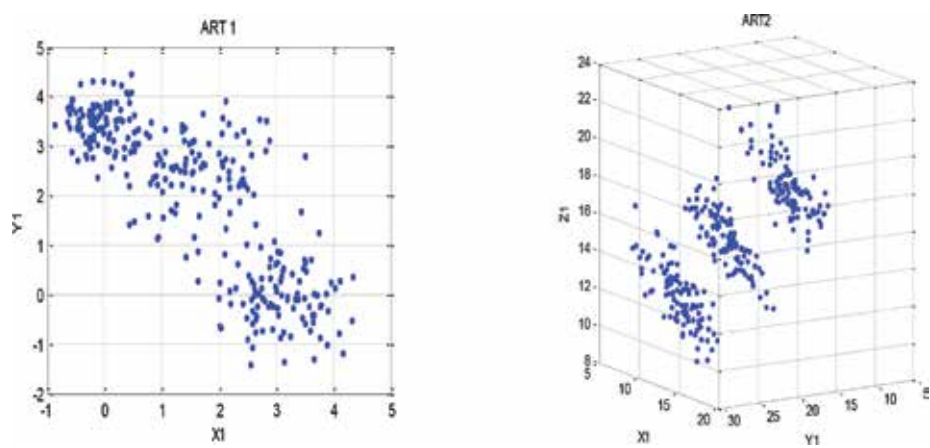
(10)

3.2. Parameters settings

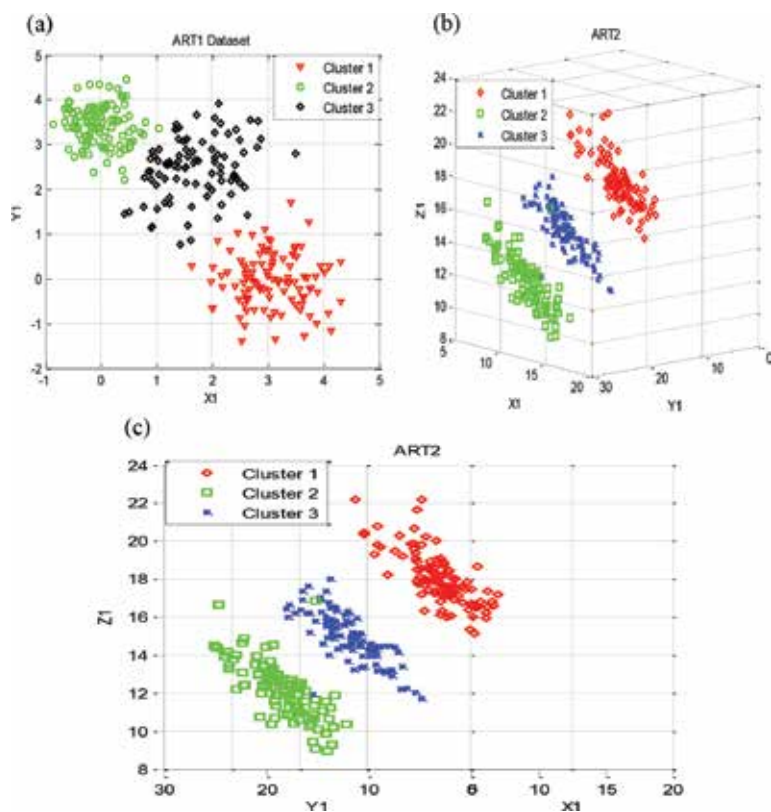
In order to evaluate the performance of the proposed algorithm, user defined parameters are to be used prior to the process. In MCSS, there are four user defined parameters such as number of CPs, rand, R and  $\epsilon$ . The details of the parameters as follows: the number of CPs is equal number of clusters present in a dataset, rand is a random function that provides a value in the range of 0 and 1, R denotes the radius of CPs and it is set as 1,  $\epsilon$  is also a user defined parameter which is used to prevent the singularity and it is set to 0.001. In addition to it, number of iterations for algorithm must be specified. Therefore, maximum iteration number is set to 100 and results are summed over 10 runs of the algorithm using different initial cluster centers for each dataset. **Table 1** summarizes the parameters setting of MCSS algorithm. It is also mentioned that the performance of the proposed algorithm is compared with the K-means, GA, PSO, ACO, and CSS. The parameter settings of these algorithms are set accordingly as reported (**Figures 1 and 2; Table 2**).

Parameters	Value
No. of CPs	No. of Clusters
rand	random value between [0, 1]
R	1
$\epsilon$	0.001

**Table 1.** Parameters setting of MCSS algorithm.



**Figure 1.** (a): Distribution of data in ART1. (b): Distribution of data in ART2.



**Figure 2.** (a): Clustering in ART1 dataset. (b): Clustering the ART2 dataset. (c): Clustering the ART1 dataset using MCSS (Vertical view as X1 and Y1 coordinate in horizontal plane and Z1 coordinate in vertical plane).

Dataset	Classes	Attributes	Total instances	Instance in each classes
ART 1	3	2	300	(100, 100, 100)
ART 2	3	3	300	(100, 100, 100)
Iris	3	4	150	(50, 50, 50)
Glass	6	9	214	(70,17, 76, 13, 9, 29)
LD	2	6	345	(145, 200)
Thyroid	3	3	215	(150, 30, 35)
Cancer	2	9	683	(444, 239)
CMC	3	9	1473	(629,334, 510)
Vowel	6	3	871	(72, 89, 172, 151, 207, 180)
Wine	3	13	178	(59, 71, 48)

**Table 2.** Description of datasets.

### 3.3. Experiment results

This subsection demonstrates the results of the proposed algorithm. The results of the proposed algorithm are compared with other existing techniques like K-means, GA, PSO, ACO, and CSS using a mixture of datasets [53–55]. Two artificial and eight real life datasets are used to obtain the results [52]. In real life datasets, iris, thyroid, and vowel datasets are categorized as low dimensional datasets, while cancer and LD datasets are moderate ones, and the rest of datasets (wine, CMC, and glass) are high dimensional. For enrich visualization and understanding, the results are discussed with one dataset at a time.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
ART 1	Best	157.12	154.46	154.06	154.37	153.91	153.18
	Average	161.12	158.87	158.24	158.52	158.29	158.02
	Worst	166.08	164.08	161.83	162.52	161.32	159.26
	Std	0.34	0.281	0	0	0	0
	F-Measure	99.14	99.78	100	100	100	100

**Table 3.** Comparison of the proposed MCSS algorithm with other clustering algorithms using ART1 dataset.

**Table 3** illustrates the results of the proposed method as well as other clustering algorithms (in terms of intra cluster distance: best, average and worst, standard deviation, and F-measure parameters) for ART1 dataset. It is seen that the K-means exhibits the poor performance among all the techniques being compared using all of the parameters. From the results, it also noticed that performance of the PSO, ACO, CSS, and MCSS are almost similar except intra cluster parameter. On the behalf of intra cluster parameter, it can be said that the MCSS algorithm achieves minimum distance in comparison to all other algorithms.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
ART2	Best	743	741.71	740.29	739.81	738.96	737.85
	Average	749.83	747.67	745.78	746.01	745.61	745.12
	Worst	754.28	753.93	749.52	749.97	749.66	748.67
	Std	0.516	0.356	0.237	0.206	0.209	0.17
	F-Measure	98.94	99.17	99.26	99.19	99.43	99.56

**Table 4.** Comparison of the proposed MCSS algorithm with other clustering algorithms using ART2 dataset.

**Table 4** summarizes the results of all the techniques for artificial dataset ART2. From the results, it is clearly shown that a significant difference occurred between the results of the proposed algorithm and other algorithms. The proposed algorithm outperforms using all of the parameters. Again, it is observed that the performance of the K-means algorithm is poor

among all the methods. It is also stated that the results of the CSS and PSO algorithms are close to the optimal solution, but with slightly high value of standard deviation parameter.

**Table 5** displays the results of the proposed algorithm and other algorithms for iris dataset. From this table, it came to notice that results obtained using GA is far from the optimal solutions, while the proposed method again gives the superior results. It is also observed that K-means algorithm gives the better results with iris dataset; especially it performs well over GA and ACO algorithms. It is also noticed that results of the K-means algorithm is very close to the PSO algorithm (in terms of F-measure parameter).

Results of all six methods for wine dataset are listed in **Table 6**. It demonstrates that MCSS algorithm obtains good results (in terms of intra cluster distance and F-measure) in comparison to others, but slightly large value of standard deviation parameter. On the other hand, it is also stated that the performance of GA, PSO, and ACO are nearly same except some variation between F-measure parameter. Again, the performance of the K-means algorithm is better than the GA, PSO, and ACO in terms of F-measure parameter but with a large value of standard deviation parameter. CSS algorithm also gives good performance with wine dataset except MCSS, and obtains low value for standard deviation parameter which shows that in each iteration, a near optimal solution is generated.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Iris	Best	97.33	113.98	96.89	97.1	96.47	96.34
	Average	106.05	125.19	97.23	97.17	96.63	96.57
	Worst	120.45	139.77	97.89	97.8	96.78	96.63
	Std	14.631	14.563	0.347	0.367	0.14	0.1
	F-Measure	0.782	0.778	0.782	0.779	0.787	0.790

**Table 5.** Comparison of the proposed MCSS algorithm with other clustering algorithms using iris dataset.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Wine	Best	16555.68	16530.53	16345.96	16530.53	16282.12	16158.56
	Average	18061	16530.53	16417.47	16530.53	16289.42	16189.96
	Worst	18563.12	16530.53	16562.31	16530.53	16317.67	16223.61
	Std	793.213	0	85.497	0	10.31	36.72
	F-Measure	0.521	0.515	0.518	0.519	0.529	0.537

**Table 6.** Comparison of the proposed MCSS algorithm with other clustering algorithms using wine dataset.

**Table 7** describes the results of all the six algorithms using LD dataset. From the results, it is clearly seen that the outcomes of the proposed algorithm is better in comparison to the other algorithms. It is also noted that K-means algorithm does not perform well with LD dataset, and results obtained using K-means are far-far away from the optimal ones. Again, it came

into revelation that the performance of PSO algorithm is better except MCSS algorithm and its results are close to the optimal solutions.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
LD	Best	11397.83	532.48	209.15	224.76	207.09	206.14
	Average	11673.12	543.69	224.47	235.16	228.27	221.69
	Worst	12043.12	563.26	239.11	256.44	242.14	236.23
	Std	667.56	41.78	29.38	17.46	18.54	12.07
	F-Measure	0.467	0.482	0.493	0.487	0.491	0.495

**Table 7.** Comparison of the proposed MCSS algorithm with other clustering algorithms using LD dataset.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Cancer	Best	2999.19	2999.32	2973.5	2970.49	2946.48	2932.43
	Average	3251.21	3249.46	3050.04	3046.06	2961.16	2947.74
	Worst	3521.59	3427.43	3318.88	3242.01	3006.14	2961.03
	Std	251.14	229.734	110.801	90.5	12.23	10.33
	F-Measure	0.829	0.819	0.819	0.821	0.847	0.859

**Table 8.** Comparison of the proposed MCSS algorithm with other clustering algorithms using cancer dataset.

Results of all the six algorithm for cancer dataset is listed in **Table 8**. As it indicates, the performance of the GA and PSO algorithms are not so good with cancer dataset. But the proposed algorithm works well and achieves respectable results as compared to others. K-means algorithm also achieves good results over GA, ACO, and PSO algorithms.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
CMC	Best	5842.2	5705.63	5700.98	5701.92	5672.46	5653.26
	Average	5893.6	5756.59	5820.96	5819.13	5687.82	5678.83
	Worst	5934.43	5812.64	5923.24	5912.43	5723.63	5697.12
	Std	47.16	50.369	46.959	45.634	21.43	17.37
	F-Measure	0.334	0.324	0.331	0.328	0.359	0.368

**Table 9.** Comparison of the proposed MCSS algorithm with other clustering algorithms using CMC dataset.

**Table 9** demonstrates the results of all the six algorithms for CMC dataset. As can be seen clearly, the proposed method achieves better results in comparison to the other algorithms using all the parameters. It is also stated that the performance of the GA is found to be poor among all the algorithms (in terms of standard deviation and F-measure parameters). Along

this, it is found that the K-means algorithm obtains maximum intra cluster distance amongst all.

**Table 10** illustrates the results of the proposed algorithm and all other algorithms for thyroid dataset. From the results, again it is observed that proposed algorithm obtains superior results in comparison to other algorithms, but gets the marginally high value of standard deviation parameter in comparison to GA, PSO, and ACO algorithms. Again, K-means results are far away from the optimal ones and CSS algorithm also obtain high value of standard deviation parameter except K-means. In the rest of the algorithms, performance of ACO algorithm is better.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Thyroid	Best	13956.83	10176.29	10108.56	10085.82	9997.25	9928.89
	Average	14 133.14	10218.82	10149.7	10108.13	10078.23	10036.93
	Worst	146424.21	10254.39	10172.86	10134.82	10116.52	10078.34
	Std	246.06	32.64	27.13	21.34	49.02	43.61
	F-Measure	0.731	0.763	0.778	0.783	0.789	0.793

**Table 10.** Comparison of the proposed MCSS algorithm with other clustering algorithms using thyroid dataset.

Results of all the six algorithm for glass dataset is summarized in **Table 11**. It indicates that CSS algorithm gives the better results for glass dataset in comparison to the other algorithms (in terms of intra cluster distance and standard deviation parameters). From the analysis of F-measure parameter, it is found that the performance of the MCSS is better than CSS. It is worthy to be noted that both the CSS and MCSS achieve good results on the cost of standard deviation parameter. Along this, it is also noticed that the GA exhibits weak performance.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Glass	Best	215.74	278.37	270.57	269.72	203.58	209.47
	Average	235.5	282.32	275.71	273.46	223.44	231.61
	Worst	255.38	286.77	283.52	280.08	241.27	263.44
	Std	12.47	4.138	4.55	3.584	13.29	17.08
	F-Measure	0.431	0.333	0.359	0.364	0.446	0.449

**Table 11.** Comparison of the proposed MCSS algorithm with other clustering algorithms using glass dataset.

**Table 12** summarizes the results of the proposed algorithm and all other algorithms for vowel dataset. From the results, it is noticed that MCSS algorithm obtains minimum intra

cluster distance amongst all but on the cost of high value of standard deviation parameter. In addition to it, it is also observed that both the MCSS and K-means algorithms exhibit similar performance in terms of F-measure parameters, but K-means obtains minimum value for standard deviation parameter. It is also stated that the quality of clustering is measured in terms of intra cluster distance. Therefore, MCSS algorithm provides good quality results in terms of intra cluster distance. Whereas, ACO method gets maximum intra cluster distance among all the methods. Over all, it is concluded that the proposed algorithm gives better performance with most of datasets in comparison to the other algorithms and quality of solutions is obtained. A statistical analysis is also carried out to prove the same.

Dataset	Parameters	K-means	GA	PSO	ACO	CSS	MCSS
Vowel	Best	149422.26	149513.73	148976.01	149395.6	149335.61	146124.87
	Average	159242.89	159153.49	151999.82	159458.14	152128.19	149832.13
	Worst	161236.81	165991.65	158121.18	165939.82	154537.08	157726.43
	Std	916	3105.544	2881.346	3485.381	2128.02	2516.58
	F-Measure	0.652	0.647	0.648	0.649	0.649	0.652

**Table 12.** Comparison of the proposed MCSS algorithm with other clustering algorithms using vowel dataset.

## 4. Conclusion

In this chapter, a magnetic charged system search algorithm is applied to solve the clustering problems. The idea of proposed algorithm came from the electromagnetic theory and it is based on the behavior of moving charged particles. A moving charged particle exerts both the forces (electric force and magnetic force) on other charged particles and in turn altered the positions of charged particles. Therefore, in MCSS algorithm, initial population is presented in the form of charged particles. It utilizes the concept of electric and magnetic forces along with newton second law of motion to obtain the updated positions of charged particles. In MCSS, both the electric force ( $E_k$ ) and magnetic force ( $M_k$ ) correspond to the local search for the solution, while the global solution is exploited using newton second law of motion. The aim of this research is to investigate the applicability of MCSS algorithm for clustering problems. To achieve the same, performance of the MCSS algorithm is evaluated on variety of datasets and compared with K-Means, GA, PSO, ACO, and CSS using intra cluster distance, standard deviation, and F-measure parameters. Experiment results support the applicability of proposed algorithm in clustering field as well as the proposed method provides good results with most datasets in comparison to the other methods. Finally, it is concluded that proposed method not only gives good results but also improves the quality of solutions.

## Author details

Yugal Kumar<sup>1\*</sup>, Sumit Gupta<sup>1\*</sup>, Dharmender Kumar<sup>2\*</sup> and Gadadhar Sahoo<sup>3\*</sup>

\*Address all correspondence to: yugalkumar.14@gmail.com; sumitkumarbsr19@gmail.com; dharm\_india@yahoo.com; gsahoo@bistmesra.ac.in

1 Department of Information Technology, Krishna Institute of Engineering and Technology, Ghaziabad, India

2 Department of Computer Science and Engineering, GJU, Hissar, India

3 Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, India

## References

- [1] Webb, A. Statistical pattern recognition. New Jersey: John Wiley & Sons, pp. 361–406 (2002).
- [2] Zhou, H., and Liu, Y. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition*, 41, no. 1 (2008), 152–175.
- [3] Sonka, M., Hlavac, V., and Boyle, R. Image processing, analysis, and machine vision. Springer, US, 1993. (1999 ).
- [4] Das, S., and Konar, A. Automatic image pixel clustering with an improved differential evolution. *Applied Soft Computing*, 9, no. 1 (2009), pp. 226–236.
- [5] Portela, N. M., Cavalcanti, G. D., and Ren, T. I. Semi-supervised clustering for MR brain image segmentation. *Expert Systems with Applications*, 41, no. 4 (2014), pp. 1492–1497.
- [6] Siang Tan, K., and Mat Isa, N. A. Color image segmentation using histogram thresholding –Fuzzy C-means hybrid approach. *Pattern Recognition*, 44, no. 1 (2011), pp. 1–15.
- [7] Teppola, P., Mujunen, S.-P. and Minkkinen, P. Adaptive Fuzzy C-Means clustering in process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 45, no. (1999), pp. 23–38.
- [8] Alpaydin, E. Introduction to machine learning. Cambridge, Massachusetts, London, England. MIT Press, 2004.
- [9] Anaya, A. R., and Boticario, J. G. Application of machine learning techniques to analyse student interactions and improve the collaboration process. *Expert Systems with Applications*, 38, no. 2 (2011), pp. 1171–1181.

- [10] Dunn III, W. J., Greenberg, M. J. and Callejas, S. S. Use of cluster analysis in the development of structure-activity relations for antitumor triazenes. *Journal of Medicinal Chemistry*, 19, no. 11 (1976), pp. 1299–1301.
- [11] Hu, G., Zhou, S., Guan, J. and Hu, X. Towards effective document clustering: a constrained  $K$ -means based approach. *Information Processing & Management*, 44, no. 4 (2008), pp. 1397–1409.
- [12] He, Y., Pan, W., and Lin, J. Cluster analysis using multivariate normal mixture models to detect differential gene expression with microarray data. *Computational Statistics & Data Analysis*, 51, no. 2 (2006), pp. 641–658.
- [13] Pappas, Thrasyvoulos N., An adaptive clustering algorithm for image segmentation. *IEEE Transactions on Signal Processing*, 40, no. 4 (1992), pp. 901–914.
- [14] Cheng, Y.M., and Leu, S. S., Constraint-based clustering and its applications in construction management. *Expert Systems with Applications*, 36 (2009), 5761–5767.
- [15] Kim, K.J. and Ahn, H. A recommender system using {GA} k-means clustering in an online shopping market. *Expert Systems with Applications*, 34 (2008), pp. 1200–1209.
- [16] Kuo, R., An, Y., Wang, H., and Chung, W., Integration of self-organizing feature maps neural network and genetic k-means algorithm for market segmentation. *Expert Systems with Applications*, 30 (2006), pp. 313–324.
- [17] Gunes, S., Polat, K., and Sebnem, Y. Efficient sleep stage recognition system based on {EEG} signal using k-means clustering based feature weighting. *Expert Systems with Applications*, 37 (2010), pp. 7922–7928.
- [18] Hung, Y.S., Chen, K.L. B., Yang, C.T., and Deng, G.F. Web usage mining for analysing elder self-care behavior patterns. *Expert Systems with Applications*, 40 (2013), pp. 775–783.
- [19] Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letter*, 31 (2010), pp. 651–666.
- [20] Barbakh, W., Wu, Y., Fyfe, C. Review of Clustering Algorithms. Non-Standard Parameter Adaptation for Exploratory Data Analysis. Springer, Berlin/Heidelberg (2009), pp. 7–28.
- [21] Camastra, F., Vinciarelli, A. Clustering Methods. *Machine Learning for Audio, Image and Video Analysis*. Springer, London (2008), pp. 117–148.
- [22] Kogan, J., Nicholas, C., Teboulle, M., Berkhin, P. A Survey of Clustering Data Mining Techniques, Grouping Multidimensional Data. Springer, Berlin Heidelberg (2006), pp. 25–71.
- [23] Maimon, O., Rokach, L., A survey of Clustering Algorithms, *Data Mining and Knowledge Discovery Handbook*. Springer, US (2010), pp. 269–298.

