Management and Industrial Engineering

Shubhabrata Datta
J. Paulo Davim *Editors*

Optimization in Industry

Present Practices and Future Scopes

Springer

Management and Industrial Engineering

Series editor

J. Paulo Davim, Aveiro, Portugal

More information about this series at http://www.springer.com/series/11690

Shubhabrata Datta · J. Paulo Davim
Editors

Optimization in Industry

Present Practices and Future Scopes

Springer

Editors
Shubhabrata Datta
Department of Mechanical Engineering
SRM Institute of Science and Technology
Chennai, India

J. Paulo Davim
Department of Mechanical Engineering
University of Aveiro
Aveiro, Portugal

ISSN 2365-0532 ISSN 2365-0540 (electronic)
Management and Industrial Engineering
ISBN 978-3-030-01640-1 ISBN 978-3-030-01641-8 (eBook)
https://doi.org/10.1007/978-3-030-01641-8

Library of Congress Control Number: 2018957251

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Optimization is a standard activity in everyday life. We use it consciously or unconsciously for almost all our daily jobs. When we try to find a way to finish an assignment in our professional life in less effort or time, it is an optimization procedure. Deciding the best way to reach the workplace from the time and distance aspects, depending on the time of the day, is also an optimization. Many people in this world try to reduce the daily cost of living for their survival. All these actions have some amount of mental calculations behind it, and most of the time the calculation is not that mathematical. The optimization process basically finds the values of the variables those control the objective we need to optimize (i.e., minimize or maximize) while satisfying some constraints. This process becomes mathematical when we employ it in the professional sectors like finance, construction, manufacturing, etc. In those cases, the number of variables is quite high, and they are correlated in a complex way.

Mathematical optimization has various components. The first is the objective function, which defines the attribute to be optimized in terms of the dependent variables or design variables. For example, in manufacturing process, it describes the profit or the cost or the product quality. The design variables are the variables which control the value of the attribute, which is being optimized. The amounts of different resources used and the time spent in a manufacturing process may be considered as the design variables. The third important component in an optimization process is the constraint. It may be single or a set of constraints, which allow the process to take on certain values of the design variables but exclude others.

Among the different approaches of optimization, the classical derivative-based approaches were the most established and common optimization methods. But in recent years, advent of metaheuristic methods has changed the domain of optimization and made those methods more acceptable due to their ability to handle complex problems and lesser possibility of getting stuck in the local optima. A metaheuristic method has the strategy to guide and modify the pure heuristics to produce better solutions which are beyond the solutions generated in heuristic processes. The metaheuristic optimization algorithms consist of the evolutionary

algorithms, the swarm intelligence, other bio- and nature-inspired algorithms, and some Physics and process-based algorithms. In this book, most of the chapters deal with several case studies related to application of these metaheuristic algorithms in different domains of industry.

Optimizations methods are extensively being applied to different sectors of the industries and professional life, which includes the information system, the financial operations, the manufacturing systems, engineering design and design optimization, operations and supply chain management, internet of things, and multicriteria decision-making. Various metaheuristic optimization tools are being used in these spheres of life. In this book, after a brief description of the metaheuristic tools in the first chapter, eleven more chapters have been dedicated for dealing with such applications of the optimization techniques in industrially relevant fields. The authors from different countries have shared their experience in the optimization of systems like banking, steel making, manufacturing processes, electrical vehicles design, and civil constructions.

The editors express their gratitude to all the authors for their effort to make excellent contributions for the book. The editors are also grateful to all the reviewers who have taken the pain to improve the quality of the chapters further and make the book even better. The colleagues, friends, and family members of both the editors are gratefully acknowledged. The editors also acknowledge the brilliance of the Springer team shaping the compilation beautifully and express thanks to them.

Chennai, India — Shubhabrata Datta
Aveiro, Portugal — J. Paulo Davim

Contents

About the Editors

Shubhabrata Datta, presently Research Professor in the Department of Mechanical Engineering, SRM Institute of Science and Technology, Chennai, India, did his Bachelors, Masters, and Ph.D. in Engineering from Indian Institute of Engineering Science and Technology, Shibpur, India in the field of Metallurgical and Materials Engineering. He has more than 25 years of teaching and research experience. His research interest is in the area of modeling and optimization in engineering systems, with an emphasis to design of materials using computational intelligence techniques. He has more than 125 publications in journals and peer-reviewed conference proceedings. He was conferred with Exchange Scientist Award from Royal Academy of Engineering, UK and worked in the University of Sheffield, UK. He also worked in Dept of Materials Science and Engineering, Helsinki University of Technology, Finland, Dept of Materials Science and Engineering, Iowa State University, Ames, USA and Heat Engineering Lab, Dept of Chemical Engineering, Åbo Akademi University, Finland as Visiting Scientist. He is a Fellow of Institution of Engineers (India), and Life Member, Materials Research Society of India. He is the Associate Editor, Journal of the Institution of Engineers (India): Series D, Member, Editorial Board of International Journal of Materials Engineering Innovation and Member, Editorial Review Board, International Journal of Surface Engineering and Interdisciplinary Materials Science.

J. Paulo Davim received his Ph.D. in Mechanical Engineering in 1997, M.Sc. in Mechanical Engineering (materials and manufacturing processes) in 1991, Mechanical Engineering (5 years) in 1986 from the University of Porto (FEUP), the Aggregate title (Full Habilitation) from the University of Coimbra in 2005, and D.Sc. from London Metropolitan University in 2013. He is Eur Ing by FEANI-Brussels and Senior Chartered Engineer by the Portuguese Institution of Engineers with an MBA and Specialist title in Engineering and Industrial Management. Currently, he is Professor at the Department of Mechanical Engineering of the University of Aveiro, Portugal. He has more than 30 years of teaching and research experience in Manufacturing, Materials, Mechanical and Industrial Engineering, with special emphasis in Machining & Tribology. He also

has interest in Management, Engineering Education and Higher Education for Sustainability. He has guided large numbers of postdoc, Ph.D., and master's students as well as has coordinated and participated in several financed research projects. He has received several scientific awards. He has worked as evaluator of projects for international research agencies as well as examiner of Ph.D. thesis for many universities in different countries. He is the Editor in Chief of several international journals, Guest Editor of journals, books Editor, book Series Editor, and Scientific Advisory for many international journals and conferences. Presently, he is an Editorial Board member of 25 international journals and acts as reviewer for more than 80 prestigious Web of Science journals. In addition, he has also published as editor (and co-editor) more than 100 books and as author (and co-author) more than 10 books, 80 book chapters, and 400 articles in journals and conferences (more than 200 articles in journals indexed in Web of Science core collection/h-index 47+/6500+ citations and SCOPUS/h-index 54+/9000+ citations).

Optimization Techniques: An Overview

Shubhabrata Datta, Sandipan Roy and J. Paulo Davim

Abstract There are several types of optimization methods having their own advantages and disadvantages. In recent times, metaheuristic optimization techniques are gaining attention and being applied to various industrial applications. In this chapter, a brief description of classes of optimization techniques is followed by an elaboration of the most popular optimization techniques, which are getting substantial uses in the industries. The above techniques include evolutionary algorithms, swarm intelligence techniques and simulated annealing.

Keywords Optimization methods · Classification · Evolutionary · Computation Swarm optimization · Ant colony optimization · Simulated annealing

1 Introduction

Optimization is a method to search the best solution in a given circumstance. Engineers need to decide on several technological and managerial issues for design, manufacturing, construction, as well as maintenance of systems. Minimization of efforts with maximization of the desired output is the most crucial aspect of all search decisions. Thus, efforts required to be given and/or to obtain the desired output is a function of certain variables and can be optimized to achieve the optimum level of that function [1, 2].

S. Datta (✉) · S. Roy
Department of Mechanical Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai 603203, India
e-mail: shubhabrata.p@ktr.srmuniv.ac.in

J. P. Davim
Department of Mechanical Engineering, University of Aveiro, Campus Santiago, 3810-193 Aveiro, Portugal

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_1

Different applications of optimizations in the field of engineering are as follows:

- Optimum design of civil engineering structures.
- Minimum-weight design of structures for earthquake, and other types of random loading.
- Design of minimum cost materials management methods.
- Design of maximum efficiency pumps, turbines, and equipment for heat transfer.
- Design optimization of electrical machinery and networks.
- Planning, scheduling, and controlling of manufacturing processes.
- Design of materials with improved performance.
- Decreasing the manufacturing cost, etc.

During the design process, an engineering system is defined by a set of variables on which the performance of the system depends. As a common aspect, some of the variables are fixed at the beginning as preassigned parameters. The other parameters are analyzed during the design optimization process, which are called design or decision variables [1]. In many cases, these design variables had to satisfied pre-specified requirements, hence, they cannot be chosen arbitrarily. To produce an acceptable design, these restrictions have to be satisfied and they are called optimization constraints. Among the constraints, behavior or functional constraints restrict the behavior of the system. Those constraints which indicate restrictions on design variables are called geometric or side constraints. In case of application to the manufacturing industry the plant constraints, describing the limits of the plant capacity, need to be considered during optimization. Any computed design usually leads to situations, which provide feasible and acceptable path toward the solution of the problem. Naturally, several other designs of the same problem could be computed with superior or inferior performances. To fulfill the purpose of optimum outcome, the best among all the acceptable designs available needs to be chosen. Thus, a decisive factor must be selected which could compare among the several acceptable designs and the optimum design could be chosen.

This decisive factor for optimizing the design is called the objective function. The objective function needs to be formulated depending on the inherent mechanism of the problem [1]. For example, in the case of structural design problems weight of automobiles or aircraft and cost in civil structures are common objective functions for minimization. Similarly, mechanical efficiency in mechanical engineering systems design is an objective function for maximization. An optimization algorithm starts with random initial solutions and improves in iterative methods. After certain iterations, it converges to the optimum solutions for the problem [3, 4]. The algorithm makes the system attracted towards the optimum solution from the solutions existing at different states, as a self-organizing system. Such an iterative system can be developed using some mathematically described equations or rules [5].

2 Classification of Optimization Methods

Optimization techniques can be classified from various aspects. It can be divided based on the presence of constraints. If the optimization problems have one or more constrained, it is called constraints optimization, and otherwise defined as an unconstrained optimization problem. Optimization can also be classified according to the number of objectives. In a single-objective optimization, either in search of maximum or minimum optimum solution, it leads to a single optimum solution. If there is more than one objective having conflict between them, the situation calls for multiobjective optimization. Here, a set of optimum non-dominated solutions evolve, which is represented in the form of Pareto front [6]. If the number of objectives is more than three, it is also called a many-objective optimization. The optimization problems can also be classified based on the nature of the objective functions as well as the constraints. The problem is called a linear or nonlinear problem based on the objective function and the constraints being linear or nonlinear. If the objective function is polynomial, it is called a geometric problem. If the constraints are integrated with the objective function and cannot be separated, then the problem is called a non-separable programming problem. When it comes to time constraints, optimization can be distinguished into two main types, online and offline optimization. If the job needs to be solved within a few seconds or milliseconds, it is called online optimization [6]. In such cases, the focus of the algorithm is speed. Robot localization, job schedule updating, transport process, and search engine are an example of such optimization.

These examples indicate that the online optimization has to be carried out in a repetitive manner to cater different orders arriving continuously to the system without significant weighting. In cases where time is not important as users are ready to wait for the optimal or close-to-optimal results, offline optimization is acceptable [6]. In case of offline optimization, the process is carried out only once. Here the optimization strategies are more important and have to be decided before the starting of the process. Again, deterministic and heuristic algorithms are two classes of the optimization algorithm.

In the case of deterministic algorithms, an obvious relation between the attributes of the system exists. It becomes hard to solve a problem deterministically when the relation between the system parameter and the objective/fitness of the system are complicated or obscure.

A heuristic algorithm [7, 8] gathers information about the system and fitness of random solution is tested and decision is taken for generating the next solution. Thus, these methods are dependent on the nature of the problem. On the other hand, a metaheuristic technique combines the heuristics and the objective function efficiently without depending on the structure of the problem [9, 10]. In this way, a metaheuristic algorithm makes the heuristic methods applicable to a wide range of problems.

High-quality solution for combinatorial optimization problems can be found by metaheuristics in a reasonable time [11].

A single-run algorithm like constructive methods or iterative improvement generate a limited number of solutions and even stop at local optima [12, 13]. If a function is difficult from the mathematical point of view when the context is discontinuous, not differentiable, or having multiple optima, a deterministic or classical optimization algorithm may converge to solutions in the local optima or from a small part of the search space. Thus, the problem with classical optimization algorithms is that generally, it is impossible to find whether the generated solution is situated in a local or global optima space. Similarly, it can also be stated that whether the provided solution is the result of the whole search space or only a part of it. This issue is more important for a multimodal problem, where multiple optima exist [14–17].

The concepts of metaheuristic optimization techniques are getting more and more accepted in real-life applications due to their robustness and capability to deal with complex design problems as discussed above. The metaheuristic techniques can be broadly divided into three groups. The first is the evolutionary computation group, consisting of genetic algorithm, genetic programming, evolutionary strategy, differential evolution, etc. The second group can be named swarm intelligence group consisting of particle swarm optimization, ant colony optimization, etc. The techniques within these two groups can also be termed as bioinspired optimization techniques. The third group consists of physical-process-inspired optimization techniques, and simulated annealing is the most common optimization tool in this group. The groups are shown in Fig. 1. Later in this chapter, some of these techniques will be discussed.

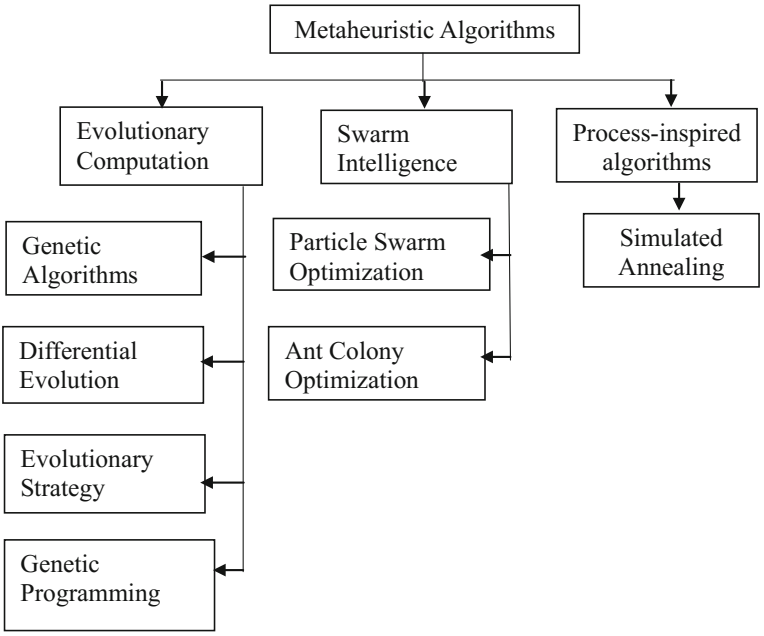


Fig. 1 Major classes of metaheuristic global optimization algorithms with important variants

3 Evolutionary Computation

Evolutionary computation (EC) uses principles inspired by the natural evolution of species and being used nowadays for solving many complex engineering problems. Four main types of this genre of algorithm are genetic algorithms [18], evolution strategies [19, 20], differential evolution [21], and genetic programming [22]. The common part of these four algorithms is that all four algorithms are populations based and use operators inspired by the concept of natural evolution. The selection operator chooses the individuals who will survive and create the next generation. If the objective function provided better fitness of an individual, then the probability of selection of the individuals is higher. The selected individuals are combined to form offspring through an operator called crossover, which merges parts of two selected individuals to generate in new individuals. The mutation operator introduces random modification within one individual to incorporate diversity of the solutions. These algorithms differ in representations of the individuals and the method of implementing the operators. For example, the mutation operator is more emphasized in the case of evolutionary strategies and genetic programming algorithm.

Compared to other methods, the most significant advantage of EC is that little knowledge about the system is required to solve the problem, and thus these can be applied to a wide range of optimization problems. Even if the fitness function is not continuous, not differentiable and highly complex then also EC can solve the problem efficiently [23]. Additional information or knowledge about the system can also be incorporated in the optimization framework as constraints to improve the performance of the algorithm and to achieve better solutions.

As the different types of EC algorithms are available, users can easily choose the methods suited for a given problem depending on the structured of the system to be optimized. These methods can easily be used for multi-objective optimization problems [24]. EC provides a described heuristic estimation of optimum solution through a process based on certain operators [25–30]. The schema of the EC Algorithm is given in the form of a flowchart in Fig. 2.

All the evolutionary computing algorithms follow the same general methods with certain variations. The structure of a solution varied from one algorithm to other. In the case of Genetic Algorithms (GA), the solutions are represented in the form of a stream. In the case of Evolution Strategies (ES), it is a real-valued vector, and in the case of Genetic Programming (GP) it is represented as a tree. This difference in the structures of candidate solution makes the different types of EC algorithms applicable for different types of optimization problems. Genetic algorithms (GA) are natural selection and natural genetics inspired optimization algorithms. It can be used for a wide range of problems with high complexity in the objective function. To solve practical engineering optimization problems, the idea of using a population of solutions was evolved during the 1950 and 1960s. John Holland of the University of Michigan initiated the idea of sexual reproduction of solutions to EA [31]. This initiation of the new method of the optimization considered makes Holland considered to be the father of Evolutionary Algorithm. His book, written in 1975

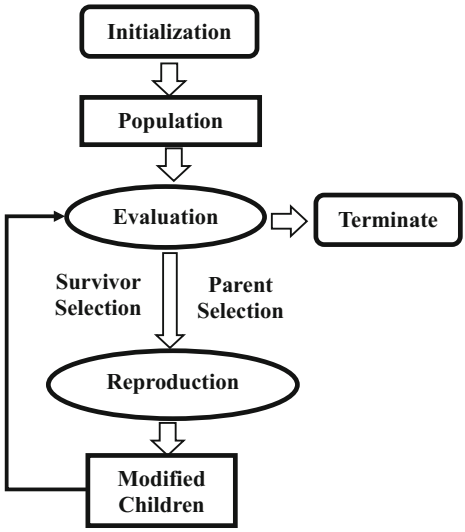


Fig. 2 The scheme of an evolutionary computing

[31] particularly gives an idea about his creative thinker approach. De Jong, in one of his papers entitled Genetic Algorithms emphasized that GA is just NOT Function Optimizers [32] and has more potential than a mere robust method for optimizing and engineering system.

Though there are certain differences in EC algorithms, the basic principles of assessing the performance of the solution and then keeping the better performing solutions for further processing and make the worst performers perish remains the same for all cases [33]. Population-based incremental learning (PBIL) [34], a type of EC algorithm has similarities to ant colony optimization which is discussed later. PBIL has a generating vector which is nothing but a vector and probabilities. Binary strings representing the solutions generated randomly has the ith bit having a probability value corresponding to the generating vector. When the solutions in the population are evaluated through the fitness function the probability values in the generating vector is updated is depending on the quality of the performance of the solutions. In case of ant colony optimization, pheromone trail values are similarly updated depending on the performance.

4 Swarm Optimization

In 1995 by Eberhart and Kennedy developed Particle Swarm Optimization (PSO), which is an optimization method inspired by the behaviors of birds flocking or fish schooling [35]. Later several variations of PSO have emerged for improving the convergence speed and solution quality. Basic PSO is found to be more appropriate for processing simple as well as static optimization problems. The PSO imitate behavior

of such animal societies those find their food without the existence of any leader in the group or swarm. In such cases, the members of the group try to follow other members of the group depending on the closeness of the member from the source of food. This can only be possible due to the presence of communication system among the members of the group. The member who is in a better condition should have the facility to inform the other members of the flocks regarding his present position which can be followed by the other members. This situation is repeated until the food source could be reached. This basic concept of animal behavior is used by PSO for finding optimal solutions. Each particle of the swarm of a particle in particle optimization represents potential solutions.

To improve the convergence speed and solution quality in PSO several variations have been developed, viz., inertia weight, velocity clamping, synchronous and asynchronous updates, constriction coefficient [36]. PSO algorithm [5] adjusts the trajectories of the agents during its search for the optimum solutions in the search space. Each agent or particle compares its current position with its own best location during its random movement. As it reaches a location better than its previous best location, it updates the best location. In this way, the particles search the global best location during the random search. Ant colony optimization (ACO) is another global optimization technique, which falls within the swarm optimization group. Ant colonies are generally formed by social insect societies. Due to the highly structured social organization, ant colonies can accomplish complex tasks. The behavior of real ant acts as the source of the information for this novel algorithm, where the principle of self-organizing of ants in a highly coordinated manner is exploited. In case of SEO populations of artificial ants collaborate between themselves to solve optimization problems. Different ant algorithms have evolved depending on various aspects of ant behavior which include division of labor, foraging, cooperative transport, and brood sorting. In all the above activities, ants' coordinates among themselves through indirect communication methods called stigmergy. In case of foraging a chemical named pheromone, a deposited on the ground by the ants to inform the other ants about its path of movements and directing them to follow the same path. This stigmergic communication system has been proved by the biologists as the methods to achieve the self-organization. One of the most flourishing examples of several ant algorithms is ACO. The communication through pheromone which detected by the antenna of ants make the ants follows the path where a larger amount of pheromone is present. This behavior pursuit by real ants can be considered as a heuristic method. The initial explorations of the ants in the neighboring area occur in a complete random manner without the presence of pheromone. But the addition of pheromone incorporates a certain amount of control over the random search. During the random travel deposition by the pheromone by the ants provide some feedback to the other ants and search methods becomes direction control to a certain extent. As pheromone is evaporating in nature an unused path disappeared with time [37].

Dorigo and Gambardela [38] proposed an improved algorithm, where the three main differences with the previous and system are transition rule, local updating, global updating. The transition rule is made to generate a relation between the explorations and priority of the problem. The local updating methods update the pheromone

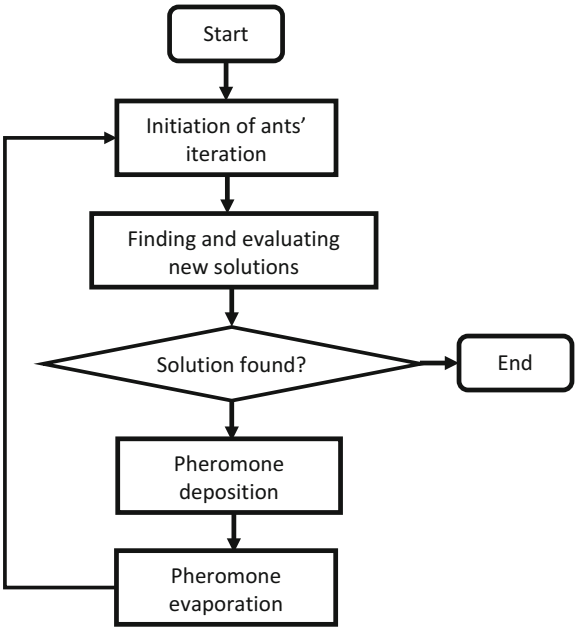


Fig. 3 Structure of ACO algorithm

deposition of the ants during the local search whereas global updating is done for the ants with superior position. Figure 3 shows a general scheme of the Ant Colony Swarm algorithm.

5 Simulated Annealing

Simulated Annealing (SA) mimics a physical process called 'annealing', which has a close relation with thermodynamics, and a method to control the microstructure and properties of metals and alloys through a process of heating and cooling [39]. In this process, the temperature is reduced slowly so that the material can achieve an internal structure or arrangements of atoms in the lowest energy configuration or equilibrium condition. In simulated annealing the objective function replaces the energy of the material being annealed. In case of a minimization problem, lower solution will have lower energy. The optimization method is made of a random move for hill-climbing. Every move gets a probability as it related to the energy or the function value using a parameter similar to Boltzmann constant. Thus, the probability value has an analogy of temperature in thermodynamics and the quality of the solution can be compared with the energy of the system. At higher probability, more uphill moves are possible. The overall probability starts high and is gradually decreased. In the process, optimum solution is reached by one or more moves.

6 Concluding Remarks

The metaheuristic algorithms have greater flexibility to handle problems with complex objective functions and constraints. This reason has made these techniques more applicable for the industrial situations. In such complex real-life situations, these techniques are being used quite successfully to solve the industrial problems related to design, quality control, productivity, and cost.

References

1. Rao, S. S. (2009). *Engineering Optimization Theory and Practice* (4th ed.). Copyright © 2009.
2. Beightler, C. S, Phillips, D. T., & Wilde, D. J. (1979). *Foundations of optimization* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.
3. Koziel, S., & Yang, X. S. (2011). *Computational optimization, methods and algorithms*. Germany: Springer.
4. Yang, X. S. (2010). *Engineering optimization: an introduction with metaheuristic applications*. Wiley.
5. Yang, X.-S. (2014). *School of science and technology*. London: Middlesex University London. Copyright © 2014 Elsevier Inc. ISBN 978-0-12-416743-8.
6. Weise, T. (2009) *Global optimization algorithms—theory and application*, Version: June 26, 2009.
7. Michalewicz, Z., & Fogel, D. B. (2004). *How to solve it: Modern heuristics*. Springer, second, revised and extended edition, December 2004. ISBN: 978-3-54022-494-5.
8. Rayward-Smith, V. J., Osman, I. H., Reeves, C. R., & Smith, G. D. (Eds.). *Modern heuristic search methods*. Wiley, December 1996. ISBN: 978-0-47196-280-9.
9. Glover, F., & Kochenberger, G. A. (Eds.). (2003). Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science. Kluwer Academic Publishers/Springer, New York, USA. ISBN: 978-1-40207-263-5, 978-0-30648-056-0, 0-3064-8056-5, 1-4020-7263-5. https://doi.org/10.1007/b101874. Series Editor Frederick S. Hillier.
10. Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, *35*(3):268–308. ISSN: 0360-0300. CODEN: CMSVAN. http://iridia.ulb.ac.be/~meta/newsite/downloads/ACSUR-blum-roli.pdf.
11. Dorigo, M., & Stützle, T. *Ant colony optimization, a bradford book*. London, England, MA: The MIT Press Cambridge. ISBN 0-262-04219-3.
12. Johnson, D. S., & McGeoch, L. A. (1997). The travelling salesman problem: A case study in local optimization. In E. H. L. Aarts & J. K. Lenstra (eds.), *Local search in combinatorial optimization* (pp. 215–310). Chichester, UK: Wiley.
13. Schreiber, G. R., & Martin, O. C. (1999). Cut size statistics of graph bisection heuristics. *SIAM Journal on Optimization, 10*(1), 231–251.
14. Shekel, J. (1971). Test functions for multimodal search techniques. In *Proceedings of the Fifth Annual Princeton Conference on Information Science and Systems* (pp. 354–359). Princeton, NJ, USA: Princeton University Press.
15. Žilinskas, A. (1978). Algorithm as 133: Optimization of one-dimensional multimodal functions. *Applied Statistics*, *27*(3), 367–375. ISSN: 00359254. https://doi.org/10.2307/2347182.
16. Ursem, R. K. (2003). Models for evolutionary algorithms and their applications in system identification and control optimization. Ph.D. thesis, Department of Computer Science, University of Aarhus, Denmark, April 1, 2003. Advisors: T. Krink & B. H. Mayoh. http://www.daimi.au.dk/~ursem/publications/RKU_thesis_2003.pdf and http://citeseer.ist.psu.edu/572321.html.

17. Schaffer, J. D., Eshelman, L. J., & Offutt, D. (1990). Spurious correlations and premature convergence in genetic algorithms. In *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA)*, pp. 102–112. In proceedings (1924).
18. Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. Addison-Wesley: Reading, MA.
19. Rechenberg, I. (1973). *Evolutions strategie—Optimierung technischer Systemenach Prinzipien der biologischen Information*. Freiburg, Germany: Fromman Verlag.
20. Schwefel, H.-P. (1981). *Numerical optimization of computer models*. Chichester, UK: Wiley.
21. Price, Kenneth, Storn, Rainer M., & Lampinen, Jouni A. (2005). *differential evolution—a practical approach to global optimization*. Berlin, Heidelberg: Springer.
22. Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. New York: Wiley.
23. F. Streichert, *Introduction to evolutionary algorithms,* presented at the Frankfurt MathFinance Workshop, April 2–4, 2002.
24. Van Veldhuizen, D. A., & Lamont, G. B. (2000). Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation, 8*(2), 125–147.
25. Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. Oxford University Press, New York. A book giving a formal treatment of evolutionary programming, evolution strategies, and genetic algorithms (no genetic programming) from a perspective of optimisation.
26. Bäck, T., & Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimisation. *Evolutionary Computation*, *1*(1), 1–23. A classical paper (with formalistic algorithm descriptions) that "unified" the field.
27. Eiben, A. E. (2002). Evolutionary computing: the most powerful problem solver in the universe? *Dutch Mathematical Archive (Nederlands Archief voor Wiskunde)*, *5/3*(2), 126–131. A gentle introduction to evolutionary computing with details over GAs and ES. To be found at http://www.cs.vu.nl/~gusz/papers/ec-intro-naw.ps.
28. Fogel, D. B. (1995). *Evolutionary computation*. IEEE Press. A book covering evolutionary programming, evolution strategies, and genetic algorithms (no genetic programming) from a perspective of achieving machine intelligence through evolution.
29. Hillier, M. S., & Hillier, F. S. (2002). Conventional optimization techniques. Chapter 1. In R. Sarker, M. Mohammadian, & X. Yao, (eds.), *Evolutionary optimization*, (pp. 3–25). Kluwer Academic Publishers. Gives a nice overview of Operations Research techniques for optimisation, including linear-, nonlinear-, goal-, and integer programming.
30. Yao, X. (2002). Evolutionary computation: A gentle introduction. Chapter 2. In R. Sarker, M. Mohammadian, & X. Yao, (eds.), *Evolutionary optimization* (pp. 27–53). Kluwer Academic Publishers. Indeed a smooth introduction presenting all dialects and explicitly discussing EAs in relation to generate-and-test methods.
31. Holland, J. (1975). *Adaption in natural and artificial systems: An introductory analysis with applications to biology, control and artificial systems*. Ann Arbor: The University Press of Michigan Press.
32. De Jong, K. A. (1993). Genetic algorithms are NOT function optimisers. In L. D. Whitley (ed.), *Foundations of genetic algorithms 2*, Morgan Kaufinann.
33. Wright, S. (1931). Evolution in mendelian populations. *Genetics, 16,* 97–159.
34. Baluja, S., & Caruana, R. (1995). Removing the genetics from the standard genetic algorithm. In A. Prieditis & S. Russell (Eds.), *Proceedings of the Twelfth International Conference on Machine Learning (ML-95)* (pp. 38–46). Palo Alto, CA: Morgan Kaufmann.
35. Kennedy, J., & Eberhart, R. C. (1995). Particle swam optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA (pp. 1942–1948).
36. Rini, D. P., Shamsuddin, S. M., & Yuhaniz, S. S. (2011). Particle swarm optimization: Technique, system and challenges. *International Journal of Computer Applications* (0975–8887), *14*(1).
37. Artificial Societies and Social Simulation using Ant Colony, Particle Swarm Optimization and Cultural AlgorithmsSource. (2010). In: Book P. Korosec (ed.), New Achievements in Evolutionary Computation (p. 318), February 2010, Croatia: INTECH. Downloaded from SCIYO.COM. ISBN 978-953-307-053-7.

38. Dorigo, M., & Gambardella, L. M. (1996). *Ant colony system: A cooperative learning approach to the traveling salesman problem*. Technical Report TR/IRIDIA/1996-5, IRIDIA, Université Libre de Bruxelles.
39. van Laarhoven, P. J., & Aarts, E. H. Simulated *Annealing: Theory and Applications (Mathematics and Its Applications) Hardcover*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Evolutionary Computation for Multicomponent Problems: Opportunities and Future Directions

Mohammad Reza Bonyadi, Zbigniew Michalewicz, Markus Wagner and Frank Neumann

Abstract Over the past 30 years, many researchers in the field of evolutionary computation have put a lot of effort to introduce various approaches for solving hard problems. Most of these problems have been inspired by major industries so that solving them, by providing either optimal or near optimal solution, was of major significance. Indeed, this was a very promising trajectory as advances in these problem-solving approaches could result in adding values to major industries. In this chapter, we revisit this trajectory to find out whether the attempts that started three decades ago are still aligned with the same goal, as complexities of real-world problems increased significantly. We present some examples of modern real-world problems, discuss why they might be difficult to solve, and whether there is any mismatch between these examples and the problems that are investigated in the evolutionary computation area.

Keywords Multicomponent optimization · Traveling thief problem
Evolutionary algorithms · Cooperative coevolution

M. R. Bonyadi (✉)
Centre for Advanced Imaging, The University of Queensland, Brisbane, Australia
e-mail: rezabny@gmail.com
URL: http://cs.adelaide.edu.au/~optlog/

Z. Michalewicz
Complexica Pty Ltd., Suite 75, 155 Brebner Drive, West Lakes SA 5021, Australia

Z. Michalewicz
Polish-Japanese Academy of Information Technology, ul. Koszykowa 86,
02-008 Warsaw, Poland

Z. Michalewicz
Institute of Computer Science, Polish Academy of Sciences, ul. Ordona 21,
01-237 Warsaw, Poland

M. R. Bonyadi · Z. Michalewicz · M. Wagner · F. Neumann
Optimisation and Logistics, The University of Adelaide, Adelaide, Australia

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management
and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_2

1 Motivation

The Evolutionary Computation (EC) community over the last 30 years has made a lot of effort designing optimization methods (specifically Evolutionary Algorithms, EAs) that are well-suited for hard problems—problems where other methods usually fail [1]. As most real-world problems[1] are quite complex, set in dynamic environments, with nonlinearities and discontinuities, with variety of constraints and business rules, with a few, possibly conflicting, objectives, with noise and uncertainty, it seems there is a great opportunity for EAs to be applied to such problems.

Some researchers investigated features of real-world problems that served as "reasons" for difficulties that EAs experience in solving them. For example, in [2] the authors discussed premature convergence, ruggedness, causality, deceptiveness, neutrality, epistasis, and robustness that make optimization problems hard to solve. However, it seems that these reasons are either related to the landscape of the problem (such as ruggedness and deceptiveness) or the optimizer itself (like premature convergence and robustness) and they are not focusing on the nature of the problem. In [3], a few different reasons behind the hardness of real-world problems were discussed that included the size of the problem, presence of noise, multi-objectivity, and presence of constraints. Apart from these studies, there have been EC conferences (e.g., GECCO, IEEE CEC, PPSN) that have had special sessions on "real-world applications". The aim of these sessions was to investigate the potentials of EC methods in solving real-world optimization problems.

Most of the features discussed in the previous paragraph have been captured in optimization benchmark problems (many of these benchmark problems can be found in OR-library[2]). As an example, the size of benchmark problems has been increased during the last decades and new benchmarks with larger problems have appeared (e.g., knapsack problems, KP, with 2,500 items and traveling salesman problems, TSP, with more than 10,000 cities). Presence of constraints has also been captured in benchmark problems (e.g., constrained vehicle routing problem, CVRP) and studied by many researchers. Some researchers also studied the performance of evolutionary optimization algorithms in dynamic environments [4, 5]. Thus, the expectation is, after capturing all (or at least some) of these pitfalls and addressing them, EC optimization methods should be effective in solving real-world problems. However, after over 30 years of research and many articles written on Evolutionary Algorithms in dedicated conferences and journals with special sessions on applications of evolutionary methods on real-world applications, still it is not that easy to find EC-based applications in real-world.

There are several reasons [6] for such mismatch between the contributions made by researchers to the field of Evolutionary Computation over many years and the number of real-world applications which are based on concepts of Evolutionary Algorithms. These reasons include:

---

[1] See [1] for details on different interpretations of the term "real-world problems".

[2] Available at: http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

(a) Experiments focus on single-component (also known as single silo) benchmark problems
(b) Dominance of Operation Research methods in industry
(c) Experiments focus on global optima
(d) Theory does not support practice
(e) General dislike of business issues in research community
(f) Limited number of EA-based companies.

It seems that the reasons (a) and (b) are primary while the reasons (c) to (f) are secondary. Let us explain.

There are thousands of research articles addressing traveling salesman problems, job shop, and other scheduling problems, transportation problems, inventory problems, stock cutting problems, packing problems, various logistic problems, to name but a few. Although most of these problems are NP-hard and deserve research efforts, they are not exactly what the real-world industries need. Most companies run complex operations and they need solutions for problems of high complexity with several components (i.e., multicomponent problems[3]). In fact, problems in real-world usually involve several smaller subproblems (several components) that interact with each other and companies are after a solution for the whole problem that takes all components into account rather than only focusing on one of the components. For example, the issue of scheduling production lines (e.g., maximizing the efficiency or minimizing the cost) has direct relationship with inventory costs, transportation costs, delivery-in-full-on-time to customers, etc., hence it should not be considered in isolation. Moreover, optimizing one component of the operation may have negative impact on other activities in other components. These days, businesses usually need "global solutions" for their operations that includes all components together, not single-component solutions. This was recognized already over 30 years ago by Operations Research (OR) community; in [7], there is a clear statement: "Problems require holistic treatment. They cannot be treated effectively by decomposing them analytically into separate problems to which optimal solutions are sought." However, there are very few research efforts which aim in that direction that is mainly due to the lack of appropriate benchmarks or test cases available. It is usually harder to work with a company on such global level because the delivery of a successful software solution involves many other (apart from optimization) skills such as understanding the company's internal processes and complex software engineering issues.

Further, there are many reasons why OR methods are widely used to deal with such problems. One reason is that the basic OR approaches (e.g., linear programming) are introduced to many students in different disciplines. This makes these approaches well-known by researchers and industries, and, consequently, widely used. Also, OR community has a few standard and powerful configurable products (e.g., CPLEX) that are used in many organizations especially for complex systems with many components [8] (see also [6] for further discussion).

[3]There are concepts similar to multicomponent problems in other disciplines, e.g., OR and management sciences, with different names such as integrated system, integrated supply chain, system planning, and hierarchical production planning.

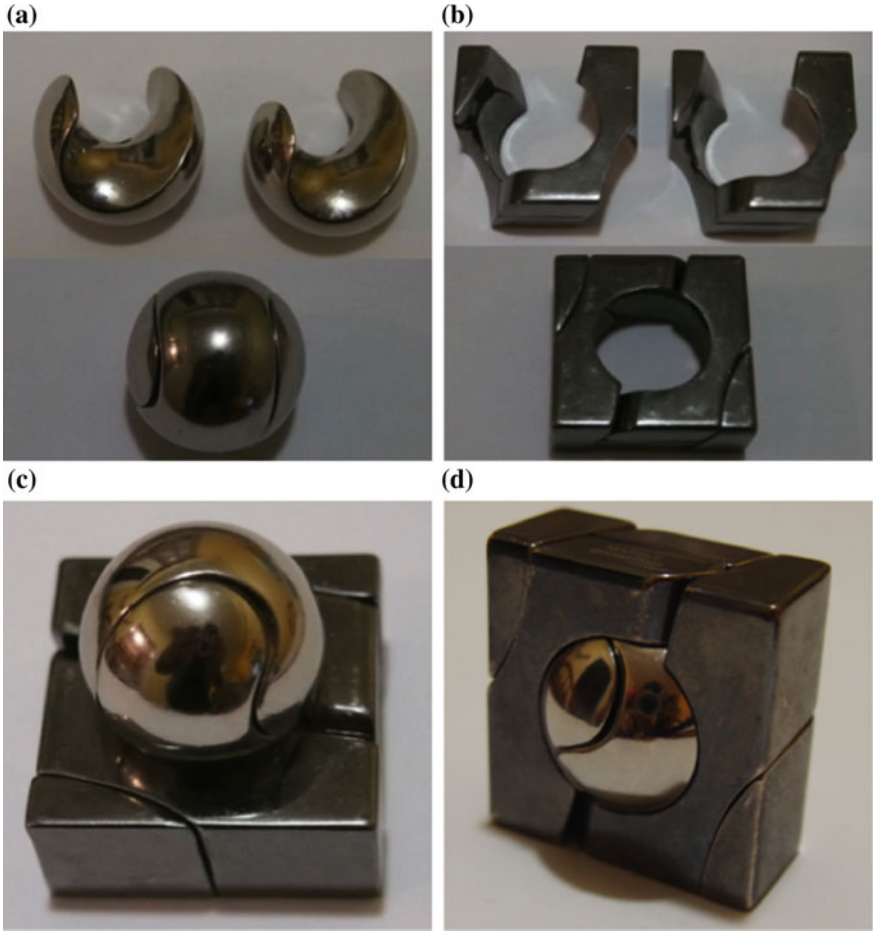


Fig. 1 The Cast Marble puzzle: **a** two pieces of a ball and the ball that is generated by setting up the pieces, **b** two pieces of a cuboid and the cuboid that is generated by setting up the pieces, **c** the ball is not fit in the hole of the cuboid if the cuboid is set up first, and **d** the solution of the puzzle

Let us illustrate the differences between single-component and multicomponent problems by presenting a puzzle (see Fig. 1).[4] There is a ball that has been cut into two parts in a special way, and a cuboid with a hole inside that has been also cut into two parts in a special way. The two parts of the ball can be easily set up together to make a complete ball (see Fig. 1a). Also, the two parts of the cuboid can be put easily together to shape the cuboid (see Fig. 1b). The size of the hole inside the cuboid is slightly larger than the size of the ball, so that, if the ball is inside the hole it can spin freely. However, it is not possible to set up the cuboid and then put the ball inside the cuboid as the entry of the hole of the cuboid is smaller than the size of the ball

[4]The name of this puzzle is the "Cast Marble", created by Hanayama company.