- [24] MacQueen, J. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1. no. 14 (1967 ) pp. 281–297.
- [25] Jain, A.K., Murty, M.N., Flynn, P.J. Data clustering: a review. ACM Computing Surveys (1999) Vol. 31, Issue No. 3, pp. 264–323.
- [26] Selim, S.Z., Alsultan, K. A simulated annealing algorithm for the clustering problem. Pattern Recognition 24 (1991), pp. 1003–1008.
- [27] Al-Sultan, K.S. A Tabu search approach to the clustering problem. Pattern Recognition 28 (1995), pp. 1443–1451.
- [28] Sung, C.S. and Jin, H.W. A tabu-search-based heuristic for clustering. Pattern Recognition 33 (2000), 849–858.
- [29] Cowgill, M.C., Harvey, R.J., and Watson, L.T. A genetic algorithm approach to cluster analysis. Comput. Math. Appl. 37 (1999), pp. 99–108.
- [30] Krishna, K., Narasimha Murty, M. Genetic K-means algorithm. IEEE Transactions on Systems, Man, Cybernet. Part B: Cybernet. 29 (1999), pp. 433–439.
- [31] Maulik, U., Bandyopadhyay, S. Genetic algorithm-based clustering technique. Pattern Recognition 33 (2000), pp. 1455–1465.
- [32] Murthy, C.A., Chowdhury, N. In search of optimal clusters using genetic algorithms. Pattern Recognition Lett. 17 (1996), pp. 825–832.
- [33] Van der Merwe, D. W. and Engelbrecht, A. P. Data clustering using particle swarm optimization, In Congress on Evolutionary Computation CEC-03, 1, pp. 215–220. IEEE (2003).
- [34] Rana, S., Jasola, S. and Kumar, R. A review on particle swarm optimization algorithms and their applications to data clustering. Artificial Intelligence Review 35, no. 3 (2011) pp. 211–222.
- [35] Shelokar, P. S., Jayaraman, V. K. and Kulkarni, B. D. An ant colony approach for clustering. Analytica Chimica Acta 509, no. 2 (2004), pp. 187–195.
- [36] Zhang, C., Ouyang, D. and Ning, J. An artificial bee colony approach for clustering. Expert Systems with Applications 37, no. 7 (2010), pp. 4761–4767.
- [37] Karaboga, D., and Ozturk, C. A novel clustering approach: artificial bee colony (ABC) algorithm. Applied Soft Computing 11, no. 1 (2011), 652–657.
- [38] Kumar, Y., and Sahoo, G. A charged system search approach for data clustering. Progress in Artificial Intelligence 2, no. 2–3 (2014), pp. 153–166.
- [39] Kumar, Y. and Sahoo, G. A chaotic charged system search approach for data clustering. Informatica, 38, no. 3 (2014), pp. 249–261.

- [40] Santosa, B., and Ningrum, M. K. Cat swarm optimization for clustering. In *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of*, pp. 54–59. IEEE (2009).
- [41] Kumar, Y. and Sahoo, G. A Hybridize approach for data clustering based on cat swarm optimization. *International Journal of Information and Communication Technology* (2015) (accepted for publication ).
- [42] Kumar, Y. and Sahoo, G. An improved cat swarm optimization algorithm for data clustering. In *International Conference on Computational Intelligence in Data Mining (ICCIDM-2014) proceedings published in the Springer's Series on Smart Innovation, Systems and Technologies, Germany* (2014).
- [43] Satapathy, S. C. and Naik, A. Data clustering based on teaching-learning-based optimization. In *Swarm, Evolutionary, and Memetic Computing*, pp. 148–156. Springer Berlin Heidelberg (2011).
- [44] Sahoo, A. J. and Kumar, Y. Modified teacher learning based optimization method for data clustering. In *Advances in Signal Processing and Intelligent Recognition Systems* (2014), pp. 429–437.
- [45] Hatamlou, A., Abdullah, S., Nezamabadi-pour, H. Application of Gravitational Search Algorithm on Data Clustering, *Rough Sets and Knowledge Technology*. Springer, Berlin/Heidelberg (2011), pp. 337–346.
- [46] Hatamlou, A., Abdullah, S., Nezamabadi-pour, H. A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm and Evolutionary Computation*, 6 (2012), pp. 47–52.
- [47] Hatamlou A. In search of optimal centroids on data clustering using a binary search algorithm. *Pattern Recognition Letters* 33 (2012), pp. 1756–1760.
- [48] Kaveh, A., Motie Share, M. A. and Moslehi, M. Magnetic charged system search: a new meta-heuristic algorithm for optimization. *Acta Mechanica* 224, no. 1 (2013), pp. 85–107.
- [49] Niknam, T. and Amiri, B. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing* 10 (2010), pp. 183–197.
- [50] Dalli, A. Adaptation of the F-measure to cluster based lexicon quality evaluation. In *Proceedings of the EAACL* (2003), pp. 51–56.
- [51] Handl, J., Knowles, J., and Dorigo, M. On the performance of ant-based clustering. *Design and Application of Hybrid Intelligent System. Frontiers in Artificial Intelligence and Applications*, 104 (2003), pp. 204–213.
- [52] <https://archive.ics.uci.edu/ml/datasets>
- [53] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), pp. 1–30.

- [54] Derrac, J. Salvador García, Daniel Molina, and Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1, no. 1 (2011), pp. 3–18.
- [55] García, S., Fernández, A., Luengo, J., and Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Information Sciences* 180, no. 10 (2010), pp. 2044–2064.
- [56] Kumar, Y. and Sahoo, G. A two-step artificial bee colony algorithm for clustering. *Neural Computing and Applications*. (2015), pp 1–15, DOI: 10.1007/s00521-015-2095-5.
- [57] Kumar, Y. and Sahoo, G.. A hybrid data clustering approach based on improved cat swarm optimization and K-harmonic mean algorithm. *AI Communications (Preprint)* (2015), pp.1-14.



---

# Topology Optimization Method Considering Cleaning Procedure and Ease of Manufacturing

---

Takeo Ishikawa

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63153>

---

## Abstract

This chapter proposes a novel topology optimization method for the material distribution of electrical machines using the genetic algorithm (GA) combined with the cluster of material and the cleaning procedure. Moreover, the obtained rotor structure was assumed to consist of the simple shape of PMs in order to consider ease of manufacturing. The rotor structure of a permanent magnet (PM) synchronous motor is designed and manufactured. The optimized rotor has 32% more average torque than that of the experimental motor with the same stator. The effectiveness of the proposed method is verified.

**Keywords:** topology optimization, genetic algorithm, permanent magnet synchronous motor, finite element method, manufacturing

---

## 1. Introduction

There are several design techniques to optimize electrical machines and electromagnetic devices. However, most of these techniques were restricted to the optimization of a couple of parameters defining the shape. I think that the first step of optimal design should design the topology of these structures by starting from an empty space. Topology optimization allows obtaining an initial conceptual structure starting with minimal information regarding the structure of the object. Topology optimization methods are very promising and, therefore, were proposed about 20 years ago [1, 2]. Since then, several papers have been published in this field. For example, reference [3] proposed a topology optimization using design sensitivity. Reference [4] proposed an ON/OFF sensitivity method and hybridized it with the GA in order to improve convergence characteristics. Reference [5] designed an electromagnetic

system by a topology optimization method considering magnetization direction. Reference [6] proposed a topology optimization method coupling with magneto-thermal systems. Reference [7] designed a 3D electromagnetic machine with soft magnetic composite core. Reference [8] proposed a topology optimization method using the GA and ON/OFF sensitivity in conjunction with a blurring technique in order to avoid small structure spots. Reference [9] took into account a mapping function to improve convexity in the topology optimization procedure. Reference [10] applied a topology optimization method to a coupled magnetic structural problem. Reference [11] designed an IPM motor using the ON/OFF method. Reference [12] optimized magnetic actuators by a level-set method. Reference [13] optimized an inductor by the evolutionary algorithm. Reference [14] proposed a 3D topological optimization method based on the multistep utilization of GA. Reference [15] presented a possible solution to the structural optimization problem using a simple heuristic search algorithm.

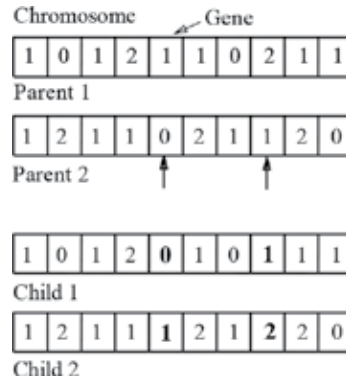
The author proposed a topology optimization method to optimize the distribution of materials within an electrical machine using the GA [16]. In addition, he proposed a concept for the cluster of materials and a cleaning procedure for the materials and designed the stator of a brushless DC motor based on this method [17]. However, that study considered only two types of materials, air, and iron, and therefore, it is similar to the ON/OFF method. He improved the previous method in order to consider more than two materials namely, air and iron as well as  $r$ -oriented,  $x$ -oriented, and  $y$ -oriented magnets [18]. Moreover, initially, the obtained rotor structure was assumed to consist of the simple shape of PMs in order to consider ease of manufacturing [19].

This chapter summarizes the novel topology optimization method for the material distribution of electrical machines using the GA combined with the cluster of material and the cleaning procedure. Moreover, the obtained rotor structure was assumed to consist of the simple shape of PMs in order to consider ease of manufacturing. The proposed method is applied to the design of the rotor structure of PM synchronous motor. It is an interior permanent magnet synchronous motor (IPMSM) used for air-conditioning. The average torque characteristics of IPMSMs are compared with the commercialized motor.

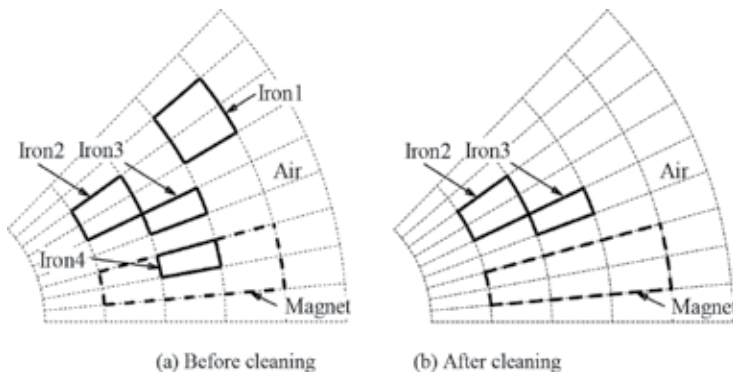
## 2. Proposed topology optimization method

In this section, the topology optimization method is briefly explained, which is implemented in this study. GA is an algorithm that imitates the evolution of living things and is suitable for problems with a large sample space. In the proposed method, the design region is split into finite element meshes, and the materials of several elements—for example, a cell—are associated with a gene in the chromosome. For example, if we consider three types of materials—for example, air, iron, and magnet, which are set to 0, 1, and 2, respectively—the chromosome is composed of some genes as shown in **Figure 1**. Two parents are selected randomly, and some genes are selected to be exchanged by a uniform crossover with a crossover ratio, and then, two new children are generated as shown in **Figure 1**. The chil-

dren inherit the good characteristics of parents by repeating the process. We proposed the concept of the cluster with many types of materials. For example, irons 2 and 3 form the same cluster because they are next to each other as shown in **Figure 2a**, and iron 4 forms another cluster. If the area of cluster is narrow, that is, the number of cells in the same cluster is smaller than or equal to an integer  $N_{min}$ , a cleaning procedure is introduced. When  $N_{min}$  is equal to 2, irons 2 and 3 remain, iron 1 is changed to the surrounding material air, and iron 4 is changed to the magnet. As a result, this cleaning method has the ability of removing floating pieces of the material.



**Figure 1.** Example of genes and uniform crossover in GA.



**Figure 2.** Cluster of materials and the cleaning method concept.

**Figure 3** shows the cross section of a four-pole PM synchronous motor with distributed windings. One-eighth of the rotor is designed for symmetry. This study iterates the GA with the newly increased length of genes. For the first iteration, a coarse topology is designed using a small number of design variables in a  $5 \times 9$  array of cells as shown in **Figure 4a**. Three magnetized directions of permanent magnet are dealt with as shown in **Figure 5**. **Figure 6** shows the treatment of the cluster of material on the boundary. The  $y$ -oriented magnet has the same magnetized direction; on the contrary, the  $x$ - and  $r$ -oriented magnets have the opposite

direction at both sides of periodic boundary. Therefore, the numbers of cells in the cluster of air, iron, and  $y$ -oriented magnet are doubled. On the other hand, the numbers of cells in the cluster of air, iron, and  $r$ -oriented magnet are doubled at both sides of symmetric boundary, because the  $r$ -oriented magnet has the same magnetized direction.

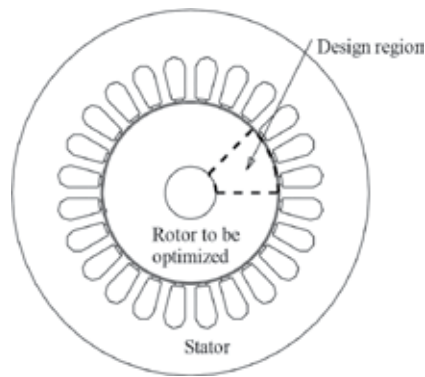


Figure 3. Motor cross section and design region.

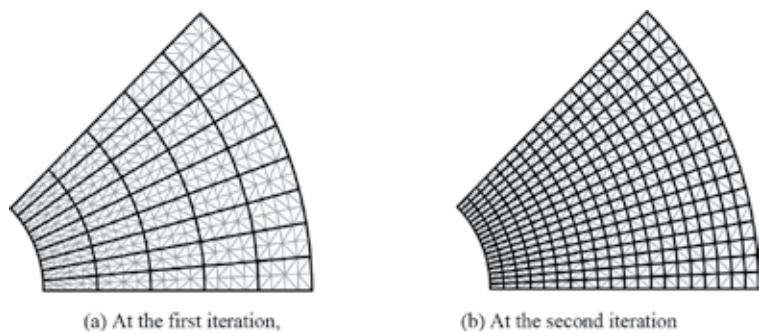


Figure 4. Cell at each iteration.

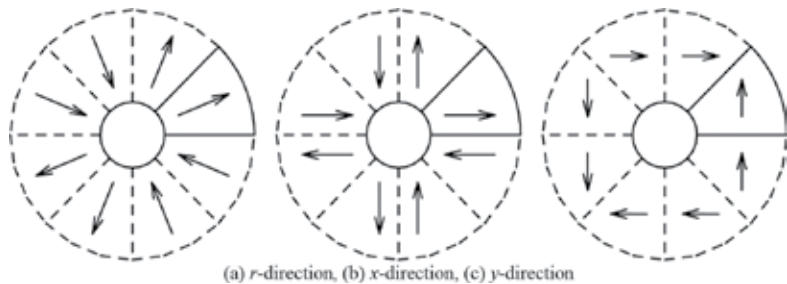
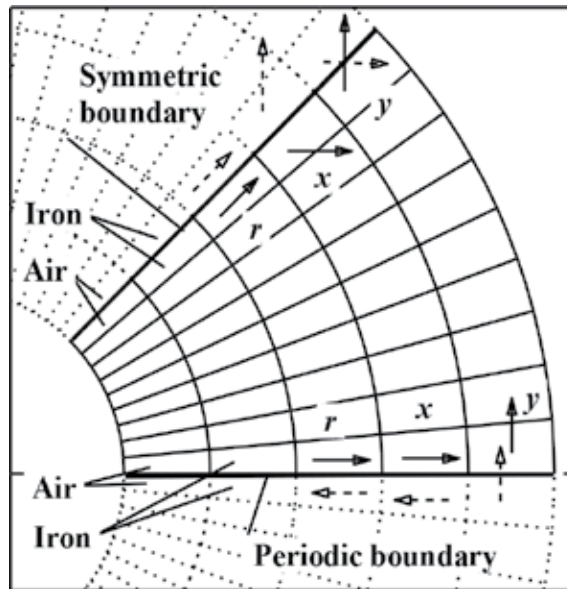
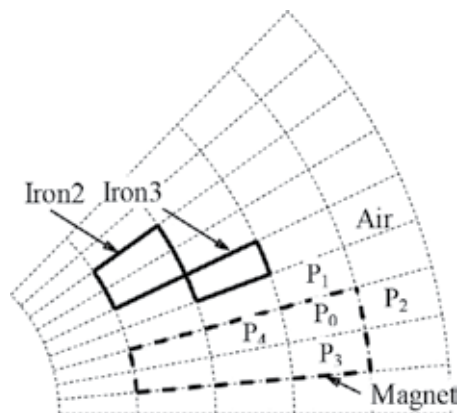


Figure 5. Magnetized direction in permanent magnets.



**Figure 6.** Cluster of material on the symmetric boundary and on the periodic boundary.

A fine topology then is designed using a large number of design variables for the second iteration in a  $20 \times 18$  array of cells as shown in **Figure 4b**. For the second iteration, a set of initial individuals in the GA inherit the individual that has the best fitness at the conclusion of the previous iteration. For example, the initial material in cell  $P_0$  is generated by a probability of  $1/5$  from materials in cells  $P_0$ ,  $P_1$ ,  $P_2$ ,  $P_3$ , and  $P_4$  shown in **Figure 7**. The whole flow-chart for the proposed method is shown in **Figure 8**. The parts highlighted by the thick line are newly added to the conventional GA with the elite selection. The parameter  $N_{min}$  of the cleaning procedure at each iteration is listed in **Table 1**.



**Figure 7.** Cluster of material on the symmetric boundary and on the periodic boundary.

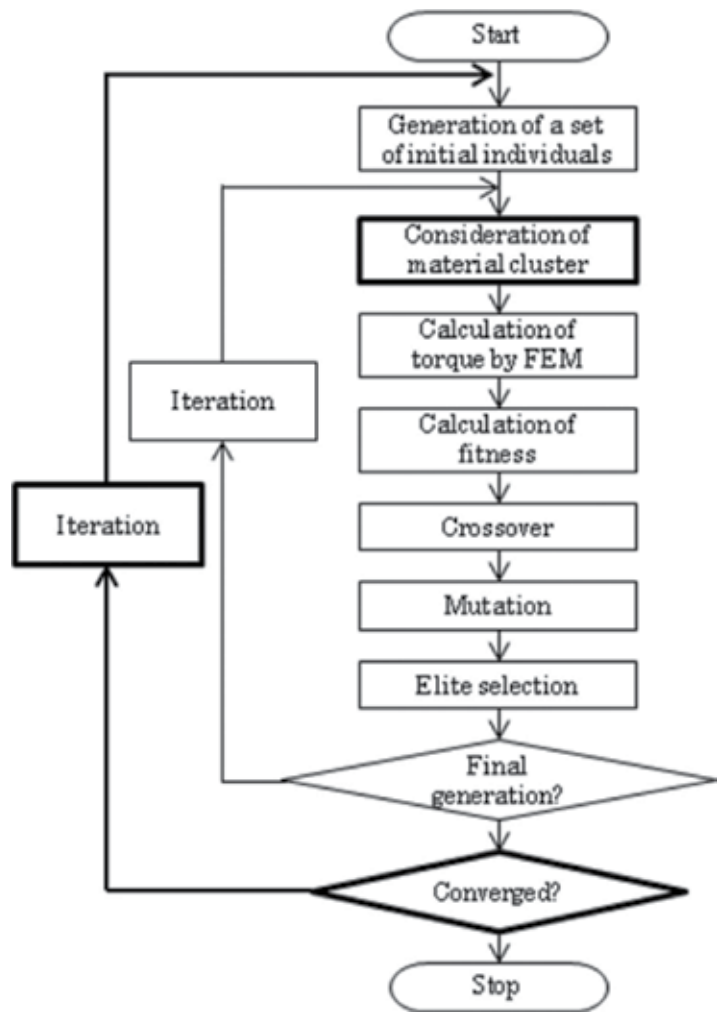


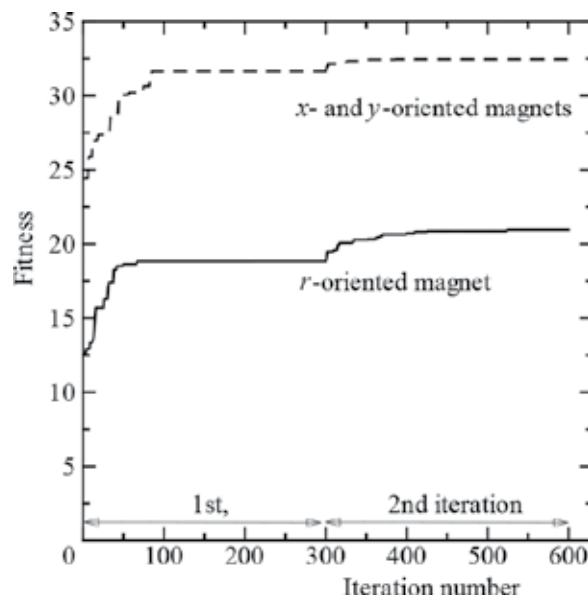
Figure 8. Flowchart for the proposed topology optimization method.

Figures	Material	$N_{min}$ at first iteration	$N_{min}$ at second iteration
Figure 10a	Air, iron, $r$ -oriented PM	1	4
Figure 10b	Air, iron, $x$ -, $y$ -oriented PM	1	4
Figure 10c	Air, iron, $x$ -, $y$ -oriented PM	1 for air and iron, 0 for PM	4
Figure 10d	Air, iron, $r$ -oriented PM	0	0

Table 1. Design parameters.

### 3. Rotor structure obtained by the topology optimization method

We optimize the topology of the rotor structure by considering two types of PMs; *r*-oriented magnet only and both *x*- and *y*-oriented magnets. The number of populations is set to 45. If a number larger than 45 is selected, there is a possibility for a better result. However, this would lead to a longer computational time. **Figure 9** shows a convergence characteristic of the applied topology optimization. We find that the fitness functions almost converge at every iteration, namely at every 300th generation.

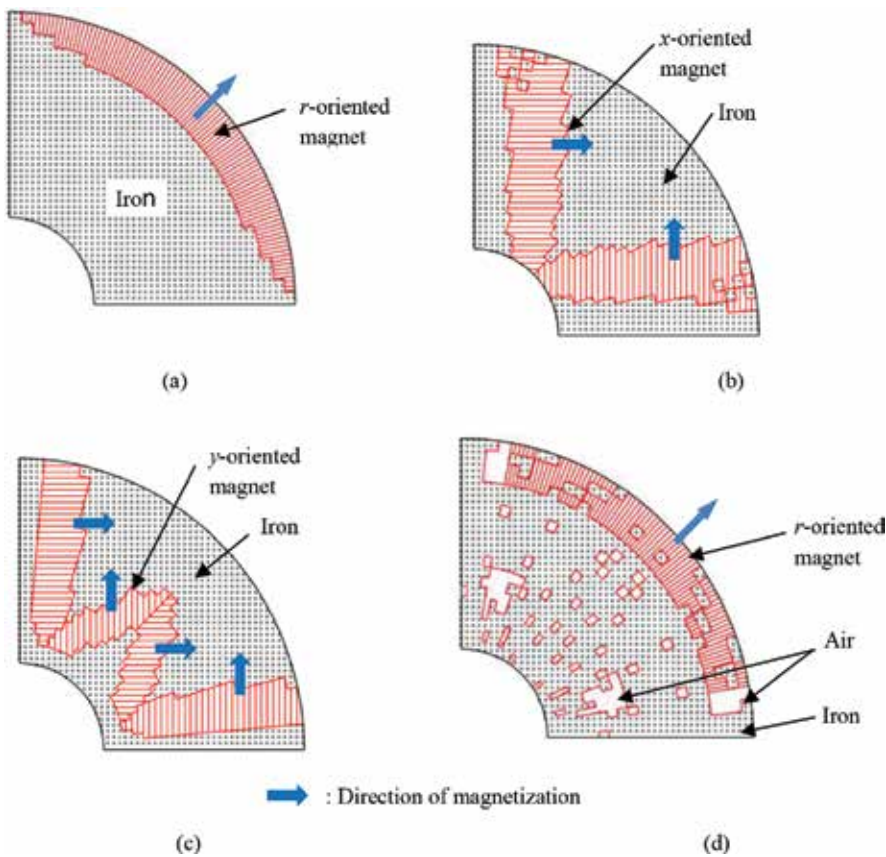


**Figure 9.** Convergence characteristic of the topology optimization.

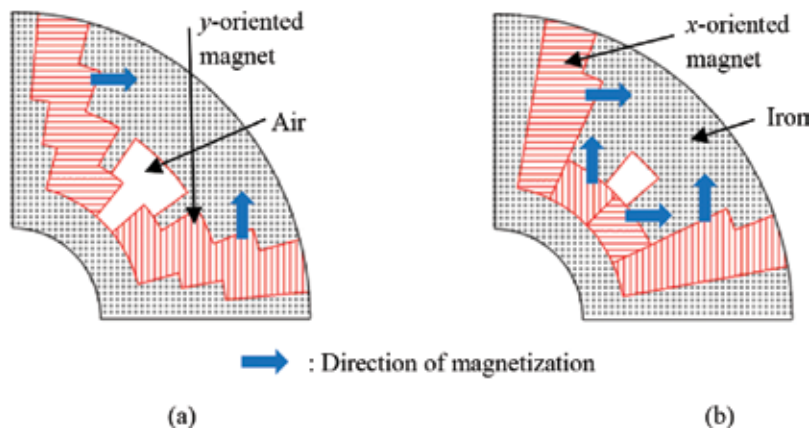
**Figure 10a** shows the obtained rotor structure, in which three types of materials—air, iron, and the *r*-oriented magnet—are taken into account.  $N_{min}$  at the first iteration is equal to 1. This means that if the number of cells in the cluster is  $<2$ , the material of the cluster is changed to the material of surrounding cells. At the second iteration,  $N_{min}$  is chosen to 4. Design parameters are shown in **Table 1**. The optimization by considering the *r*-oriented magnet only produces a kind of surface PM-type rotor. **Figure 10b, c** shows the obtained rotor structure by considering *x*- and *y*-oriented magnets that produce a kind of interior PM-type rotor. The parameters listed in **Table 1** are used for the cleaning procedure. The rotor shapes appear as V and W. The V shape was obtained when the cleaning procedure was carried out for the PM material at the first iteration, and the W shape was obtained when the cleaning procedure was not carried out for the PM material at the first iteration. **Figure 10d** shows the obtained rotor structure when the cluster of material and the cleaning method are not carried out.  $T_{ave}$  and  $V_{pm}$  for **Figure 10d** are 5.38 Nm and 23.2 cm<sup>3</sup>, respectively, and for **Figure 10a**, they are 6.16 Nm and 31.6 cm<sup>3</sup>, respectively. Although the obtained

magnet shape appears similar to that shown in **Figure 10a**, the rotor has numerous small pieces of iron in the  $r$ -oriented magnet. As mentioned above, the cleaning procedure can remove the small cluster of materials. Therefore, the structure obtained without considering the cleaning procedure has numerous small pieces of iron in the magnet and numerous pockets of air in the iron, which are shown in **Figure 10d**. This rotor has a complicated weak structure and is difficult to manufacture.

The optimized rotor shapes shown in **Figure 10b, c** appear as  $V$  and  $W$ . Let us discuss what causes this difference. **Figure 11** shows the obtained shapes at the first iteration with and without considering the cleaning procedure for the PM materials. When the cleaning procedure is not carried out for PM materials, small PM cells can easily remain in the GA procedure. As a result, **Figure 11a** does not include the  $x$ -oriented magnet at the center, whereas **Figure 11b** includes the  $x$ -oriented magnet at the center. The  $x$ -oriented magnet shown in **Figure 11b** remains in the next iteration, and the final shape, which includes the  $x$ -oriented magnet, is then obtained as shown in **Figure 10c**.



**Figure 10.** Rotor structure obtained by the proposed topology optimization with the parameter shown in **Table 1**.

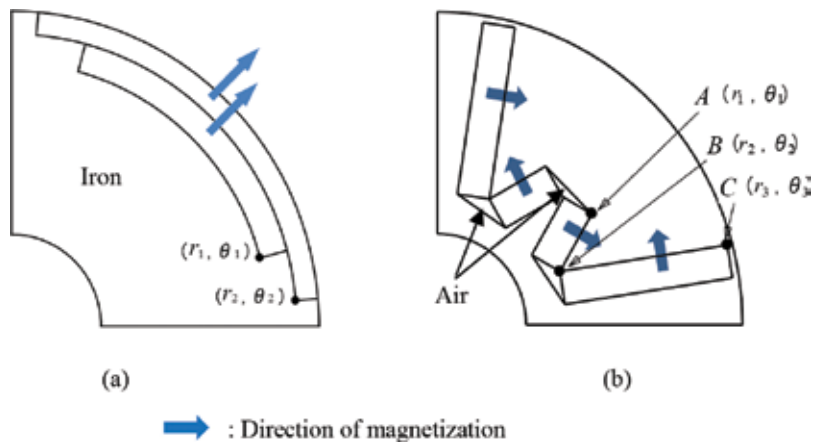


**Figure 11.** Rotor structures obtained at the first iteration by considering air, iron, x- and y-oriented PMs, (a) with and (b) without the cleaning procedure for the PM.

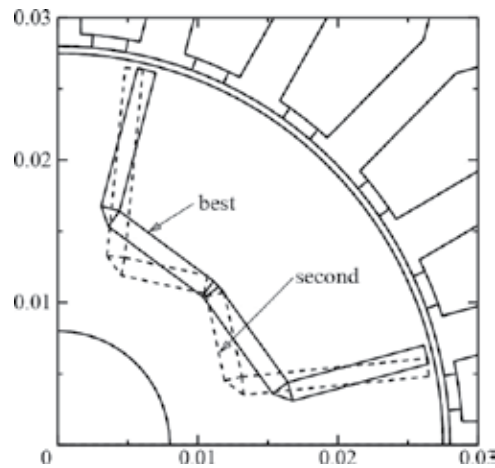
#### 4. Optimization considering ease of manufacturing

The obtained rotor structures are complex and impractical. In order to consider the ease of manufacturing, magnets and air pockets are assumed to be simple shapes, and they are then optimized using conventional techniques. For example, the rotor structures shown in **Figure 12a** can be assumed to be similar to the ones in **Figure 10a**, where the magnets are represented by four parameters. The rotor structure shown in **Figure 10b, c** can be assumed to contain four hexahedron PMs as shown in **Figure 12b**. In this shape, the rotor structure is represented by six parameters if the thickness of the magnets is uniform and the volume of the magnets is specified. We assume that the shape of each magnet is a hexahedron and the magnet is magnetized in the vertical direction as shown in **Figure 12b** for the ease of magnetization. Moreover, we assume that a core area is introduced on the surface of the rotor as shown in **Figure 12b** to insure machine strength against centrifugal force. Therefore, parameter  $r_3$  is fixed, and then, the number of design parameters becomes five.

An example to be optimized is the rotor of an experimental motor. This experimental motor is well known as the D model in IEE Japan for an air-conditioner and has an IPM-type rotor. A full-search method to optimize the rotor shape is used, because we want to verify that there is a good rotor shape similar to those shown in **Figure 10b, c**. The angle of position A is set to five values, and the radius and angle of position B and C, respectively, are also set to five values. This gives  $5^3 = 3125$  patterns to be calculated in the full-search method. This study iterates the full-search method twice, where the variation of each parameter is set to approximately half, and the phase angle of stator current is set to 15°, 20°, 25°, and 30°. The finite element mesh is generated automatically using the Delaunay method, and the torque is calculated using the Maxwell's tensor method in the full-search method.



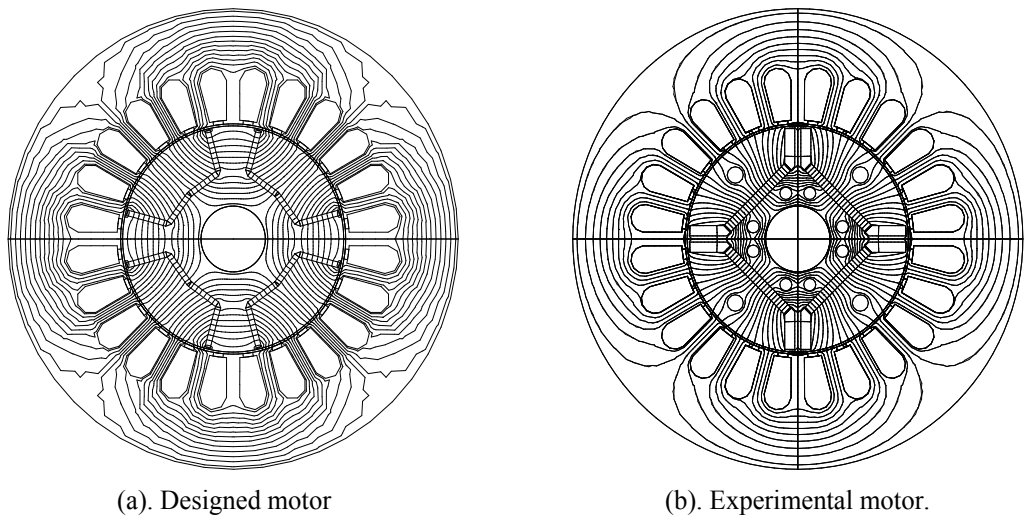
**Figure 12.** Rotor structures represented by simple magnet shape.



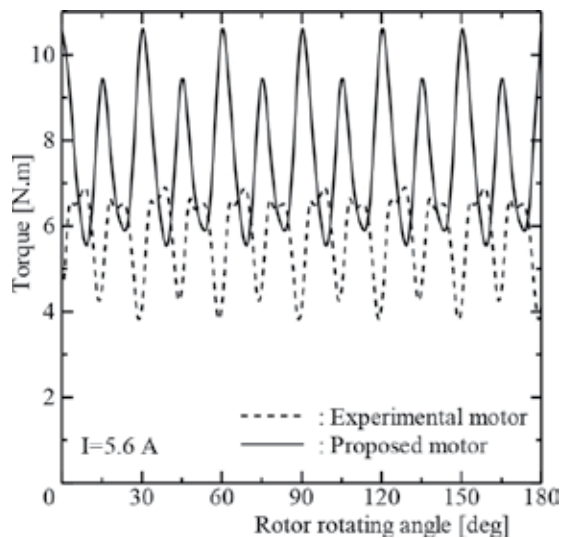
**Figure 13.** Obtained rotor structure of the air-conditioner.

**Figure 13** shows the obtained rotor structure, where the “best” rotor structure provides the largest average torque at the second iteration of the full-search method, and the “second” one provides the second-largest average torque at the first iteration. It was found that the “best” rotor structure is *U*-shaped and the “second” is *W*-shaped. **Figure 14a** shows the obtained rotor structure and its flux distribution at no load. **Figure 14b** shows the flux distribution of the experimental motor whose stator is the same as that shown in **Figure 14a**. The ratings of the experimental motor are  $3000 \text{ min}^{-1}$ , 1.5 kW, 192 V, and 5.6 A, and the stator outer diameter, the rotor diameter, and the air-gap length are 112, 55, and 0.5 mm, respectively. **Figure 14** shows that the flux lines of the optimized motor are increased by approximately 50%, because the magnets are wider and thinner than the experimental motor. It is found from **Figure 15** that the average torque is increased from 5.8 to 7.6 Nm—that is, by approximately 32%—

mainly due to the increase of magnetic flux linkage. However, the torque ripple of the proposed motor is increased because it is not considered by the fitness function. The calculated torque of the “second” rotor structure is almost the same as that of the “best” rotor structure. This means that the shape of the PM in the inside rotor is not a significant factor in the torque provided. Therefore, we believe that two types of rotor structure have been obtained by the proposed method as shown in **Figure 10b, c**.



**Figure 14.** Obtained rotor structure and flux distribution at no load.



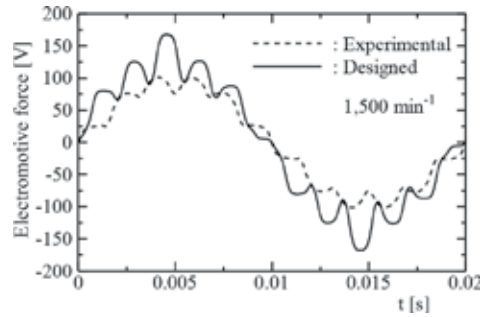
**Figure 15.** Comparison of torque with the experimental motor.

## 5. Comparison of measured results

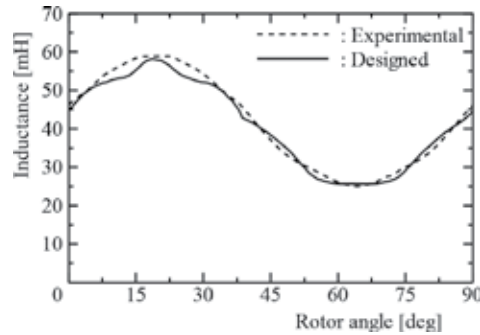
We have manufactured the designed rotor shown in **Figure 14a**. **Figure 16** shows the measured electromotive force when the rotor is rotating at a speed of  $1500 \text{ min}^{-1}$ . The effective value and fundamental component of the electromotive force are 96.3 and 133.8 V, respectively, and those for the experimental motor are 68.2 and 95.6 V, respectively. Therefore, the electromotive force of the designed motor is 1.4 times larger than that of the experimental motor. **Figure 17** shows the inductance measured by an LCR meter at 100 Hz. The obtained  $d$  and  $q$  inductances are 12.69 and 28.37 mH, respectively, for the designed motor, and 12.51 and 29.37 mH, respectively, for the experimental motor. The developed torque of IPMSM can be expressed by

$$T = p\Phi I \cos \beta + 0.5p(L_q - L_d)I^2 \sin 2\beta \quad (1)$$

where  $p$ ,  $\Phi$ ,  $I$ , and  $\beta$  are pole pairs, flux linkage, stator current, and phase angle of stator current, respectively. Although  $(L_q - L_d)$  of the designed motor is somewhat small, the flux linkage is 1.4 times greater than that of the experimental motor.

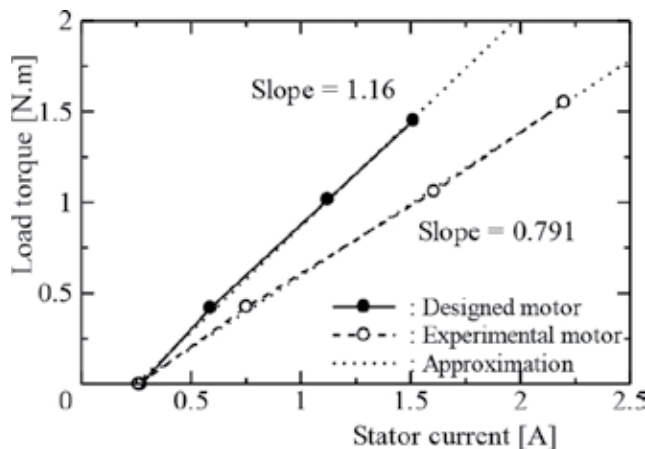


**Figure 16.** Measured electromotive force.



**Figure 17.** Measured inductance.

**Figure 18** shows the torque–current characteristics when  $\beta$  is set to approximately 0. The developed torque calculated by the measured electromotive force is  $T = 0.75I$  for the experimental motor and  $T = 1.05I$  for the designed motor. These values are approximately the same as the slope shown in **Figure 18**. The differences are generated by the fact that  $\beta$  is not controlled to be exactly 0. Therefore, it is verified that the measured torque of the designed motor is larger than that of the experimental motor under the same stator current.