(see Fig. 1c). Now, the puzzle is stated as follows: set up the cuboid with the ball inside (see Fig. 1d). Setting up the ball separately and the cuboid separately is easy. However, setting up the cuboid while the ball is set up inside the cuboid is extremely hard.[5]

This puzzle nicely represents the difference between single-component and multicomponent problems. In fact, solving a single-component problem (setting up the ball or the cuboid separately) might be easy; however, solving the combination of two simple component problems (setting up the ball while it is inside the cuboid) is potentially extremely harder.

The purpose of this letter is to encourage the EC community to put more effort into researching multicomponent problems. First, OR community is already doing this. Second, such research seems necessary if we would like to see the emergence of powerful EC-based applications in the real-world. Third, because of the flexibility of EC techniques, we believe that they are more than suitable for delivering quality solutions to multicomponent real-world problems.

So, in this letter we explore this issue further and we organize the letter as follows. In Sect. 2 two real-world examples are explained, in Sect. 3 some important observations about real-world problems are discussed, in Sect. 4 a recently presented benchmark multicomponent problem is introduced and discussed, and in Sect. 5 some discussions and directions for future research are provided.

2 Two Examples

The first example relates to optimization of the transportation of water tanks [9]. An Australian company produces water tanks with different sizes based on some orders coming from its customers. The number of customers per month is approximately 10,000; these customers are in different locations, called stations. Each customer orders a water tank with specific characteristics (including size) and expects to receive it within a period of time (usually within one month). These water tanks are carried to the stations for delivery by a fleet of trucks that is operated by the water tank company. These trucks have different characteristics and some of them are equipped with trailers. The company proceeds in the following way. A subset of orders is selected and assigned to a truck and the delivery is scheduled in a limited period of time (it is called subset selection procedure). Because the tanks are empty and of different sizes, they might be packed inside each other (it is called bundling procedure) to maximize trucks load in a trip. A bundled tank must be unbundled at special sites, called bases, before the tank delivery to stations. Note that there might exist several bases close to the stations where the tanks are going to be delivered and selecting different bases (it is called base selection procedure) affects the best overall achievable solution. When the tanks are unbundled at a base, only some of them fit

[5]The difficulty level of this puzzle was reported as 4 out of 6 by the Hanayama website, that is equal to the difficulty of Rubik's cube.

in the truck as they require more space. The truck is loaded with a subset of these tanks and carries them to their corresponding stations for delivery. The remaining tanks are kept in the base until the truck gets back and loads them again to continue the delivery process (it is called delivery routing procedure).

The aim of the optimizer is to divide all tanks ordered by customers into subsets that are bundled and loaded in trucks (possibly with trailers) for delivery and to determine an exact routing for bases and stations for unbundling and delivery activities to maximize the total "value" of the delivery at the end of the time period. This total value is proportional to the ratio between the total prices of delivered tanks to the total distance that the truck travels.

Each of the mentioned procedures in the tank delivery problem (subset selection, base selection, delivery routing, and bundling) is just one component of the problem and finding a solution for each component in isolation does not lead to the optimal solution of the whole problem. As an example, if the subset selection of the orders is solved to optimality (the best subset of tanks is selected in a way that the price of the tanks for delivery is maximized), there is no guarantee that there exists a feasible bundling such that this subset fits in a truck. Also, by selecting tanks without considering the location of stations and bases, the best achievable solutions might not be very high quality, e.g., there might be a station that needs a very expensive tank but it is very far from the base, which actually makes delivery very costly. On the other hand, it is impossible to select the best routing for stations before selecting tanks—without selection of tanks, the best solution (lowest possible tour distance) is to deliver nothing. Thus, solving each subproblem in isolation does not necessarily lead to the overall optimal solution.

Note also that in this particular case there are many additional considerations that must be taken into account for any successful application. These include scheduling of drivers (who often have different qualifications), fatigue factors and labor laws, traffic patterns on the roads, feasibility of trucks for particular segments of roads, maintenance schedule of the trucks.

The second example relates to the optimization of a wine supply chain [10], from grape to bottle. The overall aim of the wine producer is to deliver the orders in time while minimizing the useless storage. The wine producer needs to decide if the grape is at its peak of ripeness, i.e., optimal maturity, to be collected and used for wine production. This is done through a predictive model that assesses different characteristics of the grape (e.g., sugar, acidity, berry metabolites, berry proteins, taste) to determine when is the grape in its "optimal maturity" for harvest. The definition of optimal maturity may vary depending upon the style of wine being made; the working definition of quality; varietal; rootstock; site; interaction of varietal, rootstock and site; seasonal-specific factors; viticultural practices; and downstream processing events and goals. Once the ripeness was verified, the grapes are removed and sent to the weighbridge (this stage is called intake planning). After weighing the grapes, they are crushed using specific crushers to provide grape juice (this is called crusher scheduling). The grape juice is then stored in some tanks where they are fermented to provide wine, each tank may have different capacities and capabilities (this is called storage scheduling). Different types of wine may require specific tank

attributes for processing, such as refrigeration jackets or agitators. As some special types of wine may need a blend of different grapes juice, it is better to store the juice in adjacent tanks to facilitate such blending if necessary. Also, it is better to fill/use the tanks to their full capacity as a half-empty tank affects the quality of the wine inside to a degree that the wine might become useless (because of its quality) after a while. To prepare the final product, the wines in the tanks should be bottled that requires scheduling the bottling lines (this is called the bottling line scheduling). This is done through bottling lines where appropriate tank of wine (according to the placed orders) are connected to the line and the wine is bottled and sent for either storage or to direct delivery.

The aim of the optimizer is to find a feasible schedule for the intake plan to remove grapes in their pick of ripeness, schedule crushers to press these grapes, assign the crushed grapes to proper tank farms with appropriate facilities depending on the desired wine, and schedule bottling lines to the tank farms to perform bottling and deliver placed orders as soon as possible.

Each of the mentioned models in the optimization of the wine supply chain (intake planning, crushing stage, storage scheduling, and bottling line scheduling) is just one component of the overall problem and finding a solution for each component in isolation does not lead to the optimal solution of the whole problem. For example, daily decisions on the crushing should not be done in isolation from storage of wines and juices in the tank farm, as even if the crushing can be done in a very efficient way, the capacity of tanks might constraint the process. Also, the storage scheduling should consider the crushing too as the optimal choice for the storage depends on the amount and type of the processed material in the crushing stage.

In the real-world case, there are some other considerations in the problem such as scheduling of workers for removing grapes and required transports, maintenance schedule for tools and machines, deal with sudden changes and uncertainty (e.g., weather forecast) and take into account risk factors.

3 Lessons Learned—Dependencies

Let us take a closer look at the examples presented in Sect. 2. Obviously, both optimization problems contain constraints and noise; both of them might be large in terms of the number of decision variables. However, there is another characteristic present in both problems: each problem is a combination of several subproblems (components/silos). The tank delivery problem is a combination of tank selection, delivery routing, base selection, and bundling. Also, the wine industry problem is a combination of intake planning, crushers scheduling, storage scheduling, and bottling line scheduling. Because of this characteristic, we call them multicomponent problems. Each component might be hard to solve in isolation and solving each component to optimality does not necessarily direct the search toward good overall solutions if other components are not considered. In fact, solutions for each subproblem potentially affect the variables of some other subproblems because of the dependencies

among components. As an example, in the tank delivery problem, delivery routing (best route to deliver tanks) is affected by the base selection (best choice for the base to unbundle the tanks) as choosing different bases imposes different lengths of travels between the base and stations. Delivery routing is also affected by the tank selection (selecting the tanks for delivery) as the selected subset of tanks determines the stations to visit. In short, a solution for each component affects the feasibility or the best achievable solution in other components. In spite of the importance of this topic in real-world problems, the progress to address such problems with dependencies has been very limited in the EC community so far.

Dependency causes appearance of some other features in real-world problems. As an example, the dependency among components in a problem sometimes causes a special flow of data among components of that problem. For instance, generating a solution for the delivery routing in the tank delivery problem is impossible without being aware of the selected subset of tanks.

Definition 1 If generating a solution (independent from the quality) for a component A is impossible because of some data that needs to be provided by the component B, we say that A is the data follower of B.

This indeed imposes a special sequence of solution procedure for some components that need to be taken into account by the solver.

Also, dependencies among components make (mathematical) modeling of problems more complex. In fact, modeling of existing benchmark problems (such as TSP, multidimensional KP, job shop scheduling, etc.) is relatively easy and many different models for them already exist. However, a multicomponent problem involves a more complex model, even if it has been composed of components that have been modeled in previous studies. The main reason is that in a multicomponent problem a constraint that is in one of the components may influence feasible set of solutions of other components because of dependency. Thus, modeling each component in isolation and putting these models together does not express the model for the whole problem.

Another effect that is caused by dependency in multicomponent problems is the propagation of noise. Noise in benchmark problems is usually defined by a stochastic function like a normal or Poisson distribution that is simply added to the objective function or constraints. However, with the presence of dependency noise is propagated from each component to the others. Because dependency between components might follow a complex function, the propagated noise from one component to the others might also become complex (even if the original noise was based on a simple distribution) which causes difficulties in solving the problem. As an example, if the break-down distribution of the crushers in the wine supply chain problem is Poisson with some parameters and this break-down has some other distribution for the tanks maintenances, the effect of these noises on the objective function cannot be treated as a simple Poisson and it is actually hard to even estimate. Note also that investigation of noise over the whole system results in a better risk analysis in the whole operation and a better estimation tolerance in the final benefit.

An additional interesting feature that comes into play because of dependency is the concept of bottleneck component. A bottleneck component is a component in the whole system that constrains the best overall achievable solution. If there is a bottleneck component in the system, adding more resources to other components does not cause improvement of the best achievable objective value (see also [11] for details). As an example, in the wine supply chain problem if the bottleneck component is the number of crushers then investment on any other parts of the system (expanding the number of tanks, hiring more workers to remove grapes, or establishing new bottling line) has minimal affect (or even no effect) on the best overall achievable solution in the system.

Dependency is the source of some complexities in multicomponent problems. Thus, a proper definition for dependency among components is of importance. The concept of dependency among components and effects of the components on each other is similar to the concept of separability of variables in continuous space optimization problems [12]. As it was mentioned earlier, dependency stems from the effects of components on each other, i.e., changing the solution of a component affects feasibility or quality of solutions of other components. Accordingly, we suggest the following definition for dependency among components:

Definition 2 We say component B is dependent on component A (notation: $A \rightarrow B$) if
(1) A is not data follower of B (see Definition 1), and
(2) changing the solutions of component A can change the best achievable solution for the component B in terms of the overall objective value.

The part (1) of the definition prevents introduction of dependency between two components, A and B, where A needs a flow of data from B. If A is a data follower of B then B cannot be dependent to A. Assessing the first part of the definition is not a hard task. The part (2) of the definition ensures that the components are not separable. To assess if B is dependent to A, assume that there exists a solution a for the component A and, given A is fixed to a (showing by $A = a$), setting $B = b$ results in the best possible overall objective value. Now, if there exists an a that for $A = a$, $B = b' \neq b$ results in the best overall objective value, then $A \rightarrow B$ (B is dependent to A). This means that changing the solution for A actually might change the best solution for B. Dependency is shown in a diagram for the example problems (see Fig. 2).

Figure 2 shows the diagram of dependency among components of the tank delivery problem. The links in the figure refer to a dependency among different components. As an example, one can fix the solution for the base selection (of course with being aware of the solution for the subset selection) without being aware of the delivery routing solution. Also, by changing the base, the best achievable solution (shortest tours) for delivery routing is changed. Thus, the base selection is linked to delivery routing. Note that generating a solution for delivery routing is impossible without being aware of the location of the base, hence, there is no link from delivery routing to the base selection.

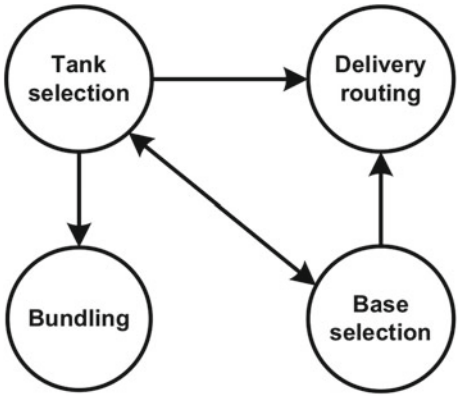


Fig. 2 Diagram of dependency among components of tank delivery problem

Hypothetically, any dependency can exist between a set of problems. These dependencies can be represented by a digraph, which can potentially form a complex network. In the simplest case, there are some problems with no dependencies. In this case, one can solve each problem separately to optimality and combine the solutions to get the global optimal solution.

4 Traveling Thief Problem (TTP)

A recent attempt [13] to provide an abstraction of multicomponent problems with dependency among components was introduced recently; it was called the traveling thief problem (TTP). This abstraction combined two problems and generated a new problem, which contains two components. The TSP and KP were selected and combined, as both problems were well-known and researched for many years in the field of optimization. TTP was defined as follows. A thief is supposed to steal m items from n cities, where the distances $d_{j,k}$ between cities j and k, profits of each item (p_i), and weights of the items (w_i) are given. The thief carries a limited-capacity W knapsack to store the collected items. The problem is to find the best plan for the thief (in terms of maximizing its total benefit) to visit all cities exactly once (a TSP component) and select the items from these cities (a KP component) so its total benefit is maximized.

To make these two components dependent, it was assumed that the current speed of the thief is affected by the current weight of the knapsack (W_c). In other words, the more items the thief selects, the slower he can move. A function $v : R \rightarrow R$ was defined that maps the current weight of the knapsack to the current speed of the thief: $v(W)$ is the minimum speed of the thief (full knapsack) and $v(0)$ is the maximum speed of the thief (empty knapsack). Further, the thief's profit is reduced (e.g., rent of the knapsack, r) by the total time he needs to complete the tour. So the total profit B of the thief is then calculated by

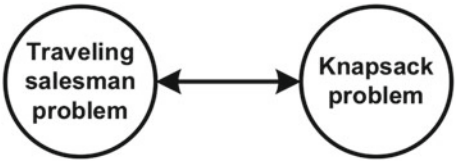


Fig. 3 TTP dependency graph

$$B = P - r \times T$$

where P is the aggregation of the profits of the selected items, r is the rent of the knapsack, and T is the total tour time.

It is possible to generate solutions for KP or TSP in TTP sequentially. Note, however, that each solution for the TSP component impacts the best possible solution for the KP because of the total profit is a function of travel time. Further, each solution for the KP component impacts the tour time for TSP as the weight of the knapsack impacts the speed of travel due to the variability of weights of items (Fig. 3).

Note that different values of the rent r and different functions v result in different instances of TTPs that might be "harder" or "easier" to solve. For example, for small (relative to P) values of r, $r \times T$ contribute a little to the value of B. In the extreme case ($r = 0$), the contribution of $r \times T$ is zero, so the best solution for a given TTP is equivalent to the best solution of the KP component. In other words, in such a case there is no need to solve the TSP component at all. By increasing the value of r, the contribution of $r \times T$ becomes larger and if the value of r is very large (relative to P) then the impact of P on B becomes negligible. In such a case the optimum solution of the TTP would be very close to the optimum solution of the given TSP (Fig. 4).

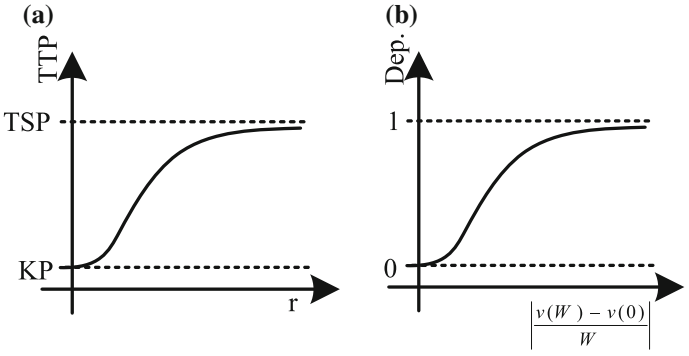


Fig. 4 **a** The impact of the rent rate r on dependency in TTP. For $r = 0$, the TTP solution is equivalent to the solution of KP, while for larger r the TTP solutions become more closer to the solutions of TSP. **b** The impact of the speed v on dependency in TTP. When v does not drop significantly for different weights of picked items ($\left|\frac{v(W)-v(0)}{W}\right|$ is small), the two problems can be decomposed and solved separately. The value Dep. = 1 (Dep. is the short for Dependency) shows that the two components are dependent while Dep. = 0 shows that two components are not dependent

Similar analysis can be done for the function v. For a given TSP and KP components, different functions v would result in different instances of TTPs. These instances, as before, might be "harder" or "easier" to solve. For example, lets assume that v is a decreasing function of weight of the knapsack, i.e., selecting items with positive weight would not increase the value of v. In such a case, if selected items do not affect the speed of the travel (i.e., $\left|\frac{v(W)-v(0)}{W}\right|$ is zero) then the optimal solution of the TTP is the composition of the optimal solutions of KP and TSP that are solved separately, as selection of items does not change the time of the travel. As the value of $\left|\frac{v(W)-v(0)}{W}\right|$ grows, the TSP and KP become more dependent on each other (i.e., selection of items have a more significant impact on the travel time), so selecting more items reduces the value of B significantly. In the extreme case ($\left|\frac{v(W)-v(0)}{W}\right|$ is infinitely large) it would be better not to pick any item (i.e., the solution for the KP would be to pick no items at all) and only solve the TSP component as efficiently as possible. This has been also discussed in [14].

By now, quite a number of researchers have reported experimental results and analytical investigations of TTP. Opportunities to decompose instances of TTP to KP and TSP were studied in [15]. Reported experimental results indicated that methods which consider both components simultaneously consistently achieve better results than methods which decompose the problem and solve them separately. Further, [16] proposed a comprehensive benchmark set for TTP and the experimental results on performances of three methods (one simple heuristic and two evolutionary based methods) were reported.

A recent case study [17] analysed the instances and compared 21 algorithms. It was observed that only a small subset of the published algorithms was needed for well-performing TTP portfolios, as some of the early approaches are already outperformed by newer ones. As the instances vary significantly in size, it is not a big surprise that the current best approaches range from swarm-intelligence approaches for small instances [18], to hyper-heuristics for mid-sized instances [19], and to customized fast implementations of two-opt variants [20].

Until 2017, exact approaches have been unknown. For a constrained version for the TTP, where the tour is fixed, a fully polynomial time approximation scheme is known [21], and very recently several exact approaches for the original TTP have been presented as well [22].

5 Discussion and Future Directions

There are a few challenges in dealing with multicomponent problems. In this section, we discuss some of these challenges and present some potential opportunities for EC-based methods to address them.

As it was mentioned earlier, a collection of optimal solutions that correspond to components of a multicomponent problem does not guarantee global optimality. This

is because of presence of dependencies among components. To solve a multicomponent problem, however, it is often necessary to decompose the problem taking into account the dependencies among components. Complexity of such decomposition is usually related to the dependencies among the components. For example, if in the cast marble puzzle the final objective was to set up the ball on top of the cuboid (a simple dependency between components) then the puzzle would be extremely easy (one could set up the ball and the cuboid separately and put the ball on the cuboid). In contrast, setting up the ball inside the cuboid (a complex dependency between components) is extremely hard as the ball and the cuboid need to be set up simultaneously one inside another. Likewise, one can define a simple dependency between KP and TSP in the TTP problem that makes the problems decomposable or makes them tighter together so that they are not easily decomposable.

The lack of abstract problems that reflect the presence of dependencies among components is apparent in the current benchmarks.[6] In fact, real-world supply chain optimization problems are a combination of many smaller subproblems dependent on each other in a network while benchmark problems are singular. Because global optimality (in the sense of the overall objective) is of interest for multicomponent problems, singular benchmark problems cannot assess quality of methods which are going to be used for multicomponent real-world problems with the presence of dependencies.

One of the challenges in solving multicomponent problems relates to the way the problem is modeled. There are two general ways to model such problems. One way is to design a monolithic model to represent all variables, constraints, and objectives of all components of the problem and then design a method to apply to such a model (see [8], for example, for a complex large linear model for a supply chain problem). The other way is to model the components separately and consider the dependencies to integrate the components. A monolithic model for a multicomponent problem carries some disadvantages. As an example, such a model is usually large and hard to define because of potential complications raising form different variables in different domains, different constraints in different components and their potential relations, etc. Also, it is hard to maintain such a model because all components have been fed into a large system (when for example a new component is added or a configuration is changed then the whole model should be redesigned). Finally, such a model usually disregards potential knowledge about the individual components. Note also that the model needs to represent the details of the system such as potential noises and dynamicity on different components that might be of different natures. These issues are, however, easier to address if the model is composed of smaller models, each model represent a component, and their integration represents a modeling for the problem, i.e., a multicomponent model. Such a model is easier to define as it follows the natural representation of the problem, is easier to maintain because of its modularity (adding, removing, or modifying a component only affects a part of the

[6]There are some problems such as multiprocessor task scheduling problem (MTSP) [23] or resource investment scheduling problem (RISP) [24] that can be also considered as two-component problems with a simple dependency between the components.

whole model), and enables the possible usage of existing knowledge, algorithms, and modeling ideas (with modifications) to deal with the problem as some components might be already well-studied.[7]

One can also consider modeling a multicomponent problem in several levels [26] and then apply the bi-level optimization (and in the more general case, the multi-level optimization) approaches to that model of the problem. This approach for modeling is, in fact, a special case of multicomponent modeling. There have been some advances related to single-objective [27] and multi-objective [28] bi-level optimization in the evolutionary computation community. Bi-level models of a problem can be split into an upper and a lower level that depend on each other. Usually, there is a leader-follower relation between the upper and lower level. For the multicomponent problems, however, such relation might not exist. Instead, we can have multiple problems that depend on each other in an arbitrary way. As a bi-level problem can be seen as a special case of a multicomponent problem, we assume that techniques developed in the area of evolutionary bi-level optimization can be very useful for designing evolutionary algorithms for multicomponent problems.

Clearly, choosing among different modeling approaches for multicomponent problems depends on the problem at hand. However, there might be situations that existing knowledge about the problem and its components can assist the designer to model the problem. As an example, if there is existing knowledge about the components of a multicomponent problem (as some components are well-studied problems), then it might be better to use the multicomponent modeling (rather than the monolithic modeling) approach as the existing knowledge may assist in solving the problem effectively. Also, if there is a well-studied bi-level model that can formulate the multicomponent problem at hand, then it might be better to use that model to make use of existing knowledge about the model. Nevertheless, there might be a need to tailor the existing models to fit the components of the problem at hand. Such tailoring should be conducted with consideration of the dependencies as, otherwise, the solution of the model might deviate (potentially significantly) from the solution of the original problem. In addition, it might be possible to model the problem at hand as a single objective or multiple objectives. This again depends on the problem and potential existing knowledge about the objectives as well as the nature of the objectives (whether they can be combined into one objective or, in the case of single objective, if that objective can be decomposed into several objectives). One should note that combining objectives might result in irregularities in the landscape of the problem and make the problem harder or easier to solve for different optimization algorithms.

Typically EC methods offer a great flexibility in terms of incorporating several factors (constraints, multiple objectives, noise, etc.) that allows a designer to retain intricacies of the problem in the model. One possible EC-based approach to deal with a multicomponent model involves cooperative coevolution: a type of

[7] An example of different modeling can be seen for MTSP: one can design a large model to solve MTSP [25] or, alternatively, the components can be modeled separately and different methods are applied to the components and the results are integrated to solve the problem [23].

multi-population Evolutionary Algorithm [29]. Coevolution is a simultaneous evolution of several genetically isolated subpopulations of individuals that exist in a common ecosystem. Cooperative coevolution uses divide and conquer strategy: all parts of the problem evolve separately; fitness of each individual of particular species is assigned based on the degree of collaboration with individuals of other species. It seems that cooperative coevolution is a natural fit for multicomponent problems with presence of dependencies. Individuals in each subpopulation may correspond to potential solutions for particular components, with their own evaluation functions, whereas the global evaluation function would also include dependencies between components. In a more generic framework, one can consider a network of solvers (including heuristics, metaheuristics, and exact methods) to deal with each component where each solver decides individually while communicates with others to meet a global goal.

Cooperative coevolution has already been used to address a few real-world multicomponent problems. For example, in [30, 31] a problem similar to the tank delivery example discussed in Sect. 2 was formulated and a method based on cooperative coevolution was experimented with. Results of experiments showed that the cooperative coevolution method can provide high-quality solutions. These applications were, however, developed "ad hoc" with the lack of any guidance from theoretical investigations. One can consider a systematic way to investigate the dependencies among components and their potential impacts on the performance of the algorithm and its parameters. Also, investigation of different approaches to deal with different components and integration of the solutions can be other research topics in this regard.

The method described in [30, 31] could also provide the solutions within a 10 min time frame, that is another important aspect that needs to be satisfied in solving real-world problems. The reason is that decisions in real-world problems need to be taken in a small amount of time (5–10 min), otherwise, those decisions are not of much use. A team of OR experts designed a method to solve a multicomponent problem with almost the same complexity of the one considered in [30, 31]. After application of the algorithm to the real-case, it was found that the algorithm needs 18 h per objective to deal with the problem.[8] Such delay for providing solutions is, however, not acceptable.

One should note that the concept of global optimality is usually not of a great importance when dealing with real-world problems. One reason is that the model that is designed for the problem (whether it is a monolithic model or several small models that need to be integrated to represent the problem) usually includes some simplifications. These simplifications can range from relaxations of constraints to accuracy adjustments of time-consuming exact simulators and to the creation of approximative surrogate models. Hence, as the solver is applied to this model, even if it solves the model to optimality, it still has some deviations from real-world (expected) solution because of the simplification. Also, because of the dynamic nature of the problems in real-world, even if the solver finds the optimum solution,

[8] Private correspondence.

that solution might not be valid anymore after a (possibly short) period of time due to unexpected events, e.g., delay of products, failure of trucks, extreme changes of weather (see also [1, 6] for further discussion).

Multicomponent problems pose new challenges for the theoretical investigations of evolutionary computation methods. The computational complexity analysis of evolutionary computation is playing a major role in this field [32, 33]. Results have been obtained for many NP-hard combinatorial optimization problems from the areas of covering, cutting, scheduling, and packing. We expect that the computational complexity analysis can provide new rigorous insights into the interactions between different components of multicomponent problems. As an example, we consider again the TTP problem. Computational complexity results for the two underlying problems (KP and TSP) have been obtained in recent years. Building on these results, the computational complexity analysis can help to understand when the interactions between KP and TSP make the optimization process harder.

In a similar way, feature-based analysis might be helpful to provide new insights and help in the design of better algorithms for multicomponent problems. Analyzing statistical feature of classical combinatorial optimization problems and their relation to problem difficulty has gained an increasing attention in recent years [34]. Classical algorithms for the TSP and their success depending on features of the given input have been studied in [35–37], and similar analysis can be carried out for the knapsack problem. Furthermore, there are different problem classes of the knapsack problem which differ in their hardness for popular algorithms [38]. Understanding the features of the underlying subproblems and how the features of interactions in a multicomponent problem determine the success of different algorithms is an interesting topic for future research which would guide the development and selection of good algorithms for multicomponent problems.

It seems multicomponent problems provide great opportunity for further research in EC community. Thus, we believe that future research in this direction can potentially close the gap between academic research in EC community and needs for optimization methodologies in industries.

References

1. Michalewicz, Z. (2012). Quo vadis, evolutionary computation? In J. Liu, C. Alippi, B. Bouchon-Meunier, G. Greenwood, & H. Abbass (Eds.), *Advances in Computational Intelligence. Lecture Notes in Computer Science* (vol. 7311, pp. 98–121). Berlin, Heidelberg: Springer.
2. Weise, T., Zapf, M., Chiong, R., & Nebro, A. (2009). Why is optimization difficult? In R. Chiong (Ed.), *Nature-inspired algorithms for optimisation. Studies in Computational Intelligence* (vol. 193, pp. 1–50). Berlin, Heidelberg: Springer.
3. Michalewicz, Z., & Fogel, D. B. (2004). *How to solve it: Modern heuristics*. Berlin: Springer.
4. Jin, Y., & Branke, J. (2005). Evolutionary optimization in uncertain environments–A survey. *IEEE Transactions on Evolutionary Computation*, *9*(3), 303–317.
5. Nguyen, T. T., & Yao, X. (2012). Continuous dynamic constrained optimization–The challenges. *IEEE Transactions on Evolutionary Computation*, *16*(6), 769–786.

6. Michalewicz, Z. (2012). The emperor is naked: Evolutionary algorithms for real-world applications. *ACM Ubiquity*, pp. 1–13.
7. Ackoff, R. L. (1979). The future of operational research is past. *Journal of the Operational Research Society*, *30*(2), 93–104.
8. Wang, S., & Sarker, B. R. (2006). Optimal models for a multi-stage supply chain system controlled by kanban under just-in-time philosophy. *European Journal of Operations Research*, *172*(1), 179–200.
9. Stolk, J., Mann, I., Mohais, A., & Michalewicz, Z. (2013). Combining vehicle routing and packing for optimal delivery schedules of water tanks. *OR Insight*, *26*(3), 167–190.
10. Michalewicz, M., Michalewicz, Z., & Spitty, R. (2010). Optimising the wine supply chain. In *Proceedings of the Fourteen Australian Wine Industry Technical Conference (14 AWITC), Adelaide, Australia*. Citeseer.
11. Bonyadi, M. R., Michalewicz, Z., & Wagner, M. (2014). Beyond the edge of feasibility: Analysis of bottlenecks. In *Simulated Evolution and Learning*, pp. 431–442. Springer Int. Publishing Switzerland.
12. Whitley, D., Rana, S., & Heckendorn, R. (1997). Island model genetic algorithms and linearly separable problems. In D. Corne & J. Shapiro (Eds.), *Evolutionary computing* (Vol. 1305, pp. 109–125). Lecture Notes in Computer Science Berlin, Heidelberg: Springer.
13. Bonyadi, M., Michalewicz, Z., & Barone, L. (2013). The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. In *Congress on Evolutionary Computation*, pp. 1037–1044.
14. Bonyadi, M. R., Michalewicz, Z., Przybylek, M. R., & Wierzbicki, A. (2014). Socially inspired algorithms for the travelling thief problem. In *Genetic and Evolutionary Computation*, pp. 421–428. ACM.
15. Mei, Y., Li, X., & Yao, X. (2014). On investigation of interdependence between subproblems of the travelling thief problem. *Soft Computing*, pp. 1–16.
16. Polyakovskiy, S., Bonyadi, M. R., Wagner, M., Michalewicz, Z., & Neumann, F. (2014). A comprehensive benchmark set and heuristics for the traveling thief problem. In *Genetic and Evolutionary Computation*, pp. 477–484. ACM.
17. Wagner, M., Lindauer, M., Mısır, M., Nallaperuma, S., & Hutter, F. (2018). A case study of algorithm selection for the traveling thief problem. *Journal of Heuristics*, *24*(3), 295–320.
18. Wagner, M., (2016). Stealing items more efficiently with ants: A swarm intelligence approach to the travelling thief problem. In M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, & T. Stützle (Eds.), *Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7–9, 2016, Proceedings* (pp. 273–281). Cham: Springer International Publishing. ISBN: 978-3-319-44427-7. https://doi.org/10.1007/978-3-319-44427-7_25.
19. El Yafrani, M., Martins, M., Wagner, M., Ahiod, B., Delgado, M., & Lüders, R. (2017). A hyperheuristic approach based on low-level heuristics for the travelling thief problem. *Genetic Programming and Evolvable Machines*. ISSN: 1573-7632. https://doi.org/10.1007/s10710-017-9308-x.
20. El Yafrani, M., & Ahiod, B. (2016). Population-based vs. single-solution heuristics for the travelling thief problem. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO' 16, Denver, Colorado, USA* (pp. 317–324). New York, NY, USA: ACM. ISBN: 978-1-4503-4206-3. http://doi.acm.org/10.1145/2908812.2908847.
21. Neumann, F., Polyakovskiy, S., Skutella, M., Stougie, L., & Wu, J. (2017). A fully polynomial time approximation scheme for packing while traveling. *CoRR*. arXiv: 1702.05217.
22. Wu, J., Wagner, M., Polyakovskiy, S., & Neumann, F. (2017). Exact approaches for the travelling thief problem. In *Proceedings of the 11th International Conference on Simulated Evolution and Learning (SEAL)* (p. 12). Shenzhen, China: Springer (Accepted for publication).
23. Bonyadi, M. R., & Moghaddam, M. E. (2009). A bipartite genetic algorithm for multi-processor task scheduling. *International Journal of Parallel Programming*, *37*(5), 462–487.
24. Xiong, J., Liu, J., Chen, Y., & Abbass, H. (2014). A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems. *IEEE Transactions on Evolutionary Computation*, *18*(5), 742–763.

25. Moghaddam, M. E., & Bonyadi, M. R. (2012). An immune-based genetic algorithm with reduced search space coding for multiprocessor task scheduling problem. *International Journal of Parallel Programming*, *40*(2), 225–257.
26. Deb, K. & Sinha, A. (2009). Solving bilevel multi-objective optimization problems using evolutionary algorithms. In *EMO*, pp. 110–124.
27. Legillon, F., Liefooghe, A., & Talbi, E.-G. (2012). Cobra: A cooperative coevolutionary algorithm for bi-level optimization. In *IEEE Congress on Evolutionary Computation*, pp. 1–8.
28. Deb, K., & Sinha, A. (2010). An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary Computation*, *18*(3), 403–449.
29. Potter, M. A., & De Jong, K. A. (1994). A cooperative coevolutionary approach to function optimization. In Parallel Problem (Ed.), *Solving from Nature* (pp. 249–257). London, UK, UK: Springer.
30. Ibrahimov, M., Mohais, A., Schellenberg, S., & Michalewicz, Z. (2012). Evolutionary approaches for supply chain optimisation. Part I: Single and two component supply chains. *International Journal of Intelligent Computing and Cybernetics*, *5*(4), 444–472.
31. Ibrahimov, M., Mohais, A., Schellenberg, S., & Michalewicz, Z. (2012). Evolutionary approaches for supply chain optimisation. Part II: Multi silo supply chains. *International Journal of Intelligent Computing and Cybernetics*, *5*(4), 473–499.
32. Auger, A., Auger, A., & Doerr, B. (2011). *Theory of randomized search heuristics: Foundations and recent developments*. River Edge, NJ, USA: World Scientific Publishing Co., Inc.
33. Neumann, F., & Witt, C. (2010). *Bioinspired computation in combinatorial optimization: Algorithms and their computational complexity* (1st ed.). New York, NY, USA: Springer, New York Inc.
34. Smith-Miles, K., Baatar, D., Wreford, B., & Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research*, *45*, 12–24.
35. Mersmann, O., Bischl, B., Trautmann, H., Wagner, M., Bossek, J., & Neumann, F. (2013). A novel feature-based approach to characterize algorithm performance for the traveling salesperson problem. *Annals of Mathematics and Artificial Intelligence*, *69*(2), 151–182.
36. Nallaperuma, S., Wagner, M., Neumann, F., Bischl, B., Mersmann, O., & Trautmann, H. (2013). A feature-based comparison of local search and the Christofides algorithm for the travelling salesperson problem. In *Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms XII, FOGA XII'13* (pp. 147–160). Adelaide, Australia, New York, NY, USA: ACM. ISBN: 978-1-4503-1990-4. http://doi.acm.org/10.1145/2460239.2460253.
37. Smith-Miles, K., van Hemert, J., & Lim, X. Y. (2010). Understanding tsp difficulty by learning from evolved instances. In International Conference (Ed.), *on Learning and Intelligent Optimization, LION'10* (pp. 266–280). Berlin, Heidelberg: Springer.
38. Martello, S., & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations*. New York, NY, USA: Wiley Inc.

Evolutionary Computing Applied to Solve Some Operational Issues in Banks

Gutha Jaya Krishna and Vadlamani Ravi

Abstract Banking industry is the backbone of the economy of any country, and it does have many operational issues as well as other financial issues. As regards to solving operational issues such as Portfolio optimization, Bankruptcy prediction, FOREX rate prediction, ATM cash replenishment, ATM/Branch location prediction, Interbank payments, liquidity prediction, etc., banking industries are moving away from conventional ways toward more automated and more robust methods. Evolutionary and Swarm Optimization (ESO) based techniques play a vital role in solving the above-mentioned operational issues because they yield global or near-global optimal results. We survey most of the works reported in this space starting from 1998 to 2016. While the application of ESO techniques to solve the business issues is well-documented, the same on the operational issues is very relevant.

Keywords Banks · Evolutionary computing · Operational issues
Swarm intelligence

1 Introduction

Evolutionary computing is a name for a set of optimization techniques propelled by biological evolution, for example, selection and genetic redesigning. Swarm computing motivated by the foraging and flocking conduct of insects, flying creatures, and animals. These methods have discovered a horde of uses in differing disciplines.

G. J. Krishna · V. Ravi (✉)
Center of Excellence in Analytics, Institute for Development and Research in Banking Technology (IDRBT), Castle Hills Road #1, Masab Tank, Hyderabad 500057, India
e-mail: rav_padma@yahoo.com

G. J. Krishna
e-mail: krishna.gutha@gmail.com

G. J. Krishna
School of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, India

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_3

They shape a subset of a greater set called optimization procedures, which incorporate traditional and differentiation-based methods, point-based metaheuristics, for example, recreated simulating-annealing [1], threshold accepting [2], tabu search [3, 4], and so forth.

Evolutionary and Swarm Optimization (ESO) algorithms are population-based random search procedures where a population of solutions is refreshed iteratively utilizing particular heuristics until the point that convergence is accomplished by an optimization problem, regardless of whether it is a continuous or discrete/combinatorial. These systems vary in the way in which the solutions are encoded and the heuristics they use while refreshing the solutions [5]. Genetic Algorithms, differential evolution, particle swarm optimization, ant colony optimization, firefly-based optimization, cuckoo search, artificial bee colony, and the harmony search [6–9] are a portion from this family.

Banks face many problems on the operational front while servicing the customers, where evolutionary computing can be useful. If we consider operational issues, conventional and human-based techniques do not yield comparatively better results than the evolutionary-based techniques. Some of the operational areas of banks that can be listed out where evolutionary computing plays a vital role are given as follows:

- Portfolio optimization,
- Bankruptcy prediction,
- FOREX management,
- Gridlock resolution,
- ATM cash replenishment,
- ATM/Branch location prediction, etc.

Portfolio optimization is implemented by minimizing risk and maximizing the return on bank investments subject to constraints. Portfolio optimization modeled in a bi-objective optimization framework with a weighted sum of objectives as a new single-objective with or without constraints. After modeling the objective function one can apply multi-objective evolutionary algorithms or single-objective evolutionary algorithms or hybrid evolutionary algorithms or even powerful algorithm like memetic algorithm [10, 11] to solve the problem.

The prediction of bankruptcy is paramount in the present day scenario where there is an increasing trend of firms becoming bankrupt. Since the late 90s, machine learning and statistical techniques have caught up the attention of researchers in predicting bankruptcy of a bank or financial firm. Optimization procedures have since a long time ago utilized for training a portion of the DM methods. Nonetheless, amid the previous couple of years, ESO techniques have turned out to be so capable and flexible that we can send them as a substitute for some DM methods. Therefore, ESO techniques alone can be used to predict the bankruptcy [12–15] as it turns out to be a binary classification problem.

Evolutionary computing has two main applications in the FOREX market. One is FOREX rate prediction [16], and other is FOREX trading rule optimization [17, 18]. In both of them, the application of evolutionary computing plays a key role as it tries

to reduce the error of future prediction based on current value. The above modeled as a time series problem, and the evolutionary algorithm can be applied based on lag and embedding dimension, also known as chaotic modeling. To further reduce the error in prediction, hybrid or memetic evolutionary algorithms can be utilized as they are more powerful and versatile.

Gridlock resolution is a problem faced during interbank payments and clearances which usually involve a large amount of money and participants [19]. The above modeled as an optimization problem, which can be solved by evolutionary computing techniques. Clearing volume is the objective function used by the evolutionary algorithms. There are greedy as well as heuristic variants of the above interbank payment clearance systems [20]. But few are reported dealing with evolutionary-based optimization algorithms. So, this is a currently hot and evolving operational area in banking where evolutionary techniques find their place.

For ATM cash replenishment banks want to use fewer resources and also meet fluctuating customer demands [21, 22]. Till now exponential weighted moving average procedure was used to model daily cash replenishment for individual ATMs. Sometimes these approaches do not work well as variations increase. In light of these various evolutionary computing, techniques can be applied to forecast the ATM cash replenishment mechanism based on the current cash demand. The above can be accomplished either in stand-alone mode or conjunction with an intelligent technique.

With the above-mentioned operational aspects of banks, it evident that the application of evolutionary computing techniques will help automate and also solve the business issues which will in turn help banks in gaining a return on investment and also an edge over competitors. With this, we propose that evolutionary computing techniques in the present and future help the banking industry in alleviating their operational issues in conducting business. We also provide suitable future directions in these areas, which will be potential research problems, useful to budding researchers and experienced faculty as well.

2 Theme of the Review Work

The theme of the review work is to present and explore the research articles where the ESO techniques are utilized to solve some of the operational issues of banks. Though some research articles are reviewed where the ESO techniques not utilized in solving the operational issues, these papers present objective functions formulation which in the future one can utilize in ESO techniques.

This chapter's main theme is to discuss how ESO techniques can be utilized to solve the problems better than conventional techniques. Conventional techniques are sometimes helped by ESO techniques to solve the operational issues, and sometimes ESO techniques alone can be used to solve the operational issues. Conventional techniques include machine learning based techniques, expert-based techniques, statistical-based techniques, age-old practices, etc. ESO techniques include point-

based local search metaheuristics and population-based global search metaheuristics techniques.

Only a few of the operational issues of banks considered here, and they are not exhaustive. The analytical issues to solve business problems well-documented, and in [23] some of these analytical issues discussed. So, in this work, we are going to focus on the operational issues of financial firms especially banks.

3 Methodology Used in the Current Work

3.1 Review Process

First, online databases are searched for the research articles. The preference of research articles are journals, book chapters, and conferences. In searching for the research articles, first operational issues are searched with keywords 1. Portfolio optimization, 2. Bankruptcy prediction, 3. FOREX rate prediction, 4. Gridlock resolution, 5. ATM cash replenishment. Next ESO techniques used in solving the above operational issues are searched. The ESO techniques searched for are Evolutionary related, Swarm related, Foraging related, Nature inspired/Process inspired, Local search related, Artificial Immune System related.

3.2 Review Methodology Considered

For this chapter, we consider only ESO techniques which have been applied to solve the operational issues as listed in the above subsection. We have considered both single and multi-objective optimization problems for the review. We have also considered memetic (hybrid) algorithms in our review work.

3.3 Flowchart of Review Methodology and Structure of Considered Techniques

See Figs. 1, 2 and 3.

4 Distribution of Articles

The year wise count of the articles is presented in Fig. 4.

Table 1 gives the references of research articles by operational issues, and Table 2 gives the references of research articles by ESO techniques. These tables also

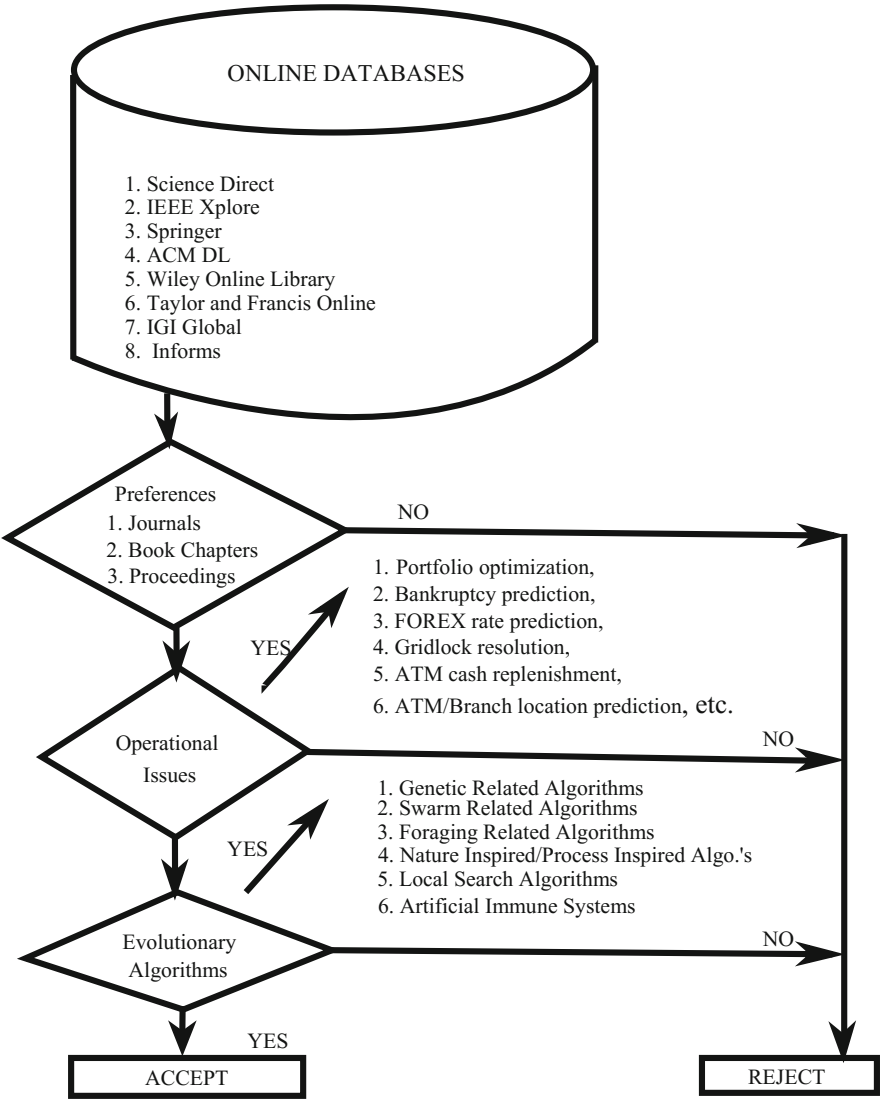


Fig. 1 Flowchart of review methodology

provide a count of research articles by operational issues and by ESO techniques (Figs. 5 and 6).

Table 1 Distribution of research articles by operational issue

Operational issue	References	Count
Portfolio optimization	[11, 24–31]	9
Bankruptcy prediction	[12–15, 32–35]	8
FOREX management	[16–18, 36–38]	6
Gridlock resolution	[19, 20]	2
ATM cash replenishment	[21, 22]	2

Table 2 Distribution of research articles by techniques

ESO technique	References	Count
Genetic algorithm (GA)	[12, 14, 17, 18, 24, 27, 28, 31, 32, 35, 37]	12
Genetic programming (GP)	[13, 16, 29, 33, 36]	5
Particle swarm pptimization (PSO)	[24, 34, 35]	3
Strength pareto evolutionary algorithm-2 (SPEA-2)	[25, 26, 30]	3
Genetic network programming (GNP)	[27–29]	3
Ant colony optimization (ACO)	[15, 34]	2
Non-dominated sorting GA (NSGA-II)	[25, 26]	2
Simple multi-objective-PSO (SMPSO)	[26, 30]	2
Bacterial foraging (BF)	[24]	1
Tabu search (TS)	[34]	1
Memetic algorithm (MA)	[10]	1
Vector evaluated GA (VEGA)	[30]	1
Multi-objective GA (MOGA)	[30]	1
General differential evolution-3 (GDE3)	[26]	1

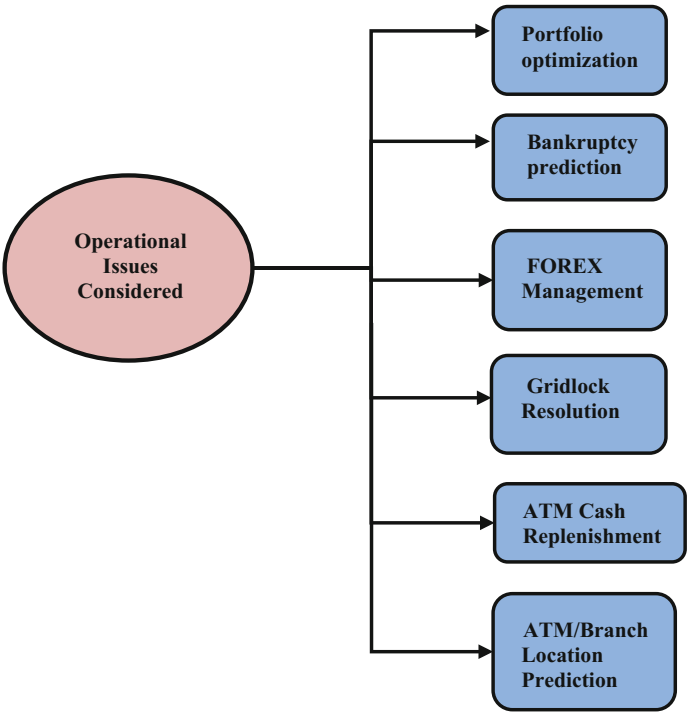


Fig. 2 Types of operational issues considered

5 Review of Articles

The current study focuses on five operational issues in banks listed below in five subsections with a review of articles in each subsection.

5.1 Portfolio Optimization

Portfolio optimization (PO) is the problem of allocation of the budget for various assets to maximize expected return on the assets and minimize corresponding financial risk.

First, the portfolio optimization algorithm has been solved using MOEA, namely, VEGA, MOGA, SPEA-2, NSGA-II with cardinality constraints, floor constraints and round lot constraints [30]. In the proposed work one of the objective is the maximization of the expected yield of the portfolio and at the same time the other objective is the minimization yield variance subject to the above-mentioned constraints. The inclusion of the above constraints makes the portfolio more realistic apart from the

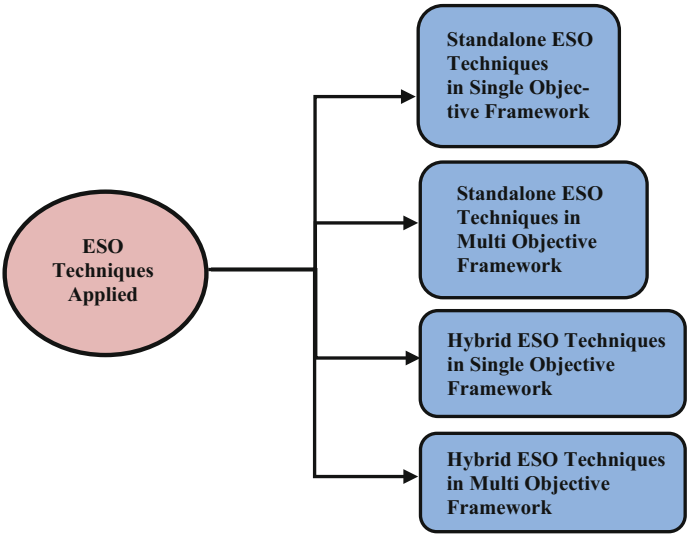
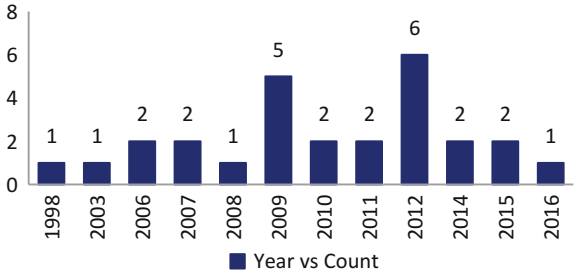


Fig. 3 Types of ESO techniques applied

Fig. 4 Distribution of articles by year



complexity it brings into solving the portfolio problem. The use of Pareto front-based MOEA techniques is that they give flexibility in choosing the optimal solutions on the Pareto front according to the decision makers choice. Results show that MOGA and SPEA-2 perform comparatively better than other utilized MOEA techniques though both the above ESO techniques are complex. SPEA-2 performs better in terms of smaller number of generations and also better Pareto fronts.

Another facet is GNP, where a multi-brands portfolio optimization algorithm utilizing GNP with control nodes (GNPcn) was proposed in [28]. The application of GNP over Neural Network (NN) is that NN cannot give us the relationship between inputs and outputs. This disadvantage of NN, which are black-boxes, made the authors of [28] choose GNP, which is an extended method compared to GA. GNP with control nodes gives efficient trading rules apart from the optimal solution. The proposed method also has higher profitability compared to GNP, GA, and the Buy and Hold methods for the portfolios generated.

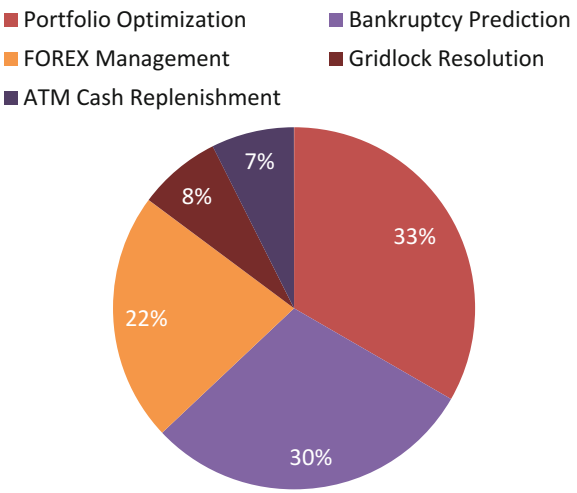
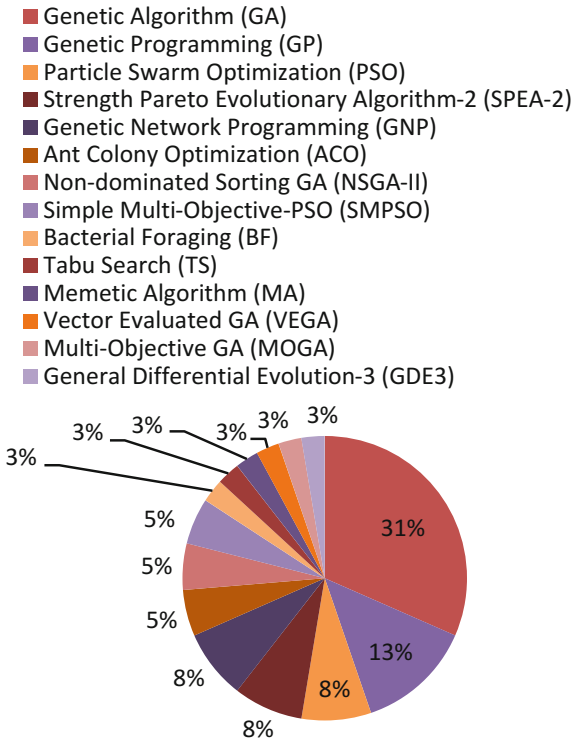


Fig. 5 Distribution of papers by operational issues

Fig. 6 Distribution of ESO techniques



Also, a time adapting GNP (TA-GNP) proposed [27] to solve portfolio optimization and compared with the traditional GNP, GA, and the Buy and Hold methods. First, TA-GNP has many control nodes and each group of control nodes assigned to each a stock brand. Second, TA-GNP can adapt to changing stock prices. By the above two, effective portfolios are generated and are better than the compared methods. The main advantage of TA-GNP is that it evolves through time, i.e., as markets change it also adapts through the changing markets.

Next, a large-scale portfolio optimization problem has been solved using Genetic Relation Algorithm with guided mutation (GRA/G) as an operator [29]. The guided mutation is performed based on the average value of the correlation coefficient of the individuals of the offspring. The effectiveness of the portfolio generated has been confirmed by comparing with the GNP. The advantage of utilizing the GRA/G is faster convergence and increased profits than Buy and Hold. Another advantage is that large-scale portfolio optimization problems can be solved with GRA/G.

Portfolio optimization is basically nonlinear, NP-hard and also has a lot of local optima. The recent ESO techniques have become very strong and versatile and one out of this family is BFO. Though BFO has many parameters, it tends to converge to the optima very fast and is also very robust to multiple local optima. Therefore, a bacterial foraging optimization (BFO) based ESO technique has been applied to solve the portfolio optimization model by introducing a factor called liquidity risk [24]. By considering liquidity risk, the PO problem becomes even more complex where the use of BFO makes it solvable even with the high complexity. In solving the portfolio optimization problem, a variant of BFO with linearly decreasing chemotaxis (BFO-LDC) is incorporated. The proposed BFO-LDC [24] outperforms BFO, PSO, and GA-based portfolio optimization methods in terms of less iterations and better fitness values in terms of more yield as well as less risk.

Further, the portfolio optimization problem was considered by applying with greedy coordinate mutation (GCM) operator on GA with the inclusion of the trading volumes [31]. The proposed method uses a real as well as binary representation of the individuals. The objective function used is Sharpe's Ratio which mathematically combines risk and return. The proposed method also incorporates trading volumes by which the news (e.g., Political news) can be also be taken into consideration. Trading volume captures the prices of the assets. The proposed method was compared with the simple GA, and the proposed method is better than the simple GA and portfolio without news in terms of less number of assets with significant weights. The advantage of the GCM operator is that it acts as local search by applying mutation in two stages.

Furthermore, the age-based third objective for the solution obtained is included, which has to be maximized apart from the two objectives namely the return and the risk maximized and minimized, respectively [26]. For solving the above portfolio optimization problem NSGA-II, GDE3, SMSPO, and SPEA-2 are utilized. The data is resampled at each time frame and applied to the MOEA. The reason is for using the above approach is to capture dynamic and real-world scenarios. The results were compared in both, without time-stamped and with time-stamped models. The time-stamped model gave better results than without time-stamped models. The main

advantage of using standard MOEA techniques is that they are more stable and robust. Also, a different Pareto front validation measures have been utilized like estimation error, stability, extreme risk and unrealized returns which make the solutions more robust.

On the other hand, a Probe Guided Mutation (PGM) Operator proposed for solving the cardinality constrained portfolio optimization problem [25]. The PGM operator is an extension of the Polynomial Mutation (PLM) in the MOEA setting. The proposed method outperforms earlier polynomial mutation operator in Multi-Objective Evolutionary Algorithm (MOEA) setting, with NSGA-II and SPEA-2 utilized. The PGM performed better than PLM in terms of the runtime of convergence. Also, the PGM operator performed better than PLM operator, and exact efficient point methods of Mixed-Integer Quadratic Program (MIQP), CPLEX in terms of approximate efficient point generated. The MOEA solutions generated are evaluated using standard metrics, i.e., Hypervolume, Spread, and Epsilon Indicator.

In addition, a memetic approach has been proposed [10] to solve portfolio optimization problem of the standard Markowitz's mean-variance model with constraints on the limit of investment of assets and with minimum trading restrictions. The above-described problem transforms into a mixed-integer quadratic problem as transactions costs and limit of some assets is considered. For solving this mixed-integer quadratic problem a memetic technique which combines GA and quadratic programming was considered. The proposed technique is compared with other investment strategies like passive management, the equally weighted portfolio management, minimum variance portfolio management and optimal portfolios without cardinality constraints ignoring transaction costs or obtained with L_1 regularization. It is concluded that stable and robust portfolios have been obtained compared to other compared methods (Table 3).

In summary, we observe the following:

- GA along with various operators has often been utilized in solving the portfolio optimization.
- The above ESO techniques are compared with conventional (like Buy and Hold) as well as other ESO-based techniques such as GA, GNP, etc.

5.2 Bankruptcy Prediction

Bankruptcy prediction of the firms, whether corporate or public is of increasing interest to the investors, borrowing firms, creditors and governments alike. In time identification of the firms which are going to get bankrupt is indeed very desirable.

There are many quantitative data mining approaches for bankruptcy prediction and very few on qualitative decision rules. Especially, qualitative decision-making approach for bankruptcy prediction has been proposed using GA instead of the quantitative approaches in [12]. In this proposed method [12] rules are generated apart from reporting the performance metrics. The GA-based approach compared with inductive

Table 3 Comparison of ESO techniques (Portfolio optimization)

References	ESO technique applied, objective function utilized, results	Comparison with techniques, results
[30]	• VEGA, MOGA, SPEA-2, NSGA-II • ***Maximize portfolio expected yield minimize portfolio variance of the yield Subject to:*** ***1.*** $\sum_{i=1}^{n} x_i$ ***and*** $x_i \geq 0$ ***where x's are 'n' assets*** ***2. Cardinality constraint (Maximum and mnimum no. of assets)*** ***3. Floor constraints (Lower limit of the assets)*** ***4. Round lot constraint (Making multiple of normal trading lot)*** • SPEA-2 gave better results	Not compared with conventional techniques
[28]	• GNP with control nodes (GNPcn) • ***Maximize profit*** • GNPcn (4,262,714 Yen)	Technique (Avg. Profit) • GNP-RL (292,716 Yen) • GNP-Candlestick (98,302 Yen) • GA (−41,585 Yen) • Buy and hold (−136,500 Yen)
[27]	• Time adapting—GNP (TA-GNP) • ***Maximize profit*** • TA-GNP (6,962,135 Yen)	Technique (Avg. Profit) • GNP-RL (292,716 Yen) • GNP-Candlestick (98,302 Yen) • GA (−41,585 Yen) • Buy & Hold (−136,500 Yen)
[29]	• GRA/G & GNP • ***Maximize profit*** • GRA/G & GNP (5,275,248 Yen)	Technique (Avg. Profit) • Buy and hold (3,566,217 Yen)
[24]	• BFO-LDC, BFO, PSO, and GA • ***Minimize*** $\lambda\left[2.33\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_{ij}} - \sum_{i=1}^{n} x_i r_i + 0.5\left(\overline{WAV} + 2.33\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma'_{ij}}\right)\right] - (1-\lambda)\sum_{i=1}^{n} x_i r_i$ ***Subject to:*** $\sum_{i=1}^{n} x_i$ ***and*** $x_i \geq 0$ ***where x's are 'n' assets*** • BFO-LDC gave good results	Not compared with conventional techniques

(continued)

Table 3 (continued)

References	ESO technique applied, objective function utilized, results	Comparison with techniques, results
[31]	• GA-GCA, GA-GCA with no news, simple GA • ***Maximize sharpe ratio*** • Dow-Jones: number of assets with significant weights (Technique) 1. 5 (GA-GCA) 2. 5 (GA-GCA with no news) 3. 10 (Simple GA) • NASDAQ: number of assets with significant weights (Technique) 1. 12 (GA-GCA) 2. 14 (GA-GCA with no news) 3. 21 (Simple GA)	Not compared with conventional techniques
[26]	• SPEA-2 (R + T), SPEA-2, NSGA-II (R + T), NSGA-II, SMPSO (R + T), SMPSO, GDE3, GDE3 (R + T) • ***Optimize estimation error, stability, unrealized returns, extreme risk*** • SPEA-2 (R + T) gave good results	Not compared with conventional techniques
[25]	• SPEA-2 (PGM, PLM), NSGA-II (PGM, PLM) • ***Optimize hypervolume, spread, and epsilon indicator*** • PGM operator gave good results for both SPEA-2 and NSGA-II	Not compared with conventional techniques
[10]	• Combines GA and quadratic programming • ***Maximize: turnover, returns, sharpe ratio*** • Proposed hybrid yield good results	Compared with: • Minimum variance portfolio • No cardinality constraints • LASSO • With and without transaction costs

learning (IL) and neural network (NN) based approaches. GA-based approach performed better than the remaining two compared approaches. Less number of rules are generated by the proposed approach than the IL and the NN-based approaches. Also the accuracy of the proposed GA-based technique is better than IL and NN. This kind of rule-based decision-making approach aids the decision makers in making better decisions.

One main advantage of using GP is that it generates a GP-tree for the individual representation which is flexible for the rule representation. Several mathematical operators can be incorporated into the tree by which better search space traversing can happen. However, in the proposed approach, the Spanish companies' data has been utilized to predict bankruptcy using GP, and the proposed technique compared with the SVM [33]. The proposed method [33] outperforms the compared SVM-

based method with many hits, true positive rate and true negative rate. The main advantage of the proposed approach is that it can be applied highly unbalanced databases.

Corporate bankruptcy prediction will be helpful to the governments, investors as well as creditors, etc., because predicting firms which will get bankrupt in future will help all the interested in the firm alike a warning sign. Here, in the proposed technique for bankruptcy prediction [13], GP was utilized for prediction of the bankruptcy of Iranian firms, as the GP-tree produced generates the discriminating rules. The proposed method [13] compared with Multiple Discriminant Analysis (MDA), and the proposed method performs better than the compared method with pretty good metrics. The MDA generates a linear discriminating equation for bankrupt and non-bankrupt firms which are compared with the rules from GP and GP produces better prediction accuracies than MDA.

Apart from GA and GP, there are other good ESO metaheuristics that can be applied to the task of bankruptcy prediction. In the proposed method [34], ACO and PSO were utilized for the bankruptcy prediction. Also, feature selection was performed on the financial task of credit risk assessment using the ACO and PSO [34]. Feature selection helps in the process of finding important and useful features from large number of feature set. Apart from the above, feature selection also improves the metrics considered for the evaluation. The metrics used for comparison are overall classification accuracy, root means square error and an average number of features. The proposed methods are in turn compared with TS and GA, and the proposed methods outperformed the compared with the above-mentioned performance metrics.

Due to the banks' preference toward a cost-sensitive classifier which is much in demand and also classifier friendly to the decision makers, the discussed approach is very useful. So, GA was utilized simultaneously for obtaining optimal features as well as parameter estimation of the model considered for bankruptcy prediction with the cost consideration [14]. In cost-sensitive learning approach, the weights of vector quantization are optimized along with minimal feature length. The proposed methodology [14] obtained optimal features with better performance in terms of expected misclassification cost. Further, the advanced genetic operators can be used along with ensemble-based methods to improve the proposed method.

In addition, a GA-based coverage optimization is utilized for the ensemble of classifiers on bankruptcy data of the Korean firms [32]. The diversity problem of ensemble learning is addressed using GA with summation of coverage rate of the classifiers as the objective with Variance Inflation Factor (VIF) as the constraint, to address multi-collinearity among the picked classifiers. The model is generated by the method compared with the stand-alone Decision Tree (DT), Neural Network (NN), and Support Vector Machine (SVM). The proposed ensemble of classifiers using GA performs comparatively better than the stand-alone machine learning techniques. The advantage is that bagging and boosting of ensemble learning is improved using a GA-based ensemble.

Further, a hybrid method combining adaptive PSO with SVM was proposed on American banking industry data [35]. The proposed hybrid outperforms the traditional machine learning techniques compared in the research article in terms of

accuracy, sensitivity, specificity, and F-measure. The compared techniques include Logistic Regression (LR), Naïve Bayes (NB), Decision Tree (DT-ID3, C4.5), ANN, etc. Also, various inertia weight variants of PSO were compared like Adaptive Inertia Weight (AIW), Fixed Inertia Weight (FIW), Linear Decreasing Inertia Weight (LDIW), and Nonlinear Decreasing Inertia Weight (NDIW). Out of all the compared variants of inertia weight, AIW gave good results on the standard optimization benchmark function.

On the other hand, ACO has been used with weighted fitness function of both Accuracy (ACC) and Area Under the Curve (AUC) for the bankruptcy prediction [15]. ACO was used to develop a discrimination model which can work on continuous data without the use of data discretization. The proposed method compared with various Machine Learning (ML) techniques like Random Forest (RF), J48, Multi-Layer Perceptron (MLP), etc. The devised technique outperformed the above-mentioned conventional machine learning techniques in terms of both performance metrics, i.e., AUC and ACC. The above ACO-based model was run on top of various attribute selection techniques like Gain Ratio-based, Chi Square-based, Correlation-based, Wrapper-based, etc. When compared with the above attribute selection methods, ACO-based technique outperformed all the compared ML techniques (Table 4).

In summary, we observe the following:

- GA has often been utilized to solve the bankruptcy prediction problem.
- The above ESO techniques are compared with conventional (like MDA, RF, J48, MLP, SVM, etc.) as well as ESO-based techniques such as ACO, PSO, etc.

5.3 FOREX Management

Two main aspects where optimization can be employed is in finding optimal trading rules for the FOREX market and FOREX rate prediction.

Initially, GP was utilized in combination with trigonometric operators and high-order statistics [16]. The performance was compared with GP, traditional function sets, stand-alone ARMA models and the Buy & Hold on the performance metrics like mean square error (MSE), hits, profit, etc. The proposed method gave comparatively better results than the compared in terms of better profit percentage, high hit percentage and less MSE.

In addition to the above, the effect of foreign exchange management on the firm's performance is analyzed using GA and user has to specify various criteria like Value at Risk (VaR), FOREX trading condition, conditions on dealing amount, and GA operator as well as penalty [37]. Management of FOREX risk is involved when trading in different currencies by the firm. VaR depicts the maximum potential loss for a given period. Various conditions like Non-constraint condition, maximum holding condition and maximum VaR (currency are sold when maximum criteria are reached) are considered in the proposed approach. The proposed approach provides

Table 4 Comparison of ESO techniques (Bankruptcy prediction)

References	ESO technique applied, objective function utilized, results	Comparison with conventional techniques, results
[12]	• GA • ***Maximize (Predictive accuracy (Rule) + Coverage (Rule))*** • Technique (Number of rules, coverage %, average accuracy %, overall accuracy %) GA (11, 18.5, 93.3, 94.0)	Technique (Number of rules, coverage %, average accuracy %, overall accuracy %) • IL (16, 15.3, 87.7, 89.7) • NN (12, 15.6, 88.4, 90.3)
[33]	• GP • ***Fitness:*** $Maximize \sum_{i=1}^{n} U_i$ ***1 for correct classification and 0 for wrong classification*** • Technique (Data) (Overall: hits %, TP %, TN%) • Best-GP (1999) (79.44, 81.48, 79.32) • Avg-GP (1999) (83.94, 62.04, 85.28) • Best-GP (2000) (79.44, 81.48, 79.32) • Avg-GP (2000) (83.24, 62.33, 84.63)	Technique (Data) (Overall: hits %, TP %, TN%) • Best-SVM (1999) (68.74, 92.59, 62.27) • Avg-SVM (1999) (68.80, 91.48, 67.41) • Best-SVM (2000) (69.10, 93.33, 67.68) • Avg-SVM (2000) (67.95, 91.00, 66.41)
[13]	• GP • ***Fitness: maximize number of hits*** • Holdout: accuracy (Bankrupt %, non-bankrupt %) GP- (95, 84)	• Holdout: accuracy (Bankrupt %, non-bankrupt %) MDA- (71, 74)
[34]	• ACO, PSO, GA, TS • ***Fitness: maximize overall accuracy and minimize root mean square error*** • ACO and PSO performed better	Not compared with conventional techniques
[14]	• vGA • ***Fitness: reduce estimated misclassification cost*** • GA performed better than ML techniques	• J4.8, MLP, SVM, KNN
[32]	• GA • ***Fitness: maximize coverage rate with VIF as a constraint*** • Coverage optimized bagging and boosting (Accuracy) • DT (76.20, 76.00) • NN (76.92, 76.52) • SVM (77.23, 77.53)	• Individual, bagging and boosting (Accuracy) • DT (70.30, 75.78, 75.10) • NN (71.02, 73.97, 73.10) • SVM (72.45, 72.85, 73.07)

(continued)

Table 4 (continued)

References	ESO technique applied, objective function utilized, results	Comparison with conventional techniques, results
[35]	• Adaptive PSO—SVM • ***Fitness: overall accuracy, precision, F-measure are all maximized*** • Adaptive PSO-SVM (96.97, 94.44, 97.14)	Technique (overall accuracy, precision, F-measure) • LR (72.73, 75, 72.72) • NB (78.79, 81.25, 78.78) • DT (78.79, 77.77. 80) • ANN (75.76, 82.35, 77.77) • SVM (81.82, 83.33, 83.33)
[15]	• ACO • ***Fitness: maximize*** $C_1 * ACC + C_2 * AUC$ • Dataset (AUC, ACC) • Dataset-I (97.14, 89.2) • Dataset-II (88.81, 77.2) • Dataset-III (91.28, 81.5)	• Compared with random forest (RF), MLP, NB, J48, etc. Apart from Dataset-I especially in case of RF, the results of the compared are better for other datasets

realistic dealing history conforming to the company's financial circumstances and also compatible with the user's requirements.

Next, GA has been used to find the optimal set of trading rules for the FOREX market [17]. Various technical indices like Relative Strength Index (RSI), Moving Averages (MA) and Exponentially Weighted Moving Averages (EWMA) are utilized to generate the trading rules for the FOREX market. The trading rules are optimized using GA. The performance metric is profit, and the proposed method has measured significant profit for the test time series. The proposed method was also compared with NN and NN-based trading rule generator was inferior to the GA-based trading rule generator. It is also compared with standard methods like Buy and Hold and no Leverage and GA-based method outperformed the compared.

Apart from the above ESO techniques, a Linear GP (LGP) is used to find the optimal currency trading in the foreign exchange market with consideration of trading strategies, trends, and profitability [36]. The advantage of using LGP over GP is that it is not tree-like (i.e., linear) and easy to use than traditional GP. The analysis of profitability shows that LGP system achieves both by buying in profit and prevent selling loss. The advantage of this model is in terms of maximizing profits, i.e., maximizing drawdown and less emphasis on winning trades. The implementation will be useful in multiple prior day trading.

Also, Mendel's principle-based GA is used to determine exchange rates under uncertainty [38]. The fitness function used is coefficient of determination of the regression analysis. The use of Mendel's operator after selection is to boost the local search capability of the GA by amplifying unstable solutions generated from the

mutation process, i.e., where good solutions are not lost by the mutation operation. By using this above principle convergence happens at faster rate as good solutions are not lost. The Mendel-GA compared with Standard GA (SGA) and Ordinary Least Squares (OLS), and Mendel-GA outperformed both the compared regarding the exchange rate determination.

Moreover, FOREX trading rules generated with a profitability measure using GA and the proposed technique compared with a Neural Network (NN) based trading rule generator [18]. In the proposed approach each solution represents ten set of trading rules (5 representing opening position +5 representing closing position) with 31 parameters encoded as genes. The fitness function used in the proposed approach is Stirling's Ratio. The profit and risk associated with the trading rule is assessed using the fitness function considered. The proposed method outperforms NN-based trading rule generator regarding profitable trading rules (Table 5).

In summary, we observe the following:

- GP and GA are often utilized to solve the FOREX Management problem.
- The above ESO techniques are compared with conventional techniques (like ARMA, Buy and Hold, NN, VaR).

5.4 Gridlock Resolution

Interbank payment clearance also called Gridlock Resolution is the hot topic where ESO techniques find a potential application.

Initially, interbank payments clearing system modeled as discrete optimization algorithm where clearing volumes are the objective function and deposits of participants are the limiting resources [20]. Though ESO technique has not been utilized, the way of modeling is a good candidate for applying ESO technique. Later, interbank payment clearing system is developed using discrete optimization using a fast heuristic [19] where ESO finds a potential application area.

5.5 ATM Cash Replenishment

ATM cash replenishment is also a hot topic where ESO can be applied. It is where the optimal amount of money to be replenished in ATMs give banks a potential lift as interest paid to the replenishing company will be reduced, and more importantly, the customer dissatisfaction levels reduced as there will be fewer ATMs with the money depleted.

The ATM cash replenishment problem addressed in [21, 22] but not in ESO-based approach. These research articles provide a basis for the objective function formulation and the application of ESO techniques.

Table 5 Comparison of ESO techniques (FOREX management)

References	ESO technique applied, objective function utilized, results	Comparison with conventional techniques, results
[16]	• GP with trigonometric operators and high-order statistics • ***Fitness: minimize MSE*** • Proposed method: (MSE, hits %, Profit %) British Pound (0.2434, 55.23, 88.78) Japanese Yen (0.2215, 53.00, 79.78)	(MSE, hits %, profit %) • ARMA British Pound (–, 50.98, 14.83) Japanese Yen (–, 49.41, −19.63) • Buy and hold British Pound (–, –, 12.15) Japanese Yen (–, –, −20.71)
[37]	• GA • ***Fitness: maximize sum of exchanged amount at the last business day*** • Compared with various user-selected criteria	Not compared with conventional techniques
[17]	• GA • ***Fitness: technical indices like RSI, MA, EWMA used to maximize profit*** • GA-based method gave better profit percentage than the compared	• Compared with NN, buy & hold, no leverage
[36]	• Linear GP (LGP) • ***Fitness: maximize drawdown*** • LGP-based method compared with various fitness cases	Not compared with conventional techniques
[38]	• Mendel's GA, SGA • ***Minimize: coefficient of determination*** • Better results than the compared	Means of the two datasets are compared with OLS. Mendel's GA outperforms OLS
[18]	• GA • ***Fitness: maximize stirling ratio*** • Results are compared with the rolling window-based technique's results	Not compared with conventional techniques

5.6 ATM/Branch Location Prediction

No work is reported in the area of ATM/Branch location prediction in the journals or publishers considered for the review. This also indicates that it is a fertile area for research.

6 Discussion and Future Directions

In the following, the discussion of the review and future directions is presented as follows:

- It is observed that numerous operational issues are solved using GAs. Therefore, other ESO techniques can be explored for solving the operational issues listed above.
- The operational issues listed above are not exhaustive. Other operational issues such as interest rate prediction and ATM/branch location prediction can be solved in future using ESO techniques.
- ESO techniques obviously take more computational time because of parameter tuning for the ESO techniques and repeated runs required to nullify the effect of randomization. However, the optimal or near optimal results obtained by ESO techniques make them more attractive for the firms than conventional techniques.
- There are some operational issues like ATM cash replenishment, gridlock resolution, etc., where ESO techniques were least exploited. Therefore, in these areas, ESO techniques find an immediate application, as these are primarily optimization problems.
- The newer realm of techniques such as quantum-inspired Evolutionary techniques to find a lot of promise to solve these problems.
- Further, many of these problems can be formulated as multi-objective optimization problems, which will pave the way for the immediate adoption of evolutionary multi-objective (EMO) algorithms to these areas.
- Fuzzy optimization finds a direct application as some of these problems can be reformulated to model the inherent fuzziness in the objectives and/or constraints.
- The Return on Investment (ROI) of the banks/financial firms will increase as seen in the above review work by considering profitability as one factor.
- This survey shows that conventional techniques which include machine learning based techniques, expert-based techniques, statistical-based techniques, age-old practices etc. which can be replaced with ESO techniques which offer global optimal or near-global optimal results.
- The survey also shows the utilization of optimization operators alongside the use of ESO techniques. These operators like greedy coordinate mutation operator, guided mutation operator, trigonometric operators, etc., speed up the execution time and in future, even better operators can be proposed with the ESO techniques.
- The trend of application of ESO techniques to the operational issues considered has gone up in 2009 and 2012 (in Fig. 4). This trend will increase in future as researchers and practitioners are moving toward ESO techniques for solving their problems.
- In the age of automation, ESO-based techniques provide a good way of automating the conventional methods. ESO techniques are a part of broad areas of soft computing and AI. Although many believe that ESO-based automation of the operational issues (e.g., ATM Cash Replenishment, Gridlock Resolution, FOREX Management, etc.) will be detrimental for the employability of humans but, it

offers a greater depth into the profitability of the firms and also in making reliable decisions for the banks/financial firms.
- ESO techniques can be employed to generate rules which is not possible in case of some conventional black-box techniques (e.g., Artificial Neural Networks, Deep Learning, etc.) of machine learning.
- ESO-based techniques tend to increase the managerial satisfaction levels of banks/financial firms, by offering the best-fit solutions to the operational issues. Examples of these can be seen in the operational issues of ATM Cash Replenishment, Gridlock Resolution, ATM/Branch Location prediction, Interest rate prediction, etc. In all the above operational issues, the customer satisfaction levels also improve greatly with the use of ESO techniques.
- This survey presents some objective function formulations which may be utilized in the future with even better, strong and versatile ESO techniques.