**Figure 18.** Measured torque characteristics.

## 6. Conclusions

In this study, we designed the rotor structure of PM synchronous motors. The proposed optimization process combines the topology optimization method and a method that considers the ease of manufacturing. We assumed four hexahedron PMs similar to the rotor shapes obtained by the proposed method for reasons of manufacturing ease. The obtained rotor of the compressor motor for the air-conditioner has 32% more average torque than that of the experimental motor with the same stator. Therefore, the effectiveness of the proposed method is verified.

## Author details

Takeo Ishikawa

Address all correspondence to: [ishi@gunma-u.ac.jp](mailto:ishi@gunma-u.ac.jp)

Division of Electronics and Informatics, Faculty of Science and Technology, Gunma University, Maebashi, Japan

## References

- [1] Z. Q. Zhu, D. Howe, and Z. P. Xia, "Prediction of open-circuit air gap field distribution in brushless machines having an inset permanent magnet rotor topology," *IEEE Trans. Magn.*, vol.30, no.1, pp.98–107, (1994)
- [2] D. N. Dyck and D. A. Lowther, "Automated design of magnetic devices by optimizing material distribution," *IEEE Trans. Magn.*, vol.32, no.3, pp.1188–1193, (1996)
- [3] J. K. Byun, S. Y. Hahn, and I. H. Park, "Topology optimization of electrical devices using mutual energy and sensitivity," *IEEE Trans. Magn.*, vol.35, no.5, pp.3718–3720, (1999)
- [4] C. H. Im, H. K. Jun, and Y. J. Kim, "Hybrid genetic algorithm for electromagnetic topology optimization," *IEEE Trans. Magn.*, vol.39, no.5, pp.2163–2169, (2003)
- [5] S. Wang, D. Youn, H. Moon, and J. Kang, "Topology optimization of electromagnetic systems considering magnetization direction," *IEEE Trans. Magn.*, vol.41, no.5, pp. 1808–1811, (2005)
- [6] H. Shim, S. Wang, and K. Hameyer, "Topology optimization of magneto thermal systems considering eddy current as Joule heat," *IEEE Trans. Magn.*, vol.43, no.4, pp. 1617–1620, (2007)
- [7] D. H. Kim, J. K. Sykulski, and D. A. Lowther, "The implications of the use of composite material in electromagnetic device topology and shape optimization," *IEEE Trans. Magn.*, vol.45, no.3, pp.1154–1157, (2009)
- [8] J. S. Choi and J. Yoo, "Structural topology optimization of magnetic actuators using Genetic algorithms and ON/OFF sensitivity," *IEEE Trans. Magn.*, vol.45, no.5, pp.2276–2279, (2009)
- [9] T. Labbe and B. Dehez, "Convexity-oriented mapping method for the topology optimization of electromagnetic devices composed of iron and coils," *IEEE Trans. Magn.*, vol.46, no.5, pp.1177–1185, (2010)
- [10] J. Lee and N. Kikuchi, "Structural topology optimization of electrical machinery to maximize stiffness with body force distribution," *IEEE Trans. Magn.*, vol.46, no.10, pp. 3790–3794, (2010)
- [11] N. Takahashi, T. Yamada, and D. Miyagi, "Examination of optimal design of IPM motor using ON/OFF method," *IEEE Trans. Magn.*, vol.46, no.8, pp.3149–3152, (2010)
- [12] S. Park and S. Min, "Design of magnetic actuator with nonlinear ferromagnetic material using level-set based topology optimization," *IEEE Trans. Magn.*, vol.46, no.2, pp.618–621, (2010)
- [13] K. Watanabe, F. Campelo, Y. Iijima, K. Kawano, T. Matsuo, T. Mifune, and H. Igarashi, "Optimization of inductors using evolutionary algorithms and its experimental validation," *IEEE Trans. Magn.*, vol.46, no.8, pp.3393–3396, (2010)

- [14] Y. Okamoto, Y. Tominaga, S. Sato, "Topological design for 3-D optimization using the combination of multistep genetic algorithm with design space reduction and nonconforming mesh connection," *IEEE Trans. Magn.*, vol.48, no.2, pp.515–518, (2012)
- [15] Hahn, I., "Heuristic structural optimization of the permanent magnets used in a surface mounted permanent-magnet synchronous machine," *IEEE Trans. Magn.*, vol.48, no.1, pp.118–127, (2012)
- [16] T. Ishikawa, M. Kaneda, and A. Yamagiwa, "Optimal material distribution design of interior permanent magnet synchronous motor using Genetic algorithm," *The 10th biennial IEEE Conference on Electromagnetic Field Computation, Perugia, Italy*, pp. 2–4, (2002)
- [17] T. Ishikawa, K. Yonetake, and N. Kurita, "An optimal material distribution design of brushless DC motor by genetic algorithm considering a cluster of material," *IEEE Trans. Magn.*, vol.47, no.5, pp. 1310–1313, (2011)
- [18] T. Ishikawa and K. Nakayama, "Topology optimization of rotor structure in brushless DC motor with concentrated windings using genetic algorithm combined with cluster of material," *IEEE Trans. Magn.*, vol.48, no.2, pp.899–902, (2012)
- [19] T. Ishikawa, K. Nakayama, N. Kurita, and F. P. Dawson, "Optimization of rotor topology in PM synchronous motors by Genetic algorithm considering cluster of materials and cleaning procedure," *IEEE Trans. on Magnetism*, vol. 50, no. 2, paper no. 7015704, (2014)



---

# **A Review and Comparative Study of Firefly Algorithm and its Modified Versions**

---

Waqar A. Khan, Nawaf N. Hamadneh,  
Surafel L. Tilahun and Jean M. T. Ngnotchouye

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62472>

---

## **Abstract**

Firefly algorithm is one of the well-known swarm-based algorithms which gained popularity within a short time and has different applications. It is easy to understand and implement. The existing studies show that it is prone to premature convergence and suggest the relaxation of having constant parameters. To boost the performance of the algorithm, different modifications are done by several researchers. In this chapter, we will review these modifications done on the standard firefly algorithm based on parameter modification, modified search strategy and change the solution space to make the search easy using different probability distributions. The modifications are done for continuous as well as non-continuous problems. Different studies including hybridization of firefly algorithm with other algorithms, extended firefly algorithm for multiobjective as well as multilevel optimization problems, for dynamic problems, constraint handling and convergence study will also be briefly reviewed. A simulation-based comparison will also be provided to analyse the performance of the standard as well as the modified versions of the algorithm.

**Keywords:** Optimization, Metaheuristic Algorithms, Parametric Modification, Mutation, Binary Problems, Simulation

---

## **1. Introduction**

An optimization problem refers to the maximization or minimization of an objective function by setting suitable values for the variables from a set of feasible values. These problems appear not only in complex scientific studies but also in our day-to-day activities. For instance,

when a person wants to go from one place to another and has multiple possible routes, a decision needs to be made on which route to take. The decision can be with the objective to minimize travel time, fuel consumption and so on. However, these kinds of problems with fewer number of alternatives can easily be solved by looking at the outcome of each of the alternatives. However, in real problems, it is not always the case to have a finite and small number of alternatives. Hence, different solution methods are proposed based on the behaviour of the problem.

Since the introduction of evolutionary algorithms, many studies have been conducted on heuristic algorithms. Introducing new algorithms has been one of the leading research areas [1]. Currently, there are more than 40 metaheuristic algorithms [2]. Most of these new algorithms are introduced by mimicking a scenario from nature. For instance, genetic algorithm is inspired by the Darwin theory of survival of the fittest [3]; particle swarm optimization is another metaheuristic algorithm mimicking how a swarm moves by following each other [4]; firefly algorithm is inspired by how fireflies signal each other using the flashing light to attract for mating or to identify predators [5] and prey predator algorithm is another new algorithm inspired by the behaviour of a predator and its prey [6]. These algorithms use different degree of exploration and exploitation based on their different search mechanisms.

Firefly algorithm is among those metaheuristic algorithms which have different applications. Its uncomplicated and easy steps with its effectiveness attract researchers from different disciplines. Different studies have been performed to modify the standard firefly algorithm to boost its performance and to make it suitable for a problem at hand. In this chapter, a comprehensive study will be presented on firefly algorithm and its modified versions. A brief discussion on extended firefly algorithm with other relevant studies will also be provided. In the next section, a discussion on optimization problems with their solution methods will be given followed by a review on studies on firefly algorithm, which includes a discussion on the standard firefly algorithm with its modified versions and other relevant studies on firefly algorithm, in Section 3. In Section 4, a comparative study based on simulation results will be presented followed by summary of the chapter in Section 5.

## 2. Optimization problems

Decision-making problems can be found beyond our daily activity. They are very common in engineering, management and in many other disciplines. Different researchers used the concept of optimization in different applications, including engineering applications, transportation planning, management applications, economics, computational intelligence, decision science, agriculture, tourism, sport science and even political science [7–18].

When these problems are formulated mathematically, they are called mathematical optimization problems. It will have a set of feasible actions, also called feasible regions, and a measure of performance of these actions called the objective. A standard single objective minimization problem can be given as in Eq. (1).

$$\min_x \{f(x) \mid x \in S \subseteq \mathbb{R}^n\} \quad (1)$$

where  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is called the objective function,  $S$  is the feasible region and the vector  $x$  is the decision variable. A vector  $\bar{x}$  is said to be an optimal solution for the minimization problem given in Eq. (1) if and only if  $\bar{x} \in S$  and  $f(\bar{x}) \leq f(x)$ ,  $\forall x \in S$ . A local solution  $x'$  is a member of  $S$  and  $f(x') \leq f(x)$ , for all  $x$  in the neighbourhood of  $x'$ .

In a broad sense, optimization solution methods can be categorized as exact and approximate solution methods. Exact solution methods are methods which use an exhaustive search for the exact solution in the solution space. They use mathematical and statistical arguments to get an exact solution. They mainly used calculus-based and iterative procedures. Perhaps Fermat is the first to use a calculus-based argument to solve optimization problems [19]. Iterative methods were first proposed and used by Newton and Gauss [20]. Since then, several exact solution methods are proposed and used in different problems. Branch and bound, simplex method and gradient descent method are good examples of exact solution methods. However, due to complex problems modelled from complex real aspects, it becomes challenging for the deterministic solution methods. This leads to the search of new 'out of the box' way of solving these problems, which in turn gives rise to the birth of metaheuristic solution algorithms.

Metaheuristic algorithms are approximate solution methods for an optimization problem which use a randomness property with an 'educated guess' in their search mechanism and try to improve the quality of the solutions at hand through the iterations, from a randomly generated set of feasible solutions, by exploring and exploiting the solution space. Even though these algorithms do not guaranty optimality, they are tested to give a reasonable and acceptable solution. Furthermore, they have the advantage of not to be affected much by the behaviour of the problem; this makes them useful in many applications. Having a variety of algorithms will give the option to choose a suitable one to solve a problem according to its behaviour.

### 3. Studies on firefly algorithm

#### 3.1. Introduction

Nature has been an inspiration for the introduction of many metaheuristic algorithms. It has managed to find solution to problems without being told but through experience. Natural selection and survival of the fittest was the main motivation behind the early metaheuristic algorithms. Different animals communicate with each other through different mode of communications. Fireflies use their flashing property to communicate. There are around 2000 firefly species with their own distinct flash patterns. They usually produce a short flash with a certain pattern. The light is produced by a biochemical process called the bioluminescence. The flashing communication is used to attract their mate and also to warn predators. Based on the pattern of the light, a suitable mate will communicate back by either mimicking the same pattern or responding with a specific pattern. It also needs to be noted that the light intensity

decreases through distance; hence, a flashing light emanating from a firefly gets a response from fireflies around it within a visual range of the flash.

In addition to enjoying the beautiful view of a summer sky created by fireflies, they have motivated and have been the centre for many scientific researches [5, 21, 22]. In the sense of optimization, if we consider the fireflies as solution on the landscape of the solution space, then the attraction and movement of fireflies can inspire an optimization algorithm in which solutions follow better (brighter) solutions. Hence, firefly algorithm is motivated and inspired by these properties.

### 3.1.1. The standard firefly algorithm

Firefly algorithm is a swarm-based metaheuristic algorithm which was introduced by Yang [5]. The algorithm mimics how fireflies interact using their flashing lights. The algorithm assumes that all fireflies are unisex, which means any firefly can be attracted by any other firefly; the attractiveness of a firefly is directly proportional to its brightness which depends on the objective function. A firefly will be attracted to a brighter firefly. Furthermore, the brightness decreases through distance based on inverse square law, as given in Eq. (2).

$$I \propto \frac{1}{r^2} \quad (2)$$

If the light is passing through a medium with a light absorption coefficient  $\gamma$ , then the light intensity at a distance of  $r$  from the source can be given as in Eq. (3).

$$I = I_0 e^{-\gamma r^2} \quad (3)$$

where  $I_0$  is light intensity at the source. Similarly, the brightness,  $\beta$ , can be given as in Eq. (4).

$$\beta = \beta_0 e^{-\gamma r^2} \quad (4)$$

A generalized brightness function for  $\omega \geq 1$  is given in Eq. (5) [5]. In fact, any monotonically decreasing function can be used.

$$\beta = \beta_0 e^{-\gamma r^\omega} \quad (5)$$

In the algorithm, a randomly generated feasible solution, called fireflies, will be assigned with a light intensity based on their performance in the objective function. This intensity will be used to compute the brightness of the firefly, which is directly proportional to its light intensity. For minimization problems, a solution with smallest functional value will be assigned with

highest light intensity. Once the intensity or brightness of the solutions is assigned, each firefly will follow fireflies with better light intensity. For the brightest firefly, it will perform a local search by randomly moving in its neighbourhood. Hence, for two fireflies, if firefly  $j$  is brighter than firefly  $i$ , then firefly  $i$  will move towards firefly  $j$  using the updating formula given in Eq. (6).

$$x_i := x_i + \underbrace{\beta_0 e^{-\gamma r_{ij}^2}}_{=\beta} (x_j - x_i) + \alpha(\varepsilon() - 0.5) \quad (6)$$

where  $\beta_0$  is the attractiveness of  $x_j$  at  $r = 0$ , in [5] the author recommended that  $\beta_0 = 1$  for implementation,  $\gamma$  is an algorithm parameter which determines the degree in which the updating process depends on the distance between the two fireflies,  $\alpha$  is an algorithm parameter for the step length of the random movement and  $\varepsilon()$  is a random vector from uniform distribution with values between 0 and 1. For the brightest firefly,  $x_b$ , the second expression in Eq. (6) will be omitted, as given in Eq. (7).

$$x_b := x_b + \alpha(\varepsilon() - 0.5) \quad (7)$$

---

### Set algorithm parameters ( $\alpha, \gamma$ )

Set simulation set-up (Number of initial solutions and maximum iteration ( $N, MaxGen$ ))

Randomly generate  $N$  initial solutions

for iteration = 1 : MaxGen

```

    Compute the brightness,  $I$ 
    Sort the solution in such a way that ,  $I_i \geq I_{i-1}, \forall i$ 
    for  $i = 1 : n - 1$ 
        For  $j = i + 1 : n$ 
            If  $I_j > I_i$ 
                | move firefly  $i$  towards firefly  $j$ 
            end if
        end for
    end for
    move firefly  $N, (x_b)$ , randomly

```

end for

Report the best solution,

---

**Table 1.** The standard firefly algorithm.

These updates of the location of fireflies continue with iteration until a termination criterion is met. The termination criterion can be maximum number of iterations, a tolerance from the

optimum value if it is known or no improvement is achieved in consecutive iterations. The algorithm is summarized in **Table 1**.

### 3.2. Modified versions of firefly algorithm with critical analysis

Firefly algorithm is efficient and an easy-to-implement algorithm. It is also suitable for parallel implementation. However, researches show that it is slow in convergence and easily gets trapped in local optimum for multimodal problems. In addition, the updates solely depend on current performance and no memory on previous best solutions and performances are kept. That may lead to losing better solutions. Furthermore, since the parameters are fixed, the search behaviour remains to be the same for any condition in all iterations. Hence modifying the standard firefly algorithm to boost its performance has been one of the research issues. Furthermore, the standard firefly algorithm is designed for continuous optimization problems; hence in order to use it for non-continuous problems it needs to be modified and adjusted.

#### 3.2.1. Modification for problems with continuous variables

Basically, there are three classes of modification. Class 1 modification is the modification on the parameters. It is the first category in which the parameters of the algorithm are modified and the same updating mechanisms or formulas are used. Class 2 contains new updating mechanisms. It includes modifications which change part or all of the updating formulas, add mutation operator and the likes. The last category, Class 3, includes modifications on the search space, perhaps with the same updating mechanism it may be easier to switch to another 'easy-to-search' space, and changes in the probability distribution when generating random numbers. The categories are not necessarily disjoint as some of the modifications may fall in multiple classes.

##### 3.2.1.1. Class 1 (parametric modification)

In the standard firefly algorithm, the parameters in Eq. (6) are user-defined constants. Like any other metaheuristic algorithms, the performance of a firefly algorithm depends on these parameter values. They control the degree of exploration and exploitation.

Some of the modifications of firefly algorithm are done by making these parameters variable and adaptive. In recent researches on the modification of firefly algorithms, the parameters  $\alpha$ ,  $\gamma$  and also  $r$  are modified. The modification of  $\alpha$  affects the random moment of the firefly, whereas modifying either  $\gamma$  or  $r$  affects the degree of attraction between fireflies. Adjusting the brightness at the origin,  $\beta_0$ , has also been done in some researches.

##### a. Modifying the random movement parameter:-

To deal with parameter identification of infinite impulse response (IIR) and nonlinear systems, firefly algorithm is modified in [23]. The modification with the random movement is based on initial and final step lengths  $\alpha_0$  and  $\alpha_\infty$  using  $\alpha : = \alpha_\infty + (\alpha_0 - \alpha_\infty)e^{-Itr}$ . In addition, additional fourth term in the updating process, given by  $\alpha\epsilon(x_i - x_b)$  where  $x_b$  is the brightest firefly of all the

fireflies, is added so that firefly algorithm will resemble and have search behaviour like particle swarm optimization. In order to implement this modification, initial and final randomization parameters,  $\alpha_0$  and  $\alpha_\infty$  need to be supplied by the user. The randomized parameter is set to decrease exponentially and within a couple of iteration it will vanish. For example, if  $\alpha_0 = 1$  and  $\alpha_\infty = 0.2$  starting from 0.089 in the first iteration it will decrease to 0.0001 in the seventh iteration. Furthermore, the additional term in the updating formula takes the solution  $x_i$  away from  $x_b$ , with a given step length  $\alpha\epsilon$ . This is in contradiction to the following concept of the current best solution of the algorithm. Assuming that the step length for the new term as well as the randomness is the same, there is only one additional parameter, either  $\alpha_0$  or  $\alpha_\infty$  in place of a single parameter  $\alpha$ .

Another firefly algorithm with adaptive  $\alpha$  is presented in [24]. The modification is given by  $\alpha^{(ltr)} = \alpha^{(ltr-1)} \left( \frac{1}{2^{ltr_{Max}}} \right)^{\frac{1}{ltr_{Max}}}$ . In addition, two new solutions are generated based on three solutions from the population chosen randomly, the one with the better brightness from the two new solutions, and  $x_i$  will replace  $x_i$  and pass for the next iteration. It is also used to solve optimal capacitor placement problem [25]. Similar modification of  $\alpha$  with additional mutation and crossover operators is also given in [26].

In [27], the randomized parameter is modified based on the number of iterations using  $\frac{0.4}{1 + e^{\frac{0.005(ltr - ltr_{Max})}{0.4}}}$ . The simulation results in 16 benchmark problems show that the modification increases the performance of the standard firefly algorithm significantly.

In extending firefly algorithm for multiobjective problems, an adaptive  $\alpha$  is proposed and used in [28]. Here  $\alpha$  is made adaptive based on the number of iteration and is given by  $\alpha = \alpha_0 0.9^{ltr}$ . Hence, the step length decreases faster than linearly.