7 Conclusions

In this article, we present a survey of applications of ESO techniques for solving the operational issues of the banks such as Portfolio Optimization, Bankruptcy Prediction, FOREX rate prediction, Gridlock resolution, ATM cash replenishment, etc. Current review work would be very useful to the bankers, financial firms as well as academicians and researchers working in the domains of Banking, Financial Services, and Insurance.

References

1. Kirkpatrick, S., Gelatt, C., Jr., & Vecchi, M. (1994). Optimization by simulated annealing. *Science, 80*(220), 76–86.
2. Dueck, G., & Scheurer, T. (1990). Threshold accepting: A general purpose optimization algorithm. *Journal of Computational Physics, 90,* 161–175.
3. Glover, F. (1989). Tabu search Part I. *INFORMS Journal on Computing, 1,* 190–206.
4. Glover, F. (1989). Tabu search—Part II. *ORSA Journal on Computing, 2*(1), 4–32.
5. Back, T., Fogel, D., & Michalewicz, Z. (1997). *Handbook of evolutionary computation*. IOP Publishing Ltd.
6. Golberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley.
7. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *1995 IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942–1948).
8. Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization,* 341–359.
9. Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. In *Proceeding European Conference on Artificial Life* (pp. 134–142).
10. Ruiz-Torrubiano, R., & Suarez, A. (2015). A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. *Applied Soft Computing, 36,* 125–142.

11. Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2013). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation, 17,* 321–344.
12. Kim, M. J., & Han, I. (2003). The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. *Expert Systems with Applications, 25,* 637–646.
13. Etemadi, H., Anvary Rostamy, A. A., & Dehkordi, H. F. (2009). A genetic programming model for bankruptcy prediction: Empirical evidence from Iran. *Expert Systems with Applications, 36,* 3199–3207.
14. Chen, N., Ribeiro, B., Vieira, A. S., et al. (2011). A genetic algorithm-based approach to cost-sensitive bankruptcy prediction. *Expert Systems with Applications, 38,* 12939–12945.
15. Lalbakhsh, P., & Chen, Y.-P. P. (2015). TACD: A transportable ant colony discrimination model for corporate bankruptcy prediction. *Enterprise Information systems,* 1–28.
16. Schwaerzel, R., & Bylander, T. (2006). Predicting currency exchange rates by genetic programming with trigonometric functions and high-order statistics. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation—GECCO'06* (p. 955). New York, New York, USA: ACM Press.
17. Mendes, L., Godinho, P., & Dias, J. (2012). A forex trading system based on a genetic algorithm. *Journal of Heuristics, 18,* 627–656.
18. Hirabayashi, A., Aranha, C., & Iba, H. (2009). Optimization of the trading rule in foreign exchange using genetic algorithm. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation—GECCO'09* (p. 1529). New York, USA: ACM Press.
19. Shafransky, Y. M., & Doudkin, A. A. (2006). An optimization algorithm for the clearing of interbank payments. *European Journal of Operational Research, 171,* 743–749.
20. Güntzer, M. M., Jungnickel, D., & Leclerc, M. (1998). Efficient algorithms for the clearing of interbank payments. *European Journal of Operational Research, 106,* 212–219.
21. Ekinci, Y., Lu, J.-C., & Duman, E. (2015). Optimization of ATM cash replenishment with group-demand forecasts. *Expert Systems with Applications, 42,* 3480–3490.
22. Ágoston, K. C., Benedek, G., & Gilányi, Z. (2016). Pareto improvement and joint cash management optimisation for banks and cash-in-transit firms. *European Journal of Operational Research, 254,* 1074–1082.
23. Krishna, G. J., & Ravi, V. (2016). Evolutionary computing applied to customer relationship management: A survey. *Engineering Applications of Artificial Intelligence, 56,* 30–59.
24. Niu, B., Fan, Y., Xiao, H., & Xue, B. (2012). Bacterial foraging based approaches to portfolio optimization with liquidity risk. *Neurocomputing, 98,* 90–100.
25. Liagkouras, K., & Metaxiotis, K. (2014). A new Probe Guided Mutation operator and its application for solving the cardinality constrained portfolio optimization problem. *Expert Systems with Applications, 41,* 6274–6290.
26. García, S., Quintana, D., Galván, I. M., & Isasi, P. (2012). Time-stamped resampling for robust evolutionary portfolio optimization. *Expert Systems with Applications, 39,* 10722–10730.
27. Chen, Y., Mabu, S., & Hirasawa, K. (2010). A model of portfolio optimization using time adapting genetic network programming. *Computers & Operations Research, 37,* 1697–1707.
28. Chen, Y., Ohkawa, E., Mabu, S., et al. (2009). A portfolio optimization model using genetic network programming with control nodes. *Expert Systems with Applications, 36,* 10735–10745.
29. Chen, Y., Mabu, S., & Hirasawa, K. (2011). Genetic relation algorithm with guided mutation for the large-scale portfolio optimization. *Expert Systems with Applications, 38,* 3353–3363.
30. Skolpadungket, P., Dahal, K., & Harnpornchai, N. (2007). Portfolio optimization using multiobjective genetic algorithms. In *2007 IEEE Congress Evolutionary Computation* (pp. 516–523). IEEE.
31. Soam, V., Palafox, L., & Iba, H. (2012). Multi-objective portfolio optimization & rebalancing using genetic algorithms with local search. In *2012 IEEE Congress Evolutionary Computation* (pp. 1–7). IEEE.
32. Kim, M.-J., & Kang, D.-K. (2012). Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction. *Expert Systems with Applications, 39,* 9308–9314.

33. Alfaro-Cid, E., Sharman, K., & Esparcia-Alcázar, A. I. (2007). A genetic programming approach for bankruptcy prediction using a highly unbalanced database. In *Application evolutionary computing* (pp. 169–178). Berlin, Heidelberg: Springer.
34. Marinakis, Y., Marinaki, M., Doumpos, M., & Zopounidis, C. (2009). Ant colony and particle swarm optimization for financial classification problems. *Expert Systems with Applications, 36,* 10604–10611.
35. Shie, F. S., Chen, M.-Y., & Liu, Y.-S. (2012). Prediction of corporate financial distress: An application of the America banking industry. *Neural Computing and Applications, 21,* 1687–1696.
36. Wilson, G., & Banzhaf, W. (2010). Interday foreign exchange trading using linear genetic programming. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation—GECCO'10* (p. 1139). New York, USA: ACM Press.
37. Kim, H., & Hur, S. (2009). Effect of foreign exchange management on firm performance using genetic algorithm and VaR. *Expert Systems with Applications, 36,* 8134–8142.
38. Chen, Y., & Zhang, G. (2013). Exchange rates determination based on genetic algorithms using mendel's principles: Investigation and estimation under uncertainty. *Information Fusion, 14,* 327–333.

Evolutionary Computation for Theatre Hall Acoustics

Cemre Cubukcuoglu, Ayca Kirimtat, Berk Ekici, Fatih Tasgetiren and P. N. Suganthan

Abstract Architectural design is a process that considers many objectives to satisfy. In general, these objectives are conflicting with each other. On the other hand, many design parameters are associated with these conflicting objectives, too. Therefore, architectural design is described as a complex task. To handle the complexity, computational optimization methods can be employed to investigate architectural design process in detail. This paper focuses on investigating Pareto-front solutions for theatre hall design using multi-objective evolutionary algorithms. To formulate the theatre hall acoustic design problem, we consider three objectives. Two objectives are minimization of both reverberation time, and total initial cost whereas the third objective is the maximization of seating capacity. In addition, several designs and acoustical performance constraints are defined. To tackle this problem, a multi-objective self-adaptive differential evolution algorithm (JDEMO) is proposed and compared with a well-known non-dominated sorting genetic algorithm-II (NSGA-II) from the literature. Computational results show that the proposed JDEMO algorithm achieves competitive results when compared to the NSGA-II.

C. Cubukcuoglu · B. Ekici
Faculty of Architecture, Yasar University, Bornova, Turkey
e-mail: C.Cubukcuoglu@tudelft.nl; cemre.cubukcuoglu@yasar.edu.tr

B. Ekici
e-mail: B.Ekici-1@tudelft.nl; berk.ekici@yasar.edu.tr

A. Kirimtat · B. Ekici
Chair Design Informatics, TU Delft, The Netherlands
e-mail: ayca.kirimtat@yasar.edu.tr

F. Tasgetiren (✉)
International Logistics, Yasar University, Bornova, Turkey
e-mail: fatih.tasgetiren@yasar.edu.tr

P. N. Suganthan
Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: EPNSugan@ntu.edu.sg

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_4

Keywords Computational design · Performance-based design · Acoustics
Theatre hall · Multi-objective optimization · Architectural design
Differential evolution · Form-finding · Evolutionary computation

1 Introduction

Acoustics is an important building performance aspect in the conceptual design phase of theatre halls. The reason is that well-performing acoustic design provides a pleasant hearing quality to audiences. On the other hand, insufficient acoustic design may cause an unfavorable experience during a lecture or a classical music composition.

From small to large scale, acoustic performance can be applied to different building types in the field of architecture. Because of this reason, different design parameters can be associated with the acoustic performance.

In the literature, there exist several studies to explore design parameters in acoustic performance for different building types in recent years. For instance, Chiang and Shu [1] studied how architectural design is affected by acoustical measures through computer simulations. Martin and Arana [2] presented vital parameters for symphony hall acoustics using two different simulation softwares, where several design parameters were investigated for the acoustic performance of symphony hall. On the other hand, Cerda et al. [3] also studied parameters, related to acoustic performance, which are intelligibility, spaciousness, and strength. They presented results for nine different halls with different shapes and functions. Another study [4] focused on large-scale halls by considering acoustic parameters based on the characteristics of the layout configuration. Moreover, Brothánek et al. [5] assessed the acoustics performance of a concert hall considering monaural and binaural parameters. Briefly, vital parameters for acoustic performance can be summarized as follows [6]:

- Layout configuration of the rooms within the building,
- Avoiding sound transmissions for the surfaces, which are adjacent,
- Sound generated by service spaces,
- Shape and size of the space,
- Furniture and equipment placed in the room.

In order to explore a most suitable set of parameters, there are several implementations based on parametric modeling and computational optimization in the literature as well. For instance, Sato et al. [7] investigated near-optimal parameter set based on Genetic Algorithms (GAs). This study considered a shoebox-shaped hall's ceiling and fan-shaped outdoor theatre. Authors presented satisfactory results by making a discussion over the complexity of the problems. In another study, Sato et al. [8], considered a shoebox hall to optimize the proportions of the shape and layout plan, respectively. For both cases, authors formulated the problem as single objective and employed GA. Another study [9] focused on several methods used for evaluating acoustic performance. These methods are real-time computer modeling,

iterative design, computational optimization methods, and auralization. The role of computational and optimization design tools for the acoustic performance is clearly underlined as an important future direction in their study. According to the reviewed literature, we can summarize several gaps in computational optimization for building acoustic performance, as follows:

- Most of the published studies are formulated as single objective problems to assess acoustics performance.
- Multi-objective optimization approaches has not discussed yet for acoustic performance.
- Only a single objective GA has developed and implemented as a computational optimization method to tackle building acoustic in the theatre hall design.

This study deals with the optimization of the acoustic performance of theatre hall design. For this reason, we extended our previous published paper [10] by considering three objective functions and different multi-objective algorithms in the theatre hall design model. The model presented in this paper considers the following additional features:

- An additional conflicting objective function, which is seating capacity of the theatre hall design presented.
- We employed a sine function to differentiate the extrusion of each sidewall module in the theatre hall design.
- With new features, total amount of decision variables are increased up to 17, hence, the complexity of the optimization problem is increased as well.
- We present NSGA-II and JDEMO algorithms to obtain non-dominated Pareto-front solutions.

Multi-objective problems (MOPs) are studied for many years. In order to solve these problems, multi-objective optimization algorithms are proposed. Evolutionary algorithms (EAs) are usually employed to solve MOPs in the literature. EAs are in use to solve the MOPs, denoted as Multi-Objective Evolutionary Algorithms (MOEAs). In particular, NSGA-II [11] and SPEA2 [12] are one of the most studied ones, which are capable of solving difficult engineering and benchmark problems.

Differential evolution (DE), suggested by [13, 14], is another EA widely employed in the current literature [15]. In DE, crossover rate (Cr), and mutation factor (Mr) have an enormous influence on the algorithm's performance. An improved version of DE, self-adaptive DE (jDE) was presented to update Mr and Cr values in each generation [16, 17]. Such a small modification on DE made the jDE a very efficient algorithm. As a multi-objective version of jDE, DEMOwSA is proposed in [18] using the properties of DEMO for MOPs [19, 20]. In this paper, we propose JDEMO with self-adaptive parameter setting proposed in [16, 17]. Then, we compare with well-known NSGA-II in order to present desirable design alternatives to decision maker.

Following sections are as follows: Sect. 2 presents the definition of the problem on hand. Section 3 explains MOEAs in order to gather non-dominated Pareto-front solutions. Section 4 shows how we develop algorithmic form finding model. Section 5 presents computational results. Finally, Sect. 6 discusses final remarks and conclusions.

2 Problem Definition

This study aims at discovering a set of alternatives by considering multiple objectives for theatre hall design. Objective functions of this problem are minimization of reverberation time and total initial cost; and, maximization of the seating capacity. Besides boundary constraints of decision variables, the design problem includes additional constraints as well. These are stage visibility, undesirable floor intersections, and convenient range for reverberation time. In addition, decision variables control the number of seats and arrangement of both seats and platforms, the movement of the stage, and the configuration of absorbing panels on sidewalls of the hall. In Table 1, the design variables of the problem on hand are listed in detail.

2.1 Objective Functions and Constraints

In the problem on hand, three objective functions subject to several constraints are formulated in the mathematical model as follows:

$$\begin{cases} Minimize\, f_1(x) = RT \\ Minimize\, f_2(x) = TC \\ Maximize\, f_3(x) = S_{cap} \\ Subject\, to\, A_s \neq B_s \neq C_s \\ \quad S_{stage} \cap S_A = \emptyset \\ \quad 1.5 \leq RT \leq 2.5 \\ \quad R_{INT} = \emptyset \end{cases}$$

where RT is reverberation time, TC is total initial cost, S_{cap} is seating capacity. $A_S, B_S,$ *and* C_S are the number of seating capacities of A, B, and C-type seats, respectively. Moreover, S_{stage} is the area covered by the stage platform where S_A is the area covered by A-type seats. Finally, R_{INT} is an intersection set of visibility rays belonging to each seat in the hall.

As the first objective, we calculate RT for each generated design in order to assess acoustical performance. RT is one of the most popular acoustic measurement factors

Table 1 Design variables of the design problem

	Decision variable	Notation	Unit	Var. type	Range
x_1	Quantity of platforms	TP	Piece	Integer	[4, 12]
x_2	Platforms angle	PA	Degree	Real	[18, 90]
x_3	Height of the platform-A	Ah	Meter	Real	[0.1, 1.0]
x_4	Height of the platform-B	Bh	Meter	Real	[0.5, 1.5]
x_5	Height of the platform-C	Ch	Meter	Real	[0.8, 1.8]
x_6	Stage height	SH	Meter	Real	[0.5, 1.5]
x_7	Location of the stage in X-axis	SM	Meter	Real	[10, 20]
x_8	# of seating A	As	Piece	Integer	[18, 36]
x_9	# of seating B	Bs	Piece	Integer	[18, 36]
x_{10}	# of seating C	Cs	Piece	Integer	[18, 36]
x_{11}	Variable a in sine function of panels for side wall 1	a	Variable	Real	[0.1, 1.5]
x_{12}	Variable b in sine function of panels for side wall 1	b	Variable	Real	[0, 10]
x_{13}	Variable c in sine function of panels for side wall 1	c	Variable	Real	[5, 10]
x_{14}	Variable a in sine function of panels for side wall 2	a'	Variable	Real	[0.1, 1.5]
x_{15}	Variable b in sine function of panels for side wall 2	b'	Variable	Real	[0, 10]
x_{16}	Variable c in sine function of panels for side wall 2	c'	Variable	Real	[5, 10]
x_{17}	Shoebox movement factor	f	Meter	Real	[−15, 15]

in the field of architecture. This measurement is calculated in seconds. In fact, RT is evaluated in time, which takes for a signal to drop by 60 dB [21].

RT is conflicting with S_{cap} and TC. In other words, as seating capacity increases total amount of absorbing elements, it improves acoustical performance. However, having more seating capacity leads to an increase in TC. Therefore, we try to minimize RT and TC whereas we try to maximize S_{cap}. Sabine [21] examined the relationship between absorption and the reverberation time with the following formula in Eq. (1):

$$RT = \frac{0.161 * V}{\sum S * \propto} \tag{1}$$

where V is the volume of the room (m^3), and S is the total area of the surface (m^2), and $\propto$ is the coefficient of the sound absorption for elements used in the space. In the Eq. (2), the absorption coefficient of materials varies with absorption characteristics.

$$\sum S* \propto = \left[\left(S_{floor}* \propto_{floor}\right) + \left(S_{sidewalls}* \propto_{sidewalls}\right) + \left(S_{backwall}* \propto_{backwall}\right) + \left(S_{frontwall}* \propto_{frontwall}\right) + \left(S_{ceiling}* \propto_{ceiling}\right) + \left(S_{seatings}* \propto_{seatings}\right) + \left(S_{platforms}* \propto_{platforms}\right) + \left(S_{panels}* \propto_{panels}\right)\right] \quad (2)$$

For the second objective function, TC is a summation of seats cost (SC), platforms cost (PC), stage cost (STC), and absorbing panels cost (AC) as follows:

$$TC = SC + PC + STC + AC \quad (3)$$

where SC, PC, STC are calculated as follows:

$$SC = \left(S_{cap}\right) * chairs\ unit\ cost \quad (4)$$

$$PC = (The\ platforms\ surface\ area) * unit\ cost \quad (5)$$

$$STC = (Stage\ of\ the\ theatre\ hall\ surface\ area) * unit\ cost \quad (6)$$

Regarding AC, side walls are other important components in the cost calculation. Extrusions of these panels support the acoustical performance. However, having more sidewall elements causes an increase in TC. Extrusions of these panels are produced by the sine function as in Eqs. (7) and (8). In this way, more design variations in the theatre hall are achieved. Since we have two different sidewalls on the right and left sides, Eq. (7) corresponds to the right panels, while Eq. (8) matches for the left panels as follows:

$$a * \sin(c * x + b) \quad (7)$$

$$a' * \sin\left(c' * x + b'\right) \quad (8)$$

where "a" refers to the depth of the panels belonging to each sidewall, variable "b", and variable "c" refer to the configurations of the box panels. The use of sine function for the panel extrusions is illustrated in Fig. 1. The calculation of AC is presented in Eq. (9).

$$AC = (total\ surface\ area\ of\ absorbing\ panels) * unit\ cost \quad (9)$$

As the third objective, we calculate S_{cap}. Different combinations of chairs on different types of platforms directly control this objective. The calculation of this function is presented in Eq. (10).

$$S_{cap} = A_S + B_S + C_S \quad (10)$$

As can be shown in the mathematical model above, the first constraint provides diversity for each seating platform through a non-equal number of seating types. Second, intersection between stage platform and front seating is avoided. Third, the convenient range for RT is considered as between 1.5 and 2.5 [22, 23].

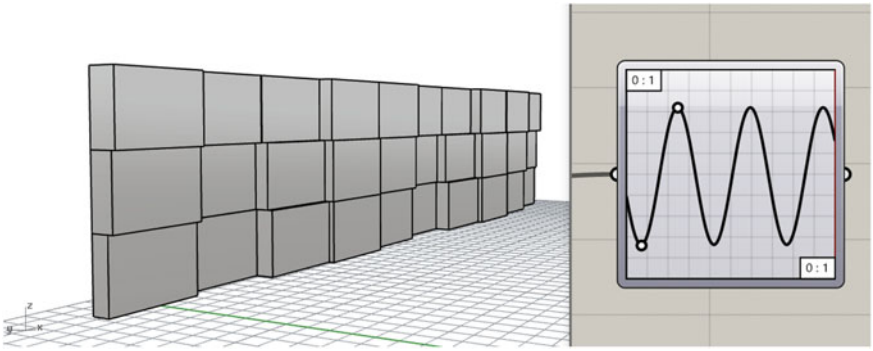


Fig. 1 Extrusions with sine function

Visibility of the stage from audiences is one of the most important aspect, too. Thus, we defined it as another constraint, which is related to different heights of the platforms and stage. To calculate this constraint function, lines between stage and chairs located on each seating group are generated as rays. Then, the intersection of each ray belongs to different seating groups is computed as follows:

$$R_{INT} = [(R_A \cap R_B) \cup (R_A \cap R_C) \cup (R_C \cap R_B)] - 2(R_A \cap R_B \cap R_C) \qquad (11)$$

where R_A, R_B, *and* R_C are rays generated between seating groups A, B, C and stage, respectively. Finally, black box constraints are defined in Table 1.

3 Multi-objective Evolutionary Algorithms

In this study, we addressed theatre hall problem by taking advantage of the MOEAs. Three-objective functions subject to several constraints are considered. To tackle this problem, we proposed JDEMO algorithm. Then, we compared results obtained by JDEMO with well-known NSGA-II algorithm.

3.1 NSGA-II Algorithm

The NSGA-II algorithm is one of the most sophisticated multi-objective optimization algorithms in the literature [11]. The NSGA-II is robust, elitist, and capable to handle constrained problems as well. It is also capable to deal with real-world multi-objective problems. As a result, the NSGA-II can distribute non-dominated Pareto-front solutions homogenously.

```
for each p ∈ P
    S_p = ∅
    n_p = 0
    for each q ∈ P
        if (p ≺ q) then                    If p dominated q
            S_p = S_p ∪ {q}                Add q to the set of solutions dominated by p
        else if (q ≺ p) then
            n_p = n_p + 1                  Increment the domination counter of p
    if n_p = 0 then                        p belongs to the first front
        p_rank = 1
        F_1 = F_1 ∪ {p}
i = 1                                      Initialize the front counter
while F_i ≠ ∅
    Q = ∅                                  Used to store the members of the next front
    for each p ∈ F_i
        for each q ∈ S_p
            n_q = n_q − 1
            if n_q = 0 then                q belongs to the next front
                q_rank = i + 1
                Q = Q ∪ {q}
i = i + 1
F_i = Q
End
```

Fig. 2 Non-dominated sorting procedure [11]

1. $L = |\Omega|$
2. *for each individual* i, *set* $\Omega[i]_{dist} = 0$
3. *for each objective* m
   a. $\Omega = sort(\Omega, m)$ *in an increasing order*
   b. $\Omega[1]_{dist} = \Omega[L]_{dist} = \infty$
   c. *for* $i = 2$ *to* $L - 1$

$$\Omega[i]_{dist} = \Omega[i]_{dist} + (\Omega[i+1][m] - \Omega[i-1][m])/(f_m^{max} - f_m^{min})$$

Fig. 3 Crowding distance calculation

In our problem, the NSGA-II begins with an initial population generated randomly. The design variables are $\{TP, PA, Ah, Bh, Ch, SH, SM, As, Bs, Cs, a, b, c, a', b', c', f\}$. Each individual represents 17 dimensions of decision variables in the problem on hand. In other words, $x_1 =$ TP is the first dimension of the first individual and so on. In order to generate offspring population, we employed simulated binary crossover (SBX) and polynomial mutation (PM). Then, we employed the fast non-dominated sorting procedure in [11] to create non-dominated fronts as shown in Fig. 2.

Another important feature of NSGA-II is the crowding distance. Given a non-dominated set Ω. The crowding distance of $\Omega[i]_{dist}$ is measured as in Fig. 3.

3.1.1 Comparison Operators

The NSGA-II uses the crowded-comparison operator ($\prec$) for unconstrained multi-objective optimization. It directs the selection process at different steps of the algorithm. Every individual i in the population has a crowding distance rank (i_{dist}) and a nomination rank (i_{rank}). Crowded-comparison operator ($\prec$) is defined as follows:

$$\begin{aligned} &if\ (i_{rank} < j_{rank})\ then\ i \prec j \\ &or\ ((i_{rank} = j_{rank})\ and\ (i_{dist} > j_{dist})) \end{aligned}$$

It means that lower (better) rank is favored between two solutions according to non-domination ranks. On the other hand, if both individuals have the same rank, lesser crowded region is favored for the selection.

This situation is somewhat different for the constrained multi-objective optimization when comparing two solutions. At this point, three cases can be observed: (1) one individual is feasible, the other is not; (2) both individuals are infeasible; and (3) both individuals are feasible. For the constrained multi-objective optimization, the NSGA-II modifies the definition of *domination* between two solutions as follows:

An individual p is considered to *constrained-dominate* an individual q under the following conditions:

(1) p is feasible, and q is infeasible.
(2) p and q are infeasible, but p has a smaller constraint violation.
(3) Both p and q are feasible and p has a smaller rank than q with crowded-comparison rule as follows:

$$\begin{aligned} &if\ (p_{rank} < q_{rank})\ then\ p \prec q \\ &or\ ((p_{rank} = q_{rank})\ and\ (p_{dist} > q_{dist})) \end{aligned}$$

Now we are ready to outline the NSGA-II in Fig. 4. An example of NSGA-II for multi-objective constrained benchmark problem is given in Appendix A.

3.2 JDEMO Algorithm

DE is an evolutionary algorithm proposed by Storn and Price [13, 14]. In DE algorithms with each individual P_{ij} with $j = 1, \ldots, D$ dimensions, a random target population (P) with size $N = |P|$ is uniformly established within the boundaries of each decision variable. In a very basic form denoted as DE/rand/1/bin, three individuals are selected from the target population. The difference vector of two individuals is multiplied by the mutation scale factor Mr and added to the a third individual to generate mutant individual as follows:

1. *Set $t = 0$ and create a random initial population P^t with size N*
2. *Perform SBX and PM on P^t to obtain Q^t with size N*
3. *If the termination criteria is satisfied, stop and return P^t.*
4. *Set $R^t = P^t \cup Q^t$ to combine two populations*
5. *Perform non − dominated sorting procedure for R^t to set the non − dominated fronts $f_1, f_2, .., f_k$.*
6. *For $i = 1, .., k$, repeat the following steps:*
   a. *Calculate crowding distance for each solution in f_i.*
   b. *Create P^{t+1} as follows:*
      i. *if $|P^{t+1}| + |f_i| \leq N$, then set $P^{t+1} = P^{t+1} \cup f_i$*
      ii. *if $|P^{t+1}| + |f_i| > N$, then add the least crowded $N - |P^{t+1}|$ solutions from f_i to P^{t+1}.*
7. *Use binary tournament selection with constrained − domination rule to select parents from P^{t+1}.*
8. *Apply SBX and PM to P^{t+1} and obtain new offspring population Q^{t+1} with size N*
9. *Set $t = t + 1$ and go to Step 3.*

Fig. 4 NSGA-II outline

$$V_{ij}^{t+1} = P_{r_1 j}^t + Mr * \left(P_{r_2 j}^t - P_{r_3 j}^t \right) \quad \forall i = 1, \ldots, |P|; \forall j = 1, \ldots, D \tag{12}$$

Then, a trial individual can be generated from target and mutant individuals using a binomial crossover operator as follows:

$$Q_{ij}^{t+1} = \begin{cases} V_{ij}^{t+1} \; if \; r_{ij} \leq Cr \; or \; j = D_j \\ P_{ij}^t \quad otherwise \end{cases} \quad \forall i = 1, \ldots, |P|; \forall j = 1, \ldots, D \tag{13}$$

where r_{ij} is a uniform random number in [0, 1] and D_j is a random integer in $[1, D]$, which makes sure that at least one dimension comes from the mutant individual.

In this paper, we propose a multi-objective DE algorithm denoted as JDEMO, which is inspired from DEMO in [19, 20]. The only difference between them is to employ the self-adaptive parameter updating of the jDE algorithm [16, 17]. As well known, DE algorithms have two parameters: (1) crossover rate, Cr; (2) mutation scale factor, Mr. These two parameters substantially affect the performance of DE algorithms. In the jDE algorithm, these two parameters are updated at each generation. Initially, these parameters are assigned to $Cr_i = 0.5$ and $Mr_i = 0.9$ for each individual in the population. However, with a small probability, these two parameters are updated at each generation as follows:

$$Mr_i^{t+1} = \begin{cases} Mr_l + r_1 . Mr_u \; if \; r_2 < p_1 \\ Mr_i^t \qquad\qquad otherwise \end{cases} \tag{14}$$

1. *Set $t = 0$ and establish a random target population P^t with size N.*
2. *Apply DE mutation and crossover operators to P^t in order to obtain trial population Q^t with size N.*
3. *If the stopping criterion is satisfied, stop and return X^t.*
4. *Combine two populations as $R^t = P^t \cup Q^t$*
5. *Apply the fast non − dominated sorting to combined population R^t and determine the non − dominated fronts $\mathcal{F}_1, \mathcal{F}_2, .., \mathcal{F}_k$ in R^t.*
6. *For $i = 1, .., k$, do the following steps:*
   a. *Calculate the crowding distance of solutions in $\mathcal{F}_i$.*
   b. *Create P^{t+1} as follows:*
      i. *if $|P^{t+1}| + |\mathcal{F}_i| \leq N$, then set $P^{t+1} = P^{t+1} \cup \mathcal{F}_i$*
      ii. *if $|P^{t+1}| + |\mathcal{F}_i| > N$, then add the least crowded $NP - |P^{t+1}|$ solutions from $\mathcal{F}_i$ to P^{t+1}.*
7. *Use binary tournament selection with constrained domination rule to select parents from P^{t+1}*
8. *Apply DE mutation and crossover operators to X^{t+1} and obtain trial population Q^{t+1} of size N.*
9. *Set $t = t + 1$ and return to Step 3.*

Fig. 5 JDEMO Algorithm

$$Cr_i^{t+1} = \begin{cases} r_3 & \text{if } r_4 < p_2 \\ Cr_i^t & \text{otherwise} \end{cases} \tag{15}$$

where $r_j \in \{1, 2, 3, 4\}$ are uniform random numbers in the range $[0, 1]$. t_1 and t_2 denote the probabilities to adjust the Mr_i and Cr_i values. They are taken as $p_1 = p_2 = 0.1$ and $Mr_1 = 0.1$ and $Mr_u = 0.9$.

Since the NSGA-II is a very powerful algorithm with its unique non-dominated sorting algorithm and crowding distance feature as well as constrained-dominate rule, we employ these features in JDEMO algorithm too as explained in the previous section. In other words, the NSGA-II employs SBX crossover and PM mutation operators when it generates an offspring population. However, the JDEMO algorithm uses the DE mutation and crossover operators (Eqs. 12 and 13) when generating the offspring population. Now, we are ready to outline JDEMO as given in Fig. 5.

4 Generative Model

In the generative model of the theatre hall building, a traditional shoebox model for acoustical halls is concerned. This shoebox model was 50 m deep and length, 10 m height. The side walls are made of a painted concrete, with acoustical box-shaped panels with absorbing material. The materials of each element in hall space are presented as follows:

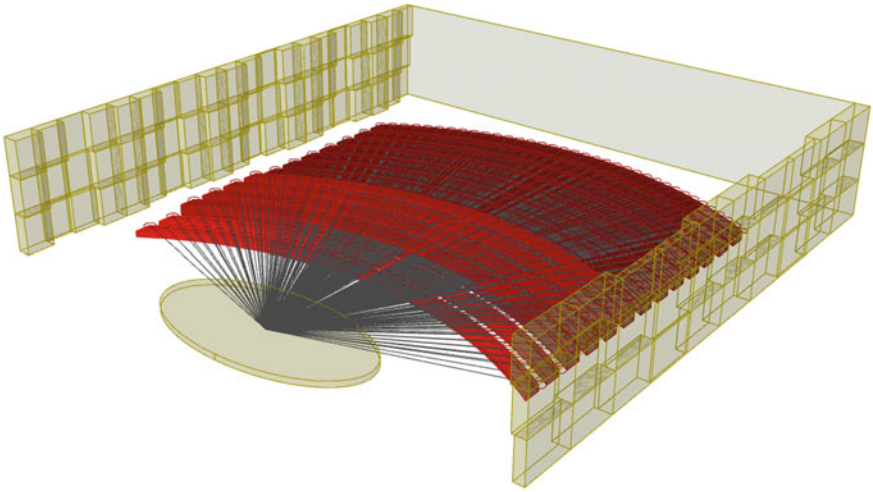


Fig. 6 Representation of the theatre hall in concept design stage

- Rear wall: Fabric curtain
- Front wall: "¼" plate glass
- Side walls: Acoustical wall fabric
- Floor: Carpet
- Ceiling: Acoustical ceiling tiles
- Platform below seating: Carpet
- Performance Stage: Wood parquet.

For audience seating layout, there are three types of seating platforms. In the newly developed model in this paper, distances between "stage and each platform" determine a grouping strategy for platforms. Seating type A is the one, which has a distance to the stage smaller than 6 m. On the other hand, distance to the stage between 6 and 15 m corresponds to seating type B. Finally, the furthest seating type, so-called type C, has larger than 15 m distance from the performance stage. This configurational layout is illustrated in Fig. 6.

Each seating group has different height parameter as such *Ah*, *Bh*, and *Ch*. In addition, density of the chairs on each platform type varies from each other. These facts are much related with visibility of audiences on each platform group. On the other hand, bigger platform height values give rise to increment of total cost, but decrement of reverberation time. In addition, angles of each platform, which are alike, are also design variables of this problem. In the new model, range of the platform angle is larger than the previous model. By this way, platforms are capable to open until 90°. The reason for this fact is the new theatre hall model that has a larger volume than the previous one.

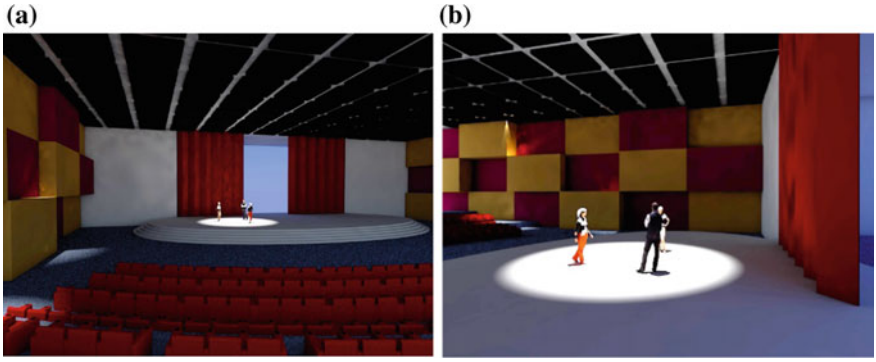


Fig. 7 **a** and **b** Computer renderings of the theatre hall

Fig. 8 Computer rendering of theatre hall

Regarding the platform stage, the shape is defined as an ellipse with the radii of 10 m to 5 m. Moreover, stage height is another design variable. The ranges for stage height and location of the stage are larger than the previous model. Renders of the generative theatre hall model can be seen in Figs. 7 and 8.

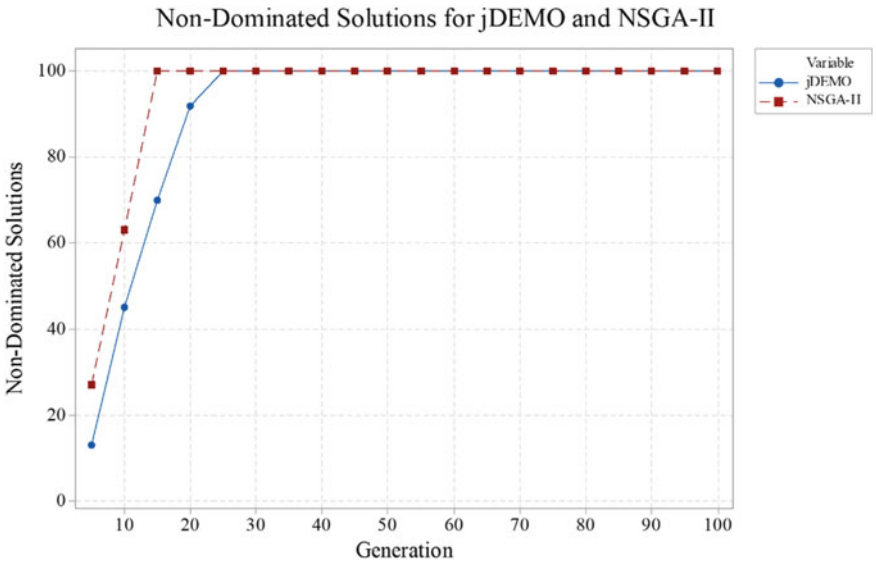


Fig. 9 Number of Non-dominated solutions during 100th generation

5 Computational Results and Discussion

In this paper, JDEMO algorithm is implemented to the multi-objective theatre hall problem and then compared to the results gathered by the well-known NSGA-II. In the early stage of the whole design process, we considered the acoustics performance of the building on hand. During the optimization process, these algorithms are separately run for a single replication on a computer, which has 2.6 GHz intel core i7-6700HQ processor, 8 GB $\times$ 2 DDR3 memory and 256 GB SSD. The population size is fixed at 100. As a termination criterion, we run both algorithms for 100 generations. The number of non-dominated solutions of JDEMO and NSGA-II for 100 generations are given in Fig. 9. Although there is no significant change after 30th generation for both algorithms as seen in this figure, we also analyzed the behavior of fitness functions during 100 generations to determine the termination criterion.

The behavior of the fitness functions in terms of minimum, average, and maximum values of non-dominated solutions in the population is analyzed in detail. After 80th generation, the values of objective functions are not changing that much. In the 100th generation, we observed that fitness function values are converged. Thus, a termination criterion is defined as 100th generation as shown in Figs. 10, 11, and 12.

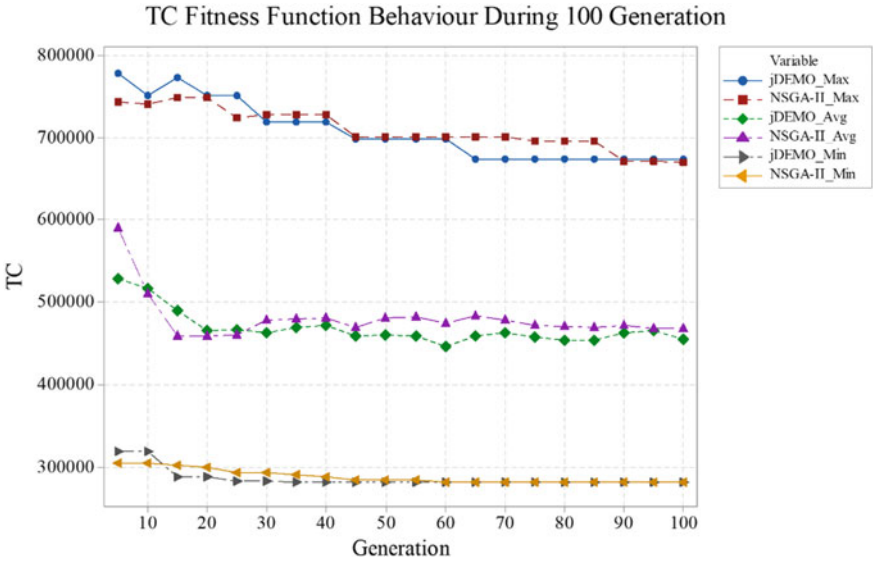


Fig. 10 JDEMO and NSGA-II Minimum, Maximum, and Average values for total initial cost

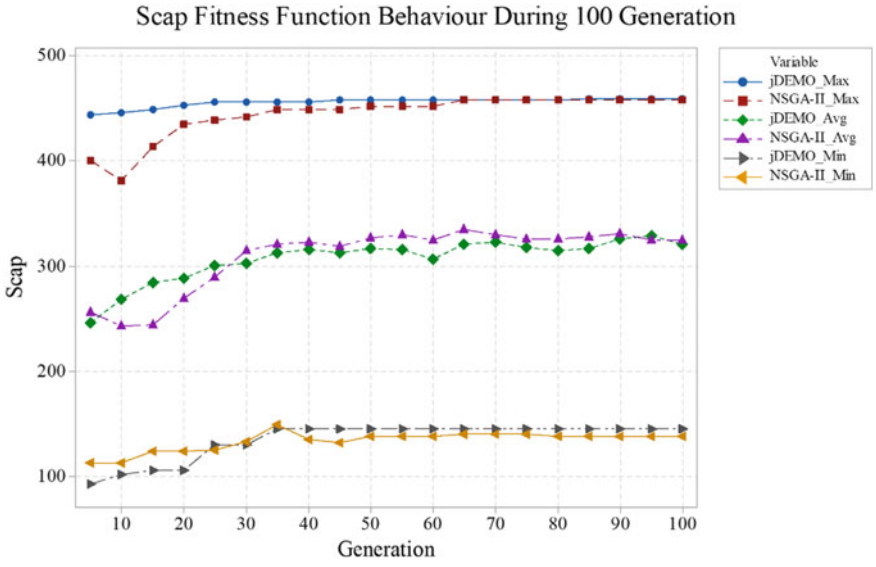


Fig. 11 JDEMO and NSGA-II Minimum, Maximum, and Average values for seating capacity

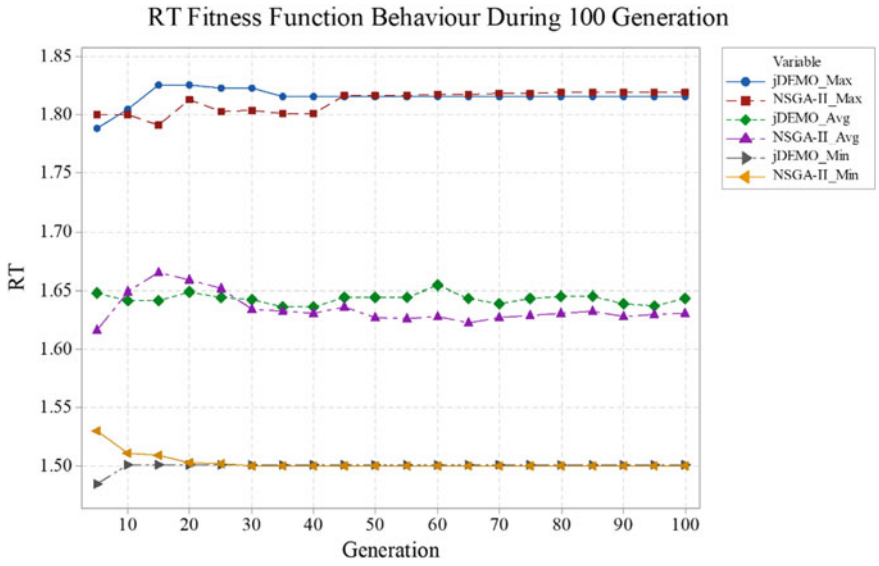


Fig. 12 JDEMO and NSGA-II Minimium, Maximum, and Average values for reverberation time

Finally, the non-dominated solutions of JDEMO and NSGA-II after 100 generations are obtained as shown in Fig. 13 and Fig. 14, respectively. In the scatter plot below, red dots correspond to the non-dominated individuals obtained from the NSGA-II; blue dots represent the results obtained by JDEMO algorithm.

Both algorithms have presented very competitive results in terms of objective function ranges. The JDEMO algorithm achieves minimum reverberation time value, which is very close to 1.4. On the other hand, the NSGA-II is slightly better in terms of total cost objective because it reaches at the minimum value under 300,000 TL. Nevertheless, both algorithms have presented very similar objective function ranges, non-dominated solutions, and different decision variable combinations. This fact has an apparent impact on the quality of the architectural design alternatives. To be more precise, the direct influence of the design parameters on the building form makes the changes in design parameters more important.

For the clarification of the previous statements, the interval plot of the decision variables after 100th generation with 95% of confidence interval for the mean is shown in Fig. 15. This plot is analyzed in terms of each algorithm's performance on the multi-objective theatre hall problem.

In Fig. 15, x coordinate shows the notations of each design variable belonging to each algorithm; y coordinate shows the normalized data of the design variables' ranges. In Fig. 15, we can easily summarize the difference between each algorithm in terms of their results of decision variables. Based on this plot, we can say that some decision variables are converged, while others are diverged. In particular, $x_1, x_3, x_4, x_5, x_8, x_9, x_{10}$ have the longest vertical lines for both algorithms. It means

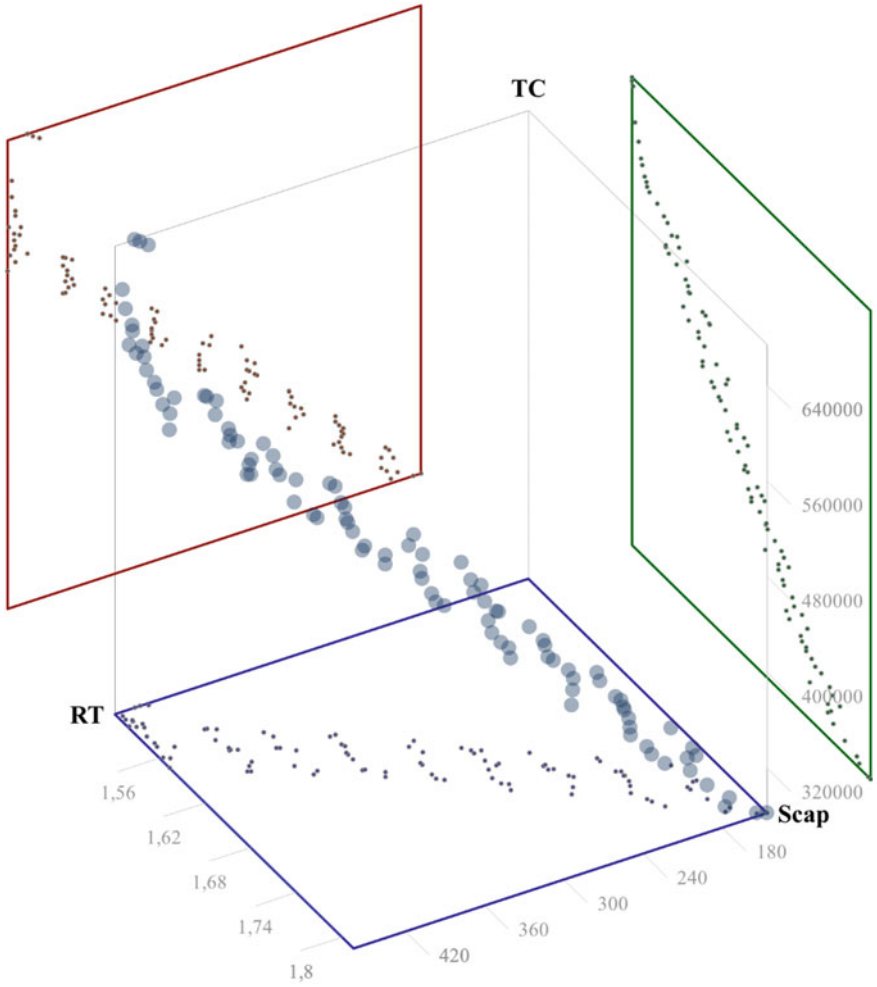


Fig. 13 Non-dominated solutions of JDEMO for 100th generation

that the decision variables pertained to seating units have a great impact on the objectives. In addition, A-type seating platform group affected the results obtained by both algorithms. B-type and C-type seating platform groups are not so much affected the objectives as much as A-type. This can be explained by the closeness of A-type chairs group to the stage causes a larger amount of A-type units produced. By this way, A-type seating group absorb more sound (which is positive for reverberation time) but cause more total initial cost at the same time. On the other hand, the variables related to sidewalls acoustics panels, which are x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, give an interesting result on each algorithm. Although the ranges of the results are very similar in both algorithms for both sidewalls, the minimum and maximum values of this

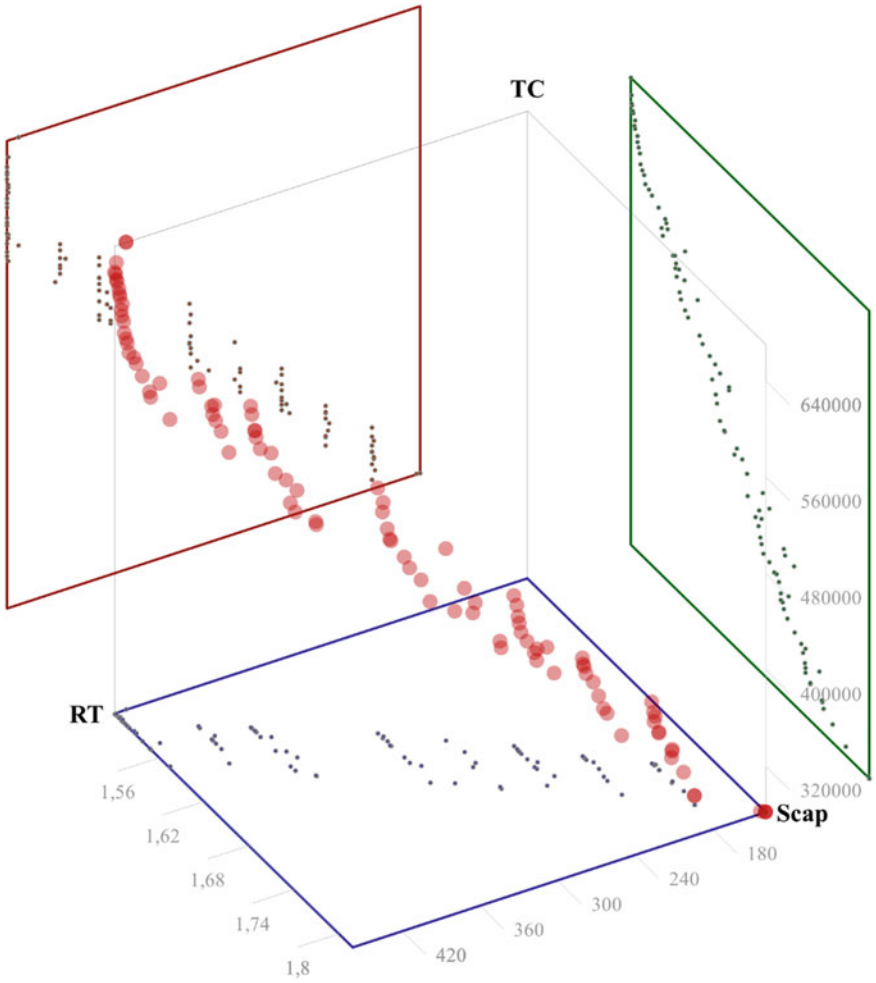


Fig. 14 Non-dominated solutions of NSGA-II for 100th generation

results regarding to each sidewall are changing in between. Right side walls have larger values as seen in the plot. It means that the right side walls' panels have more variation in the design of boxes. Moreover, some solutions have different characteristics on two algorithms. For example, NSGA-II gives higher ranges in the most cases than JDEMO. Another example is that JDEMO is capable to present better ranges in the results for the decision variables x_6 and x_7. Therefore, it can be said that JDEMO has more impact on visibility constraint than NSGA-II. On the other hand, NSGA-II presents better combinations in the design of box panels which means that the variables belonging to sine function are more changeable in NSGA-II.

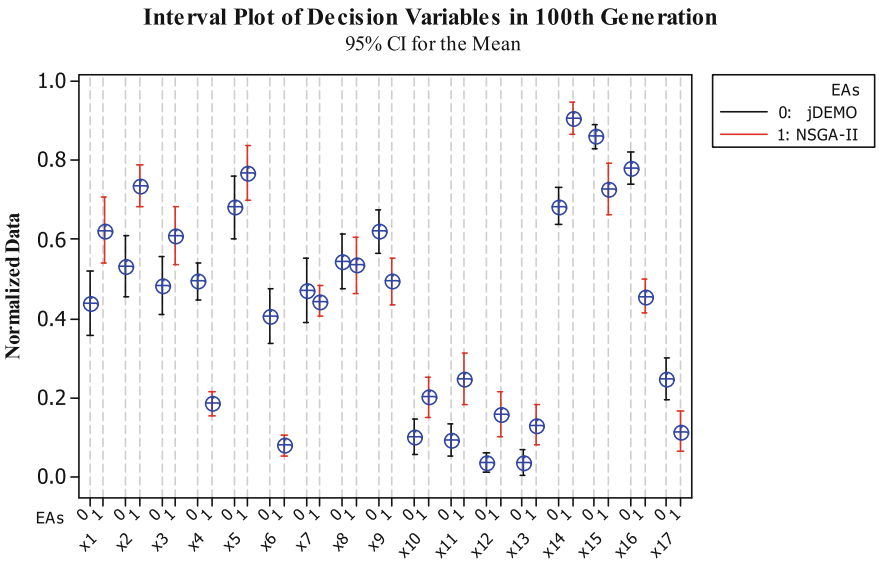


Fig. 15 Interval plot of decision variables in 100th generation

6 Conclusion

In this study, we proposed two different optimization algorithms, s-called NSGA-II and JDEMO in order to get a Pareto front with various non-dominated solutions. We achieved plausible theatre hall alternatives that may be considered as efficient designs that minimize total initial cost and reverberation time, and maximize the seating capacity. Each algorithm converges very fast and both algorithms are capable of presenting feasible design alternatives as well. However, with respect to the computational analysis, results differ from each other in terms of range of design/decision variables. This fact directly influences the form of the architectural design. Finally, it can be said that NSGA-II and JDEMO algorithms are competitive to solve the multi-objective theatre hall-design problem.

7 Appendix A

For the SRN function in [11], we present the NSGA-II algorithm with Simulated Binary Crossover (SBX), Polynomial Mutation (PM), and Non-dominated Sorting algorithm. The population size is ten and we generate five offsprings in the example.

Problem	*n*	*Variable Bounds*	*Objective Functions*	*Constraints*
SRN	2	$x_i \in [-20, 20]$	$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$	$g_1(x) = {x_1}^2 + {x_2}^2 \leq 225$
		$i = 1, 2$	$f_2(x) = 9x_1 - (x_2 - 1)^2$	$g_2(x) = x_1 - 3x_2 \leq -10$

Initial population

SRN	Parameters xi = [-20,20]		Fitness Function		Constraint Function		V1(x)=225-g1(x)	If V1(x) ≥ 0, CV=0 else, CV=V1(x)	V2(x)=-10-g2(x)	If V2(x) ≥ 0, CV=0 else, CV=V2(x)	If CV = 0, feasible else, infeasible
	i1	i2	f1(x)	f2(x)	g1(x)	g2(x)	V1(x)	CV g1(x)	V2(x)	CV g2(x)	CV sum
x1	-3,42	-16,58	340,43	-339,84	286,59	46,32	-61,59	-61,59	-56,32	-56,32	117,91
x2	3,52	14,45	185,21	-149,22	221,19	-39,83	3,81	0,00	29,83	0,00	0,00
x3	-18,21	2,66	413,20	-166,65	338,68	-26,19	-113,68	-113,68	16,19	0,00	113,68
x4	-9,35	7,56	173,86	-127,18	144,58	-32,03	80,42	0,00	22,03	0,00	0,00
x5	19,64	17,61	589,06	-99,13	695,84	-33,19	-470,84	-470,84	23,19	0,00	470,84
x6	11,46	7,51	133,87	60,76	187,73	-11,07	37,27	0,00	1,07	0,00	0,00
x7	1,21	-18,66	389,14	-375,63	349,66	57,19	-124,66	-124,66	-67,19	-67,19	191,85
x8	6,25	-0,51	22,34	53,97	39,32	7,78	185,68	0,00	-17,78	-17,78	17,78
x9	5,68	-19,54	437,43	-370,77	414,07	64,30	-189,07	-189,07	-74,30	-74,30	263,37
x10	15,68	10,94	287,95	42,32	365,55	-17,14	-140,55	-140,55	7,14	0,00	140,55

Simulated Binary Crossover for 4 dimensions of Offspring

Start SBX procedure

choose parent p1 and p2

generate a random value u between 0 and 1

Generate a random $n \in N^+$

if $u \leq 0.5$

$$\beta = 2u^{\frac{1}{n+1}}$$

else

$$\beta = \frac{1}{2(1-u)^{\frac{1}{n+1}}}$$

calculate c1 and c2

$$c1 = 0.5 \times [(1+\beta) \times p1 + (1-\beta) \times p2]$$

$$c2 = 0.5 \times [(1-\beta) \times p1 + (1+\beta) \times p2]$$

End

SBX for x2 and x8

	p1 (x2)	p2 (x8)	u	β	n	c1	c2
i1	3,52	6,25	0,21	0,84	4,00	3,74	6,03
i2	14,45	-0,51				6,97	0,68

SBX for x6 and x10

	p1 (x2)	p2 (x8)	u	β	n	c3	c4
i1	11,46	15,68	0,44	0,96	2,00	11,55	15,59
i2	7,51	10,94				7,58	10,87

Polynomial Mutation for 1 dimension of Offspring

Start PM procedure

choose parent $p1 x_i^{(U)} x_i^{(L)}$ *generate a random number u between* 0, *and* 1

$x_i^{(U)} x_i^{(L)}$ *generate a random number u between* 0, *and* 1

generate a random number u between 0, *and* 1

generate a random number t between 0, *and* 1

calculate mutated solution p' as follows

$$p' = \begin{cases} p + \delta_L \left(p - x_i^{(L)}\right), & for\ u \leq 0.5, \\ p + \delta_R \left(x_i^{(U)} - p\right), & for\ u > 0.5. \end{cases}$$

calculate two parameters δ_L *or* δ_R *as follows* $\delta_L = (2u)^{\frac{1}{1+}} - 1$, *for* $t \leq 0.5$,

$\delta_L = (2u)^{\frac{1}{1+}} - 1$, *for* $t \leq 0.5$,

$\delta_R = 1 - \left(2(1-u)\right)^{\frac{1}{1+}} - 1$, *for* $t > 0.5$.

End

PM for x3

(x3) i=1	min i=1	max i=1			δ_R $t > 0.5$	p' i=1
-18,21	-18,21	19,64				-17,76
(x3) i=2	min i=2	max i=2	0,20	20,00	0,01	p' i=2
2,66	-19,54	17,61				2,84

Initial population with 5 Offspring

SRN	Parameters xi = [-20,20]		Fitness Function		Constraint Function		V1(x)=225-g1(x)	If V1(x) ≥ 0, CV=0 else, CV=V1(x)	V2(x)=-10-g2(x)	If V2(x) ≥ 0, CV=0 else, CV=V2(x)	If CV = 0, feasible else, infeasible
	i1	i2	f1(x)	f2(x)	g1(x)	g2(x)	V1(x)	CV g1(x)	V2(x)	CV g2(x)	CV sum
x1	-3,42	-16,58	340,43	-339,84	286,59	46,32	-61,59	-61,59	-56,32	-56,32	117,91
x2	3,52	14,45	185,21	-149,22	221,19	-39,83	3,81	0,00	29,83	0,00	0,00
x3	-18,21	2,66	413,20	-166,65	338,68	-26,19	-113,68	-113,68	16,19	0,00	113,68
x4	-9,35	7,56	173,86	-127,18	144,58	-32,03	80,42	0,00	22,03	0,00	0,00
x5	19,64	17,61	589,06	-99,13	695,84	-33,19	-470,84	-470,84	23,19	0,00	470,84
x6	11,46	7,51	133,87	60,76	187,73	-11,07	37,27	0,00	1,07	0,00	0,00
x7	1,21	-18,66	389,14	-375,63	349,66	57,19	-124,66	-124,66	-67,19	-67,19	191,85
x8	6,25	-0,51	22,34	53,97	39,32	7,78	185,68	0,00	-17,78	-17,78	17,78
x9	5,68	-19,54	437,43	-370,77	414,07	64,30	-189,07	-189,07	-74,30	-74,30	263,37
x10	15,68	10,94	287,95	42,32	365,55	-17,14	-140,55	-140,55	7,14	0,00	140,55
c1	3,74	6,97	40,66	-2,00	62,55	-17,17	162,45	0,00	7,17	0,00	0,00
c2	6,03	0,68	18,36	54,19	36,86	3,99	188,14	0,00	-13,99	-13,99	13,99
c3	11,55	7,58	136,48	60,62	190,84	-11,20	34,16	0,00	1,20	0,00	0,00
c4	15,59	10,87	284,13	42,94	361,23	-17,01	-136,23	-136,23	7,01	0,00	136,23
p'	-17,76	2,84	396,02	-163,25	323,63	-26,27	-98,63	-98,63	16,27	0,00	98,63

Non-dominated Sorting Procedure

Start nond sorting procedure
  for each $p \in P$
    $S_p = \emptyset$
    $n_p = 0$
    for each $q \in P$
      if $(p \prec q)$ *then* — If p dominated q
        $S_p = S_p \cup \{q\}$ — Add q to the set of solutions dominated by p
      else if $(q \prec p)$ *then*
        $n_p = n_p + 1$ — Increment the domination counter of p
    if $n_p = 0$ *then* — p belongs to the first front
      $p_{rank} = 1$
      $F_1 = F_1 \cup \{p\}$
  $i = 1$ — Initialize the front counter
  while $F_i \neq \emptyset$
    $Q = \emptyset$ — Used to store the members of the next front
    for each $p \in F_i$
      for each $q \in S_p$
        $n_q = n_q - 1$
        if $n_q = 0$ *then* — q belongs to the next front
          $q_{rank} = i + 1$
          $Q = Q \cup \{q\}$
  $i = i + 1$
  $F_i = Q$
End

Constrained Handling Approach

A solution i is said to constrained-dominate a solution j, if any of the following conditions is true:

1) *Solution i is feasible and solution j is not.*
2) *Solution i and j are both infeasible, but solution i has a smaller overall constraint violation.*
3) *Solution i and j are feasible and solution i dominates solution j.*

Non-dominated Constrained Sorting for Initial Population with 5 Offspring

p	q	np	Sp
x1	x2	n1=1	
	x3	n1=2	
	x4	n1=3	
	x5		S1={x5}
	x6	n1=4	
	x7		S1={x5,x7}
	x8	n1=5	
	x9		S1={x5,x7,x9}
	x10		S1={x5,x7,x9,x10}
	c1	n1=6	
	c2	n1=7	
	c3	n1=8	
	c4		S1={x5,x7,x9,x10,c4}
	p'	n1=9	
S1={x5,x7,x9,x10,c4}, n1=9 , F1={Ø}			

p	q	np	Sp
x2	x1		S2={x1}
	x3		S2={x1,x3}
	x4		
	x5		S2={x1,x3,x5}
	x6		
	x7		S2={x1,x3,x5,x7}
	x8		S2={x1,x3,x5,x7,x8}
	x9		S2={x1,x3,x5,x7,x8,x9}
	x10		S2={x1,x3,x5,x7,x8,x9,x10}
	c1		
	c2		S2={x1,x3,x5,x7,x8,x9,x10,c2}
	c3		
	c4		S2={x1,x3,x5,x7,x8,x9,x10,c2,c4}
	p'		S2={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}
S2={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}, n2=0 , F1={x2}			

p	q	np	Sp
x3	x1		S3={x1}
	x2	n3=1	
	x4	n3=2	
	x5		S3={x1,x5}
	x6	n3=3	
	x7		S3={x1,x5,x7}
	x8	n3=4	
	x9		S3={x1,x5,x7,x9}
	x10		S3={x1,x5,x7,x9,x10}
	c1	n3=5	
	c2	n3=6	
	c3	n3=7	
	c4		S3={x1,x5,x7,x9,x10,c4}
	p'	n3=8	
S3={x1,x5,x7,x9,x10,c4}, n3=8 , F1={x2}			

p	q	np	Sp
x4	x1		S4={x1}
	x2		
	x3		S4={x1,x3}
	x5		S4={x1,x3,x5}
	x6		
	x7		S4={x1,x3,x5,x7}
	x8		S4={x1,x3,x5,x7,x8}
	x9		S4={x1,x3,x5,x7,x8,x9}
	x10		S4={x1,x3,x5,x7,x8,x9,x10}
	c1		
	c2		S4={x1,x3,x5,x7,x8,x9,x10,c2}
	c3		
	c4		S4={x1,x3,x5,x7,x8,x9,x10,c2,c4}
	p'		S4={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}
S4={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}, n4=0 , F1={x2,x4}			

p	q	np	Sp
x5	x1	n5=1	
	x2	n5=2	
	x3	n5=3	
	x4	n5=4	
	x6	n5=5	
	x7	n5=6	
	x8	n5=7	
	x9	n5=8	
	x10	n5=9	
	c1	n5=10	
	c2	n5=11	
	c3	n5=12	
	c4	n5=13	
	p'	n5=14	
S5={Ø}, n5=14 , F1={x2,x4}			

p	q	np	Sp
x6	x1		S6={x1}
	x2		
	x3		S6={x1,x3}
	x4		
	x5		S6={x1,x3,x5}
	x7		S6={x1,x3,x5,x7}
	x8		S6={x1,x3,x5,x7,x8}
	x9		S6={x1,x3,x5,x7,x8,x9}
	x10		S6={x1,x3,x5,x7,x8,x9,x10}
	c1	n6=1	
	c2		S6={x1,x3,x5,x7,x8,x9,x10,c2}
	c3		
	c4		S6={x1,x3,x5,x7,x8,x9,x10,c2,c4}
	p'		S6={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}
S6={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}, n6=1 , F1={x2,x4}			

p	q	np	Sp
x7	x1	n7=1	
	x2	n7=2	
	x3	n7=3	
	x4	n7=4	
	x5		S7={x5}
	x6	n7=5	
	x8	n7=6	
	x9		S7={x5,x9}
	x10	n7=7	
	c1	n7=8	
	c2	n7=9	
	c3	n7=10	
	c4	n7=11	
	p'	n7=12	
S7={x5,x9}, n7=12 , F1={x2,x4}			

p	q	np	Sp
x8	x1		S8={x1}
	x2	n8=1	
	x3		S8={x1,x3}
	x4	n8=2	
	x5		S8={x1,x3,x5}
	x6	n8=3	
	x7		S8={x1,x3,x5,x7}
	x9		S8={x1,x3,x5,x7,x9}
	x10		S8={x1,x3,x5,x7,x9,x10}
	c1	n8=4	
	c2	n8=5	
	c3	n8=6	
	c4		S8={x1,x3,x5,x7,x9,x10,c4}
	p'		S8={x1,x3,x5,x7,x9,x10,c4,*p'*}
S8={x1,x3,x5,x7,x9,x10,c4,*p'*}, n8=6 , F1={x2,x4}			

p	q	np	Sp
x9	x1	n9=1	
	x2	n9=2	
	x3	n9=3	
	x4	n9=4	
	x5		S9={x5}
	x6	n9=5	
	x8	n9=6	
	x9	n9=7	
	x10	n9=8	
	c1	n9=9	
	c2	n9=10	
	c3	n9=11	
	c4	n9=12	
	p'	n9=13	
S9={x5}, n9=13 , F1={x2,x4}			

p	q	np	Sp
x10	x1	n10=1	
	x2	n10=2	
	x3	n10=3	
	x4	n10=4	
	x5		S10={x5}
	x6	n10=5	
	x7		S10={x5,x7}
	x8	n10=6	
	x9		S10={x5,x7,x9}
	c1	n10=7	
	c2	n10=8	
	c3	n10=9	
	c4	n10=10	
	p'	n10=11	
S10={x5,x7,x9}, n10=11 , F1={x2,x4}			

p	q	np	Sp
c1	x1		S11={x1}
	x2		
	x3		S11={x1,x3}
	x4		
	x5		S11={x1,x3,x5}
	x6		S11={x1,x3,x5,x6}
	x7		S11={x1,x3,x5,x6,x7}
	x8		S11={x1,x3,x5,x6,x7,x8}
	x9		S11={x1,x3,x5,x6,x7,x8,x9}
	x10		S11={x1,x3,x5,x6,x7,x8,x9,x10}
	c2		S11={x1,x3,x5,x6,x7,x8,x9,x10,c2}
	c3		S11={x1,x3,x5,x6,x7,x8,x9,x10,c2,c3}
	c4		S11={x1,x3,x5,x6,x7,x8,x9,x10,c2,c3,c4}
	p'		S11={x1,x3,x5,x6,x7,x8,x9,x10,c2,c3,c4,*p'*}
S11={x1,x3,x5,x6,x7,x8,x9,x10,c2,c3,c4,*p'*}, n11=0 , F1={x2,x4,c1}			

p	q	np	Sp
c2	x1		S12={x1}
	x2	n12=1	
	x3		S12={x1,x3}
	x4	n12=2	
	x5		S12={x1,x3,x5}
	x6	n12=3	
	x7		S12={x1,x3,x5,x7}
	x8		S12={x1,x3,x5,x7,x8}
	x9		S12={x1,x3,x5,x7,x8,x9}
	x10		S12={x1,x3,x5,x7,x8,x9,x10}
	c1	n13=4	
	c3	n13=5	
	c4		S12={x1,x3,x5,x7,x8,x9,x10,c4}
	p'		S12={x1,x3,x5,x7,x8,x9,x10,c4,*p'*}
S12={x1,x3,x5,x7,x8,x9,x10,c4,*p'*}, n12=5 , F1={x2,x4,c1}			

p	q	np	Sp
c3	x1		S13={x1}
	x2		
	x3		S13={x1,x3}
	x4		
	x5		S13={x1,x3,x5}
	x6		
	x7		S13={x1,x3,x5,x7}
	x8		S13={x1,x3,x5,x7,x8}
	x9		S13={x1,x3,x5,x7,x8,x9}
	x10		S13={x1,x3,x5,x7,x8,x9,x10}
	c1	n13=1	
	c2		S13={x1,x3,x5,x7,x8,x9,x10,c2}
	c4		S13={x1,x3,x5,x7,x8,x9,x10,c2,c4}
	p'		S13={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}
S13={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}, n13=1, F1={x2,x4,c1}			

p	q	np	Sp
O4	x1	n14=1	
	x2	n14=2	
	x3	n14=3	
	x4	n14=4	
	x5		S14={x5}
	x6	n14=5	
	x7		S14={x5,x7}
	x8	n14=6	
	x9		S14={x5,x7,x9}
	x10		S14={x5,x7,x9,x10}
	c1	n14=7	
	c2	n14=8	
	c3	n14=9	
	p'	n14=10	
S14={x5,x7,x9,x10}, n14=10, F1={x2,x4,c1}			

p	q	np	Sp
p'	x1		S15={x1}
	x2	n15=1	
	x3		S15={x1,x3}
	x4	n15=2	
	x5		S15={x1,x3,x5}
	x6	n15=3	
	x7		S15={x1,x3,x5,x7}
	x8	n15=4	
	x9		S15={x1,x3,x5,x7,x9}
	x10		S15={x1,x3,x5,x7,x9,x10}
	c1	n15=5	
	c2	n15=6	
	c3	n15=7	
	c4		S15={x1,x3,x5,x7,x9,x10,c4}
S15={x1,x3,x5,x7,x9,x10,c4}, n15=7, **F1={x2,x4,c1}**			

First front found

F1:	x2
	x4
	c1

Defining other fronts:

p	q	nq	nq-1	Qrank	Q
x2	x1	9	8		
	x3	8	7		
	x5	14	13		
	x7	12	11		
	x8	6	5		
	x9	13	12		
	x10	11	10		
	c2	5	4		
	c4	10	9		
	p'	7	6		
S2={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
x4	x1	8	7		
	x3	7	6		
	x5	13	12		
	x7	11	10		
	x8	5	4		
	x9	12	11		
	x10	10	9		
	c2	4	3		
	c4	9	8		
	p'	6	5		
S4={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
c1	x1	7	6		
	x3	6	5		
	x5	12	11		
	x6	1	0	1+1=2	**F2**
	x7	10	9		
	x8	4	3		
	x9	11	10		
	x10	9	8		
	c2	3	2		
	c3	1	0	1+1=2	**F2**
	c4	8	7		
	p'	5	4		
S11={x1,x3,x5,x6,x7,x8,x9,x10,c2,c3,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
x6	x1	6	5		
	x3	5	4		
	x5	11	10		
	x7	9	8		
	x8	3	2		
	x9	10	9		
	x10	8	7		
	c2	2	1		
	c4	7	6		
	p'	4	3		
S6={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
c3	x1	5	4		
	x3	4	3		
	x5	10	9		
	x7	8	7		
	x8	2	1		
	x9	9	8		
	x10	7	6		
	c2	1	0	2+1=3	**F3**
	c4	6	5		
	p'	3	2		
S13={x1,x3,x5,x7,x8,x9,x10,c2,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
c2	**x1**	4	3		
	x3	3	2		
	x5	9	8		
	x7	7	6		
	x8	**1**	**0**	**3+1=4**	**F4**
	x9	8	7		
	x10	6	5		
	c4	5	4		
	p'	2	1		
S12={x1,x3,x5,x7,x8,x9,x10,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
x8	x1	3	2		
	x3	2	1		
	x5	8	7		
	x7	6	5		
	x9	7	6		
	x10	5	4		
	c4	4	3		
	p'	**1**	**0**	**4+1=5**	**F5**
S8={x1,x3,x5,x7,x9,x10,c4,*p'*}					

p	q	nq	nq-1	Qrank	Q
p'	x1	2	1		
	x3	**1**	**0**	**5+1=6**	**F6**
	x5	7	6		
	x7	5	4		
	x9	6	5		
	x10	4	3		
	c4	3	2		
S15={x1,x3,x5,x7,x9,x10,c4}					

p	q	nq	nq-1	Qrank	Q
x3	**x1**	**1**	**0**	**6+1=7**	**F7**
	x5	6	5		
	x7	4	3		
	x9	5	4		
	x10	3	2		
	c4	2	1		
S3={x1,x5,x7,x9,x10,c4}					

p	q	nq	nq-1	Qrank	Q
x1	x5	5	4		
	x7	3	2		
	x9	4	3		
	x10	2	1		
	c4	**1**	**0**	**7+1=8**	**F8**
S1={x5,x7,x9,x10,c4}					

p	q	nq	nq-1	Qrank	Q
c4	x5	4	3		
	x7	2	1		
	x9	3	2		
	x10	**1**	**0**	**8+1=9**	**F9**
S14={x5,x7,x9,x10}					

p	q	nq	nq-1	Qrank	Q
x10	x5	3	2		
	x7	**1**	**0**	**9+1=10**	**F10**
	x9	2	1		
S10={x5,x7,x9}					

p	q	nq	nq-1	Qrank	Q
x7	x5	2	1		
	x9	**1**	**0**	**10+1=11**	**F11**
S7={x5,x9}					

p	q	nq	nq-1	Qrank	Q
x9	**x5**	**1**	**0**	**11+1=12**	**F12**
S9={x5}					

Summary
F1: x2, x4, c1 F2: x6, c3 F3: c2 F4: x8
F5: p' F6: x3 F7: x1 F8: c4 F9: x10
F10: x7 F11: x9 F12: x5

Crowding Distance Assignment ($\mathfrak{T}$)

start $\mathfrak{T}$ *procedure*
$l = |\mathfrak{T}|$ number of solutions in $\mathfrak{T}$
for each i, *set* $\mathfrak{T}[i]_{distance} = 0$ initialize distance
for each objective m
$\mathfrak{T} = sort(\mathfrak{T}, m)$ sort using each objective value
$\mathfrak{T}[1]_{distance} = \mathfrak{T}[l]_{distance} = \infty$ so that boundary points are always selected
for $i = 2$ *to* $(l-1)$ for all other points

$$\mathfrak{T}[i]_{distance} = \mathfrak{T}[i]_{distance} + \frac{\mathfrak{T}[i+1].m - \mathfrak{T}[i-1].m}{f_m^{max} - f_m^{min}}$$

End

Sort according to first objective f1(x)		Sort according to second objective f2(x)		Crowding Distance Assignment			
				$\frac{(i+1)-(i-1)}{(l_{max})-(l_{min})}$		$l_{distance} + \frac{(i+1)-(i-1)}{(l_{max})-(l_{min})}$	
c2	18,36	x7	-375,63	c2	∞	x7	∞
x8	22,34	x9	-370,77	x8	0,04	x9	0,32
c1	40,66	x1	-339,84	c1	0,20	x1	0,25
x6	133,87	x3	-166,65	x6	0,17	x3	0,47
c3	136,48	p'	-163,25	c3	0,07	p'	0,05
x4	173,86	x2	-149,22	x4	0,09	x2	0,23
x2	185,21	x4	-127,18	x2	0,19	x4	0,14
c4	284,13	x5	-99,13	c4	0,18	x5	∞
x10	287,95	c1	-2,00	x10	0,10	c1	0,42
x1	340,43	x10	42,32	x1	0,18	x10	0,20
x7	389,14	c4	42,94	x7	0,10	c4	0,18
p'	396,02	x8	53,97	p'	0,04	x8	0,06
x3	413,20	c2	54,19	x3	0,07	c2	∞
x9	437,43	c3	60,62	x9	0,31	c3	0,08
x5	589,06	x6	60,76	x5	∞	x6	∞

SRN	Parameters xi = [-20,20]		Fitness Function		Constraint Function		V1(x)= 225-g1(x)	If V1(x)≥0, CV=0 else, CV=V1(x)	V2(x)= -10-g2(x)	If V2(x)≥0, CV=0 else, CV=V2(x)	If CV = 0, feasible else, infeasible	Non-d Sorting	Crowding Distance
	i1	i2	f1(x)	f2(x)	g1(x)	g2(x)	V1(x)	CV g1(x)	V2(x)	CV g2(x)	CV sum	rank	$\mathfrak{T}$
x1	-3,42	-16,58	340,43	-339,84	286,59	46,32	-61,59	-61,59	-56,32	-56,32	117,91	F7	0,25
x2	3,52	14,45	185,21	-149,22	221,19	-39,83	3,81	0,00	29,83	0,00	0,00	F1	0,23
x3	-18,21	2,66	413,20	-166,65	338,68	-26,19	-113,68	-113,68	16,19	0,00	113,68	F6	0,47
x4	-9,35	7,56	173,86	-127,18	144,58	-32,03	80,42	0,00	22,03	0,00	0,00	F1	0,14
x5	19,64	17,61	589,06	-99,13	695,84	-33,19	-470,84	-470,84	23,19	0,00	470,84	F12	∞
x6	11,46	7,51	133,87	60,76	187,73	-11,07	37,27	0,00	1,07	0,00	0,00	F2	∞
x7	1,21	-18,66	389,14	-375,63	349,66	57,19	-124,66	-124,66	-67,19	-67,19	191,85	F10	∞
x8	6,25	-0,51	22,34	53,97	39,32	7,78	185,68	0,00	-17,78	-17,78	17,78	F4	0,06
x9	5,68	-19,54	437,43	-370,77	414,07	64,30	-189,07	-189,07	-74,30	-74,30	263,37	F11	0,32
x10	15,68	10,94	287,95	42,32	365,55	-17,14	-140,55	-140,55	7,14	0,00	140,55	F9	0,20
c1	3,74	6,97	40,66	-2,00	62,55	-17,17	162,45	0,00	7,17	0,00	0,00	F1	0,42
c2	6,03	0,68	18,36	54,19	36,86	3,99	188,14	0,00	-13,99	-13,99	13,99	F3	∞
c3	11,55	7,58	136,48	60,62	190,84	-11,20	34,16	0,00	1,20	0,00	0,00	F2	0,08
c4	15,59	10,87	284,13	42,94	361,23	-17,01	-136,23	-136,23	7,01	0,00	136,23	F8	0,18
p'	-17,76	2,84	396,02	-163,25	323,63	-26,27	-98,63	-98,63	16,27	0,00	98,63	F5	0,05