Self-adaptive step firefly algorithm is another modification done to the third term of the updating process by Yu et al. [29]. The step length  $\alpha$  is updated based on the historical information and current situation using  $\alpha_i^{(ltr+1)} = 1 - \frac{1}{\sqrt{f_b^{(ltr)} - (f_i^{(ltr)})^2 + (f_i^{(ltr)})^2 + 1}}$  where

$$h_i^{ltr} = \frac{1}{\sqrt{(f_{i_{best}}^{(ltr-1)} - f_{i_{best}}^{(ltr-2)})^2 + 1}} \text{ for } f_b^{ltr} = f(x_b), f_i^{ltr} = f(x_i) \text{ after iteration } ltr, f_{i_{best}}^{(ltr-1)} \text{ and } f_{i_{best}}^{(ltr-2)} \text{ the best}$$

performance of solution  $x_i$  until  $ltr - 1$  and  $ltr - 2$  iterations. Sixteen two-dimensional benchmark problems are used showing that the proposed approach produces better result with smaller standard deviation. It is a promising idea to control the randomized parameter based on the solution's previous and current performances. The update works in such a way that whenever the solution approaches the brightest firefly its step length will decrease, since the performance of the solutions needs to be saved and the memory complexity should be studied.

Another study of modification of the random movement parameter based on the historic performance of the solution is presented in [30]. Based on its best position until current iteration,  $x_{i,best}$  and the global best solution until current iteration,  $x_{gbest}$

$$\alpha_i^{(ltr+1)} = \alpha_i^{(ltr)} - (\alpha_i^{(ltr)} - \alpha_{min}) e^{-\frac{ltr |x_{gbest} - x_{i,best}|}{MaxGen}}.$$

### b. Modifying the attraction parameters

The attraction of one firefly by another depends on the light intensity at the source of the brighter firefly as well as on the distance between them and the light absorption coefficient of the medium.

For a small change in the distance between two fireflies results in a quick decrease of the attraction term. To deal with this problem, Lin et al. [31] introduced a virtual distance which will put  $r$  in between 0 and 1 and is defined by  $r' = \frac{r - r_{\min}}{r_{\max} - r_{\min}}$ , where  $r_{\min} = 0$  and

$r_{\max} = \sqrt{\sum_{i=1}^d (x_{\max}(i) - x_{\min}(i))^2}$  for  $x_{\max}(i)$  and  $x_{\min}(i)$  being the maximum and minimum values of the  $i^{\text{th}}$  dimension over all fireflies, respectively. Furthermore,  $\beta$  is set as  $\beta = \beta_0 \gamma (1 - r')$ . In later iterations, the swarm tends to converge around an optimal solution. It means that the distance  $r$  decreases and so does  $r_{\max}$ . However, in most cases, the decreasing rate of  $r$  is faster than  $r_{\max}$ , resulting in a slight increase in  $\beta$ . In order to overcome the possibility of the attraction term dominating the attraction term, the authors proposed a new updating equation in later iterations using  $x_i := x_i + \beta(x_j - x_i)\alpha\epsilon$ . Indeed the new updating formula omits the random movement of the firefly. The firefly will only move towards a brighter firefly with a step length of  $\beta\alpha\epsilon$ .

Tilahun and Ong [32] suggested that, rather than making  $\beta_0 = 1$ , it should be a function of the light intensity given by  $\beta_0 = e^{I_{0,j} - I_{0,i}}$  for a firefly  $i$  to be attracted to  $j$ , where  $I_{0,j}$  and  $I_{0,i}$  are intensity of fireflies  $j$  and  $i$  at  $r = 0$ . In addition, moving the brightest firefly randomly may decrease the brightness; hence a direction which improved the brightness will be chosen from  $m$  random directions. If such a direction is not among these  $m$  directions, it will stay in its current position. The complexity of the algorithm may increase with respect to the new parameter  $m$ , and therefore it should be taken into consideration.

Due to the non-repetition and ergodicity of chaos, it can carry out overall searches at higher speeds. Hence, Gandomi et al. [33] proposed a modification on  $\beta$  and  $\gamma$  using chaos functions. This approach has attracted many researchers, and it has been used in different problem domains. The approach is successfully applied using Chebyshev chaos mapping for MRI brain tissue segmentation in [34], for heart disease prediction using Gaussian mapping [35], reliability-redundancy optimization [36] and for solving definite integral problems in [37]. In [38], chaotic mapping is used for  $\beta$  or  $\gamma$ . In addition,  $\alpha$  is made to decrease based on the intensity of solutions using  $\alpha = \alpha_{\max} - (\alpha_{\max} - \alpha_{\min}) \frac{I_{\max} - I_{\text{mean}}}{I_{\max} - I_{\min}}$  where  $I_{\max}$ ,  $I_{\text{mean}}$ , and  $I_{\min}$  are the maximum, the average and the minimum intensities of the solutions.

Another modification in this category is done in [39]. In this chapter,  $\beta$  is modified using  $\beta = (\beta_{\max} - \beta_{\min})e^{-\gamma r^2} + \beta_{\min}$ , where  $\beta_{\min}$  and  $\beta_{\max}$  are user-supplied values. Similar modification is also done in [40].

### c. Modifying both the random and attraction movement parameters

To overcome this challenge arising with an increase in the problem dimension and the size of the feasible region, Yan et al. [41] proposed a modification for the standard firefly algorithm. This modification is done on the generalized distance term given in Eq. (5), in which  $r^w$  is replaced by  $r^{K\sqrt{n(Range)}}$ , where  $K$  is a constant parameter,  $n$  is the dimension of the problem and  $Range$  is the maximum range of the dimensions. The parameter  $\alpha$  also reduces with iteration from a starting value  $\alpha_0$  to a final value  $\alpha_{end}$  linearly. In addition, a firefly is attracted to another firefly if it is brighter and if it is winking. The winking is based on a probability  $p_w = 0.5 + 0.1count\_i$ , where  $count\_i$  is the value of a firefly  $i$  winking state counter. The larger the counter the greater will be the probability of shifting the state. The maximum counter is five, and after that it will be reset to 0.

In order to solve economic dispatch problem, firefly algorithm is modified in [42]. To increase the exploration property, the authors replaced the Cartesian distance by the minimum variation distance. In addition, they used mutation operator on  $\alpha$  but no explanation on how the mutation works is given.

To deal with premature convergence, firefly algorithm has also been modified based on the light intensity [43]. The light intensity difference is defined by  $\xi = \frac{\Delta I_j^{(t)}}{\max\{I\} - \min\{I\}}$  for an iteration  $t$ . Based on  $\xi$ , a modification is done on  $\gamma$ ,  $\beta$  and  $\alpha$  as follows,  $\gamma = \frac{\gamma_0}{r_{\max}^2}$  where  $r_{\max} = \max\{d(x_i, x_j) \mid \forall i, j\}$ ,  $\beta_0 = \begin{cases} \xi, & \xi > \eta_1 \\ \eta_1, & \xi \leq \eta_1 \end{cases}$  where  $\eta_1$  is a new parameter and  $\alpha = \alpha_0(0.02r_{\max})$  where  $\alpha_0 = \begin{cases} \xi, & \xi > \eta_2 \\ \eta_2, & \xi \leq \eta_2 \end{cases}$  for another new algorithm parameter  $\eta_2$ . The modification shows that for two fireflies, the brighter one will have a small attraction and randomness step length compared to the brighter ones.

For the optimal sizing and siting of voltage-controlled distribution generator in distributed network, firefly algorithm is modified and is used in [44]. The problem is to minimize the power loss by selecting optimal location for distributed generations and the power produced. In the modification  $\beta_0 = 1$ , whereas  $\gamma$  and  $\alpha$  are modified based on the problem property (location and maximum power per location in each iteration). This modification is done based on the problem characteristic. The effectiveness and quality of a solution for a metaheuristic algorithm depend on the proper tuning of an algorithm parameter as well as on the behaviour of the landscape of the objective function.

Another modification of the standard firefly algorithm to be listed in this category is done in [45]. The randomized parameter  $\alpha$  has been made adaptive using  $\alpha = \alpha_{\max} - \frac{Iter(\alpha_{\max} - \alpha_{\min})}{MaxGen}$ . Furthermore, the distance function has been made to be influenced not by their location in the landscape of the feasible region but the brightness or functional values of the fireflies using  $f(x_b) - f(x_i)$ . For two fireflies with similar performance in the objective function, they are considered to be near each other.

For path planning of autonomous underwater vehicle, the parameters  $\gamma$  and  $\alpha$  of the standard firefly algorithm are modified by  $\gamma = \gamma_b + \frac{itr}{MaxGen}(\gamma_e - \gamma_b)$  for  $\gamma_e > \gamma_b$  and  $\alpha = \alpha_b + \frac{itr}{MaxGen}(\alpha_e - \alpha_b)$  for  $\alpha_e < \alpha_b$  [46]. Furthermore, the updating formula is defined as  $x_i := x_i + \beta(x_i - x_j) + \alpha r \epsilon$ . As the iteration increases,  $\alpha$  decreases and  $\gamma$  increases linearly, implying both the randomness movement and the attraction decrease as a function of the iteration. In the updating formula, the random movement is multiplied by the distance.

A similar approach in which the parameters  $\alpha, \beta$  and  $\gamma$  are encoded in the solution is proposed in [47]. Unlike in [44], the update is done using  $\psi := \psi + \sigma_\psi N(0, 1)$  where  $\sigma_\psi := \sigma_\psi e^{\tau' N(0,1) + \tau N(0,1)}$  for learning parameters  $\tau, \tau'$  and  $\psi = \{\alpha, \beta, \gamma\}$ . Another modification that can be listed in this category is done in [48]. The parameter  $\gamma$  is modified using  $\gamma_{\max} - (\gamma_{\max} - \gamma_{\min}) \left( \frac{itr}{MaxGen} \right)^2$  for  $2 \leq \gamma_{\max} \leq 4$  and  $0.5 \leq \gamma_{\min} \leq 1$ . In addition, for a new parameter  $\lambda$ ,  $\alpha = \alpha_{\max} - (\alpha_{\max} - \alpha_{\min}) \left( \frac{itr - 1}{G_0 - 1} \right)^\lambda$  where  $G_0$  is an iteration number in which  $\alpha = \alpha_{\min}$ . This results in a decrease in  $\alpha$  quicker than linear function if  $\lambda$  is in the range  $(0, 1)$ , linearly if  $\lambda = 1$  and slower than linear function if  $\lambda > 1$ . Furthermore, in order to overcome the trapping of the solutions in local optimal solution, Gauss distribution is applied to move the brightest solution,  $x_b$ , i.e.  $x_b := x_b + x_b N(\mu, \sigma)$ . This will be applied if the variance of the solutions before a predetermined  $M$  iteration is less than a given precision parameter  $\eta$ . The authors also suggested that chaos, particularly cubic mapping, can be used to improve the distribution of the initial solutions.

In [49],  $\gamma$  and  $\alpha$  are computed using  $\gamma = 0.03 |G_1|$  and  $\alpha = 0.03 |G_2|$  where  $G_1$  and  $G_2$  are generated from Gaussian or normal distribution with mean 0 and variance 1. Supported by two case studies for multivariable proportional–integral–derivative (PID) controller tuning, a similar study was also done in [50]. They used Tinkerbell mapping to tune  $\gamma$ , using  $\gamma = |G| \bar{x} \frac{itr}{MaxGen}$  where  $\bar{x}$  has normalized values generated by the Tinkerbell map in the range  $[0, 1]$ . In addition to that,  $\alpha$  is modified to decrease linearly using  $\alpha = (\alpha_{final} - \alpha_{initial}) \frac{itr}{MaxGen} + \alpha_{initial}$ .

### 3.2.1.2. Class 2 modifications (new updating mechanisms)

The updating mechanism in the standard firefly algorithm is guided by Eqs. (6) and (7). In Class 1 modification, the same updating equations are used but with adaptive preference. Class 2 modifications include modification on the updating equations including modification in the updating process of the best (the brightest) and the worst (the dimmer) solutions changing part of the updating equations and some modification with additional mutation operator.

#### a. Modifying the movement of the brightest or dimmer firefly

In a high dimensional problem, the exploration is weak which results in premature convergence. To deal with this, two modifications are proposed in [51] for the standard firefly algorithm. That is, for the initial random  $N$  solutions, their opposites will be generated, and the best  $N$  solutions will be chosen from the  $N$  solutions and their opposites where an opposite

number for  $x$  is given by  $x_{\min} + x_{\max} - x$ . The brightest solution  $x_b$  will be updated as follows:

```

 $y = x_b$ 
for  $i = 1 : D$  (for all dimensions)
    for  $j = 1 : N$  (for all the solutions)
         $y(i) = x_j(i)$ 
        if  $[f(y) \text{ is better than } f(x_b)]$   $x_b = y$  end if
    end for
end for

```

Similar to the previous modification, here also the best solution will improve or will not change in each of the iterations.

Opposition-based learning is also used in [52], to update the dimmer solution  $x_w$ , a solution with worst performance, using  $x_w = \begin{cases} x_b, & \varepsilon < p \\ x_{\min} + x_{\max} - x_w, & \text{Otherwise} \end{cases}$  for an algorithm parameter  $p$ .

Indeed, it relocates the worst solution to a new position that may give the algorithm a good explorative behaviour.

#### *b. Mutation incorporation*

Jumper firefly algorithm is a modified firefly algorithm in which a memory on the performance of each of the solution is kept [53]. A criterion called hazard condition is defined, and solutions will be tested based on their previous performance. If they are in hazardous condition, they will be randomly replaced by a new solution. Hence, based on the hazard condition, a mutation can be done by replacing the weak solution based on previous performance by a new solution.

Another modification in this category is done by Kazemzadeh-Parsi [54], where each iteration  $k$  random 'newborn' solutions will be generated to replace the weak solutions. In addition to this mutation, highly ranked  $k_1$  solutions from the previous iteration will replace the same number of weak solutions. The other modification is that, rather than having consecutive movement of a firefly towards brighter fireflies, a single combined direction, the average of the directions  $\left(\frac{1}{T} \sum_{i=1}^T x_i\right)$  for brighter fireflies  $x_j$ 's, will be computed and used. Similar approach is used in [55]. In the first case, where newborn solutions replace weak solutions, the number of solutions should not be large; otherwise it behaves as an exhaustive search. In the second case, whenever some solutions are replaced by others from the previous solutions, the solutions coming from the previous iteration will more or less perform similar search behaviour as what has been done in the previous iteration.

Another modification of firefly algorithm by introducing new solutions as a mutation or crossover is given in [26, 56]. In addition to adaptive parameter  $\alpha$ , they introduced two mutation operators and five crossover operators based on the mutated solutions. The first

mutation operator works by combining randomly selected three solutions,  $x_{q1}$ ,  $x_{q2}$  and  $x_{q3}$ , different from  $x_i$ , from the solution set  $[x_{mute1} = x_{q1} + \varepsilon(x_{q2} - x_{q3})]$  and the second based on the mutated solution from the first mutation operators, the best and worst solutions (for an iteration  $t$   $[x_{mute2} = x_{mute1} + \varepsilon t(x_b - x_w)]$ ). Based on these two solutions, five solutions will be generated, and the best one from the mutated as well as the new five solutions replace  $x_i$ . Similar modification of  $\alpha$  and two mutation types are also proposed in [24]. In [57], the parameter  $\alpha$  is made to adapt using chaotic mapping and mutation operators.

### c. New updating strategy

This is another category of Class 2 modification in which the updating formula, given by Eqs. (6) and (7), is modified or changed. The first modification, to mention in this category, is the modification proposed in [57]. For a firefly  $i$  attracted by another firefly  $j$ , the search is updated to be in the vicinity of  $x_j$ , as given by  $x_i := x_j + \beta(x_j - x_i) + \alpha\varepsilon$ . Furthermore, after the update, only improving solutions are accepted. Since the update is done based on the location of  $x_j$ , the exploration property of points in between the two solutions will not be done, and it will be trapped in local optimum solution provided  $x_j$  is a local solution, and the step lengths are small. Through iterations, the solutions will be forced to be in a neighbourhood of the best solution. The diversity of the solutions will also be low.

A similar modification in the vicinity of the brighter firefly is given in [58]. They proposed two updating formulas, with and without division, as the authors name them. The updating formula, without division, is given by  $x_i := x_j + \alpha\varepsilon$ . Once the fireflies are sorted according to their brightness, increasing with their index, the updating formula will be defined by  $x_i := x_j + \frac{\alpha}{j}\varepsilon$ , which will decrease  $\alpha$  as the brightness increases and which will give a good intensification or exploitation property. In addition, the parameters  $\alpha$  and  $\gamma$  are made adaptive. This put this modification in both Class 1 and Class 2 in our classification. Similar discussion holds for a similar work done in [57]. In addition, unlike in [59], there is an attraction term in the direction from the brighter  $x_j$  towards  $x_i$ , that means moving the brighter firefly in a non-promising direction replaces  $x_i$ . Hence, in this sense, the modification in [58] is better as it moves the solution not in a non-promising direction but randomly.

For a data clustering problem, the standard firefly algorithm is modified firefly algorithm [60, 61]. They proposed a new updating formula to increase the influence of the brightest firefly.

The new updating formula is given by  $x_i := x_i + \beta(x_j - x_i) + \beta_0 e^{-\gamma r_{i,gbest}^2} (x_{gbest} - x_i) + \alpha\varepsilon$ . This means that a firefly is not only attracted to brighter fireflies but also by the best solution found so far,  $x_{gbest}$ . Suppose there are  $l$  brighter fireflies, brighter than  $x_i$ , at the end of the iteration the

attraction term will be  $\sum_{j=1}^l \beta(x_j - x_i) + l\beta_0 e^{-\gamma r_{i,gbest}^2} (x_{gbest} - x_i)$ . Hence, repeatedly moving a firefly to the best solution increases the attraction step length, and based on the feasible region, it usually may not be acceptable. Furthermore, the global solution found can be an optimal solution, and it may result in the solutions to be forced to follow that local solution rather than exploring other regions in each loop of iteration. In [61], four ten-dimensional benchmark problems and

four twenty-dimensional benchmark problems along with Iris data set are used for clustering. Similar modification is done in [62]. In addition, to update  $\alpha$  in a decreasing manner, the updating formula for a solution  $x_i$  based on the brighter firefly  $x_b$  and the best solution from memory  $g_{best}$  with a new algorithm parameter  $\lambda$  is modified as  $x_i := x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \beta_0 e^{-\gamma r_{i,best}^2} (x_b - x_i) + \lambda \varepsilon (x_i - g_{best}) + \alpha \varepsilon$ .

Fuzzy firefly algorithm is another modification of the standard firefly algorithm [63]. Even though they start with a wrong claim by saying "*in the standard firefly algorithm, only one firefly in each iteration can affect others and attract its neighbours*", they try to increase the exploration property of the algorithm by adding an additional term in which the top  $k$  fireflies attract according to  $x_i := x_i + [\beta_0 e^{-\gamma e_{ij}^2} (x_j - x_i) + \sum_{h=1}^k A(h) \beta_0 e^{-\gamma e_{ih}^2} (x_h - x_i)] \alpha \varepsilon$ , where  $A(h) = \frac{f(x_b)}{l(f(x_h) - f(x_b))}$  with  $l$  being a new algorithm parameter. The effect of the best  $k$  fireflies is doubled, and if this updating mechanism is done for each brighter firefly  $x_j$  then its effect is more than double. Furthermore, multiplying the random term with the second expressions affects their step length and deletes the random movement. Hence, it forces the fireflies to follow best  $k$  solutions. Exploring other regions is not possible with this update.

Another modification with a new updating formula is proposed in [64] and is given by 
$$x_i := \begin{cases} x_i + \beta(x_j - x_i) + \alpha \varepsilon (x_{\max} - x_{\min}), & \text{if } \varepsilon > 0.5 \\ \frac{\text{MaxGen-Itr}}{\text{MaxGen}} (1 - \eta) x_i + \eta x_b, & \text{otherwise} \end{cases}$$
 where  $\beta_0$  is computed based on the location of  $x_i$  normalized by the locations of fireflies in the search space and  $\gamma$  is computed with a direct relation with  $\beta_0$  and additional two parameters;  $\eta$  is a value based on the difference between the location of the fireflies.