SRN	Parameters xi = [-20,20]		Fitness Function		Constraint Function		V1(x)= 225-g1(x)	If V1(x)≥0, CV=0 else, CV=V1(x)	V2(x)= -10-g2(x)	If V2(x)≥0, CV=0 else, CV=V2(x)	If CV = 0, feasible else, infeasible	Non-d Sorting	Crowding Distance
	i1	i2	f1(x)	f2(x)	g1(x)	g2(x)	V1(x)	CV g1(x)	V2(x)	CV g2(x)	CV sum	rank	$\mathfrak{T}$
c1	3,74	6,97	40,66	-2,00	62,55	-17,17	162,45	0,00	7,17	0,00	0,00	F1	0,42
x2	3,52	14,45	185,21	-149,22	221,19	-39,83	3,81	0,00	29,83	0,00	0,00	F1	0,23
x4	-9,35	7,56	173,86	-127,18	144,58	-32,03	80,42	0,00	22,03	0,00	0,00	F1	0,14
x6	11,46	7,51	133,87	60,76	187,73	-11,07	37,27	0,00	1,07	0,00	0,00	F2	∞
c3	11,55	7,58	136,48	60,62	190,84	-11,20	34,16	0,00	1,20	0,00	0,00	F2	0,08
c2	6,03	0,68	18,36	54,19	36,86	3,99	188,14	0,00	-13,99	-13,99	13,99	F3	∞
x8	6,25	-0,51	22,34	53,97	39,32	7,78	185,68	0,00	-17,78	-17,78	17,78	F4	0,06
p'	-17,76	2,84	396,02	-163,25	323,63	-26,27	-98,63	-98,63	16,27	0,00	98,63	F5	0,05
x3	-18,21	2,66	413,20	-166,65	338,68	-26,19	-113,68	-113,68	16,19	0,00	113,68	F6	0,47
x1	-3,42	-16,58	340,43	-339,84	286,59	46,32	-61,59	-61,59	-56,32	-56,32	117,91	F7	0,25
c4	15,59	10,87	284,13	42,94	361,23	-17,01	-136,23	-136,23	7,01	0,00	136,23	F8	0,18
x10	15,68	10,94	287,95	42,32	365,55	-17,14	-140,55	-140,55	7,14	0,00	140,55	F9	0,20
x7	1,21	-18,66	389,14	-375,63	349,66	57,19	-124,66	-124,66	-67,19	-67,19	191,85	F10	∞
x9	5,68	-19,54	437,43	-370,77	414,07	64,30	-189,07	-189,07	-74,30	-74,30	263,37	F11	0,32
x5	19,64	17,61	589,06	-99,13	695,84	-33,19	-470,84	-470,84	23,19	0,00	470,84	F12	∞

Next Generation

SRN		Parameters xi = [-20,20]		Fitness Function		Constraint Function		V1(x) = 225-g1(x)	If v1(x) ≥ 0, CV = 0 else, CV = V1(x)	V2(x) = -10-g2(x)	If V2(x) ≥ 0, CV = 0 else, CV = V2(x)	If CV = 0, feasible else, infeasible
Previous Individuals	New Individuals	i1	i2	f1(x)	f2(x)	g1(x)	g2(x)	V1(x)	CV g1(x)	V2(x)	CV g2(x)	CV sum
c1	x1	3,74	6,97	40,66	-2,00	62,55	-17,17	162,45	0,00	7,17	0,00	0,00
x2	x2	3,52	14,45	185,21	-149,22	221,19	-39,83	3,81	0,00	29,83	0,00	0,00
x4	x3	-9,35	7,56	173,86	-127,18	144,58	-32,03	80,42	0,00	22,03	0,00	0,00
x6	x4	11,46	7,51	133,87	60,76	187,73	-11,07	37,27	0,00	1,07	0,00	0,00
c3	x5	11,55	7,58	136,48	60,62	190,84	-11,20	34,16	0,00	1,20	0,00	0,00
c2	x6	6,03	0,68	18,36	54,19	36,86	3,99	188,14	0,00	-13,99	-13,99	13,99
x8	x7	6,25	-0,51	22,34	53,97	39,32	7,78	185,68	0,00	-17,78	-17,78	17,78
p'	x8	-17,76	2,84	396,02	-163,25	323,63	-26,27	-98,63	-98,63	16,27	0,00	98,63
x3	x9	-18,21	2,66	413,20	-166,65	338,68	-26,19	-113,68	-113,68	16,19	0,00	113,68
x1	x10	-3,42	-16,58	340,43	-339,84	286,59	46,32	-61,59	-61,59	-56,32	-56,32	117,91

References

1. Chiang, W., & Shu, Y. (2003). Acoustical design of stages with large plane surfaces in rectangular recital halls. *Applied Acoustics, 64,* 863–884.
2. Martin, R., & Arana, M. (2006). Predicted and experimental results of acoustic parameters in the new Symphony Hall in Pamplona. *Spain, Applied Acoustics, 67,* 1–14.
3. Cerda, S., Gimenez, A., Romero, J., Cibrian, R., & Miralles, J. L. (2009). Room acoustical parameters: A factor analysis approach. *Applied Acoustics, 70,* 97–109.
4. Akama, T., Suzuki, H., & Omoto, A. (2010). Distribution of selected monaural acoustical parameters in concert halls. *Applied Acoustics, 71,* 564–577.
5. Brothanek, M., Jandak, V., Jiricek, O., & Svec, P. (2012). Monaural and binaural parameters of Rudolfinum concert halls in Prague. *Applied Acoustics, 73,* 1201–1208.
6. Sabine, W. C., & Egan, M. D. (1994). Collected papers on acoustics. *The Journal of the Acoustical Society of America, 95*(6), 3679–3680.
7. Sato, S., Hayashi, T., Takizawa, A., Tani, A., Kawamura, H., & Ando, Y. (2004). Acoustic design of theatres applying genetic algorithms. *Journal of Temporal Design in Architecture and the Environment*, *4*(1).
8. Sato, S., Otori, K., Takizawa, A., Sakai, H., Ando, Y., & Kawamura, H. (2002). Applying genetic algorithms to the optimum design of a concert hall. *Journal of Sound and Vibration, 258,* 517–526.
9. Bassuet, A., & Dellatorre, L. (2014). Computational and optimization design in geometric acoustics. *Building Acoustics, 21*(1), 75–85.
10. Cubukcuoglu, C., Kirimtat, A., Tasgetiren, M. F., Suganthan, P., & Pan, Q.-K. (2016). Multiobjective harmony search algorithm for layout design in theatre hall acoustics. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (Vol. 2016, pp. 2280–2287).
11. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6,* 182–197.
12. Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. In *Eurogen* (Vol. 2001, pp. 95–100).
13. Storn, R., & Price, K. (1995). *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical report, Berkeley, CA, Technical Report TR-95–012. [2] Storn, R. & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *The Journal of Global Optimization*, *11*(4), 341–359.
14. Price, K., Storn, R., & Lampinen, J. (2005). *Differential evolution: A practical approach toglobal optimization*. Berlin: Springer.
15. Mullick, S. S., Suganthan, P. N., Das, S. (2016). Recent advances in differential evolution—an updated survey. *Swarm and Evolutionary Computation*.
16. Brest, J., Greiner, S., Boskovic, B., Mernik, M., & ˇZumer, V. (2006). Self adapting control parameters in differential evolution: a comparative study on numerical Benchmark problems. *IEEE Transactions on Evolutionary Computation*, *10*(6), 646–657.
17. Brest, J. (2009). Constrained real-parameter optimization with e-self-adaptive differential evolution. In E. Mezura-Montes (Ed.), *Constraint-handling in evolutionary optimization*. *Studies in computational intelligence series* (Vol. 198). Springer.
18. Zamuda, A., Brest, J., Boskovic, B., & Zumer, V. (2007). Differential evolution for multiobjective optimization with self-adaptation. In *The 2007 IEEE Congress on Evolutionary Computation CEC2007* (pp. 3617–3624).
19. Robic, T., & Filipic, B. (2005). DEMO: Differential evolution for multiobjective optimization. In *Proceedings of the Third International Conference on Evolutionary Multi-criterion Optimization—EMO 2005*, ser. Lecture Notes in Computer Science (Vol. 3410, pp. 520–533). Springer.
20. Tusar, T., & Filipic, B. (2007). Differential evolution versus genetic algorithms in multiobjective optimization, in *Proceedings of the Fourth International Conference on Evolutionary*

Multi-criterion Optimization—EMO 2007, ser. Lecture Notes in Computer Science (Vol. 4403, pp. 257–271). Springer.
21. Sabine, W. C. (1900). *Reverberation*. The American Architect.
22. Aretz, Marc, & Orlowski, Raf. (2009). Sound strength and reverberation time in small concert halls. *Applied Acoustics, 70*(8), 1099–1110.
23. Parkin, P. H.; Scholes, W. E., & Derbyshire, A. G. (1952). *The reverberation times of ten British concert halls* (Vol. 2, Number 3, pp. 97–100(4)). Acta Acustica united with Acustica.

Structural Optimization for Masonry Shell Design Using Multi-objective Evolutionary Algorithms

Esra Cevizci, Seckin Kutucu, Mauricio Morales-Beltran, Berk Ekici and M. Fatih Tasgetiren

Abstract In this study, the implementation of evolutionary algorithms to the form-finding problem of masonry shell models is presented using Autoclaved Aerated Concrete material. Regarding the significance of design decisions, the study is focused on the conceptual stage of the design process. In this context, the applied method is addressed as multi-objective real-parameter constrained optimization. For the sake of dealing with the shell design problem, two objective functions are considered: minimization of global displacement and minimization of mass. Two multi-objective evolutionary algorithms, namely, Non-Dominated Sorting Genetic Algorithm II and Real-coded Genetic Algorithm with mutation strategy of Differential Evolution Algorithms are compared in terms of computational and architectural performance. As a result, the solutions generated by these algorithms are found much competitive.

E. Cevizci · S. Kutucu (✉) · M. Morales-Beltran · B. Ekici
Department of Architecture, Yasar University, Selcuk Yasar Campus, İzmir, Turkey
e-mail: seckin.kutucu@yasar.edu.tr

E. Cevizci
e-mail: esracevizci@outlook.com

M. Morales-Beltran
e-mail: mauricio.morales-beltran@yasar.edu.tr; M.G.MoralesBeltran@tudelft.nl

B. Ekici
e-mail: berk.ekici@yasar.edu.tr; B.Ekici-1@tudelft.nl

M. Morales-Beltran
Structural Design, AE+T, TU Delft, Delft, The Netherlands

B. Ekici
Chair of Design Informatics, TU Delft, Delft, The Netherlands

M. Fatih Tasgetiren
Department of International Logistics Management, Yasar University,
Selcuk Yasar Campus, İzmir, Turkey
e-mail: fatih.tasgetiren@yasar.edu.tr

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management
and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_5

Keywords Architectural design · Form-finding · Parametric design
Computational design · Optimization · Evolutionary algorithms
Asymmetric shells structures

1 Introduction

The Masonry shells have an important place in the architecture over the years. They are one of the most common buildings of the world's architectural heritage. As of today, masonry shells are still present in the agenda of contemporary architecture. Nevertheless, the design of these structures is considered a complex task because it requires deep structural knowledge [1]. Historically speaking, basic form-finding methods, such as hanging chain, graphic statics, and physical modeling have been influential but not an efficient approach for the form-finding task.

In the course of time, shell structures have become part of the landscape of buildings, and thus they are more familiar to human beings. By serving to complex purposes, shell structures entail more requirements that make them even more complex to solve. In order to deal with this associated complexity, the exploration of the structural system within the shell geometry becomes the key to find answers. The introduction of computational modeling techniques in the last two decades proposed new approaches to the design of these structures in architecture [2–5].

Autoclaved Aerated Concrete (AAC) is a highly efficient material for application and construction, in terms of compressive stress resistance, lightweight, porosity, and heat impermeability. It is an alternative to materials such as stone and brick for the masonry buildings, in terms of insulation and lightweight for architectural constructions. Beyond being produced primarily as a brick with insulation properties, AAC possesses interesting structural characteristics: it is lightweight, it provides efficiency in materiality, and it presents potential for a trash-wise use, like pulverized fuel ash [6]. Moreover, AAC is known as a high-quality material that has been extensively used as walling element for residential, commercial, and industrial buildings. However, there are limited studies that have been made for ACC in the field of designing or constructing shell structures [7–10].

Generally speaking, structures are designed to be safe, stable, economic, and if possible, light. In stiffness-based design methods, the increase of stiffness is necessary to reduce the structural response, e.g., displacements. Such an increase is achieved at expense of increasing the size of structural elements, which in turn, increases its mass. This means that while designing safer structures, they usually become heavier. This extra mass will attract more load over the structure, hence requiring more stiffness. So in order to find a balanced trade-off, these two conflicting aspects should be considered within the optimization problem. The minimization of both mass (m) and displacement (v) must be introduced then in the objective function.

This study focuses on understanding and evaluating the performance of the two evolutionary algorithms in developing three different freeform masonry shells as three different cases by using AAC blocks. Departing from vaults, which are self-supporting systems and stable through their shape formed by funicular arches, the question is how to link structural stability with geometry. Therefore, the main goal is to present a set of alternatives for decision maker considering design of these funicular shells. To do this, the minimization of both v and m are considered as the objective functions, subjected to constraints related with space quality and structural safety. The study is framed as a problem on multi-objective real-parameter constrained optimization for the conflicting objectives. To cope with the problem, Real-coded Genetic Algorithm with mutation strategy of Differential Evolution Algorithm (CGA_DE) is proposed and compared with well-known Non-Dominated Sorting Genetic Algorithm II (NSGA_II) .

The rest of the study is structured as follows: Sect. 2 explains the problem formulation of the study including notations. Section 3 clarifies the development of the model. Section 4 presents the two multi-objective evolutionary algorithms (MOEAs) used for the study. Section 5 remarks the computational results of each case study. Section 6 clarifies discussion of the cases and finally Sect. 7 states the conclusion and future remarks of the research.

2 Problem Definition

The aim of the study is to explore alternative funicular shell structures. For this reason, two important objectives v and m are minimized under several load conditions for three different masonry structure cases. By generating three different shell models for three different cases, it is aimed to study the masonry shell optimization on more than one architectural case and test the algorithms in variety of shaped models.

In case 1 ($C1$), the shell system has a projected triangular shaped topology on the ground. For case 2 ($C2$), the topology is considered to be projected as rectangular shaped. Consequently, in case 3 ($C3$), the topology is designed as a projected pentagon. Points of the corners of these topologies are set and two directions of each point are determined as decision variables. For the first direction, points are denoted as $P_{1,a}$, $P_{2,a}$ and so on, whereas for the second direction, notation is defined as $P_{1,b}$, $P_{2,b}$ and so forth.

By the arrangement of the directions and points, the pillars of the shell where they stand on the ground are defined. Between each point located on the same direction, catenary chains are assigned. The length LC and the rotation angle RC of these catenary chains are determined as decision variables as well as the thickness of the shell material MT. Due to different base geometries, different amount of decision variables are assigned to each case study. There are 13 decision variables for $C1$, 17 decision variables for $C2$ and 21 decision variables for $C3$. By this way, optimization problems, which present various complexities, are considered. Figure 1

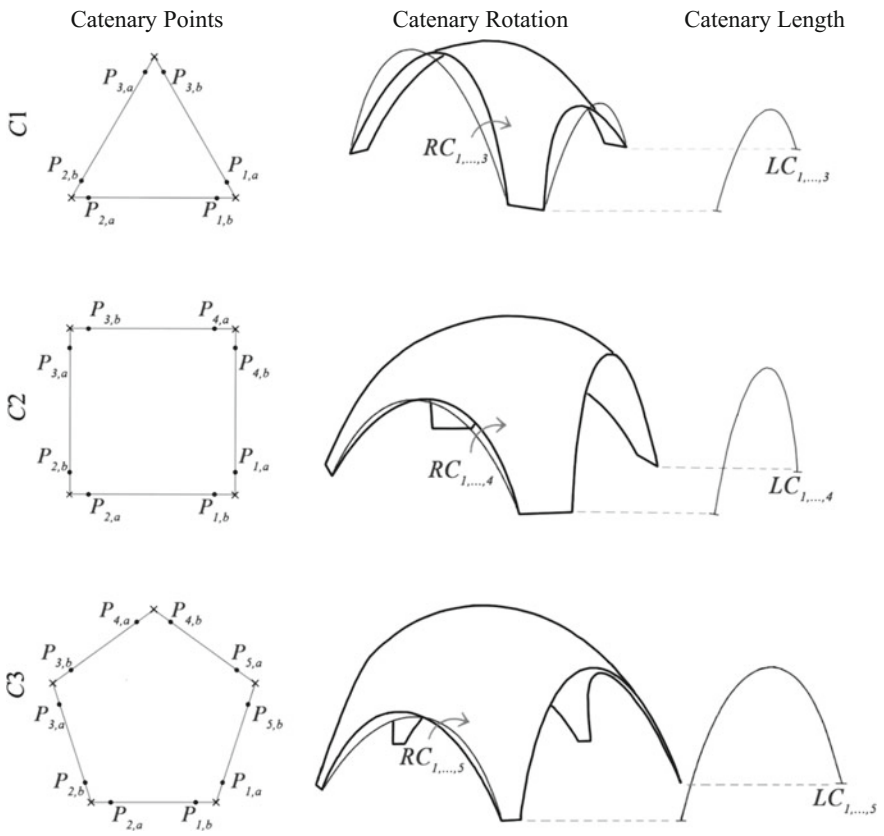


Fig. 1 Funicular shell diagrams of each case

shows the illustration of decision variables explained above for each $C1$,$C2$, and $C3$, respectively.

2.1 Objective Functions

Objective functions, which are minimizing both v and m subject to several constraints, are formulated as follows:
Min (v, m)
where v and m are given by

$$v = \frac{F}{K} \tag{1}$$

$$m = \sum_{a=1}^{b} W E_a \tag{2}$$

where F is the loading force, K is the bending stiffness of the shell structure, WE_a is the specific weight of ath element of shell structure and b is total number of the shell structure elements.

Subject to:

$$v \leq \frac{H}{50} \tag{3}$$

$$H \geq 2.5\,\text{m} \tag{4}$$

$$H_i \geq 2 \quad \text{and} \quad H_j \geq 2 \tag{5}$$

where H is the total height of the shell structure, H_i and H_j are the heights of two catenaries in each case study. To calculate the first constraint in Eq. (3), first of all, the distance of the highest point of the shell to the ground H is measured. Each generated shell structure performing a v value. With this constraint it is defined to be 50 times smaller than the H for each generated shell design alternative. In Eq. (4), a constraint for the highest part of the shell structure H is also determined. The reason for this is to propose livable spaces for human beings. Regarding this, another constraint function is also considered. In Eq. (5) i and j correspond two different catenaries in each case study. For i and j, minimum height is determined as 2 meters in $C1$, $C2$, and $C3$, respectively, as follows:

$$i \neq j, i = 1, \ldots, 3 \; and \; j = 1, \ldots, 3 \text{ for Case 1 } (C1) \tag{6}$$

$$i \neq j, \; i = 1, \ldots, 4 \; and \; j = 1, \ldots, 4 \text{ for Case 2 } (C2) \tag{7}$$

$$i \neq j, \; i = 1, \ldots, 5 \; and \; j = 1, \ldots, 5 \text{ for Case 3 } (C3) \tag{8}$$

Decision variables for each case $C1$, $C2$, $C3$ are given in Table 1.

2.2 Displacement Calculations of the Structure

In order to find the equilibrium state of the shell, the vector of v is calculated by the help of Karamba [11], which is a 3D plugin tool that is helpful in finite elements calculations of static problems in Grasshopper 3D (GH). The equation as follows is used:

$$F = K \times v \tag{9}$$

Table 1 Decision variables

Not.	Explanation	Type	Range	$C1$	$C2$	$C3$
$P_{1,a}$	First direction for Point 1	Real	[0.30 m, 1.50 m]	✓	✓	✓
$P_{1,b}$	Second direction for Point 1	Real	[0.30 m, 1.50 m]	✓	✓	✓
$P_{2,a}$	First direction for Point 2	Real	[0.30 m, 1.50 m]	✓	✓	✓
$P_{2,b}$	Second direction for Point 2	Real	[0.30 m, 1.50 m]	✓	✓	✓
$P_{3,a}$	First direction for Point 3	Real	[0.30 m, 1.50 m]	✓	✓	✓
$P_{3,b}$	Second direction for Point 3	Real	[0.30 m, 1.50 m]	✓	✓	✓
$P_{4,a}$	First direction for Point 4	Real	[0.30 m, 1.50 m]		✓	✓
$P_{4,b}$	Second direction for Point 4	Real	[0.30 m, 1.50 m]		✓	✓
$P_{5,a}$	First direction for Point 5	Real	[0.30 m, 1.50 m]			✓
$P_{5,b}$	Second direction for Point 5	Real	[0.30 m, 1.50 m]			✓
LC_1	Length of Catenary Chain 1	Real	[0.00 m, 10.00 m]	✓	✓	✓
LC_2	Length of Catenary Chain 2	Real	[0.00 m, 10.00 m]	✓	✓	✓
LC_3	Length of Catenary Chain 3	Real	[0.00 m, 10.00 m]	✓	✓	✓
LC_4	Length of Catenary Chain 4	Real	[0.00 m, 10.00 m]		✓	✓

(continued)

Table 1 (continued)

Not.	Explanation	Type	Range	$C1$	$C2$	$C3$
LC_5	Length of Catenary Chain 5	Real	[0.00 m, 10.00 m]			✓
RC_1	Rotation of Catenary Chain 1	Real	[*0.00°*, *20.00°*]	✓	✓	✓
RC_2	Rotation of Catenary Chain 2	Real	[*0.00°*, *20.00°*]	✓	✓	✓
RC_3	Rotation of Catenary Chain 3	Real	[*0.00°*, *20.00°*]	✓	✓	✓
RC_4	Rotation of Catenary Chain 4	Real	[*0.00°*, *20.00°*]		✓	✓
RC_5	Rotation of Catenary Chain 5	Real	[*0.00°*, *20.00°*]			✓
MT	Shell Thickness	Real	[5.00 cm, 20.00 cm]	✓	✓	✓

Self-weight of all architectural elements used for the construction is the permanent actions. Furthermore, imposed loads such as the pressure of the wind and the load of the snow are variable actions. As reported in Eurocode EN 1990, general formula for the action combinations is given as [12]:

$$\sum_j \gamma_{G,j} \times G_{k,j} + \gamma_{Q,1} \times Q_{k,1} + \sum_{i>1} \gamma_{Q,i} \times \Psi_{0,i} \times Q_{k,i} \tag{10}$$

where G_k represents the permanent actions and Q_k represents the variable actions on structures. According to Eurocode, for the combination of G_k and Q_k, partial and reduction factors are defined based on the low chance of having, both wind and snow action at full magnitude at the same time. The partial factor for $G_{k,}$ is γ_G, the partial factor for Q_k is γ_Q and the combination factor is Ψ_0. Hence, numerous combinations are probable for the loading of the construction. $Q_{k,1}$ is considered as the leading variable action and $Q_{k,i}$ is considered as the other variable actions.

For this study, the loads acting on the structure is considered as dead load (DL), seismic load (E) and live load (LL) and the load combination is determined as follows:

$$F = (1.35 \times DL) + (0.9 \times LL) \pm \mathrm{E} \tag{11}$$

Two different load combinations are considered due to two independent wind directions. In order to calculate these load combinations, DL, E, and LL need to be

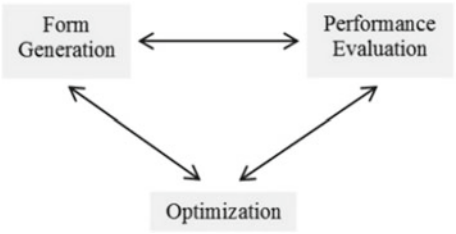


Fig. 2 PDCD process [16]

estimated first. Considered *DL* of the structure is the AAC blocks material weight times the volume of the structure (*VS*), which is retrieved from the computational model for each generated shell design. As a conservative approach, *LL* for this shell structure is taken 2 kN/m^2 and *E* is accepted as 20% of *DL* of the structure. Calculation of *DL* and *E* is presented as follows:

$$DL = VS \times \sum_{a=1}^{b} WE_m \tag{12}$$

$$E = 0.2 \times DL \tag{13}$$

3 Parametric Model

Parametric modeling is used for form-finding of the shell geometry. This model is developed in Rhinoceros 3D [13, 14] modeling software with Grasshopper 3D (GH) [15], which has a graphical algorithmic interface running as a plugin in Rhinoceros. In this parametric environment, the shell models of each case have been generated in three steps: form generation, performance evaluation and optimization. These are the main phases of the study, which constitute the algorithmic design methodology and cover an iterated loop. The method can be articulated as one of the essential manners in the research field of computational design which is named as "*Performance Driven Conceptual Design*" (*PDCD*) process [16]. Form generation is the first step and includes the generative development of shell model. Performance evaluation is the second step and contains objective functions and constraints. Finally, optimization step is the third step and deals with the evolutionary computation. The method aims to maximize the advantages and minimize the disadvantages of earlier assessments, obtained in the conceptual period of the design. Figure 2 shows the loop of PDCD process.

For the performance evaluation step of the design method, objective functions and constraints are organized numerically. According to the working principle of GH, outcomes of objective functions and constraints of shell design are based on all decision variable combinations. For that reason, optimization step is performed to find an appropriate Pareto-front. For this reason, NSGA_II and CGA_DE algorithms are

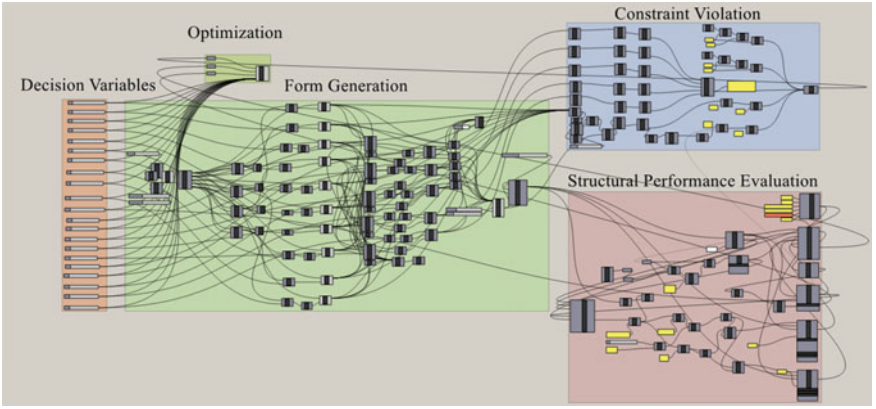


Fig. 3 Grasshopper model of the study

used to link the decision variables, constraints, and objective functions of the study. The working environment is shown in Fig. 3.

4 Algorithms

The problem is addressed to find the Pareto-front set for shell design using computational optimization. Concerning this, two multi-objective optimization algorithms are performed to find the results. In the current literature, various studies are published about multi-objective optimization problems (MOPs). Most of these problems are tackled with evolutionary algorithms (EAs). In this field, NSGA_II [17] and SPEA-2 [18], denoted as MOEAs, are very popular to cope with MOPs. Generally, MOEAs use EAs' genetic operators to handle the problem. Moreover, MOPs are usually studied as benchmark, and real-world optimization problems in the engineering domain. On the other hand, architectural problems [19–21] are also relevant as MOPs to use MOEAs in recent years. In the following sections, proposed CGA_DE algorithm and well-known NSGA_II algorithms are explained in detail.

4.1 Real-Coded Genetic Algorithm with Mutation Strategy of Differential Evolution (CGA_DE)

Genetic Algorithms (GAs), which are well-known heuristic search methods, are inspired from the biology [22]. Most important operators of GAs are natural selection and evolution. In GAs, individuals in D dimensions are defined as chromosomes to generate an initial population. Two individuals are chosen and mated from the

Fig. 4 GA procedure

```
Generate initial population P^g at generation g
Evaluate individuals in P^g
While (not termination) do
    {
        Select two individuals from P^g
        Crossover individuals to produce offspring Q^g
        Mutate some individuals in Q^g
        Add offspring Q^g to individuals in P^g
        Evaluate (P^g + Q^g) individuals in P^g
        Select P^g individuals from (P^g + Q^g)
    }
    End While
End
```

Fig. 5 Arithmetic crossover operator

```
for j = 1 to D
        Q_i^{j,g+1} = CR_i × X_a^{j,g} + (1 − CR_i) × X_b^{j,g}
endfor
```

population at each generation. Afterward, new individuals are generated (called child or offspring) using crossover operator. In addition, to escape from local minima and maxima, some individuals are mutated. Finally, new generated population (offspring) is added to previous population (parent) for selecting new individuals for the next generation. The overall scheme of the GA is presented in Fig. 4.

Real-coded GA needs a crossover and mutation operators. For crossover, simple arithmetic crossover operator is employed to generate offspring Q_i^g. In other words, two individuals, X_a and X_b, are selected from the parent population. Thereafter, each dimension value of the offspring is taken from the first individual with a CR_i rate whereas from the second individual with a $(1 - CR_i)$ rate. The outline of the crossover operator is given in Fig. 5.

The real-coded GA mentioned above is actually for single objective real-parameter optimization problems. To handle multi-objective shell design problem, this algorithm is extended to MOEA. In the following lines, a self-adaptive real-coded GA with Differential Evolution mutation operator (CGA_DE) is proposed for the problem on hand.

In CGA_DE, initial population is constructed uniformly and randomly within the given boundaries for 13, 17, and 21 dimensions in $C1$, $C2$, and $C3$, respectively. The arithmetic crossover operator is used to generate offspring population as in Fig. 5. For the mutation operator, a distinct strategy is proposed by employing the mutation operator of DE/rand/1/bin. To do that, three individuals are randomly chosen from the parent population. Thereafter, uniform random number r between 0 and 1 is multiplied by the difference of two individuals in order to add to another individual randomly chosen as in Eq. (14).

$$Q_i^{j,g+1} = X_a^{j,g} + r \times \left(X_b^{j,g} - X_c^{j,g} \right) \tag{14}$$

In the CGA_DE, some new random individuals into the parent population are immigrated with an amount of $MR_i \times \left|P^t\right|$. This means that if the MR_i is equal to 0.02 with the population size $\left|P^t\right| = 100$, two individuals are generated by Eq. (14) in order to randomly immigrate into parent population.

Brest et al. [23] proposed self-adaptive procedure in DE algorithms. In this chapter, the same idea is considered to propose CGA_DE. Initially, CR_i and MR_i values are assigned to 0.5 and 0.01, respectively. Afterwards, these values are updated at each generation g for each individual i as in Eqs. (15) and (16).

$$MR_i^{g+1} = \begin{cases} MR_{min} + r_1.MR_{max} & if\ r_2 < p_1 \\ MR_i^g & otherwise \end{cases} \tag{15}$$

$$CR_i^{g+1} = \begin{cases} CR_{min} + r_1.CR_{max} & if\ r_2 < p_1 \\ CR_i^g & otherwise \end{cases} \tag{16}$$

where $r_j \in \{1, 2\}$ are uniform random numbers in the range [0, 1]. p_1 denotes the probability to adjust the CR_i and MR_i values. Parameters are taken as $p_1 = 0.1$, $CR_{min} = 0.1$ and $CR_{max} = 0.9$. In addition, MR_{min} and MR_{max} are taken as 0.01 and 0.05.

Concerning multi-objectivity, we take advantage of non-dominated sorting procedure and constrained-domination rule of the NSGA-II algorithm [17] for proposed CGA_DE. The non-dominated sorting procedure is given in Fig. 6.

Another key feature of the NSGA-I I algorithm is the crowding distance. Suppose that a non-dominated set δ is obtained. The crowding distance of $\delta[i]_{dist}$ is calculated as in Fig. 7.

For unconstrained multi-objective optimization, crowded-comparison operator ($\prec$) is used to direct the selection process at the numerous stages. Every individual i in the population has a domination rank (i_{rank}) assigned in non-dominated sorting procedure and a crowding distance rank ($i_{distance}$) calculated as in Fig. 7. Based on these two, crowded-comparison operator ($\prec$) is defined as follows:

$$if\,(i_{rank} < j_{rank})then\ i \prec j$$
$$or((i_{rank} = j_{rank})\ and\ (i_{distance} > j_{distance}))$$

This means that lesser (better) rank is preferred between two individuals with different non-domination ranks. On the other hand, if both individuals have the same non-domination rank, individual with smaller crowded region is preferred. For the constrained multi-objective optimization, NSGA_II modifies the definition of *domination* between two solutions as follows:

An individual i is considered to constrained-dominate an individual j under the following conditions:

1. Individual i is feasible and individual j is infeasible.
2. Both individuals are infeasible, but individual i has a smaller constraint violation.

$for\ each\ p \in P$
  $S_p = \emptyset$
  $n_p = 0$
  $for\ each\ q \in P$
    $if\ (p \prec q)\ then$ — If p dominated q
      $S_p = S_p \cup \{q\}$ — Add q to the set of solutions dominated by p
    $else\ if\ (q \prec p)\ then$
      $n_p = n_p + 1$ — Increment the domination counter of p
  $if\ n_p = 0\ then$ — p belongs to the first front
    $p_{rank} = 1$
    $F_1 = F_1 \cup \{p\}$
$i = 1$ — Initialize the front counter
$while\ F_i \neq \emptyset$
  $Q = \emptyset$ — Used to store the members of the next front
  $for\ each\ p \in F_i$
    $for\ each\ q \in S_p$
      $n_q = n_q - 1$
      $if\ n_q = 0\ then$ — q belongs to the next front
        $q_{rank} = i + 1$
        $Q = Q \cup \{q\}$
$i = i + 1$
$F_i = Q$
End

Fig. 6 Non-dominated sorting procedure

1. $L = |\delta|$
2. $for\ each\ individual\ i, set\ \delta[i]_{dist} = 0$
3. $for\ each\ objective\ m$
   a. $\delta = sort(\delta, m)\ in\ an\ ascending\ order$
   b. $\delta[1]_{dist} = \delta[L]_{dist} = \infty$
   c. $for\ i = 2\ to\ L - 1$

$$\delta[i]_{dist} = \delta[i]_{dist} + (\delta[i+1][m] - \delta[i-1][m])/\left(f_m^{max} - f_m^{min}\right)$$

Fig. 7 Crowding distance calculation

3. Both individuals, i and j are feasible and individual i dominates individual j with crowded-comparison rule as follows:

$$if\,(i_{rank} < j_{rank})then\ i \prec j$$
$$or((i_{rank} = j_{rank})\ and\ (i_{distance} > j_{distance}))$$

The outline of the proposed CGA_DE algorithm is described in Fig. 8.

1. $Set\ g = 0\ and\ create\ a\ random\ parent\ population\ P^g\ for\ N$
2. $Perform\ arithmetic\ crossover\ and\ DE\ mutation\ operator\ as\ in\ Fig.5\ and\ Eq.(14)\ on\ P^g\ to\ obtain\ Q^g\ with\ size\ N$
3. $If\ the\ termination\ criteria\ is\ satisfied, stop\ and\ return\ P^g.$
4. $Set\ R^g = P^g \cup Q^g\ to\ combine\ two\ populations$
5. $Perform\ non-dominated\ sorting\ procedure\ for\ R^g\ to\ set\ the\ non-dominated\ fronts\ f_1, f_2, .., f_k.$
6. $For\ i = 1, .., k\ \ repeat\ the\ following\ steps:$
   a. $Calculate\ crowding\ distance\ for\ each\ solution\ in\ f_i.$
   b. $Create\ P^{g+1}\ as\ follows:$
      i. $if\ |P^{g+1}| + |f_i| \leq N, then\ set\ P^{g+1} = P^{g+1} \cup f^g$
      ii. $if\ |P^{g+1}| + |f_i| > N, then\ add\ the\ least\ crowded\ N - |P^{g+1}|\ solutions\ from\ f_i\ to\ P^{g+1}.$
7. $Use\ binary\ tournament\ selection\ with\ constrained\ domination\ rule\ to\ select\ parents\ from\ P^{g+1}.$
8. $Apply\ arithmetic\ crossover\ and\ DE\ mutation\ operator\ to\ P^{g+1}\ and\ obtain\ new\ offspring\ population\ Q^{g+1}\ for\ N\ size$
9. $Set\ g = g + 1\ and\ go\ to\ Step\ 3.$

Fig. 8 CGA_DE algorithm

4.2 Non-dominated Sorting Genetic Algorithm II (NSGA_II)

NSGA_II is a well-known MOEAs developed by [17]. The power of NSGA_II is coming from the unique properties of elitism, superiority of feasibility and non-dominated sorting. Thus, it is presenting homogenously distributed Pareto-optimal solutions. The main approaches of NSGA_II algorithm are presented as follows:

- O (MN^2) sorting algorithm to obtain non-dominated solutions.
- Estimation of "Crowding Distance" subject to cuboid volume between neighboring elements, which have the same rank value.
- Application of crowding distance, superior of feasibility, as well as ranking for diversity preserving through binary tournament selection.
- Elitism strategy by combining offspring members and elite parents.
- Application of genetic operator "Simulated Binary Crossover (SBX)".
- Finally employing polynomial mutation operator (PM).

The process of the NSGA_II is shown in Fig. 9:

1. $Set\ g = 0\ and\ create\ a\ random\ parent\ population\ P^g\ with\ N\ size.$
2. $Perform\ SBX\ and\ PM\ on\ P^g\ for\ reaching\ child\ population\ Q^g\ with\ N\ size.$
3. $If\ termination\ criterion\ is\ satisfied, stop\ and\ return\ P^g.$
4. $Set\ R^g = P^g \cup Q^g\ to\ combine\ parent\ and\ child\ populations.$
5. $Apply\ the\ fast\ nondominated\ sorting\ algorithm\ to\ the\ R^g\ for\ defining\ the\ non-dominated\ fronts\ f_1, f_2, .., f_k\ in\ R^g.$
6. $For\ i = 1, .., k\ \ do\ following\ steps:$
   a. $Calculate\ crowding\ distance\ of\ the\ solutions\ in\ f_i.$
   b. $Create\ P^{g+1}\ as\ follows:$
      i. $if\ |P^{g+1}| + |f_i| \leq N, then\ set\ P^{g+1} = P^{g+1} \cup f_i$
      ii. $if\ |P^{g+1}| + |f_i| > N, then\ add\ the\ least\ crowded\ N - |P^{g+1}|\ solutions\ from\ f_i$ to P^{g+1}.
7. $Use\ binary\ tournament\ selection\ with\ constrained-domination\ rule\ to\ choose\ parents\ from\ P^{g+1}.$
8. Apply $SBX\ and\ PM\ to\ P^{g+1}\ and\ get\ new\ child\ population\ Q^{g+1}\ with\ N\ size.$
9. $Set\ g = g + 1\ and\ go\ to\ Step\ 3.$

Fig. 9 NSGA_II algorithm process

5 Computational Results

CGA_DE and NSGA_II algorithms were run on a computer, which has Intel I7 at 4.4 GHz with 16 GB Ram and 250 GB solid-state drive, to cope with the problem on hand. For all optimization runs, population size was taken as 100.

For generating one individual in the computer, required time was between 2.5 and 5.0 s depending upon the complexity of the geometry in each case. Karamba 3D evaluate structural performance of shell geometries using mesh surface. Such a smooth geometry needs too many mesh surfaces. For this reason, high computational time was required for not only generating but also evaluating large amount of shell meshes.

In order to define the termination criterion, non-dominated solution size in the search space was obtained at every 10 generations. Moreover, minimum, maximum, and average, values of m and v objective functions were recorded as well. By this way, the behavior of objective functions in each case was observed. Based on non-dominated solution size and behavior of objective functions, termination criterion was defined in each case. Afterwards final set of Pareto-fronts were presented. In order to give a better insight, minimum, maximum, and average values of decision variables for the solutions in Pareto-fronts were also reported as tables.

For the sake of computational performance, Hypervolume (HV) was considered as performance metric [24, 25]. HV is a common metric to assess the volume of non-dominated solutions with respect to the reference point. HV also was known as convergence-diversity metric [26] because it evaluates both closeness to the utopic point as well as distribution through the Pareto-front with a various solutions. Following equation was used to calculate HV for each Pareto-front in case studies:

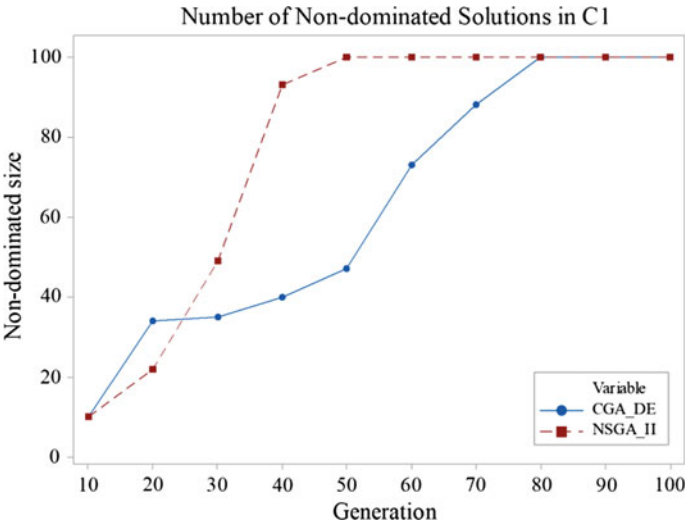


Fig. 10 Number of non-dominated solutions in Case 1

$$HV(PF, RP) = volume\left(\bigcup_{i=0}^{|n|} u_i\right) \tag{17}$$

where PF is the Pareto-front, RP is the reference point, n is the number of solutions on PF, u_i is the volume of the space defined between the solution i and RP. In each case study, RP was defined as point dominated by all non-dominated solutions in the search space. All these aforementioned analyses are presented for each case including detailed results in the following sections.

5.1 Case 1

Number of non-dominated solutions during the optimization process for $C1$ is illustrated in Fig. 10. According to this figure, NSGA_II algorithm found 100 Non-dominated solutions after 50 generation. On the other hand, CGA_DE reached 100 non-dominated solutions after 80 generation. It was clear that NSGA_II converged faster than CGA_DE algorithm for $C1$.

Behavior of m objective function for $C1$ is presented in Fig. 11. It was clearly seen that maximum value for m presented no alteration through the optimization process of NSGA_II. On the other hand, CGA_DE suggested different results at the early generation. After 80th generation, CGA_DE started to present very similar results, as well. At 100th generation NSGA_II reached 1168.76 kg as maximum m value, whereas CGA_DE discovered as 1409.89 kg in the Pareto-front.

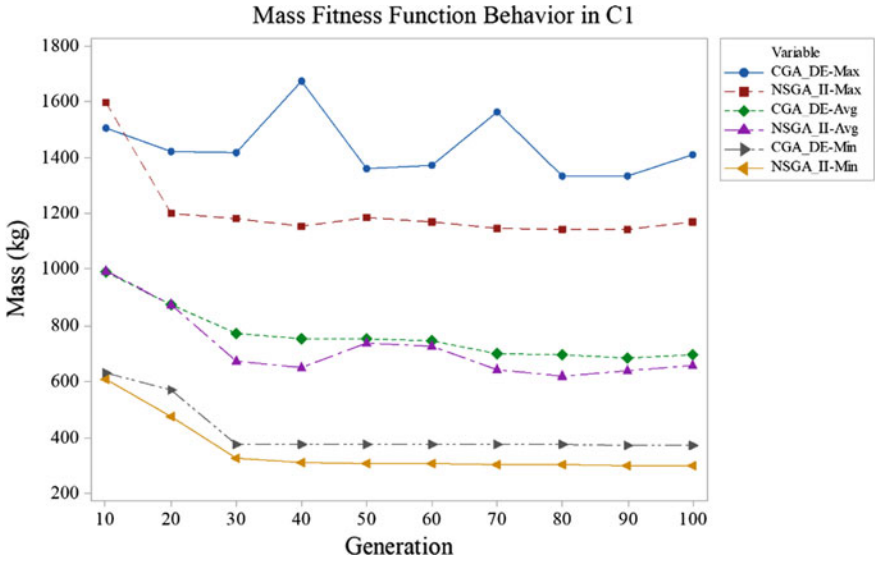


Fig. 11 Mass fitness function behavior of NSGA_II and CGA_DE in Case 1

Concerning average m values, both algorithms started from approximately 1000 kg. During the optimization process, CGA_DE and NSGA_II algorithms presented very similar results, which were around 800 kg. It was also seen that after 80th generation, algorithms suggested very similar results. At 100th generation average value suggested by NSGA_II was slightly smaller than CGA_DE.

From the standpoint of minimum m values, CGA_DE found the minimum value after 30th generation. However, NSGA_II reached the minimum value at 40th generation. At 100th generation, CGA_DE presented 373.78 kg as minimum m value, while NSGA_II found 310.49 kg. Generally speaking, for all values shown in Fig. 11, NSGA_II discovered smaller values than CGA_DE.

Behavior of v objective function for *C1* is presented in Fig. 12. After 60th generation, CGA_DE presented same results for the maximum v. On the other hand, NSGA_II suggested similar results after 60th generation. At 100th generation, CGA_DE found maximum v value as 0.026 m whereas, NSGA_II reached 0.050 m.

For the average values, both algorithms started from approximately 0.02 m. During the optimization process, average v value decreased in optimization process of both algorithms. After 70th generation, it was seen that average results were converged. At 100th generation, NSGA_II solutions were presented slightly bigger v values than CGA_DE. However, it was observed that both algorithms presented very similar behavior during the 100th generation.

Likewise the average values, both algorithms presented very similar behavior for the minimum v values. At the 10th generation, it was noted that both algorithms suggested very similar results. After the 60th generation, NSGA_II and CGA_DE

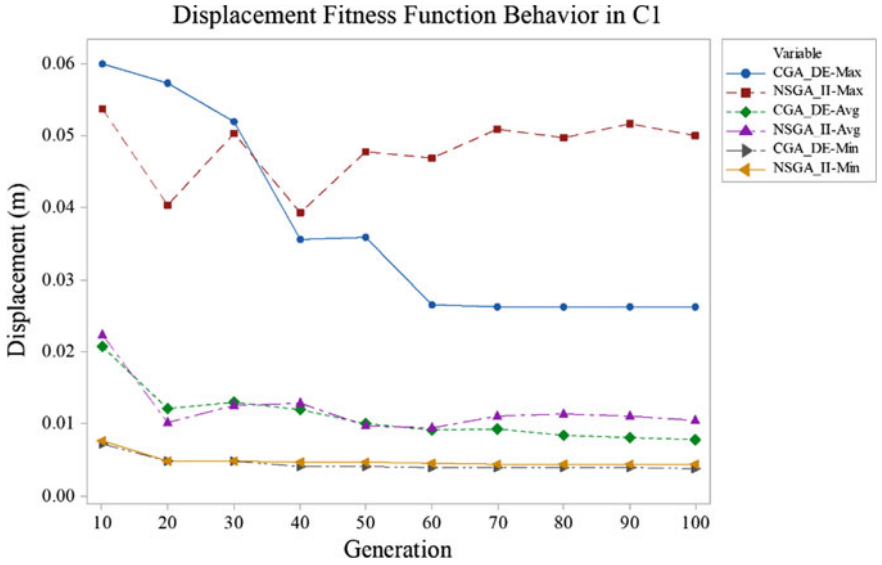


Fig. 12 Displacement fitness function behavior of NSGA_II and CGA_DE in Case 1

presented slightly different results. At 100th generation, it was seen that NSGA_II suggested 0.004 m as minimum v, whereas CGA_DE found 0.003 m.

To sum up, non-dominated solutions of both algorithms presented same results after 80th generation. In addition, m and v objective functions presented very similar behaviors for minimum, average, and maximum values between 80th and 100th generations. Due to these facts, termination criterion was defined as 100th generation for *C1*.

As result of optimization process for *C1*, Pareto-fronts at 100th generation for CGA_DE and NSGA_II are presented in Fig. 13.

Minimum, maximum, and average values of decision variables belong to Pareto-fronts of *C1* are given in Table 2. Regarding these values, it was seen that $P_{1,a}$, $P_{1,b}$, $P_{2,a}$, $P_{2,b}$, $P_{3,a}$, $P_{3,b}$, LC_1, LC_2, LC_3, and RC_1 presented narrow ranges for both NSGA_II and CGA_DE solutions. On the other hand, RC_2 for CGA_DE and RC_3 for NSGA_II suggested wider range. However, it was observed that MT presented wide range. It can be said that this variable had a great impact for the presented Pareto-fronts belong to both algorithms.

As final analysis for *C1*, HVs were calculated in accordance with Eq. (17). For 100th generation, HV was computed for CGA_DE as 0.95 and for NSGA_II as 0.92. As a result, computational performance of CGA_DE was slightly better than NSGA_II for *C1*.

Table 2 Decision Variable Values for $C1$

	$C1$					
	CGA_DE			NSGA_II		
	Min	Max	Avg	Min	Max	Avg
$P_{1,a}$	0.86	0.90	0.87	0.70	0.72	0.71
$P_{1,b}$	0.58	0.67	0.60	0.50	0.53	0.52
$P_{2,a}$	0.87	0.88	0.88	0.89	0.94	0.90
$P_{2,b}$	0.75	0.79	0.77	0.86	0.88	0.87
$P_{3,a}$	0.71	0.76	0.74	0.43	0.48	0.45
$P_{3,b}$	0.72	0.74	0.73	0.61	0.66	0.63
$P_{4,a}$						
$P_{4,b}$						
$P_{5,a}$						
$P_{5,b}$						
LC_1	8.09	8.13	8.10	8.61	9.00	8.66
LC_2	8.16	8.25	8.19	8.52	8.62	8.53
LC_3	7.92	8.06	7.96	8.20	8.24	8.24
LC_4						
LC_5						
RC_1	9.69	10.76	9.89	19.12	19.53	19.19
RC_2	14.21	17.37	14.88	16.85	16.87	16.86
RC_3	9.17	10.90	10.08	16.97	18.04	17.51
RC_4						
RC_5						
MT	5.00	18.31	9.01	5.00	19.23	10.86

5.2 Case 2

Number of non-dominated solutions during the optimization process for $C2$ is illustrated in Fig. 14. According to this figure, NSGA_II and CGA_DE algorithms found 100 non-dominated solutions after 60 generation. Last 40 generations both algorithms kept the same non-dominated solution size.

Behavior of m objective function for $C2$ is presented in Fig. 15. From the standpoint of maximum m values, it can be said that both algorithms presented high values at 10th generation. Afterwards, maximum m value started to decrease until the 40th generation. NSGA_II reached to 3214.39 kg at 100th generation. On the other hand, CGA_DE presented the same results between 80th and 100th generations. Maximum m result was found by CGA_DE as 3569.3 kg at 100th generation.

Regarding average m values, both algorithms presented very similar behavior during the 100th generation. At 10th generation, it was observed that NSGA_II presented higher average m value than CGA_DE. After 70th generation, average m

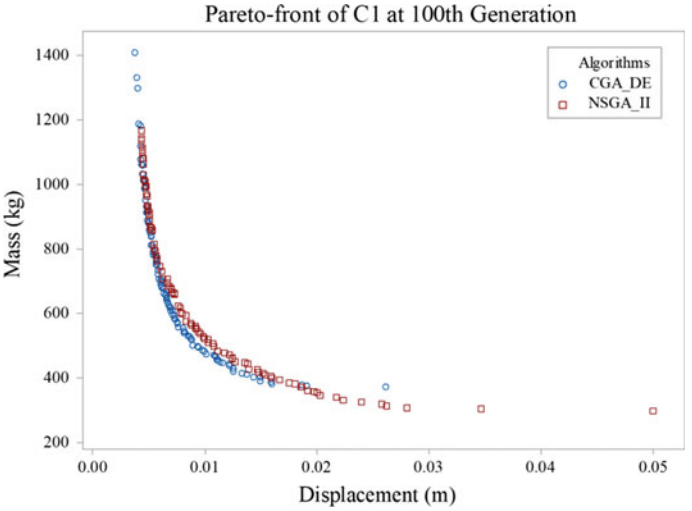


Fig. 13 Pareto-front of Case 1 at 100th generation

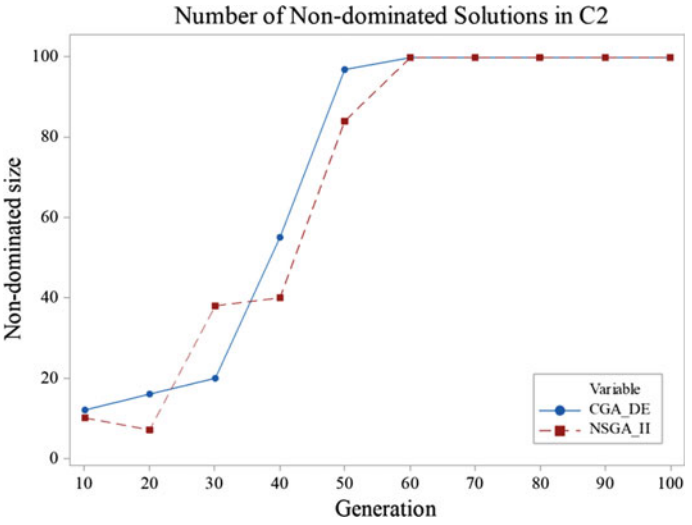


Fig. 14 Number of non-dominated solutions in Case 2

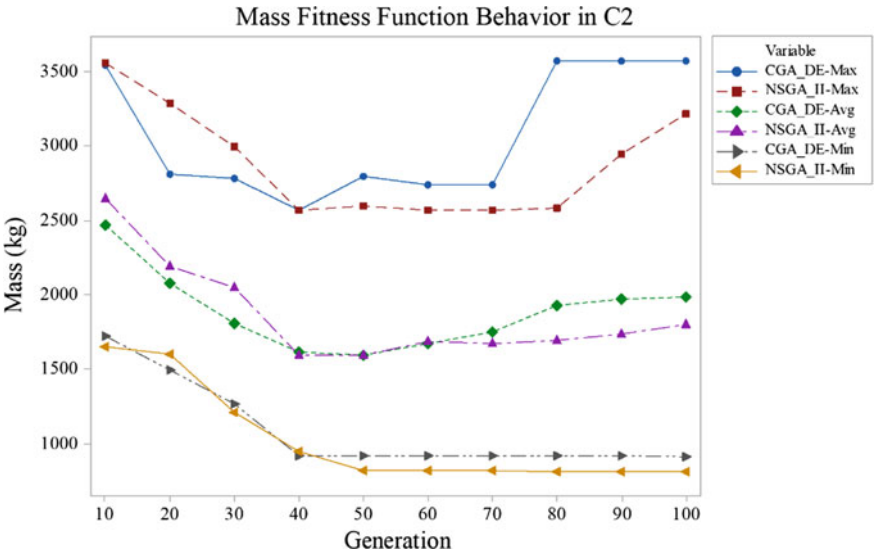


Fig. 15 Mass fitness function behavior of NSGA_II and CGA_DE in Case 2

values of CGA_DE was higher than NSGA_II results. At 100th generation, CGA_DE reached 1983.30 kg whereas, NSGA_II presented 1802.20 kg.

Concerning minimum *m* values, both algorithms presented approximately 1650 kg at 10th generation. During the 40th generation, NSGA_II and CGA_DE started to decrease these values with a similar behavior. At 50th generation, both algorithms founded minimum *m* values in the optimization process of C2. At 100th generation, it was observed that NSGA_II founded 814.90 kg for minimum *m*. On the other hand, CGA_DE suggested minimum *m* value as 916,05 kg at the same generation.

Behavior of *v* objective function for *C2* is presented in Fig. 16. Between 20th and 80th generations, both algorithms presented similar behavior. However, during the last 20 generations, maximum *v* values of NSGA_II started to increase, whereas *v* value of CGA_DE decreased. Even different results, maximum *v* values between 80th and 100th founded by both algorithms were slightly different. At 100th generation, CGA_DE found maximum *v* value as 0.044 m whereas, NSGA_II reached 0.052 m.

For the average values at 10th generation, NSGA_II and CGA_DE algorithms presented 0.037 m, and 0.030 m, respectively. During the optimization process, it was observed that average *v* values were decreased. Moreover, both algorithms presented similar behavior during the decrement. After 90th generation, it was seen that both algorithms presented no alteration. At 100th generation, NSGA_II suggested 0.0182 m as average *v* value, whereas CGA_DE found 0.0155 m.

For the minimum *v* values, both algorithms presented very similar behavior starting from the 10th generation. After 70th generation, behavior of NSGA_II and CGA_DE slightly differentiated from each other. At 100th generation, NSGA_II suggested 0.009 m, while CGA_DE found 0.008 m for average *v* value.

Table 3 Decision Variable Values for $C2$

	$C2$					
	CGA_DE			NSGA_II		
	Min	Max	Avg	Min	Max	Avg
$P_{1,a}$	0.95	0.97	0.96	0.43	0.54	0.52
$P_{1,b}$	0.92	0.96	0.93	0.83	0.94	0.85
$P_{2,a}$	0.80	0.84	0.82	0.83	0.86	0.84
$P_{2,b}$	0.90	0.92	0.91	0.83	0.86	0.85
$P_{3,a}$	0.84	0.87	0.85	0.70	0.70	0.70
$P_{3,b}$	0.81	0.89	0.84	0.55	0.79	0.62
$P_{4,a}$	0.84	0.88	0.86	0.65	0.83	0.80
$P_{4,b}$	0.91	0.94	0.93	0.82	0.83	0.83
$P_{5,a}$						
$P_{5,b}$						
LC_1	6.90	6.95	6.93	7.46	7.48	7.47
LC_2	6.66	6.89	6.78	7.17	7.53	7.26
LC_3	6.94	7.01	6.98	7.17	7.36	7.19
LC_4	7.13	7.18	7.15	7.14	7.15	7.15
LC_5						
$R\,C1$	7.24	7.89	7.43	19.50	19.51	19.51
RC_2	13.39	13.72	13.55	18.55	19.91	19.28
RC_3	11.23	13.32	12.13	19.90	20.00	19.95
RC_4	13.34	14.66	14.37	16.25	17.10	17.04
RC_5						
MT	5.00	19.37	10.77	5.00	19.69	11.03

Briefly, both algorithms founded 100 non-dominated solutions at 60th generation and kept the same size until 100th generation. Moreover, NSGA_II and CGA_DE showed no alteration between 90th and 100th generations for minimum, maximum, and average m and v objective function values. Therefore, termination criterion was defined as 100th generation for $C2$.

According to the termination criterion for $C2$ explained above, Pareto-fronts at 100th generation for CGA_DE and NSGA_II are presented in Fig. 17.

Decision variables of reported Pareto-fronts for $C2$ are given in Table 3. Based on these values, it was observed that $P_{1,a}$, $P_{1,b}$, $P_{2,a}$, $P_{2,b}$, $P_{3,a}$, $P_{3,b}$, $P_{4,a}$, $P_{4,b}$, LC_1, LC_2, LC_3, LC_4, and RC_1 suggested narrow ranges for both Pareto-front sets. On the other hand, RC_2, and RC_4 presented wide ranges for NSGA_II. In addition, CGA_DE presented wide range for RC_3. However, it was seen that MT provided widest range amongst all decision variables. Likewise the $C1$, MT variable had a great impact for the Pareto-front sets belong to both algorithms.

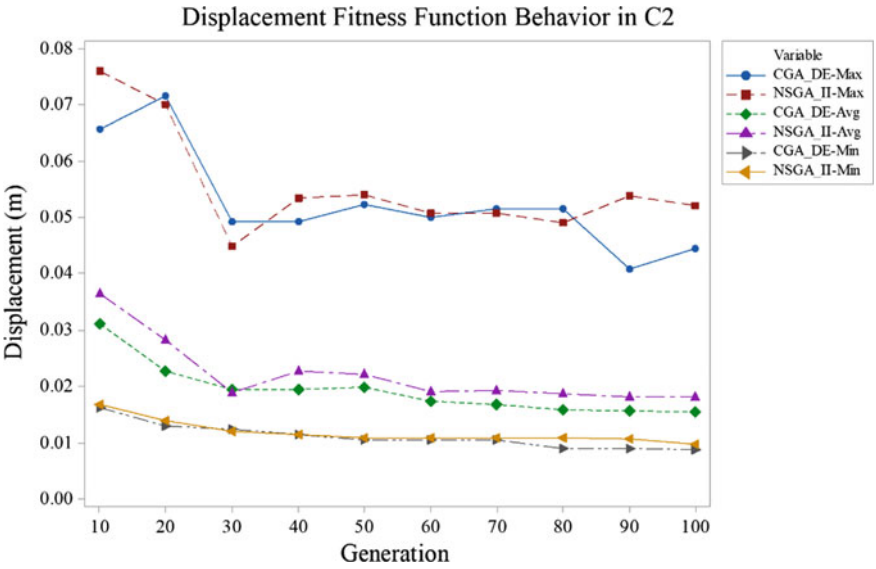


Fig. 16 Displacement Fitness Function Behavior of NSGA_II and CGA_DE in Case 2

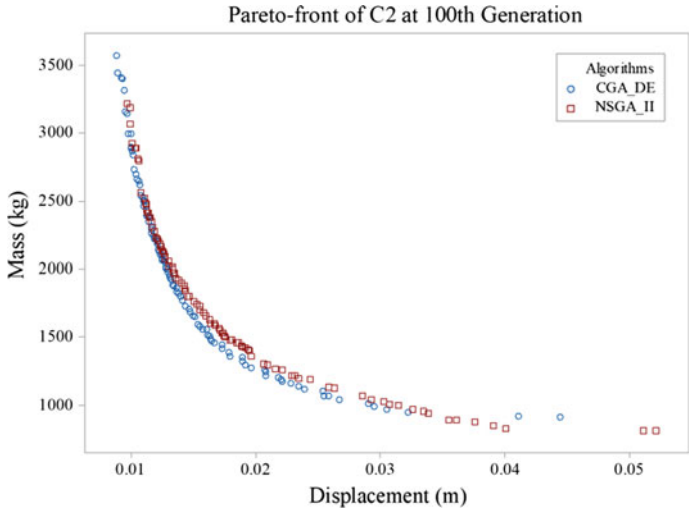


Fig. 17 Pareto-front of Case 2 at 100th generation

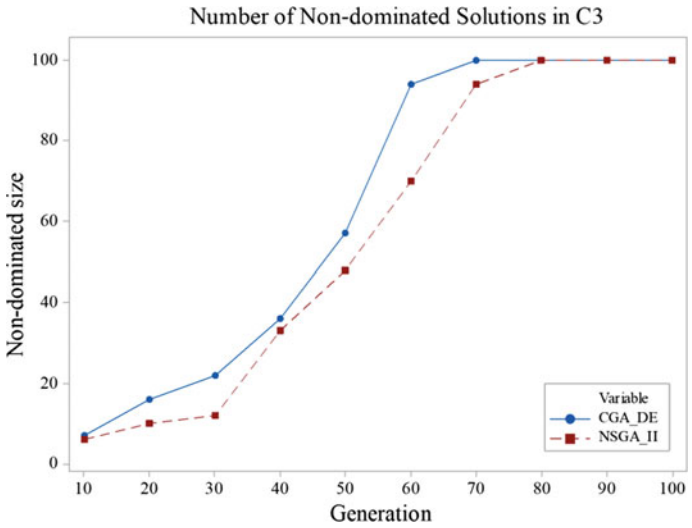


Fig. 18 Number of non-dominated solutions in Case 3

Last but not least, HVs were calculated based on Eq. (17) for this case, as well. According this, HV was computed for CGA_DE as 0.88 and for NSGA_II as 0.86. From the standpoint of computational performance, CGA_DE was slightly better than NSGA_II for *C2*.

5.3 Case 3

Non-dominated solution size during the optimization process for *C*3 is illustrated in Fig. 18. According to this figure, NSGA_II found 100 non-dominated solutions at 70th generation, whereas CGA_DE algorithm reached the same size at 80th generation. During the last 20 generations both algorithms kept 100 non-dominated solutions in the Pareto-front sets.

Behavior of *m* objective function for *C3* is presented in Fig. 19. For maximum *m* values, both algorithms suggested the highest values at 10th generation. Thereafter, these values decreased until 50th generation. During the last 50 generations, values of CGA_DE were higher than values of NSGA_II. In addition, both algorithms presented no significant alteration after 50th generation. NSGA_II reached to 5837.45 kg, whereas CGA_DE founded 6160.95 kg at 100th generation.

Similar to maximum *m* behavior, the highest average values were founded at 10th generation. Until the 80th generation, both algorithms decreased these values. During last 20 generations, average *m* results of NSGA_II and CGA_DE algorithms were slightly different. At 100th generation, NSGA_II presented 3758.47 kg whereas, CGA_DE reached 3733.14 kg.

Likewise the maximum and average *m* values, both algorithms presented highest values at 10th generation, as well. Until 70th generation, these values were decreased. In the last 30 generations, significant alteration was not observed. At 100th generation, NSGA_II founded 2049.81 kg as minimum *m* value for *C3*. On the other hand, CGA_DE suggested minimum *m* value as 1816.72 kg at the same generation.

Behavior of *v* objective function for *C3* is presented in Fig. 20. Both algorithms offered highest values at 10th generation for maximum *v*. At 40th generation, it was observed that NSGA_II decreased maximum *v* to 0.036 m, whereas CGA_DE decreased the same value to 0.052 m. It was also seen that highest difference between values of both algorithms presented at 40th generation. 30 generation later, maximum *v* values of both algorithms were slightly different from each other. At 100th generation, with no significant alteration, NSGA_II presented maximum *v* value as 0.064 m, and CGA_DE suggested maximum *v* value as 0.068 m.

For the average values, both algorithms presented highest values at 10th generation, as well. Until 40th generation, average *v* values were decreased with a similar behavior. Afterwards, these results were increased when the optimization process reached to 90th generation. Last 10 generations, it was observed that NSGA_II and CGA_DE algorithms presented no significant difference. At 100th generation, average *v* value was founded by NSGA_II as 0.034 m, whereas CGA_DE suggested as 0.035 m.

For the minimum *v* values, it was observed that minimum *v* values decreased with a similar behavior during the optimization process. At 100th generation, NSGA_II and CGA_DE founded 0.021 m, and 0.020 m, respectively.

To sum up, between 90th and 100th generations no substantial alteration were observed in terms of non-dominated solution size, minimum, maximum, and average objective function values. Therefore, algorithms were terminated at 100th generation.

Based on previous analysis of termination criterion, Pareto-fronts of *C3* at 100th generation for CGA_DE and NSGA_II are presented in Fig. 21.

Decision variables of reported Pareto-fronts for *C3* are given in Table 3. According to these values, $P_{1,a}$, $P_{1,b}$, $P_{2,a}$, $P_{2,b}$, $P_{3,a}$, $P_{3,b}$, $P_{4,a}$, $P_{4,b}$, $P_{5,a}$, $P_{5,b}$, LC_1, LC_2, LC_3, LC_4, LC_5, RC_1, RC_2, RC_3, and RC_4 suggested narrow ranges for both Pareto-front sets. On the other hand, RC_5 presented wide range for both NSGA_II, and CGA_DE algorithms. However, likewise the previous observations in C1, and C2, it was seen that *MT* provided widest ranges for both Pareto-front sets (Table 4).

Finally, according to Eq. (17) HVs were calculated. For CGA_DE HV was computed as 0.71 and for NSGA_II HV was computed as 0.78. As a result, it can be said that optimization performance NSGA_II was slightly better than CGA_DE for *C3*.

6 Discussion

In order to discuss the results, initial and optimized values of the shell design problem were compared. For this reason, it was required to select one solution in the Pareto-front for each MOEAs in all cases. The importance of each objective function varies

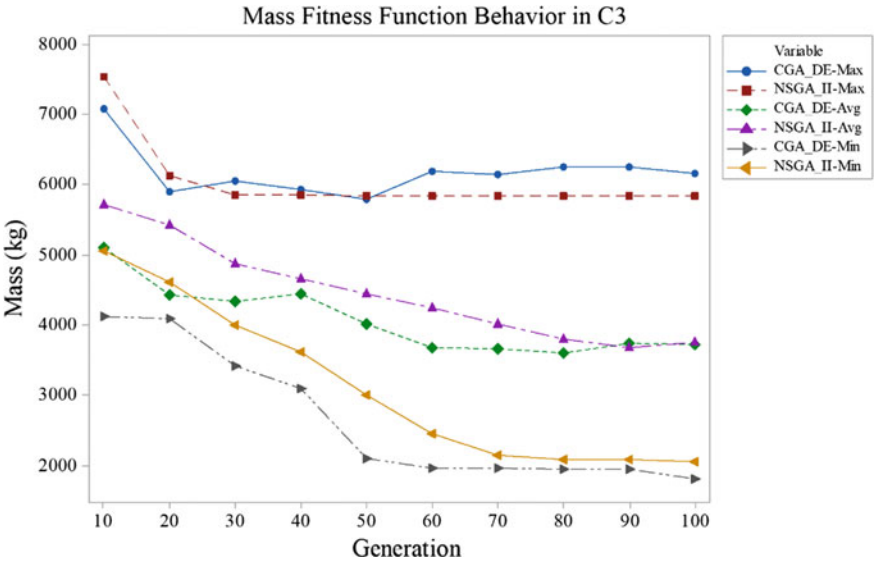


Fig. 19 Mass fitness function behavior of NSGA_II and CGA_DE in Case 3

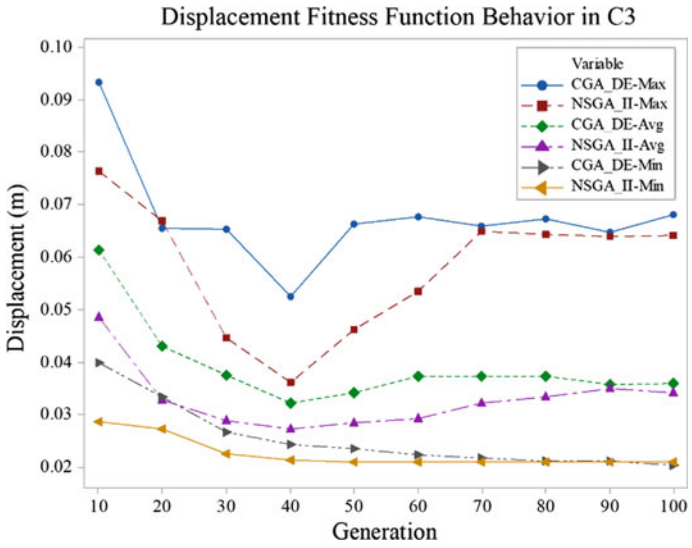


Fig. 20 Displacement fitness function behavior of NSGA_II and CGA_DE in Case 3

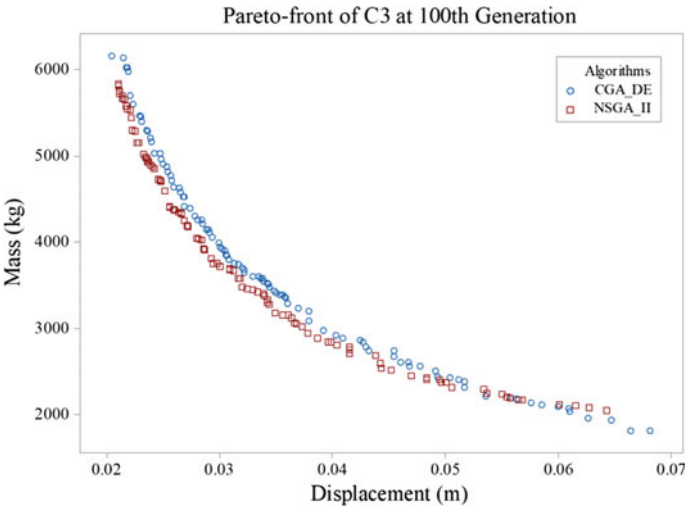


Fig. 21 Pareto-front of Case 3 at 100th generation

in accordance with the decision maker (designer). In this study, weighted summation (WS) [27, 28] was calculated by giving equal weights for each objective function in $C1$, $C2$, and $C3$ as follows:

$$F_{WS}(x) = \frac{1}{2}\sum_{i=1}^{2}\frac{f_i(x) - f_i(x)^{min}}{f_i(x)^{max} - f_i(x)^{min}} \tag{16}$$

where $f_1(x)$ and $f_2(x)$ are m and v, respectively. After calculation of WS, the ones with smallest values were picked due to the optimization problem defined as minimization for both objectives. Selected solutions from each case belong to NSGA_II and CGA_DE algorithms are presented in Fig. 22.

Before the optimization process, the initial parameters and corresponding fitness values for each case were saved. Initial and optimized values for decision variables and objective functions are compared in Table 5. Selected alternatives and their initial states are presented in Fig. 23, 24 and 25.

Concerning the Table 5, it can be noted that while the areas of the shells enlarged from $C1$ to $C3$, the values of objective functions also increased. In addition, this fact caused to increment of the weights of shell structures. Moreover, due to the huge area covered by shell geometries, optimized v values of *C3* were bigger than *C2*, and *C1*. On the other hand, optimized v values of *C1* observed as the smallest value amongst all cases. Regarding the pillars of the shell structure, decision variable of points close to 0.30 m suggest smaller support surface. Oppositely, larger support surface is provided when points close to 1.50 m. Consequently, it was seen that the values of points increased starting from $C1$, $C2$, and $C3$. This means that while the mass of the shell increased, the pillars of the shell became larger.

Table 4 Decision Variable Values for $C3$

	$C3$					
	CGA_DE			NSGA_II		
	Min	Max	Avg	Min	Max	Avg
$P_{1,a}$	1.01	1.06	1.03	1.18	1.19	1.18
$P_{1,b}$	1.07	1.08	1.08	0.93	1.01	0.96
$P_{2,a}$	0.86	0.89	0.88	0.31	0.40	0.31
$P_{2,b}$	0.97	0.98	0.98	0.30	0.31	0.31
$P_{3,a}$	0.72	0.72	0.72	0.45	0.57	0.47
$P_{3,b}$	0.70	0.76	0.73	1.18	1.22	1.19
$P_{4,a}$	0.96	0.98	0.97	0.36	0.43	0.38
$P_{4,b}$	1.01	1.03	1.01	0.31	0.54	0.43
$P_{5,a}$	0.69	0.81	0.74	0.86	0.94	0.91
$P_{5,b}$	0.84	0.87	0.86	0.82	0.92	0.86
LC_1	5.29	5.40	5.34	4.94	4.97	4.95
LC_2	5.83	5.92	5.86	6.44	6.56	6.48
LC_3	6.72	6.78	6.76	5.92	5.96	5.95
LC_4	6.09	6.14	6.11	7.51	7.54	7.51
LC_5	6.51	6.59	6.53	7.26	7.28	7.27
RC_1	13.62	14.19	13.89	11.42	11.76	11.66
RC_2	14.12	14.86	14.59	16.65	17.19	17.03
RC_3	10.18	10.56	10.40	16.73	17.25	17.07
RC_4	19.57	20.00	19.96	16.10	16.24	16.17
RC_5	9.37	10.50	9.96	16.21	18.28	17.31
MT	5.94	20.00	12.19	7.14	20.00	12.97

According to Table 5, the length of the catenaries decreased from $C1$ to $C3$. For each case, catenary chains were rotated in order to achieve smaller v values using smaller m. Related to this observation, it was seen that rotation values of catenaries were more than 10 degrees in most of the cases. Besides, there was an increment on the thickness of the shell system from $C1$ to $C3$. This means that while the topology surface became larger, the material thickness increased, accordingly.

From the standpoint of results achieved at the end of optimization process, significant enhancement observed for all cases. For *C1* and *C2*, while NSGA_II presented smaller m for optimized solutions, CGA_DE suggested smaller v values. However, for *C3*, this situation was observed in the opposite way. Based on these results, the appropriate approach to optimize shell structure problem is to employ different heuristic optimization algorithms due to No Free Launch Theorem [29]. By this way, designer can compare results belong to different algorithms in order to give the most adequate final solution.

Table 5 Decision Variable Values of All Cases

	$C1$			$C2$			$C3$		
	Initial	CGA_DE	NSGA_II	Initial	CGA_DE	NSGA_II	Initial	CGA_DE	NSGA_II
$P_{1,a}$	0.30	0.87	0.70	0.30	0.96	0.52	0.30	1.01	1.18
$P_{1,b}$	0.30	0.60	0.51	0.30	0.92	0.84	0.30	1.08	0.93
$P_{2,a}$	0.30	0.88	0.89	0.30	0.82	0.84	0.30	0.89	0.31
$P_{2,b}$	0.30	0.77	0.87	0.30	0.91	0.85	0.30	0.98	0.31
$P_{3,a}$	0.30	0.73	0.43	0.30	0.86	0.70	0.30	0.72	0.45
$P_{3,b}$	0.30	0.73	0.62	0.30	0.85	0.61	0.30	0.73	1.19
$P_{4,a}$				0.30	0.87	0.77	0.30	0.97	0.38
$P_{4,b}$				0.30	0.93	0.83	0.30	1.01	0.37
$P_{5,a}$							0.30	0.71	0.88
$P_{5,b}$							0.30	0.85	0.84
LC_1	8.00	8.10	8.68	8.00	6.93	7.47	8.00	5.33	4.95
LC_2	8.00	8.18	8.54	8.00	6.75	7.25	8.00	5.86	6.44

(continued)

Table 5 (continued)

	$C1$			$C2$			$C3$		
	Initial	CGA_DE	NSGA_II	Initial	CGA_DE	NSGA_II	Initial	CGA_DE	NSGA_II
LC_3	8.00	7.96	8.23	8.00	6.97	7.19	8.00	6.76	5.92
LC_4				8.00	7.14	7.14	8.00	6.12	7.51
LC_5							8.00	6.51	7.27
RC_1	0.00	9.94	19.22	0.00	7.46	19.51	0.00	14.05	11.75
RC_2	0.00	14.77	16.86	0.00	13.51	18.77	0.00	14.66	17.06
RC_3	0.00	10.33	17.44	0.00	12.26	19.98	0.00	10.40	17.21
RC_4				0.00	14.37	17.07	0.00	20.00	16.13
RC_5							0.00	10.14	17.79
MT	5.00	7.22	7.05	5.00	7.92	7.97	5.00	10.13	11.10
Mass	469.69	558.22	427.23	948.66	1457.53	1305.54	1953.1	3092.22	3185.92
Disp.	0.22	0.008	0.014	1.43	0.017	0.021	0.46	0.038	0.035

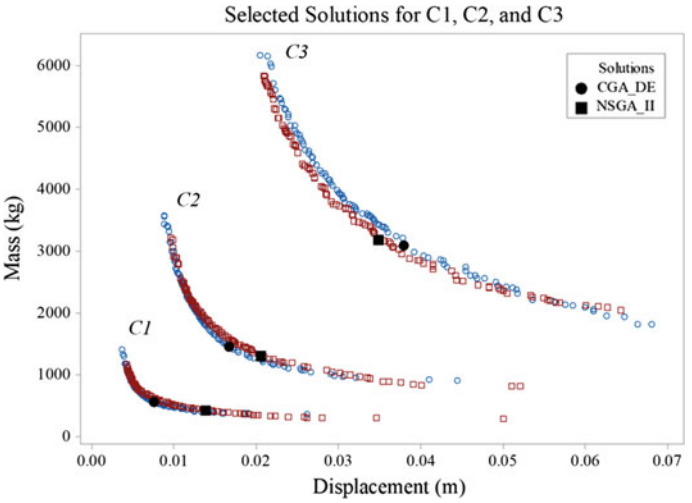


Fig. 22 Selected solutions after computational optimization

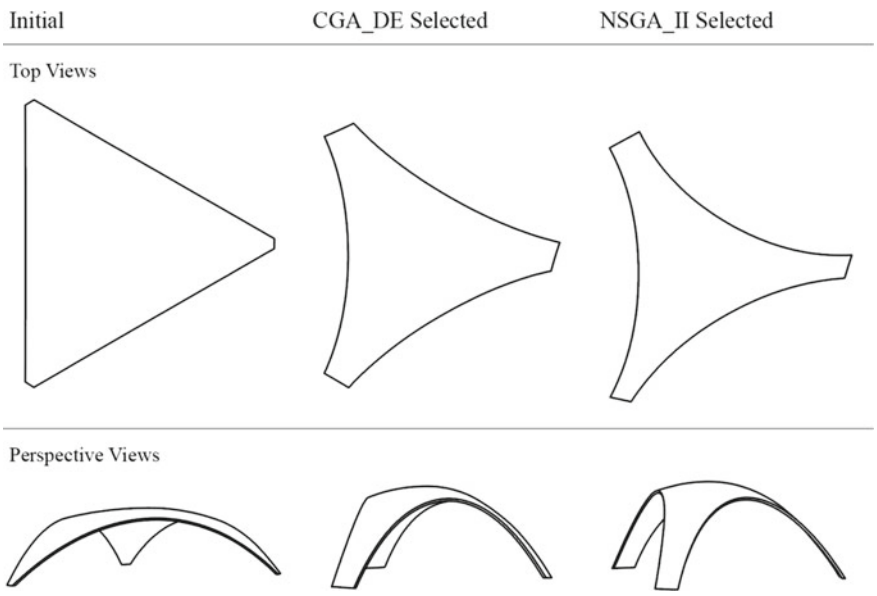


Fig. 23 Visualization of initial and selected solutions for $C1$

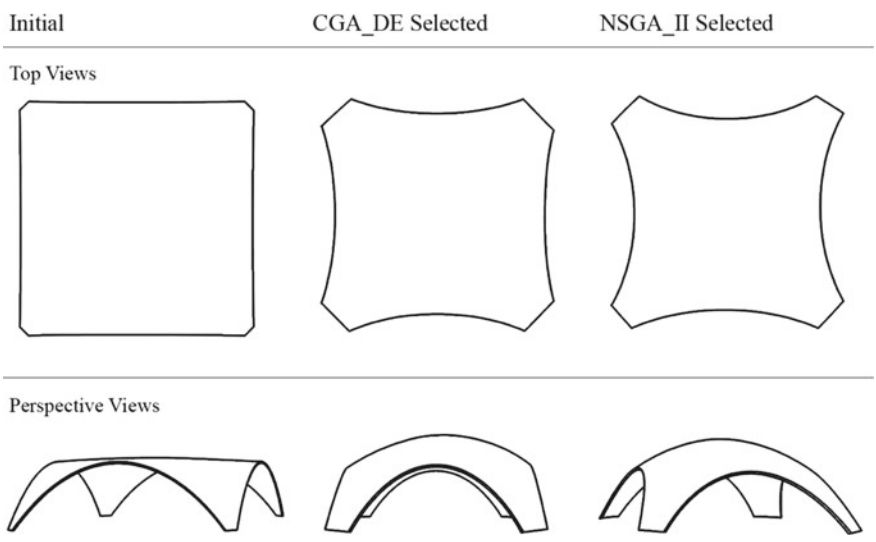


Fig. 24 Visualization of initial and selected solutions for $C2$

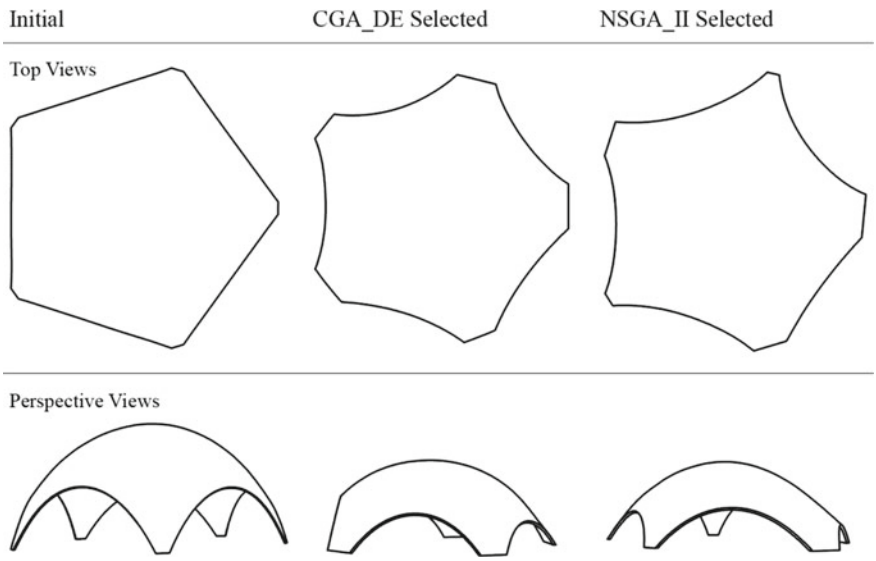


Fig. 25 Visualization of initial and selected solutions for $C3$

7 Conclusions

This study presents the application of NSGA_II and CGA_DE for shell structures form-finding problem in terms of multi-objective real-parameter constrained optimizations. In this context, three cases including 13, 17, and 21 decision variables were considered for minimization of *m* and *v* objectives. According to the results obtained after 100 generation, both NSGA_II and CGA_DE algorithms presented set of desirable shell design solutions. Using the HV as optimization performance metric, CGA_DE provided slightly better non-dominated solutions than NSGA_II in the first two cases. However, NSGA_II slightly outperformed CGA_DE in the third case.