Diversity-guided firefly algorithm is one of the recent modified versions [65]. The modification is done to make the solutions as diverse as possible with a given threshold. The updating mechanism of the standard firefly is used until diversity of the solution falls beyond the given threshold. The diversity is measured by  $\frac{1}{NL} \sum_{i=1}^N \|x_i - \bar{x}\|$  where  $L$  is the longest diagonal of the feasible region, and  $\bar{x}$  is the average position of all fireflies. If the diversity is less than a predefined threshold value, then the updating formula will be  $x_i := x_i + \beta(x_j - x_i) + \alpha \varepsilon (x_i - \bar{x})$ . The modification proposed is effective in diversifying the solutions by replacing the random movement by moving the solutions away from the average position of fireflies.

In [66, 67], a mutated firefly algorithm is proposed in such a way that the brighter firefly donates some of its features based on a new algorithm parameter called probability of mutation,  $p_m$ . The features and their amount copied from the bright firefly are not mentioned. However, based on the context, it seems some components of the vectors for  $x_i$  will be replaced by the corresponding components from the brighter firefly  $x_j$ . In [66], this mutation operator will replace the updating formula given in Eq. (6) whereas in [67], the mutation will be done after the update is taken place.

In [68], a firefly located at  $x_i$  first checks the direction towards other brighter fireflies and looks for the one that improves its performance. If there exists such a solution in which  $x_i$  moves

towards the firefly, its brightness increases. Checking the direction towards each of the solution may increase the complexity. Furthermore, in order to escape a local solution some solution should be allowed to decrease its performance; hence in this modification it is highly affected by local optimum solutions especially in misleading problems.

Another modification in this category is introduced to deal with economic dispatch optimization of thermal units [69]. A memory is used to record the best solution found so far. Based on cultured differential evolution, the updating formula is modified as  $x_i := x_i + a\beta(gbest - x_i) + b\alpha(\epsilon) - 0.5$  where  $a = \frac{f(x_i) - f(gbest)}{f_{max} - f_{min}}$  and  $b = x_{max} - x_{min}$  for  $x_{max}$  and  $x_{min}$  being the maximum and minimum component of vector  $x$ , respectively.

Another modification in this category is presented in [70]. The updating formula becomes  $x_i := w x_i + \beta(x_j - x_i) + \alpha\epsilon()$  for a weighting parameter  $w$  given by  $w = w_{max} - (w_{max} - w_{min}) \frac{itr}{MaxGen}$ .

### 3.2.1.3. Class 3 modifications (change in search space or probability density functions)

This class of modifications is on the abstract level modification and includes two types of modifications. The first one is changing the solution space to an easy search space, and the second one is on the types of probability distribution that is used to generate a random vector direction for the random movement.

#### a. Change in search space

In the modified version presented in [71], each component of a solution will be represented by quaternion  $x_i(k) = (y_1^{(i)}, y_2^{(i)}, y_3^{(i)}, y_4^{(i)})$  for all components  $k$  of  $x_i$ , and the updating will be done over the quaternion space. In order to compute the brightness, the Euclidian space is used by changing the quaternion space to the search space by taking the norm function,  $x_i(k) = \|(y_1^{(i)}, y_2^{(i)}, y_3^{(i)}, y_4^{(i)})\|$ . Even though the search space increases fourfold, it is interesting to zoom in into each component and perform the search for optimal solution. However, since a norm is used to convert quaternion space to the search space, a mechanism to deal with negative values should be studied. A more mathematical support should be provided along with complexity study.

#### b. Change in probability distribution function

Perhaps the first work which tries to adapt the randomness movement in the updating process is by Farahani et al. [72, 73]. Even though they started with a wrong claim by saying 'In standard Firefly algorithm, firefly movement step length is a fixed value. So all the fireflies move with a fixed length in all iterations' by ignoring the random variable  $\epsilon$  that makes the step length between 0 and  $\alpha$ . They updated the step length to decrease with iteration and introduced a new parameter which updates each solution using  $x_i := x_i + \alpha\epsilon(1 - p)$ , where  $p$  is a random vector from Gaussian distribution. This will increase the randomness property of the algorithm as it randomly moves once using the usual updating equation. The same modification is also employed in [74].

By enhancing the random movement of a firefly algorithm, Levy firefly algorithm is introduced in [75]. This is the first modification made to firefly algorithm with the Levy distribution guiding the random movement by generating a random direction as well as the step length. The update formula is modified as  $x_i := x_i + \beta(x_j - x_i) + \alpha \text{sign}(\varepsilon) \otimes \text{Levy}$ ;  $\otimes$  indicates a component-wise multiplication between the random vector from the Levy distribution and the sign vector. Similarly, in [76], Levy distribution is used to guide the random term of the updating formula.

In addition, the parameter  $\alpha$  is made to decrease with iteration, using  $\alpha = \frac{\alpha_{\max}}{It_r^2}$ . Furthermore, what they call information exchange between top fireflies will be done. That is, two solutions are randomly chosen from the top fireflies, and a new solution on the line joining the two fireflies near the brightest one will be generated. Similar approach of using Levy distribution with the step length is generated using a chaotic random number and has applications in image enhancement [77]. The same updating using Levy distribution and same formula for  $\alpha$  is used in [78]. In addition to these updates, a communication between top fireflies is used in [79]. Levy distribution along with other probability distribution is suggested for the randomized parameter and used in [80, 81].

### 3.2.2. Modifications for problems with non-continuous variables

Even though firefly algorithm is introduced for continuous problems, due to its effectiveness it has been modified for non-continuous problems as well. In this section, we will look at three classes of modification. The first one is when modifications are made to solve binary problems. The second is for integer-valued problems which include problems whose variable can have discrete values. The last one is mixed problems in which some of the variables are continuous and the rest are non-continuous.

#### 3.2.2.1. Modifications for binary problems

To deal with set covering problem, a binary firefly algorithm is proposed in [82]. There is no modification in the updating process except converting the solution to either one or zero. Three ways of conversion are proposed in [82]. The conversion works dimension wise. After a solution  $x_i$  is updated, for each component  $p$  of  $x_i$ , three rules based on a transfer function  $T$ , which will change the new value of  $x_i$  in the interval  $[0, 1]$ , are given with eight transfer functions. The first rule of conversion is given by  $x_i(p) = \begin{cases} 1, & \varepsilon < T(x_i(p)) \\ 0, & \text{Otherwise} \end{cases}$ . The second is

$x_i(p) = \begin{cases} (x_i^{(t)}(p))^{-1}, & \varepsilon < T(x_i^{t+1}(p)) \\ x_i^{(t)}(p), & \text{Otherwise} \end{cases}$ , where  $(x_i^{(t)}(p))^{-1}$  is the complement of  $x_i^{(t)}(p)$  i.e. if  $x_i^{(t)}(p) = 1$

then  $(x_i^{(t)}(p))^{-1} = 0$ , otherwise  $(x_i^{(t)}(p))^{-1} = 1$ ,  $x_i^{(t)}(p)$  and  $x_i^{(t+1)}(p)$  are the  $p^{\text{th}}$  components of  $x_i$  from the previous iteration and after the update. The last rule is given by

$x_i(p) = \begin{cases} (x^*(p)), & \varepsilon < T(x_i^{t+1}(p)) \\ 0, & \text{Otherwise} \end{cases}$  where  $x^*$  is the best memory from memory.

Another modification for binary problems which works dimension wise, in each dimension, is presented in [83]. The update formula of the standard firefly algorithm is used. After the

update, the solution will lie in the interval  $[0, 1]$  using  $\tanh(x_i(p))$  function for the  $p^{\text{th}}$  component of solution  $x_i$ . Based on the user-defined threshold value,  $x_i(p)=1$ ; if the result is greater than the threshold value, then it will be 0. Another similar work of using sigmoid function is given in [84, 85]. In [86–88], tangent hyperbolic sigmoid function is used for discretizing the solution and also in the updating process using  $x_i = \begin{cases} x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \varepsilon, & \text{if } \varepsilon < \left| \tanh(\lambda r) \right| \\ x_i, & \text{else} \end{cases}$ , where  $\lambda$  is a parameter close to one.

Another discrete firefly algorithm, in order to deal with job, schedule problem is proposed in [89]. Each firefly  $x_i$  in the problem will have two classes of index as  $x_i(p, q)$  where  $p$  represents the jobs and  $q$  their priority in that particular firefly. In order to change real values to binary after the update as sigmoid function,  $\frac{1}{1 + e^{-x_i(p,q)}}$  is used. Based on the values of the sigmoid function for each job  $p$ , the one with higher probability in  $q$  will be assigned with value 1 and the rest priority for that particular job with 0.

In [90], for a dynamic knapsack problem, firefly algorithm has been modified. The conversion of the solutions is done based on the property of the problem using priority-based encoding. In addition to making the algorithm to suit for the problem, some modifications are done to increase its effectiveness. One of the modifications is that a firefly  $i$  moves towards firefly  $j$  if  $j$  is brighter and  $\varepsilon < \text{rank}_i \frac{\text{mode}(l_{tr-1}, \text{MaxGen})}{\text{MaxGen}}$ , where  $\text{rank}_i$  is the rank of firefly  $i$  in the solution population. If the condition is not met, i.e. if  $\varepsilon \geq \text{rank}_i \frac{\text{mode}(l_{tr-1}, \text{MaxGen})}{\text{MaxGen}}$ , no updating mechanism is mentioned in the chapter. A similar modification is used in [91]. In addition to the discretization done in [90], the authors in [91] proposed adaptive step length given by  $\alpha = \left[ 1 - \phi e^{-\text{mod}\left(\frac{l_{tr}}{l_{tr+1}}, 1\right)} \right] \alpha$  for a scaling parameter  $\phi$ . Furthermore, after the updates, two additional moves are introduced. The first one is a random flight by 10% of the top fireflies with 0.45 probability. The move will be accepted only if it is improving. The second is a local search of  $x_{b_r}$  after 10% of the iterations  $x_{b_r}$  will do a local search, and the update will be accepted if it is improving. The additional local searches help to improve the quality of the solution. Furthermore, they also mentioned that chaotic mapping can be used to generate random numbers.

Another modification in this category is presented in [92]. In addition to the discretization, they have made  $\alpha$  and  $\gamma$  adaptive using  $\alpha = \alpha_{\max} \frac{l_{tr}(\alpha_{\max} - \alpha_{\min})}{\text{MaxGen}}$  and  $\gamma = \gamma_{\max} e^{\frac{l_{tr}}{\text{MaxGen}} \ln \frac{\gamma_{\min}}{\gamma_{\max}}}$ . Furthermore, the random movement is replaced by  $\alpha L(x_b) |x_i - x_b|$  for a random number  $L(x_b)$  from Levy flight centred at  $x_b$ . Three discretization methods, the sigmoid, elf function and rounding function are used to change values in the range  $[0, 1]$  along with three updating processes. The first one is done on the continuous space, and sigmoid function will be used to change the results to binary; the second one is the update done on the discrete space, and the discretized results can be used and the third one, instead of using the updating formula, uses a probabilistic method based on sigmoid function.

### 3.2.2.2. Modifications for integer optimization problems

In [93], firefly algorithm has been modified to deal with software modularization as a graph-partitioning problem. Initially, random integer-encoded solutions are generated. The hamming distance, the number of different entries between two solutions with the same index, is used to measure the distance between two solutions. The update is done by switching a number of entries of a firefly by the entries from a brighter firefly.

Another modification in this category is done in [94]. The modification is based on a concept of random key, which is proposed in [95]. The method uses a mapping of a random number space,  $[0,1]^D$ , to the problem space.

In [96, 97], the standard firefly algorithm is modified for loading pattern enhancement. The generation of random solutions uses random permutation, and the distance between fireflies,  $d(x_i, x_j)$ , is measured using hamming's distance. The updating process is separated and made sequentially; first the  $\beta$  step, a move due to the attraction, and next the  $\alpha$  step, a move due to the random movement. In the  $\beta$  step, first same entries with same index for both fireflies,  $x_i$  and  $x_j$ , are preserved and then an entry will be copied from  $x_j$  if  $\varepsilon < \beta$ , where  $\beta = \frac{1}{1 + \gamma d(x_i, x_j)^2}$ ; otherwise the  $x_i$  entry will be used. The  $\alpha$  step is done using  $x_i := \text{Int}(x_i + \alpha \varepsilon)$ , with a swapping mechanism to preserve feasibility. A similar modification of sequentially applying  $\beta$  step and

$\alpha$  step is also used in [98], with additional modification on  $\beta$  and  $\alpha$  using  $\beta = e^{-\frac{(\max\{P_i\} - P_j)^2}{\max\{P_i\}}}$  for  $P_{ij} = \varepsilon + \frac{1}{\lceil \text{rank}_i - \text{rank}_j \rceil}$  and  $\alpha = \lfloor D - \frac{D}{\text{MaxGen}} \rfloor$ . It is a good idea to adapt and increase the step length with the dimension of the problem. For instance, when  $D = 12$ , the step length  $\alpha$  will start from 11 and decrease to zero in last iterations. However, if the feasible region is in  $[0, 4]$ , the search in more than 60% of the time  $\alpha$  will be at least 4. Hence, the moves in the first 60 plus % are not acceptable because it will force the solution out of the feasible region. Hence, the modification needs to consider the size of the feasible region. Another similar modification with the modifications done in [97] is given in [99] with additional modification to keep the best solution and use it in the updating process. That is based on  $\rho = 0.5 + \frac{0.5 \text{It}_r}{\text{MaxGen}}$ , and if  $\varepsilon > \rho$  the brighter firefly  $x_j$  will be replaced by the global best from memory.

For travel salesman problem, firefly algorithm has been modified in [100]. Initial solutions are generated using permutation of  $D$ , and each solution is represented as a string of chromosomes of these numbers. The distance between two solutions is computed using  $r = \frac{10A}{D}$ , where  $A$  is the number of different arcs. The movement is done randomly selecting the length of movement between 2 and  $r$  and then using inversion mutation towards better solution; if there is no better solution, a random move will be done. Each firefly will produce  $m$  solutions and the best  $N$  solutions will be chosen to pass to the next generation.

Another modification in this category is proposed in [101]. The decision variables,  $x_i$ 's, represent assembly sequence. In the update, the random movement is omitted, and the attraction move is done in the discrete space. The attraction direction is computed for each

dimension  $k$  using  $s_{ji}(k) = \begin{cases} x_j(k), & \text{if } x_j(k) \neq x_i \\ 0, & \text{else} \end{cases}$ , and the update will be done by  $x_i := x_i + S_{ji}$  where

$$S_{ji} = \begin{cases} s_{ji}, & \text{if } \alpha | \varepsilon - 0.5 | < \beta_0 e^{-\gamma r^2} \\ 0, & \text{else} \end{cases}.$$

In addition, the visual range,  $dv$ , which will control a firefly to be influenced by fireflies in its visual range, is introduced. The visual range is computed by

$$dv = \begin{cases} \frac{3Itr(dv_{\max} - dv_{\min})}{2(MaxGen - 1)} + dv_{\min}, & \text{if } Itr < \frac{2}{3}MaxGen \\ dv_{\max}, & \text{otherwise} \end{cases}$$

for parameters  $dv_{\max}$  and  $dv_{\min}$ . This means that a firefly will not be attracted to any brighter firefly but to a brighter firefly in a visible range.

Firefly algorithm has been discretized for supply selection problem in [102]. The sum of the absolute differences between the entries is used to measure the distance  $r = \sum_{k=1}^D |x_j(k) - x_i(k)|$ . In

addition, the movement is modified based on the property of the problem using rounding up for step length. In most cases, a modification specific to a problem is effective for that particular problem or a class of problems. However, it is hard to generalize the problems in other domains. It would be interesting to generalize the approach to be tested in other problem domains as well.

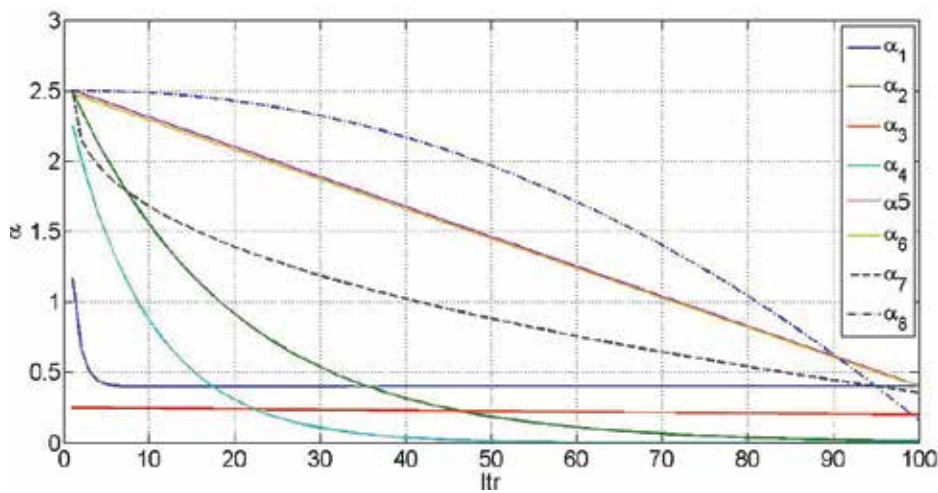
### 3.2.2.3. Modifications for mixed optimization problems

Perhaps the first modification to the standard firefly algorithm in this category is presented in [103]. The updating of solutions is conducted using the updating mechanism of the standard firefly algorithm. To deal with the discrete variables, constraint handling mechanism is used based on penalty function. In addition, the authors proposed two ways to generate a diverse set of random initial solutions. An adaptive random step length is also proposed using similar updating way in [104]. The same approach is improved in [105] by adding a scaling parameter for the random movement based on the difference between the maximum and minimum values for each variable. Portfolio optimization can be expressed as a mean-variance problem which belongs to the group of quadratic mixed-integer programming problems. In [106, 107], firefly algorithm has been extended with the use of rounding function and constraint handling approach. Deb's method [108] is also used for constraint handling. In addition,  $\alpha$  is modified

$$\text{using } \alpha := \alpha \left[ 1 - \left( 1 - \left( \frac{10^{-4}}{9} \right)^{\frac{1}{MaxGen}} \right) \right].$$

## 3.3. Discussion

Like any metaheuristic algorithm, firefly algorithm is prone to parameter values. It is noticed that changing the parameters based on the search state is effective. Hence, modification on parameters is a direct forward idea to improve the performance of firefly algorithm. As the search proceeds, in order to have a conversion with good precision, the randomness movement must decrease. Hence, the randomness step length,  $\alpha$ , is modified to be adaptive in which its value decreases with iteration [23–28, 41, 45, 46, 48]. **Figure 1** shows the graph of the modifications.



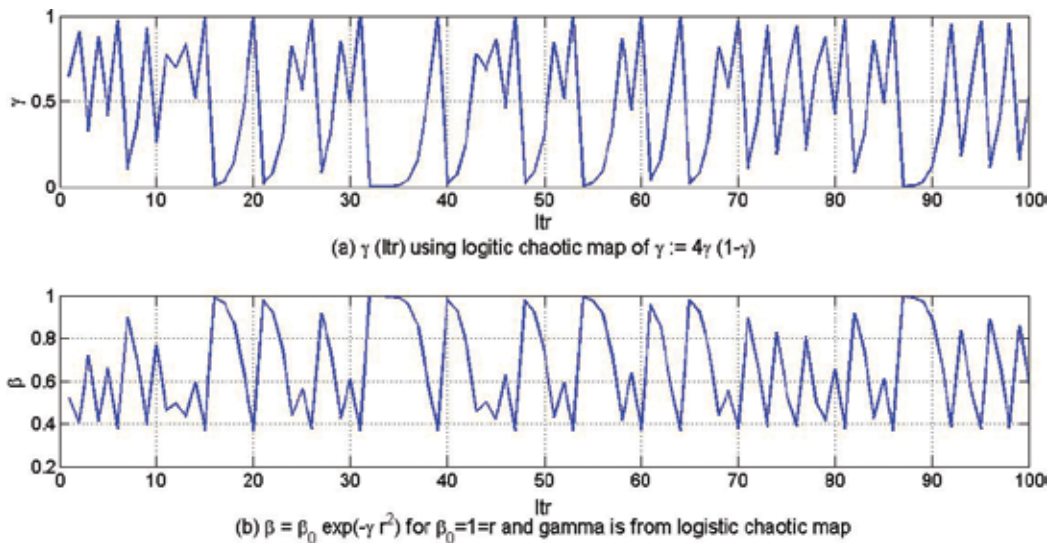
**Figure 1.** With initial and final values of 2.5 and 0.4;  $\alpha_1$  [23],  $\alpha_2$  [24–26],  $\alpha_3$  [27],  $\alpha_4$  [28],  $\alpha_5$  [41],  $\alpha_6$  [45, 46, 50],  $\alpha_7$  [48] with  $\lambda = 0.4$  and  $\alpha_8$  [48] with  $\lambda = 2.1$ .

The decreasing scenario for  $\alpha$  starting from the first iteration may not always be a good idea. Perhaps, it is better to keep a constant  $\alpha$  for a number of iterations and start the decreasing scenario based on the performance of the solutions, especially when the solution approaches an optimum point. This can be one of the possible ideas for future work. Some modifications are done making the parameter  $\alpha$  adaptive based on the performance of the solution [29, 38, 43]. Some of the modifications also involve a random term, and it behaves neither in decreasing nor in an increasing way [47, 49, 59]. In addition, other approaches such as encoding the parameters in the solution [109] and modifying the parameters based on the problem [44] are also proposed.

The attraction term has also been modified in different ways. Adaptive light absorption constant of the medium changing with iteration is given in some studies. This modifications use increasing function [46], decreasing function [43, 67] or neither increasing nor decreasing function [33–38, 49, 50] of  $\gamma$ . The modification is neither increasing nor decreasing especially when a chaotic distribution [33–38] or normal distribution [34, 49, 50] is used to compute the update. Increasing  $\gamma$  implies the decrease of the attraction step length, and its decrease implies an increase in the step length. The attraction step length  $\beta$  has also been modified. A chaotic mapping is used to modify  $\beta$  in some of the studies [33–38]. It should be noted that using a chaotic map or updating  $\gamma$  does update  $\beta$ . For instance, **Figure 2** shows that when  $\gamma$  is updated using a logistic map, the resulting  $\beta$  is also chaotic.

Hence,  $\gamma$  or  $\beta$  should not be updated at the same time. In addition to  $\gamma$  updating,  $\beta$  has also been done based on minimum and final values [39, 40], depending on the location of the solution [31] and the light intensity of solutions [32]. In addition, different approaches are used to measure the distance between the fireflies [31, 41, 45]. Modifying the feasible region should be considered as a very big step length that may take the solution far away from the brighter solution and possibly out of the feasible region. The property of the random step should also

be properly tuned in agreement with the attraction; otherwise, the random movement may dominate the attractiveness step length.



**Figure 2.** The effect of chaotic map update of  $\gamma$  on  $\beta$  [34].

The movement of the best solution should be tuned properly. If it is allowed to decrease then its best performance may get lost. Hence, the approaches used to preserve the best solution are ended effectively [32, 52].

Mutation is another good approach to diversify the solutions which in turn increase the exploration behaviour of the algorithm [24, 53–58]. However, generating many solutions may hinder the search as it will take long to run. In addition, accepting weak solution should also be incorporated in deceiving problems; a solution needs to decrease in order to escape local solutions.

Modifying the update equation is another interesting modification featured in some studies [56, 58–61, 63–67, 69]. These studies suggest that the update should be done in the vicinity of the brighter firefly [58, 60]. This is not always a good idea as the region in between the two solutions will not be explored. Some of the studies indicate an increase in the attraction towards brighter fireflies [61, 64]. It simply means that increasing the step length of the attraction may dominate the random movement or even take the solution out of the feasible region. A memory is utilized to save the best solution found and additional attraction term towards that global solution is added in [63, 69]. It is a good idea in which the best solution will not be lost through iteration. To increase the diversity of the solution, an effective modification is proposed in [110]. Using such kind of modification, the diversity of the solutions will be preserved, and the exploration behaviour of the algorithm will be improved.

Basically, two updating strategies are proposed for the non-continuous case. The first one is using the same updating formula and changing the results to discrete values afterwards [82, 94, 104]. And the second is to modify the updating formula on the discrete space [97–101]. The first problem is susceptible of trapping in local solution and misses the optimal solution. The optimal solution for a continuous version of a discrete problem may not always be an optimal solution for the discrete problem. Hence, the algorithm will tend to converge to the optimal solution of the continuous version of the problem. Hence, the second approach has an advantage in such cases.

## 4. Simulation results

The comparison of results is performed between the standard firefly algorithm and non-parameter modified version, i.e. Class 2 and Class 3 modifications. The modified versions selected for simulation are based on two criteria, the first one clear modification, that is the modification should be clearly described, and the second one is with small number of new parameters. In some of the modifications, a number of new algorithm parameters are introduced and tuning this parameter by itself needs another study so they are not included in the simulation. The modified versions used for simulation include Firefly Algorithm 1 [32], FFA2, [52], FFA3 [53], FFA4 [26, 57], FFA5 [24, 59], FFA6 [58] FFA7 [60], FFA8 [61, 62], FFA9 [69], FFA10 [63] where  $x_i$ -gbest is replaced by gbest- $x_i$ , FFA11 [110], FFA12 [72–74], FFA13 [75–79], FFA14 from [70].

### 4.1. Benchmark problems and simulation setup

Five benchmark problems are selected from different categories as presented in **Table 2**. The simulations are performed on Intel® Core™i3-3110M CPU @ 2.40 Ghz 64 bit operating system. MATLAB 7.10.0 (R2010a) is used for these simulations. The algorithm parameters are set as given in **Table 2** for dimensions 2 and 5.

### 4.2. Simulation results and discussion

The simulation results, as presented in **Table 3**, show that some of the algorithms are very expensive in terms of computational time but give a good result, and others have small running time. For instance, in second problem, when the dimension is 2 on average, FFA3 outperforms all with average CPU time of 8.8, whereas FFA1 and FFA2 give a good approximation with smaller average CPU time. In general, it can be seen that FFA4 is very effective but not with the computational time. FFA1 and FFA2 give good approximate results with smaller CPU time compared to FFA4. However, when the dimension increases, FFA2 outperforms FFA1. Perhaps it is due to the fixed random direction  $m$  for all the simulations.

Problems	Ref.	Properties of the problem	Parameters and set-up	
			D = 2	D = 5
$f_1$ $e^{-\sum_{i=1}^D x_i^2} - 2e^{-\sum_{i=1}^D (\frac{x_i}{15})^6} \prod_{i=1}^D \cos^2 x_i$ $-20 \leq x_i \leq 20$	[111]	Multimodal Continuous Differentiable Non-separable	N = 50 $\alpha = 4$ $\gamma = 2$	N = 200 $\alpha = 5$ $\gamma = 2$
$f_2$ $\sum_{i=1}^D  x_i \sin x_i + 0.1 x_i $ $-10 \leq x_i \leq 10$	[112]	Multimodal Continuous Non-differentiable Separable Non-scalable	N = 25 $\alpha = 3$ $\gamma = 2$	N = 100 $\alpha = 4$ $\gamma = 2$
$f_3$ $\sum_{j=1}^D [\sum_{i=1}^5 p(i) x_j^{5-i}]$ for $p = [0.03779 - 0.84056 - 14.427.134]$ $-1 \leq x_i \leq 12$	[113]	Multimodal Discontinuous Non-differentiable Separable	N = 20 $\alpha = 1.5$ $\gamma = 2$	N = 100 $\alpha = 3$ $\gamma = 2$
$f_4$ $-200e^{0.02\sqrt{\sum_{i=1}^D x_i^2}}$ $-32 \leq x_i \leq 32$	[112]	Unimodal Continuous Differentiable Non-separable Non-Scalable	N = 60 $\alpha = 6$ $\gamma = 2$	N = 250 $\alpha = 7$ $\gamma = 2$
$f_5$ $\sum_{i=1}^D \varepsilon_i  x_i ^i$ $-5 \leq x_i \leq 5$	[112]	Unimodal Continuous Non-differentiable Separable Scalable Stochastic	N = 20 $\alpha = 1.5$ $\gamma = 2$	N = 70 $\alpha = 2.5$ $\gamma = 2$

Table 2. Selected benchmark problems and simulation set-up.

	F1				F2				F3				F4				F5				
	D	2	5		2	5		2	5		2	5		2	5		2	5		5	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	
FFA	$f(x^*)$	-0.0195	0.1002	0.00	0.00	0.6696	0.4483	0.0504	0.0135	-6.6745	1.2874	-6.5854	3.6570	-193.75	3.4097	-166.46	7.3072	0.0158	0.0148	0.0477	0.0373
	CPU	1.4	0.3	1.4	0.0	0.3	0.1	0.1	0.0	0.2	0.1	2.7	0.1	1.8	0.4	8.4	2.5	0.2	0.0	0.3	0.1
FFA1	$f(x^*)$	-0.7185	0.4220	0.00	0.00	0.0039	0.0085	0.0109	0.0052	-7.6507	0.00	-17.135	1.8541	-199.61	0.1757	-195.44	1.1286	0.0005	0.0006	0.0002	0.0003
	CPU	1.5	0.3	1.5	0.1	0.4	0.1	0.1	0.0	0.3	0.1	2.8	0.2	1.8	0.4	8.8	2.4	0.3	0.0	0.3	0.0
FFA2	$f(x^*)$	-0.7974	0.4028	0.00	0.00	0.0003	0.0003	0.0000	0.0000	-7.6507	0.00	-19.126	0.0	-199.98	0.0136	-199.98	0.0076	0.0047	0.0092	0.0048	0.0102
	CPU	2.6	0.4	3.5	0.3	1.1	0.1	0.1	0.0	0.4	0.1	4.5	0.4	2.2	0.5	11.3	2.9	0.4	0.1	0.9	0.2
FFA3	$f(x^*)$	-0.0128	0.0903	0.00	0.00	0.4402	0.3768	0.0397	0.0158	-7.2621	0.7406	-8.6751	2.9248	-195.20	2.6004	-177.04	6.5591	0.0085	0.0085	0.0135	0.0125

	F1				F2				F3				F4				F5				
	D		5		2		5		2		5		2		5		2		5		
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	
FFA4	CPU	1.3	0.2	1.4	0.1	0.3	0.1	0.1	0.0	0.2	0.1	2.6	0.3	1.6	0.3	8.1	2.2	0.2	0.0	0.2	0.1
	$f(x^*)$	-1.00	0	-1.00	0.00	0.0	0.0	0.00	0	-7.6507	0.00	-17.977	1.9499	-200	0.0	-200	0.0	0.0	0.0	0.00	0.00
	CPU	44.2	6.7	60.4	3.8	8.8	0.6	1.6	0.2	4.2	0.3	126.5	47	52.56	10.3	574.8	96.4	4	0.4	5	0.8
FFA5	$f(x^*)$	-0.0372	0.1711	0.00	0.00	0.0078	0.0169	0.0113	0.0038	-7.6507	0.00	-17.610	1.2629	-199.58	0.2164	-194.51	1.3993	0.0013	0.0013	0.0009	0.0007
	CPU	37.6	5.8	51.0	2.4	7.8	0.8	1.4	0.1	3.6	0.3	102.5	43.1	45.6	9.3	493.6	80.1	3.4	0.3	4.2	0.8
	$f(x^*)$	0.0000	0.00	0.00	0.00	0.7534	0.5307	0.0277	0.0091	-7.6507	0.00	-13.323	0.8509	-198.43	0.7722	-193.95	1.4193	0.0051	0.0045	0.0014	0.0008
FFA6	CPU	1.4	0.2	1.5	0.1	0.3	0.1	0.1	0.0	0.3	0.1	2.7	0.3	1.6	0.3	8.6	2.5	0.2	0.1	0.3	0.1
	$f(x^*)$	0.0000	0.00	0.00	0.00	0.0664	0.0330	0.0007	0.0002	-7.6507	0.00	-12.286	1.2151	-199.98	0.0136	-196.05	1.8438	0.0001	0.0002	0.0000	0.0001
	CPU	0.7	0.1	0.8	0.1	0.2	0.1	0.1	0.0	0.2	0.1	1.7	0.2	0.9	0.2	4.0	1.3	0.1	0.0	0.2	0.0
FFA8	$f(x^*)$	0.0000	0.00	0.00	0.00	0.7113	0.5634	0.0499	0.0149	-7.2207	0.8647	-6.2323	3.2674	-193.81	3.7645	-167.58	6.8116	0.0084	0.0085	0.0425	0.0372
	CPU	3.6	0.6	6.8	0.1	1.3	0.1	0.1	0.0	0.9	0.1	7.6	0.7	4.9	0.8	29.4	8.5	0.8	0.1	1	0.1
	$f(x^*)$	-0.0303	0.15	0.00	0.00	1.1369	0.6702	0.1115	0.0251	-5.4108	1.7004	-1.0048	13.039	-188.39	5.3655	-104.37	12.818	0.0486	0.0399	0.0951	0.0915
FFA10	CPU	0.3	0.1	0.2	0.0	0.2	0.1	0.0	0.0	0.2	0.1	0.5	0.1	0.4	0.1	0.7	0.1	0.1	0.0	0.1	0.0
	$f(x^*)$	0.0000	0.00	0.00	0.00	1.0466	0.6202	0.0403	0.0203	-6.7824	1.0078	-13.245	1.6688	-175.52	12.883	-172.68	10.459	0.0064	0.0055	0.0014	0.0012
	CPU	3.5	0.6	11.1	0.1	1.2	0.1	0.1	0.0	0.9	0.1	9.3	0.9	4.7	0.9	64.2	18.1	1.0	0.1	1.4	0.2
FFA11	$f(x^*)$	0.0000	0.00	0.00	0.00	0.6357	0.4193	0.0514	0.0129	-6.7098	1.0934	-6.1737	3.3915	-194.49	3.3451	-165.94	7.3183	0.0208	0.0186	0.0471	0.0324
	CPU	1.3	0.3	1.4	0.1	0.3	0.1	0.1	0.0	0.3	0.1	2.7	0.3	1.7	0.3	8.3	2.3	0.2	0.1	0.3	0.1
	$f(x^*)$	0.0000	0.00	0.00	0.00	0.6611	0.5103	0.0493	0.0141	-6.5986	1.3430	-6.8659	4.0703	-193.96	3.4841	-169.34	8.7051	0.0161	0.0160	0.0595	0.0452
FFA12	CPU	1.5	0.2	1.5	0.1	0.4	0.1	0.1	0.0	0.3	0.1	2.9	0.3	1.8	0.4	8.4	2.4	0.3	0.0	0.3	0.1
	$f(x^*)$	0.0000	0.00	0.00	0.00	2.1137	1.4840	0.0977	0.0315	-3.3360	3.2207	-1.1917	12.914	-189.53	4.8277	-148.59	12.7616	0.1356	0.1374	1.4724	2.4610
	CPU	5.4	0.9	16.1	0.2	2.4	0.2	0.3	0.1	1.7	0.1	28	2.5	7.7	1.4	122	30.2	1.7	0.2	1.8	0.2
FFA14	$f(x^*)$	-0.6313	0.29	0.2399	0.4	0.0960	0.0829	0.0107	0.0037	-2.3289	7.0785	-0.4888	11.437	-198.32	0.8034	-193.92	1.4485	0.0046	0.0047	0.0010	0.0006
	CPU	1.5	0.3	1.5	0.1	0.4	0.1	0.1	0.0	0.3	0.1	2.7	0.3	1.8	0.4	8.5	2.5	0.2	0.0	0.3	0.1

**Table 3.** Simulation results.

## 5. Conclusion

In this chapter, a detailed review of modified versions of firefly algorithm is presented. The modifications are used to boost its performance for both continuous and non-continuous problems. Three classes of modifications are discussed for continuous problems. The first one being parameter level modification which will improve the performance of the algorithm. The second class is on the updating mechanism level, in which new updating equation or mechanisms are introduced. The last class is in the abstract level in which change of solution space

and probability distribution of the randomness term are discussed. The strength and weakness of the approaches are also presented. Simulation results show that mutation-incorporated firefly algorithm gives better result with larger computational time, whereas versions of firefly algorithm with opposition-based learning and elitist movement for the brighter firefly give approximate solution with smaller computational time. Hence, if a proper way of implementation is used, mutation operator and elitist move of brighter firefly algorithm along with possible implementation of opposition-based approach may perform better.

## Author details

Waqar A. Khan<sup>1\*</sup>, Nawaf N. Hamadneh<sup>2</sup>, Surafel L. Tilahun<sup>3</sup> and Jean M. T. Ngnotchouye<sup>3</sup>

\*Address all correspondence to: wkhan\_2000@yahoo.com

1 Department of Mechanical and Industrial Engineering, College of Engineering, Majmaah University, Majmaah, Saudi Arabia

2 College of Science and Theoretical Studies, Saudi Electronic University, Riyadh, Saudi Arabia

3 School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg Campus, South Africa

## References

- [1] Yang X-S. Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation*. 2011;3(2):77–84.
- [2] Tilahun SL, ONG HC. Prey-predator algorithm: a new metaheuristic algorithm for optimization problems. School of Mathematical Sciences, [thesis]. Penang Universiti Sains Malaysia, Malaysia 2013.
- [3] Negnevitsky M. *Artificial intelligence: a guide to intelligent systems*. New York: Pearson Education Limited, England, 2005.
- [4] Kennedy J, Eberhart R. Particle swarm optimization. *International Conference on Neural Networks IV*; IEEE; Perth, Australia; 1995. 1942–8.
- [5] Yang X-S. *Nature-inspired metaheuristic algorithms*. Luniver Press; UK, 2010.
- [6] Tilahun, SL and Ong, HC, *International Journal of Information Technology & Decision Making*, 14 (6), 1331 – 1352, 2015.

- [7] Joshi R. Optimization techniques for transportation problems of three variables. *IOSR Journal of Mathematics*. 2013;9:46–50.
- [8] Tilahun SL, Ong HC. Bus timetabling as a fuzzy multiobjective optimization problem using preference-based genetic algorithm. *Promet-Traffic & Transportation*. 2012;24(3): 183–91.
- [9] Ropponen A, Ritala R, Pistikopoulos EN. Broke management optimization in design of paper production systems. *Computer Aided Chemical Engineering*. 2010;28:865–70.
- [10] Pike J, Bogich T, Elwood S, Finnoff DC, Daszak P. Economic optimization of a global strategy to address the pandemic threat. *Proceedings of the National Academy of Sciences*. 2014;111(52):18519–23.
- [11] Hamadneh N, Tilahun SL, Sathasivam S, Choon OH. Prey-predator algorithm as a new optimization technique using in radial basis function neural networks. *Research Journal of Applied Sciences*. 2013;8(7):383–7.
- [12] Ong HC, Tilahun S. Integration fuzzy preference in genetic algorithm to solve multi-objective optimization problems. *The Far East Journal of Mathematical Sciences*. 2011;55:165–179.
- [13] Tilahun, SL, Kassa, SM and Ong, HC, A New Algorithm for Multilevel Optimization Problems Using Evolutionary Strategy, Inspired by Natural Adaptation, (Ed.: Anthony, P, Ishizuka, M and Lukose, D), *PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence*, Kuching, Malaysia, September 3-7, 2012. Verlag berlin Heidelberg: Springer; 2012, 577–88.
- [14] Tilahun SL, Asfaw A. Modeling the expansion of *Prosopis juliflora* and determining its optimum utilization rate to control the invasion in Afar Regional State of Ethiopia. *International Journal of Applied Mathematical Research*. 2012;1(4):726–43.
- [15] Shimoyama K, Seo K, Nishiwaki T, Jeong S, Obayashi S. Design optimization of a sport shoe sole structure by evolutionary computation and finite element method analysis. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*. 2011;225(4):179–88.
- [16] Hamadneh N, Khan WA, Sathasivam S, Ong HC, Design optimization of pin fin geometry using particle swarm optimization algorithm. *PLOS ONE*, 8(5): e66080, 2013 DOI 10.1371/journal.pone.0066080.
- [17] Hamadneh N, Sathasivam S, Tilahun SL, Choon OH. Learning logic programming in radial basis function network via genetic algorithm. *Journal of Applied Sciences (Faisalabad)*. 2012;12(9):840–7.
- [18] Solo AMG. The new interdisciplinary fields of political engineering and computational politics. *Political Campaigning in the Information Age*. IGI Global, USA, 226–227, 2014.
- [19] Tikhomirov VM. Stories about maxima and minima. *The American Mathematical Society;Universities Press (India) Pvt. Limited*, 1990. Trans. Shenitzer, A, American

Mathematical Society, Mathematical World Volume: 1; 1991; 187 pp; Softcover MSC: Primary 00; 01; 46; 49; Print ISBN: 978-0-8218-0165-9 Product Code: MAWRLD/1 - See more at: <http://bookstore.ams.org/mawrld-1/#sthash.9th31NiP.dpuf> USA, 1990.

- [20] Yamamoto T. Historical developments in convergence analysis for Newton's and Newton-like methods. *Journal of Computational and Applied Mathematics*. 2000;124(1):1–23.
- [21] Babaoglu O, Binci T, Jelasity M, Montresor A, editors. Firefly-inspired heartbeat synchronization in overlay networks. Cambridge, MA; 2007 SASO'07 1st International Conference on Self-Adaptive and Self-Organizing Systems; Cambridge, MA; 2007 SASO'07 1st International Conference on Self-Adaptive and Self-Organizing Systems, IEEE. DOI: 10.1109/SASO.2007.25, Cambridge, Washington, DC, USA 2007.
- [22] Miao Y. Resource scheduling simulation design of firefly algorithm based on chaos optimization in cloud computing. *International Journal of Grid and Distributed Computing*. 2014;7(6):221–8.
- [23] Shafaati M, Mojallali H. Modified firefly optimization for IIR system identification. *Journal of Control Engineering and Applied Informatics*. 2012;14(4):59–69.
- [24] Shakarami MR, Sedaghati R. A new approach for network reconfiguration problem in order to deviation bus voltage minimization with regard to probabilistic load model and DGs. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*. 2014;8(2):430–5.
- [25] Olamaei J, Moradi M, Kaboodi T, editors. A new adaptive modified firefly algorithm to solve optimal capacitor placement problem. 2013 18th Conference on Electrical Power Distribution Networks (EPDC); IEEE; 2013.
- [26] Kavousi-Fard A, Samet H, Marzbani F. A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting. *Expert Systems with Applications*. 2014;41(13):6047–56.
- [27] Yu S, Yang S, Su S. Self-adaptive step firefly algorithm. *Journal of Applied Mathematics*, 2013(2013): 8.
- [28] Yang X-S. Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*. 2013;29(2):175–84.
- [29] Yu S, Yang S, Su S. Self-adaptive step firefly algorithm. *Journal of Applied Mathematics*, 2013(2013): 8. <http://dx.doi.org/10.1155/2013/832718>.
- [30] Yu S, Su S, Lu Q, Huang L. A novel wise step strategy for firefly algorithm. *International Journal of Computer Mathematics*. 2014;91(12):2507–13.
- [31] Lin X, Zhong Y, Zhang H. An enhanced firefly algorithm for function optimisation problems. *International Journal of Modelling, Identification and Control*. 2013;18(2): 166–73.

- [32] Tilahun SL, Ong HC. Modified firefly algorithm. *Journal of Applied Mathematics*. 2012(2012): 1–12. DOI:10.1155/2012/467631.
- [33] Gandomi A, Yang X-S, Talatahari S, Alavi A. Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*. 2013;18(1):89–98.
- [34] Jansi S, Subashini P. A novel fuzzy clustering based modified firefly algorithm with chaotic map for MRI brain tissue segmentation. *MAGNT Research Report*. 2015;3(1): 52–8.
- [35] Long NC, Meesad P, Unger H. A highly accurate firefly based algorithm for heart disease prediction. *Expert Systems with Applications*. 2015;42(21):8221–31.
- [36] Coelho LDS, de Andrade Bernert DL, Mariani VC, editors. A chaotic firefly algorithm applied to reliability-redundancy optimization. 2011 IEEE Congress on Evolutionary Computation (CEC), IEEE; 2011.
- [37] Abdel-Raouf O, Abdel-Baset M, El-henawy I. Chaotic firefly algorithm for solving definite integral. *International Journal of Information Technology and Computer Science (IJITCS)*. 2014;6(6):19.
- [38] Khalil A-W. Improved firefly algorithm for unconstrained optimization problems. *International Journal of Computer Applications Technology and Research*. 2014;4(1): 77–81.
- [39] Selvarasu R, Surya Kalavathi M, Rajan A, Christober C. SVC placement for voltage constrained loss minimization using self-adaptive Firefly algorithm. *Archives of Electrical Engineering*. 2013;62(4):649–61.
- [40] Meena S, Chitra K. Modified approach of firefly algorithm for non-minimum phase systems. *Indian Journal of Science and Technology*. 2015 ;8(23):1–8. DOI: 10.17485/ijst/2015/v8i23/72264
- [41] Yan X, Zhu Y, Wu J, Chen H. An improved firefly algorithm with adaptive strategies. *Advanced Science Letters*. 2012;16(1):249–54.
- [42] Sulaiman MH, Daniyal H, Mustafa MW, editors. Modified firefly algorithm in solving economic dispatch problems with practical constraints. 2012 IEEE International Conference on Power and Energy (PECon); IEEE; 2012.
- [43] Wang B, Li D-X, Jiang J-P, Liao Y-H. A modified firefly algorithm based on light intensity difference. *Journal of Combinatorial Optimization*. 2016; 31(3), 1045–1060.
- [44] Othman MM, Hegazy YG, Abdelaziz AY. A modified firefly algorithm for optimal sizing and siting of voltage controlled distributed generators in distribution networks. *Periodica Polytechnica Electrical Engineering and Computer Science*. 2015;59(3):104–9.

- [45] Subramanian R, Thanushkodi K. An efficient firefly algorithm to solve economic dispatch problems. *International Journal of Soft Computing and Engineering*. 2013;2(1):52–5.
- [46] Liu C, Zhao Y, Gao F, Liu L. Three-dimensional path planning method for autonomous underwater vehicle based on modified firefly algorithm. *Mathematical Problems in Engineering*. 2015;2015:1–10. <http://dx.doi.org/10.1155/2015/561394>
- [47] Fister I, Yang X-S, Brest J, Fister Jr I. In: Yang X-S, Cui Z, Xiao R, Gandomi A-H, Karamanoglu M, editors. *Memetic self-adaptive firefly algorithm* (Ed.: Yang, X-S, Cui, Z, Xiao, R Gandomi, AH and Karamanoglu, M) in *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*. 2nd ed. london. Elsevier; :73–102, ISBN: 978-0-12-405163-8, 2013.
- [48] Fu Q, Liu Z, Tong N, Wang M, Zhao Y, editors. A novel firefly algorithm based on improved learning mechanism. In *International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2015)*; Atlantis Press; 2015.
- [49] dos Santos Coelho L, Mariani VC. Improved firefly algorithm approach applied to chiller loading for energy conservation. *Energy and Buildings*. 2013;59:273–8.
- [50] dos Santos Coelho L, Mariani VC. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Computers & Mathematics with Applications*. 2012;64(8):2371–82.
- [51] Roy AG, Rakshit P, Konar A, Bhattacharya S, Kim E, Nagar AK, editors. Adaptive firefly algorithm for nonholonomic motion planning of car-like system. *2013 IEEE Congress on Evolutionary Computation (CEC)*, IEEE; 2013.
- [52] Verma OP, Aggarwal D, Patodi T. Opposition and dimensional based modified firefly algorithm. *Expert Systems with Applications*. 2016;44:168–76.
- [53] Yu S, Zhu S, Ma Y, Mao D. Enhancing firefly algorithm using generalized opposition-based learning. *Computing*. 2015 :97(7) 741–754
- [54] Kazemzadeh-Parsi M. A modified firefly algorithm for engineering design optimization problems. *Iranian Journal of Science and Technology*. 2014;38(M2):403–21.
- [55] Kazemzadeh-Parsi MJ. Optimal shape design for heat conduction using smoothed fixed grid finite element method and modified firefly algorithm. *Iranian Journal of Science and Technology Transactions of Mechanical Engineering*. 2015;39(M2):367.
- [56] Kazemzadeh-Parsi MJ, Daneshmand F, Ahmadvard MA, Adamowski J. Optimal remediation design of unconfined contaminated aquifers based on the finite element method and modified firefly algorithm. *Water Resources Management*. 2015;29(8): 2895–912.
- [57] Mohammadi S, Mozafari B, Solimani S, Niknam T. An adaptive modified firefly optimisation algorithm based on Hong's Point Estimate Method to optimal operation

- management in a micro grid with consideration of uncertainties. *Energy*. 2013;51:339–48.
- [58] Kazemzadeh AS, Kazemzadeh AS. Optimum design of structures using an improved firefly algorithm. *International Journal of Optimization and Civil Engineering*, 1(2), 327–340, 2011; 2:327–340.
- [59] Amiri B, Hossain L, Crawford JW, Wigand RT. Community detection in complex networks: multi-objective enhanced firefly algorithm. *Knowledge-Based Systems*. 2013;46:1–11.
- [60] Amaya I, Cruz J, Correa R. A modified firefly-inspired algorithm for global computational optimization. *Dyna*. 2014;81(187):85–90.
- [61] Hassanzadeh T, Meybodi MR, editors. A new hybrid approach for data clustering using firefly algorithm and K-means. 2012 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISIP), IEEE; 2012.
- [62] Hongwei Z, Liwei T, Dongzheng W. Research on improved firefly optimization algorithm based on cooperative for clustering. *International Journal of Smart Home*, 9(3), 2015; 9(3):205–214.
- [63] Goel S, Panchal V, editors. Performance evaluation of a new modified firefly algorithm. 2014 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions); IEEE; 2014.
- [64] Hassanzadeh T, Kanan HR. Fuzzy FA: a modified firefly algorithm. *Applied Artificial Intelligence*. 2014;28(1):47–65.
- [65] Cheung NJ, Ding X-M, Shen H-B. Adaptive firefly algorithm: Parameter analysis and its application, *PloS one*. 2014;9(11): 1–12, e112634
- [66] Arora S, Singh S, Singh S, Sharma B, editors. Mutated firefly algorithm. 2014 International Conference on Parallel, Distributed and Grid Computing (PDGC); IEEE; 2014.
- [67] Arora S, Singh S. Performance research on firefly optimization algorithm with mutation. *International Conference on Communication, Computing & Systems (ICCCS–2014)*; 2014. 168–72.
- [68] Banati H, Bajaj M. Fire fly based feature selection approach. *International Journal of Computer Science Issues*, 8(4): 473–480, 2011.
- [69] Maidl G, de Lucena DS, dos Santos Coelho L. Economic dispatch optimization of thermal units based on a modified firefly algorithm, In proceeding of the 2nd International Congress of Mechanical Engineering (COBEM 2013), November 3-7, 2013, Ribeirão Preto, SP, Brazil, 7118–7123, ABCM. 2013.
- [70] Tian Y, Gao W, Yan S, editors. An improved inertia weight firefly optimization algorithm and application. 2012 International Conference on Control Engineering and Communication Technology (ICCECT); IEEE; 2012.

- [71] Fister I, Yang X-S, Brest J. Modified firefly algorithm using quaternion representation. *Expert Systems with Applications*. 2013;40(18):7220–30.
- [72] Farahani SM, Abshouri A, Nasiri B, Meybodi M. A Gaussian firefly algorithm. *International Journal of Machine Learning and Computing*. 2011;1(5):448–53.
- [73] Farahani S, Abshouri A, Nasiri B, Meybodi M, editors. An improved firefly algorithm with directed movement. *Proceedings of 4th IEEE International Conference on Computer Science and Information Technology*; Chengdu; 2011, 248–251.
- [74] Kanimozhi T, Latha K. An adaptive approach for content based image retrieval using Gaussian firefly algorithm. *Emerging intelligent computing technology and applications Volume 375 of the series Communications in Computer and Information Science*, 213–218. Springer Berlin Heidelberg; 2013.
- [75] Yang, X-S, Firefly algorithm, Levy flights and global optimization, in: *Research and Development in Intelligent Systems XXVI* (Eds Bramer, M, Ellis, R and Petridis, M), Verlag berlin Heidelberg: Springer London, 209–218 (2010).
- [76] Wang G, Guo L, Duan H, Liu L, Wang H. A modified firefly algorithm for UCAV path planning. *International Journal of Hybrid Information Technology*. 2012;5(3):123–44.
- [77] Dhal KG, Quraishi I, Das S. A chaotic Lévy flight approach in bat and firefly algorithm for gray level image enhancement. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*. 2015;7(7):69.
- [78] Wang G-G, Guo L, Duan H, Wang H. A new improved firefly algorithm for global numerical optimization. *Journal of Computational and Theoretical Nanoscience*. 2014;11(2):477–85.
- [79] Fateen S-EK, Bonilla-Petriciolet A. Intelligent firefly algorithm for global optimization, In *Cuckoo Search and Firefly Algorithm*, Volume 516 of the series *Studies in Computational Intelligence*, 315–330, Verlag berlin Heidelberg: Springer International Publishing Switzerland; 2014.
- [80] Hassanzadeh T, Vojodi H, Moghadam AME, editors. A multilevel thresholding approach based on Levy-flight firefly algorithm. *2011 7th Iranian Machine Vision and Image Processing (MVIP), IEEE*; 2011.
- [81] Sahoo A, Chandra S, editors. Levy-flight firefly algorithm based active contour model for medical image segmentation. *2013 6th International Conference on Contemporary Computing (IC3); IEEE*; 2013.
- [82] Crawford B, Soto R, Olivares-Suarez M, Palma W, Paredes F, Olguín E, et al. A binary coded firefly algorithm that solves the set covering problem. *Science and Technology*. 2014;17(3):252–64.
- [83] Chandrasekaran K, Simon SP, Padhy NP. Binary real coded firefly algorithm for solving unit commitment problem. *Information Sciences*. 2013;249:67–84.

- [84] Yang Y, Mao Y, Yang P, Jiang Y, editors. The unit commitment problem based on an improved firefly and particle swarm optimization hybrid algorithm. Chinese Automation Congress (CAC), 2013, IEEE; 2013.
- [85] Chhikara RR, Singh L, editors. An improved discrete firefly and t-test based algorithm for blind image steganalysis. 2015 6th International Conference on Intelligent Systems, Modelling and Simulation (ISMS); IEEE; 2015.
- [86] Farhoodnea M, Mohamed A, Shareef H, Zayandehroodi H. Optimum placement of active power conditioners by a dynamic discrete firefly algorithm to mitigate the negative power quality effects of renewable energy-based generators. *International Journal of Electrical Power & Energy Systems*. 2014;61:305–17.
- [87] Farhoodnea M, Mohamed A, Shareef H, Zayandehroodi H. Optimum placement of active power conditioner in distribution systems using improved discrete firefly algorithm for power quality enhancement. *Applied Soft Computing*. 2014;23:249–58.
- [88] Farhoodnea M, Mohamed A. Optimal placement and sizing of active voltage conditioner in a smart grid using discrete firefly algorithm. *Third International Conference on Advances in Engineering Sciences & Applied Mathematics (ICAESAM'2015)*; London, UK; 2015. 20–5.
- [89] Sayadi M, Ramezani R, Ghaffari-Nasab N. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*. 2010;1(1):1–10.
- [90] Baykasoğlu A, Ozsoydan FB. An improved firefly algorithm for solving dynamic multidimensional knapsack problems. *Expert Systems with Applications*. 2014;41(8): 3712–25.
- [91] Baykasoğlu A, Ozsoydan FB. Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied Soft Computing*. 2015;36:152–64.
- [92] Costa MFP, Rocha AMA, Francisco RB, Fernandes EM. Heuristic-based firefly algorithm for bound constrained nonlinear binary optimization. *Advances in Operations Research*. 2014 (2014): 1–12. <http://dx.doi.org/10.1155/2014/215182>
- [93] Mamaghani AS, Hajizadeh M, editors. Software modularization using the modified firefly algorithm. 2014 8th Malaysian Software Engineering Conference (MySEC); Langkawi: IEEE; 2014, p. 321 – 324. DOI: 10.1109/MySec.2014.6986037
- [94] Oliveira IMSD, Schirru R. A modified firefly algorithm applied to the nuclear reload problem of a pressurized water reactor. *International Nuclear Atlantic Conference - INAC 2011*; Brazil; 2011.
- [95] Bean JC. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*. 1994;6(2):154–60.

- [96] Durkota K. Implementation of a discrete firefly algorithm for the QAP problem within the sage framework. BSc Thesis, Czech Technical University. 2011; 393–403.
- [97] Poursalehi N, Zolfaghari A, Minuchehr A. Multi-objective loading pattern enhancement of PWR based on the discrete firefly algorithm. *Annals of Nuclear Energy*. 2013;57:151–63.
- [98] Ishikawa M, Matsushita H. Discrete firefly algorithm using familiarity degree. 2013 Shikoku-Section Joint Convention Record of the Institutes of Electrical and Related Engineers TOKUSHIMA, IEICE Tech. Rep., vol. 113, no. 271, NLP2013-89, 105–108, Oct. 2013.
- [99] Poursalehi N, Zolfaghari A, Minuchehr A. A novel optimization method, effective discrete firefly algorithm, for fuel reload design of nuclear reactors. *Annals of Nuclear Energy*. 2015;81:263–75.
- [100] Jati GK, Suyanto, editors. Evolutionary discrete firefly algorithm for travelling salesman problem. Second International Conference, ICAIS 2011; Klagenfurt, Austria; 2011; P. 393–403.
- [101] Li M, Zhang Y, Zeng B, Zhou H, Liu J. The modified firefly algorithm considering fireflies' visual range and its application in assembly sequences planning. *The International Journal of Advanced Manufacturing Technology*. 2016;82(5–8):1381–403.
- [102] Kota L. Optimization of the supplier selection problem using discrete firefly algorithm. *Advanced Logistic systems*. 2012;6(1):117–26.
- [103] Gandomi AH, Yang X-S, Alavi AH. Mixed variable structural optimization using firefly algorithm. *Computers & Structures*. 2011;89(23):2325–36.
- [104] Bacanin N, Brajevic I, Tuba M. Firefly algorithm applied to integer programming problems. *Proceeding of Recent Advances in Mathematics*, 2013; 143–148.
- [105] Baghlani A, Makiabadi M, Sarcheshmehpour M. Discrete optimum design of truss structures by an improved firefly algorithm. *Advances in Structural Engineering*. 2014;17(10):1517–30.
- [106] Bacanin N, Tuba M. Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint. *The Scientific World Journal*. 2014(2014): 1–16. <http://dx.doi.org/10.1155/2014/721521>.
- [107] Tuba M, Bacanin N, editors. Upgraded firefly algorithm for portfolio optimization problem. 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (UKSim); Cambridge, IEEE; 2014, 113 – 118.
- [108] Deb K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*. 2000;186(2):311–38.
- [109] Selvarasu R, Kalavathi MS. Tcsc placement for loss minimisation using self adaptive firefly algorithm. *Journal of Engineering Science and Technology*. 2015;10(3):291–306.

- [110] Yu S, Su S, Huang L. A simple diversity guided firefly algorithm. *Kybernetes*. 2015;44(1):43–56.
- [111] Gavana A. Global optimization benchmarks and AMPGO [Internet]. Available from: [http://infinity77.net/global\\_optimization/test\\_functions\\_nd\\_X.html#go\\_benchmark.XinSheYang03](http://infinity77.net/global_optimization/test_functions_nd_X.html#go_benchmark.XinSheYang03) [Accessed: 05/01/2016]
- [112] Jamil M, Yang X-S. A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling and Numerical Optimisation*. 4(2), 150–194, 2013.
- [113] Tilahun SL, Ong HC, Ngnotchouye JM. Extended prey predator algorithm with a group hunting scenario. *Advances in Operations Research*. Article Id 7325263, (in press) 2016.





*Edited by Ozgur Baskan*

This book covers state-of-the-art optimization methods and their applications in wide range especially for researchers and practitioners who wish to improve their knowledge in this field. It consists of 13 chapters divided into two parts: (I) Engineering applications, which presents some new applications of different methods, and (II) Applications in various areas, where recent contributions of state-of-the-art optimization methods to diverse fields are presented.

Photo by gguy44 / iStock

**IntechOpen**

ISBN 978-953-51-5077-0



9 789535 150770