Amongst final solutions belong to CGA_DE and NSGA_II, preferred alternatives from the point of designer are shown in Fig. 26. Even though CGA_DE presented higher HV results for *C1* and *C2*, alternatives discovered by NSGA_II are more satisfactory. The reason for this is proportions of shell geometries suggested by NSGA_II are more desirable than CGA_DE. On the other hand, proportions of shell geometries presented by CGA_DE are more desirable while NSGA_II founded higher HV.

In conclusion, it can be said that results support the statement that design decisions considered in the conceptual phase take a substantial role for architectural design process. Designers and architects are advised to handle these decision variables and its constraints for the sake of reaching desirable objectives. EAs present a systematic method to deal with complex design problems. Nevertheless, the use of different algorithms may lead to different structural performances. Thus, a comparative study between different algorithms is highly needed in this context. Simultaneously, architects and engineers are encouraged to consider more than one algorithm for their design scenario.

Regarding the AAC material, beyond being a dividing wall, a new application area is proposed. For further studies, more complex shell geometries can be used, whereas the enclosed areas and spans are extended. AAC type, block sizes, and shapes can be considered as decision variables of new research studies, thereby efficiency of the material usage can be thoroughly examined. Finally, extended objective functions, such as surface pattern, topography, energy performance, and economics of the generated geometry can be taken into account to developed enhanced designs.

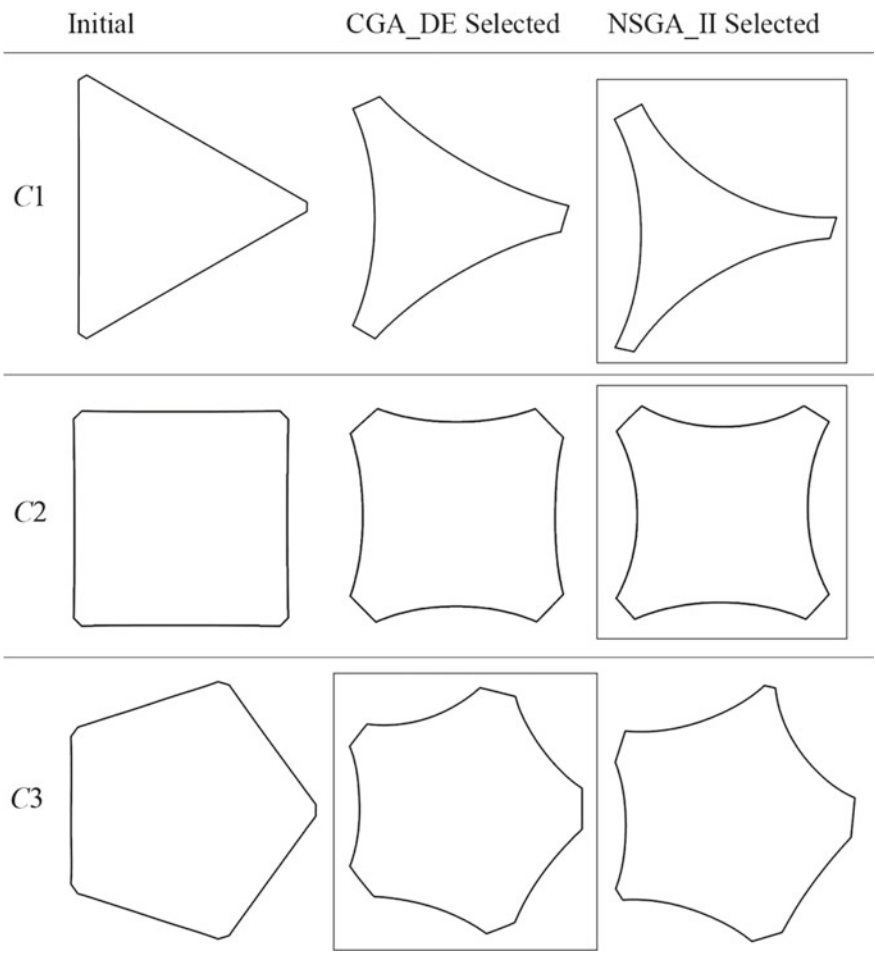


Fig. 26 Designer's choice from selected solutions for $C1$, $C2$ and $C3$

Acknowledgements This work has been supported by Yasar University (YU) "*Bilimsel Araştırma Projesi*" (Scientific Research Project) with the grant number BAP-037. We would like to express our sincere gratitude to the scientific committee of YU.

References

1. Panozzo, D., Block, P., & Sorkine-Hornung, O. (2013). Designing unreinforced masonry models. *ACM Transactions on Graphics (TOG), 32*, 91.
2. Adriaenssens, S., Block, P., Veenendaal, D., & Williams, C. (2014). *Shell structures for architecture: Form finding and optimization*. Taylor & Francis.
3. Block, P., & Lachauer, L. (2011). Closest-fit, compressiononly solutions for free form shells. In *IABSE/IASS London Symposium*, International Association for Shell Spatial Structures.
4. Rippmann, M., Lachauer, L., & Block, P. (2012). Interactive vault design. *International Journal of Space Structures, 27,* 219–230.
5. Rippmann, M., & Block, P. (2013). Funicular shell design exploration. In *ACADIA*, Waterloo, Canada.
6. Narayanan, N., & Ramamurthy, K. (2000). Structure and properties of aerated concrete: A review. *Cement & Concrete Composites, 22,* 321–329.
7. Wittmann, F. H., & Balkema, A. (1992) *Advances in autoclaved aerated concrete*. Citeseer.
8. Brisch, J. H., Nelson, R. L., & Francis, H. L. (1999) *Masonry: Materials, testing, and applications*. American Society for Testing Materials.
9. Aroni, I. S. (1993). *Autoclaved aerated concrete-properties, testing and design*. CRC Press.
10. Pacheco-Torgal, F., Lourenço, P. B., Labrincha, J., & Kumar, S. (2014). *Eco-efficient masonry bricks and blocks: Design, properties and durability*. Woodhead Publishing.
11. Preisinger, C., & Heimrath, M. (2014). Karamba—A toolkit for parametric structural design. *Structural Engineering International, 24,* 217–221.
12. Gulvanessian, H., Calgaro, J.-A., & Holický, M. (2002). *Designer's guide to EN 1990: Eurocode: Basis of structural design*. Thomas Telford.
13. Becker, M., & Golay, P. (1999). *Rhino NURBS 3D modeling*. New Riders.
14. McNeel, R. (2009). Rhinoceros 3D (Vol. 15).
15. McNeel, R. (2013). Grasshopper 3D.
16. Sariyildiz, I. (2012, November 15–17). Performative computational design. In *Keynote speech in: Proceedings of ICONARCH-I: International Congress of Architecture-I*, Konya, Turkey.
17. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA_II. *IEEE Transactions on Evolutionary Computation, 6,* 182–197.
18. Zitzler, E., Laumanns, M., & Thiele, L. (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. In *Eurogen, 2001* (pp. 95–100).
19. Cubukcuoglu, C., Chatzikonstantinou, I., Tasgetiren, M. F., Sariyildiz, I. S., & Pan, Q.-K. (2016). A multi-objective harmony search algorithm for sustainable design of floating settlements. *Algorithms, 9,* 51.
20. Ugurlu, C., Chatzikonstantinou, I., Sariyildiz, S., & Tasgetiren, M. F. (2015). Identification of sustainable designs for floating settlements using computational design techniques. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2303–2310).
21. Ugurlu, C., Chatzikonstantinou, I., Sariyildiz, S., & Tasgetiren, M. F. (2015). Evolutionary computation for architectural design of restaurant layouts. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2279–2286).
22. Goldberg, D. E. (2006). *Genetic algorithms*. Pearson Education India.
23. Brest, J., Greiner, S., Bošković, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10,* 646–657.
24. Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation, 19,* 45–76.
25. Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3,* 257–271.
26. Jiang, S., Ong, Y.-S., Zhang, J., & Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics, 44,* 2391–2404.

27. Delgarm, N., Sajadi, B., Delgarm, S., & Kowsary, F. (2016). A novel approach for the simulation-based optimization of the buildings energy consumption using NSGA-II: Case study in Iran. *Energy and Buildings, 127,* 552–560.
28. Delgarm, N., Sajadi, B., Kowsary, F., & Delgarm, S. (2016). Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization (PSO). *Applied Energy, 170,* 293–303.
29. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1,* 67–82.

Evolutionary Algorithms for Designing Self-sufficient Floating Neighborhoods

Ayca Kirimtat, Berk Ekici, Cemre Cubukcuoglu, Sevil Sariyildiz and Fatih Tasgetiren

Abstract Floating neighborhoods are innovative and promising urban areas for challenges in the development of cities and settlements. However, this design task requires a lot of considerations and technical challenges. Computational tools and methods can be beneficial to tackle the complexity of floating neighborhood design. This paper considers the design of a self-sufficient floating neighborhood by using computational intelligence techniques. In this respect, we consider a design problem for locating each neighborhood function in each cluster with a certain density within a floating neighborhood. In order to develop a self-sufficient floating neighborhood, we propose multi-objective evolutionary algorithms, namely, a self-adaptive real-coded genetic algorithm (CGA) as well as a self-adaptive real-coded genetic algorithm (CGA_DE) employing mutation operator of differential evolution algorithm. The only difference between CGA and CGA_DE is the fact that CGA uses random immigration of certain individuals into the population as a mutation operator whereas in the mutation phase of CGA_DE algorithm, the traditional mutation operator DE/rand/1/bin of DE algorithms. The arrangement of individual functions to develop each neighborhood function is further elaborated and formed by using Voronoi diagram algorithm. An application to design a self-sufficient floating neighborhood in Urla district, which is on the west coast of Turkey, İzmir, is presented.

A. Kirimtat · B. Ekici
Department of Architecture, Yasar University, İzmir, Turkey
e-mail: ayca.kirimtat@yasar.edu.tr

B. Ekici
e-mail: berk.ekici@yasar.edu.tr; B.Ekici-1@tudelft.nl

C. Cubukcuoglu
Department of Interior Architecture and Environmental Design, Yasar University, İzmir, Turkey
e-mail: cemre.cubukcuoglu@yasar.edu.tr; C.Cubukcuoglu@tudelft.nl

B. Ekici · C. Cubukcuoglu · S. Sariyildiz
Chair of Design Informatics, TU Delft, Delft, The Netherlands
e-mail: I.S.Sariyildiz@tudelft.nl

F. Tasgetiren (✉)
Department of International Logistics and Management, Yasar University, İzmir, Turkey
e-mail: fatih.tasgetiren@yasar.edu.tr

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_6

Keywords Computational design · Performance-based design · Self-sufficient Floating city · Multi-objective optimization · Urban design · Genetic algorithm Differential evolution · Form-finding · Voronoi diagram

1 Introduction

In the past decades, relevant and innovative solutions for cities have been exploring among researchers, city planners, as well as engineers. The reason is to tackle environmental problems caused by rising sea levels, natural disasters, and harmful effects of human activities. It is predicted by researchers that sea levels will continue to rise around the world [1]. For these reasons, we will see to face the problem of land shortage in the near future. In addition, extraordinary natural events such as earthquakes and heat waves will occur more frequent and more intense in the future. Many of the vulnerable cities, which are often in coastal locations, can be unpleasantly affected by problems mentioned above. In addition, a large number of world population is living in the coastal areas. For this reason, we have to consider the challenges in the development of cities and settlements to save humanity and our future life. Some of European and East Asian cities such as Tokyo, Shanghai, London, Rotterdam, and Hamburg take precautions against those challenges [2]. However, more advanced design solutions are still being sought in the world. Regarding this, floating settlements have emerged as new promising urban areas. Since the oceans are technically seen as international zones, they are defined as our last chance to remain in life on the earth [3]. In opposition to the traditional land renovation methods, floating settlements provide an exciting and environmentally friendly solution for land creation [3]. They have many advantages in terms of various aspects stated as follows:

- Protecting the marine eco-system,
- Construction on the sea is fast and easy,
- Easily removed or expanded,
- Durable against the seismic shocks,
- Presenting economical solutions especially when the sea depth is high or the seabed is very soft [4].

There are relatively few examples of floating settlements, such as

- A collaborative design of The SeaSteading Institute with Delta Sync [3],
- Floating city design proposal of AT Design Office in China [5],
- A concept design proposed by Baca Architects [6].

In the examples mentioned above, most of them deal with structural aspects, energy-efficiency, self-sufficiency, growth, movability, seakeeping, safety, water experience, and cost-efficiency. Based on these approaches, we aimed to consider both architectural and engineering design goals. We foresee that combining these two disciplines in the conceptual phase can play a key role in the development of future floating settlements.

Fig. 1 Project region: Urla, İzmir, Turkey

As an extension of [7], this paper develops a novel design methodology for the design of a self-sufficient floating neighborhood through implementations of evolutionary algorithms. Briefly, we combine knowledge from Floating Settlements Design practice with Computational Intelligence. Regarding the formulation of the complex problem in our study, we start with clustering the proposed site region into different zones. Then, we focus on locating each neighborhood function in each cluster with certain percentages. The neighborhood functions are residential, agricultural, public, and green areas. We aim to fit 30.000 people in this floating neighborhood. In the problem of distribution of the neighborhood functions in each cluster, inputs are the percentages of each function in each cluster and the cluster capacities. On the other hand, objective functions are to ensure walkability, scenery of the residents, as well as to enable the design to be cost-effective. We aim at finding desirable distributions of the functions in each cluster that contribute the self-sufficient design of floating neighborhoods. To solve this complex problem, we applied two different evolutionary algorithms, namely CGA and CGA_DE. The arrangement of individual functions to develop each neighborhood function is further elaborated and formed by using Voronoi diagram algorithm. An application to design a self-sufficient floating neighborhood in Urla, which is a coastal region located on the west of Turkey, İzmir is presented. So-called project region can be found in Fig. 1.

The rest of the paper is organized as follows: Sect. 2 introduces the design methodology and problem formulation developed in this chapter. Section 3 presents the evolutionary algorithms that have been applied in this chapter. Section 4 discusses

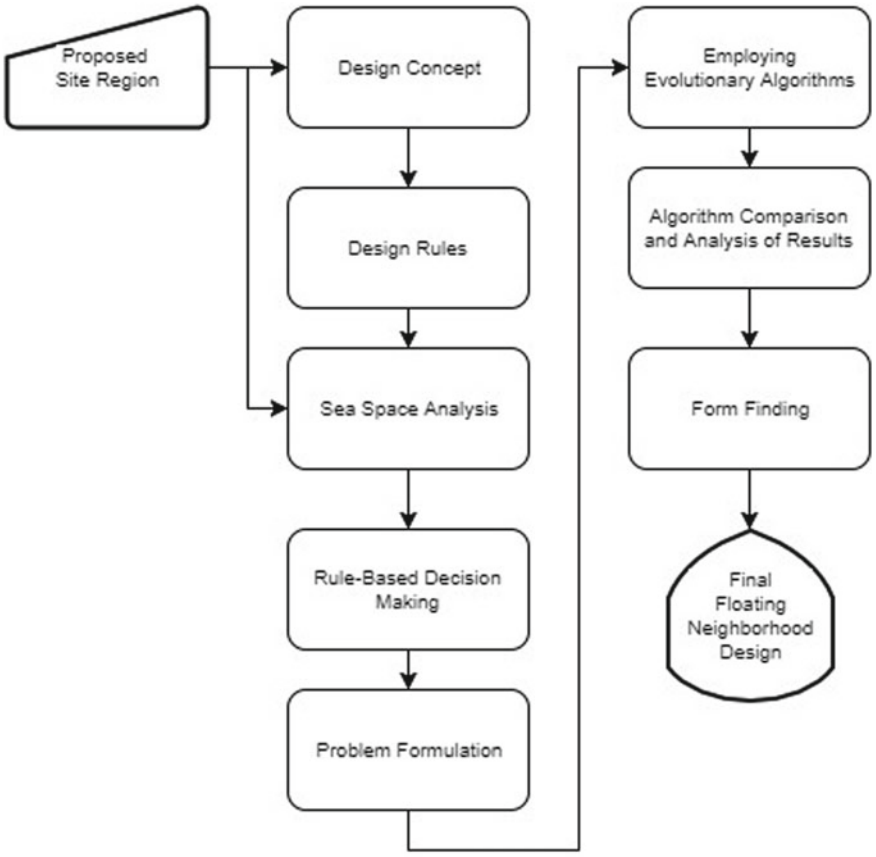


Fig. 2 Floating neighborhood design workflow

the computational results. Section 5 presents the form-finding step. Finally, Sect. 6 gives the conclusions.

2 Design Methodology

In this section, we introduce our proposed self-sufficient floating neighborhood design and its fundamentals. The design process consists of several steps as shown in Fig. 2.

2.1 Design Concept

The design concept is in relation with proposed project region. According to our preliminary site analysis, the urban area is required for further investigation in the conceptual phase. Thus, the design concept for the floating neighborhoods involves the following features:

- Proposed design should be **self-sufficient**. The food will be provided from the neighborhoods themselves by the help of agricultural areas.
- Distances between each function should be **walkable**. There will be fewer vehicles for transporting from one function to another.
- **Scenic** view should be desirable for both the residents in the project region and in the proposed floating neighborhood.
- The construction on the sea should be **cost-efficient** and required low budgets.

2.2 Design Rules

We infer the design rules through "if-then" statements, which is a common approach for representing design knowledge. In this approach, the design inferences are based on a process of obtaining new knowledge through existing knowledge. We could rather define them as our propositions, too. The design rules are mainly related to seabed characteristics of the project region, construction budget for residential areas, distances among the neighborhood functions and some specific proximity requirements. "if-then" rules considering in this study are listed as follows:

- If (sea depth is very shallow), then it is not proper for any neighborhood.
- If (sea depth is shallow) and (budget for residential function is low), then total budget is cost-effective.
- If (agricultural functions are very close to each other), and (residential functions are far from each other), then it is scenic for Urla people.
- If (green area functions are very close to each other), and (residential functions are far from each other), then it is scenic for Urla people.
- If there is a (very short distance between public and residential functions), and (very short distance between residential function and agricultural function), then it is a walkable city.
- If there is a (very short distance between residential function and public function), and (very short distance between residential function and green function), then it is a walkable city.

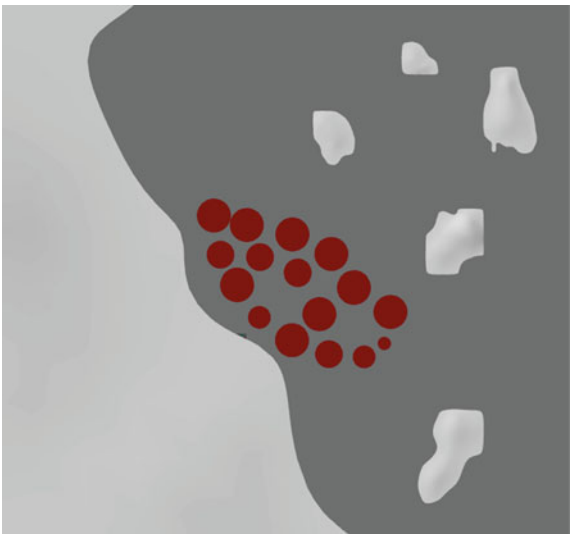


Fig. 3 Placements of 16 clusters on the intervention area

2.3 Sea Space Analysis

Sea space is analyzed to determine the proper intervention area for the location of the floating neighborhood. This decision is related to the water depth information. We selected the intervention area along the coastline that has water depths in between 5 and 20 m. Selected area is also based on our "if-then" rules stated above. As shown in Fig. 3, we divided the intervention area into 16 zones, so-called clusters where we distribute each neighborhood function in them.

2.4 Rule-Based Decision-Making

Rule-based decision-making is a way to determine which neighborhood function will be placed in which cluster with how much density. In other words, the decisions regarding the distribution of the neighborhood functions are based on "if-then" rules considering design goals, which are scenery, walkability, and cost-effectiveness. For instance, one cluster can separate its functions as 35% of green areas, 45% of residential areas, 20% of agricultural areas, and 0% of public areas. On the other hand, another cluster's percentages could be divided as 0% of green areas, 20% of residential areas, 40% of agricultural areas, and 40% of public areas. These percentages directly affect the performance of the design alternatives.

2.5 Problem Formulation

In this section, we explain the mathematical model of our proposed floating neighborhood design problem. Notations of decision variables, objectives, and constraints are given in Table 1.

According to the notations above, the mathematical model is described as

$$\begin{aligned} &\min\left(\tfrac{1}{Ec}, \tfrac{1}{Sc}, \tfrac{1}{Wa}\right) \\ &\text{subject to} \\ &100,000\,\text{m}^2 < aa < 120.000\,\text{m}^2 \end{aligned} \tag{1}$$

$$0 < p_{xy} < 1 \tag{2}$$

$$ra \geq 300,000\,\text{m}^2 \tag{3}$$

$$ca \geq 900,000\,\text{m}^2 \tag{4}$$

Decision variables of the floating neighborhood design problem consist of 64 variables, which are percentages to distribute four functions into 16 clusters.

Cost-Effectiveness, denoted as Ec, is one of the problem objectives as given in Eqs. (5)–(9). There is a close relationship between water depths and budget requirements. Locating the neighborhoods on shallow sea depths is an economical solution for settlements on the sea. Due to the cost-effectiveness priority, clusters are forced to locate on sea depth with (0–5 m) or (5–20 m). On the other hand, this objective aims at keeping total budget of the floating neighborhood between 1 million TL and 1.5 million TL for residential areas because of the highest budget requirements of them. These cost values are measured based on the proposed number of 30,000 people living in the neighborhood.

$$ss = \max\left(0, \min\left(1, \frac{sd - 20}{5 - 20}\right)\right) \tag{5}$$

$$su = \max\left(0, \min\left(1, \frac{sd - 5}{0 - 5}\right)\right) \tag{6}$$

$$abl = \max\left(0, \min\left(1, \frac{db - 1.5}{1 - 1.5}\right)\right) \tag{7}$$

where

$$db = ra * cu \tag{8}$$

$$Max\ Ec = ss + su + abl \tag{9}$$

Scenery objective consists of two main criteria: scenery for the residents in the floating neighborhood (Scf) and scenery from Urla to floating neighborhood (Scu). In order to achieve Scf part of this objective, we aim at keeping all functions close to the agricultural and the green areas to enable more sea view for residents in the

Table 1 Problem notations

Notations	Descriptions
p_{xy}	Density of function (x) in cluster (y)
x	Index of functions $x = 1, \ldots, 4$
y	Index of clusters $y = 1, \ldots, 16$
aa	Capacity of each cluster
sd	Sea depth (m)
Ec	Cost-effectiveness
Sc	Scenery
Wa	Walkability
ss	Shallow water
su	Shallow water
db	Total budget for residential areas
cu	Unit cost of residential areas $\left(\frac{\text{cost}}{\text{m}^2}\right)$
ra	Total area of residential functions $\left(\text{m}^2\right)$
ca	Total area of agricultural functions $\left(\text{m}^2\right)$
abl	The degree of low budget satisfaction
Scf	Scenery for residents in floating neighborhood
Scu	Scenery for Urla people
Ox	x coordinate of offshore
Oy	y coordinate of offshore
Cx	x coordinate for coastline of Urla
Cy	y coordinate for coastline of Urla
Rx	x coordinate of residentials
Ry	y coordinate of residentials
Ax	x coordinate of agricultural areas
Ay	y coordinate of agricultural areas
Gx	x coordinate of green areas
Gy	y coordinate of green areas
Px	x coordinate of public areas
Py	y coordinate of public areas
W_{xixj}	Walkability between functions xi and xj 1: residential, 2: agricultural, 3: green, 4: public
$D_{R,O}$	Manhattan distance between residential and offshore
$D_{A,C}$	Manhattan distance between agricultural and coast
$D_{G,C}$	Manhattan distance between green and coast
$D_{R,C}$	Manhattan distance between residential and coast
$D_{R,G}$	Manhattan distance between residential and green
$D_{R,A}$	Manhattan distance between residential and agricultural
$D_{R,P}$	Manhattan distance between residential and public

floating neighborhood. Regarding the *Scu* part of this objective, we aim at locating the residential areas more close to the water shore, in contrast, locating the agricultural and green areas more close to the Urla coastline. To assess the distances between functions and other places, we make use of Manhattan distance calculations as shown in Eqs. (10)–(13). After the distances are calculated, we aim at keeping those distance values in a certain interval as stated in below equation from (14) to (17). This interval is in between 300 and 1500 m for $D_{A,C}, D_{G,C}, D_{R,C}$. On the other hand, the acceptable range for the distance between residential and offshore is 0–1500 m.

$$D_{R,O} = |Rx - Ox| + |Ry - Oy| \tag{10}$$

$$D_{A,C} = |Ax - Cx| + |Ay - Cy| \tag{11}$$

$$D_{G,C} = |Gx - Cx| + |Gy - Cy| \tag{12}$$

$$D_{R,C} = |Rx - Cx| + |Ry - Cy| \tag{13}$$

$$0 < D_{R,O} < 1500 \tag{14}$$

$$300 < D_{A,C} < 1500 \tag{15}$$

$$300 < D_{G,C} < 1500 \tag{16}$$

$$300 < D_{R,C} < 1500 \tag{17}$$

Mathematical expressions that aim at keeping the distances between ranges are given from Eqs. (18)–(22).

$$d_1 = \max\left(0, \min\left(1, \frac{D_{R,O} - 1500}{0 - 1500}\right)\right) \tag{18}$$

$$d_2 = \max\left(0, \min\left(1, \frac{D_{A,C} - 1500}{300 - 1500}\right)\right) \tag{19}$$

$$d_3 = \max\left(0, \min\left(1, \frac{D_{G,C} - 1500}{300 - 1500}\right)\right) \tag{20}$$

$$d_4 = \max\left(0, \min\left(1, \frac{D_{R,C} - 1500}{300 - 1500}\right)\right) \tag{21}$$

$$\max Sc = \left(\frac{1}{min(d_1, (\min(\max(d_2, d_3), d_4))}\right) \tag{22}$$

Walkability is another objective of this design problem. Numerical walkability scores that assigned to each location according to their proximity related features are gathered from walk score. Walk score website [8] searches for walkable cities based on easy access to public transit, better commutes, and proximity to the people and places you love. Based on this approach, equation from (23) to (27) show the calculations of distances between functions. Then, we try to keep those distances in a walkable range.

$$D_{R,G} = |Rx - Gx| + |Ry - Gy| \tag{23}$$

$$D_{R,A} = |Rx - Ax| + |Ry - Ay| \tag{24}$$

$$D_{R,P} = |Rx - Px| + |Ry - Py| \tag{25}$$

$$300 < D_{xi,xi} < 1000 \quad i = 1, \ldots, 4 \; and \; j = 1, \ldots, 4 \tag{26}$$

$$\text{maximize } W_{xi,xj} = \max\left(0, \min\left(1, \frac{D_{xi,xj} - 1000}{300 - 1000}\right)\right) \quad for \; \forall \, xi \; and \; xj. \tag{27}$$

Problem Constraints are categorized into three parts as "Cluster Capacity Constraint", "Black Box Constraint" and "Functions' Areas Constraint". Cluster capacity constraint is given in Eq. (1), black box constraint is given in Eq. (2) and area constraints for neighborhood functions are given in Eqs. (3) and (4).

3 Proposed Evolutionary Algorithms

In this chapter, we present multi-objective evolutionary algorithms to solve the floating settlement design problem. Evolutionary algorithms (EAs) are popular optimization algorithms since they have been implemented to multi-objective problems (MOP). For this reason, they are entitled as multi-objective evolutionary algorithms (MOEA) . Amongst them, NSGA-II [9] and SPEA-2 [10] are the most studied ones to deal with MOPs in the literature. As an extension of [7], CGA and CGA_DE are developed and implemented in order to solve the complex floating neighborhood design problem.

Genetic algorithms (GA) are search heuristics based on the biological process of natural selection and evolution [11]. In GAs, individuals with decision variables in D dimensions are encoded into chromosomes to obtain an initial population that should be evolved over generations. At each generation, two individuals are chosen and mated from the population. Then, two individuals are crossed over to generate new solutions called offspring or child. Some individuals are mutated to escape from local minima. Ultimately, offspring population is added to parent population in order to select new individuals for the next generation. Figure 4 shows the overall scheme of the genetic algorithm (GA).

Real-coded GA needs a crossover and mutation operators. As a crossover operator, we employ a simple binomial crossover operator to generate offspring Q_i^t. In other words, two individuals, X_a and X_b, are selected from the parent population. Then, each dimension of the offspring is either taken from the first or second individual with a certain crossover probability CR_i^t. The outline of the crossover operator is given in Fig. 5 to generate offspring Q_i as well as an example is given in Table 2.

Regarding the mutation operator, some dimensions of offspring $Q_i^{j,t+1}$ can be mutated or perturbed with a small mutation probability as shown in Table 2.

Fig. 4 GA procedure

$$
\begin{aligned}
&Establish\ initial\ population\ P^t\ at\ generation\ t\\
&Evaluate\ individuals\ in\ P^t\\
&While\ (not\ termination) do\\
&\{\\
&\quad Select\ two\ individuals\ from\ P^t\\
&\quad Crossover\ individuals\ to\ produce\ offspring\ Q^t\\
&\quad Mutate\ some\ individuals\ in\ Q^t\\
&\quad Add\ offspring\ Q^t\ to\ individuals\ in\ P^t\\
&\quad Evaluate\ (P^t + Q^t) individuals\ in\ P^t\\
&\quad Select\ P^t\ individuals\ from\ (P^t + Q^t)\\
&\}\\
&End\ While\\
&End\ Algorithm
\end{aligned}
$$

Fig. 5 Crossover operator

$$
\begin{aligned}
&for\ j = 1\ to\ D\\
&\quad if\ r_j < CR_i\ then\\
&\quad\quad Q_i^{j,t+1} = X_a^{j,t}\\
&\quad else\\
&\quad\quad Q_i^{j,t+1} = X_b^{j,t}\\
&endfor
\end{aligned}
$$

Table 2 Binomial crossover and mutation operator

j	1	2	3	4	5
CR_i	0.70	0.70	0.70	0.70	0.70
r_j	0.80	0.25	0.92	0.67	0.11
$x_a^{j,t}$	0.15	0.70	0.35	0.45	0.95
$x_b^{j,t}$	0.65	0.75	0.10	0.25	0.05
$Q_i^{j,t+1}$	0.65	0.70	0.10	0.45	0.95
r_j	0.80	0.01	0.18	0.75	0.15
MR_i	0.02	0.02	0.02	0.02	0.02
$Q_i^{j,t+1}$	0.65	0.32	0.10	0.45	0.95

3.1 CGA and CGA_DE Algorithms

The real-coded GA mentioned above is actually for single objective real-parameter optimization problems. Now, in this section, we extend it to multi-objective floating neighborhood design problem. We propose a self-adaptive real-coded CGA and CGA_DE algorithms for the problem on hand. In both algorithms, initial popula-

tion is constructed uniformly and randomly within the given boundaries for each of 64 dimensions. The binomial crossover operator is used in both algorithms to generate offspring population. The difference between two algorithms comes from the different mutation operators used. In the CGA, we immigrate some random new individuals into the parent population with an amount of $MR_i \times \left|P^t\right|$. It means that if the MR_i is equal to 0.02 with the population size $\left|P^t\right| = 100$, two individuals selected from the population are randomly and uniformly regenerated within the boundaries of each dimension. On the other hand, we employ the traditional mutation operator DE/rand/1/bin of DE algorithms. In other words, a certain percent of the offspring population randomly goes under a mutation operator, after generating the offspring population. For the mutation operator, we propose a distinct strategy by employing the mutation operator of DE. Three individuals are randomly chosen from the parent population. Then, the difference between two individuals is multiplied by a uniform random number r between 0 and 1 in order to add to another individual randomly chosen as in Eq. (28).

$$Q_i^{j,t+1} = X_a^{j,t} + r \times \left(X_b^{j,t} - X_c^{j,t}\right) \tag{28}$$

Self-adaptive procedure in DE algorithms is first proposed by Brest et al. [12, 13]. We employ the same idea in this paper. Initially, CR_i and MR_i values are assigned to 0.9 and 0.05. These values are updated for each individual I at each generation t as in as in Eqs. (29) and (30).

$$MR_i^{t+1} = \begin{cases} MR_{min} + r_1 \cdot MR_{max} & if\ r_2 < p_1 \\ MR_i^t & otherwise \end{cases} \tag{29}$$

$$CR_i^{t+1} = \begin{cases} CR_{min} + r_1 \cdot CR_{max} & if\ r_2 < p_1 \\ CR_i^t & otherwise \end{cases} \tag{30}$$

where $r_j \in \{1, 2\}$ are uniform random numbers in the range [0, 1]. p_1 denotes the probability to adjust the CR_i and MR_i values. Parameters are taken as $p_1 = 0.1$, $CR_{min} = 0.1$, and $CR_{max} = 0.9$. In addition, MR_{min} and MR_{max} are taken as 0.01 and 0.05.

For both algorithms, we take advantage of non-dominated sorting procedure and constrained-domination rule of the NSGA-II algorithm [9], which is one of the most sophisticated multi-objective algorithms in the literature. The non-dominated sorting procedure is given in Fig. 6.

However, we employ the fast non-dominated sorting algorithm to create non-dominated fronts in both algorithms proposed in this paper.

Another key feature of the NSGA-II algorithm is the crowding distance. Suppose that we obtained a non-dominated set δ. The crowding distance of $\delta[i]_{dist}$ is calculated as in Fig. 7.

```
for each p ∈ P
    S_p = ∅
    n_p = 0
    for each q ∈ P
        if (p ≺ q) then                    If p dominated q
            S_p = S_p ∪ {q}                Add q to the set of solutions dominated by p
        else if (q ≺ p) then
            n_p = n_p + 1                  Increment the domination counter of p
    if n_p = 0 then                        p belongs to the first front
        p_rank = 1
        F_1 = F_1 ∪ {p}
i = 1                                      Initialize the front counter
while F_i ≠ ∅
    Q = ∅                                  Used to store the members of the next front
    for each p ∈ F_i
        for each q ∈ S_p
            n_q = n_q − 1
            if n_q = 0 then                q belongs to the next front
                q_rank = i + 1
                Q = Q ∪ {q}
i = i + 1
F_i = Q
End
```

Fig. 6 Non-dominated sorting procedure

1. $L = |\delta|$
2. $for\ each\ individual\ i, set\ \delta[i]_{dist} = 0$
3. $for\ each\ objective\ m$
   a. $\delta = sort(\delta, m)\ in\ an\ ascending\ order$
   b. $\delta[1]_{dist} = \delta[L]_{dist} = \infty$
   c. $for\ i = 2\ to\ L - 1$
      $\delta[i]_{dist} = \delta[i]_{dist} + (\delta[i+1][m] - \delta[i-1][m])/(f_m^{max} - f_m^{min})$

Fig. 7 Crowding distance calculation

3.1.1 Comparison Operators

For unconstrained multi-objective optimization, the NSGA-II employs the crowded-comparison operator ($\prec$). It directs the selection process at various stages of the algorithm. Every individual i in the population has a nomination rank (i_{rank}) and a crowding distance rank ($i_{distance}$). It defines a crowded-comparison operator ($\prec$) as follows:

$$if\ (i_{rank} < j_{rank}) then\ i \prec j$$
$$or\ ((i_{rank} = j_{rank})\ and\ (i_{distance} > j_{distance}))$$

1. $Set\ t = 0\ and\ create\ a\ random\ parent\ population\ P^t\ with\ N$
2. $Perform\ binomial\ and\ mutation\ operator\ as\ in\ Fig.\ 5\ and\ Table\ II\ on\ P^t\ to\ obtain\ Q^t\ with\ size\ N$
3. $If\ the\ termination\ criteria\ is\ satisfied, stop\ and\ return\ P^t.$
4. $Set\ R^t = P^t \cup Q^t\ to\ combine\ two\ populations$
5. $Perform\ non-dominated\ sorting\ procedure\ for\ R^t\ to\ set\ the\ non-dominated\ fronts\ f_1, f_2, .., f_k.$
6. $For\ i = 1, .., k, repeat\ the\ following\ steps:$
   a. $Calculate\ crowding\ distance\ for\ each\ solution\ in\ f_i.$
   b. $Create\ P^{t+1}\ as\ follows:$
      i. $if\ |P^{t+1}| + |f_i| \leq N, then\ set\ P^{t+1} = P^{t+1} \cup f^t$
      ii. $if\ |P^{t+1}| + |f_i| > N, then\ add\ the\ least\ crowded\ N - |P^{t+1}|\ solutions\ from\ f_i\ to\ P^{t+1}.$
7. $Use\ binary\ tournament\ selection\ with\ constrained\ domination\ rule\ to\ select\ parents\ from\ P^{t+1}.$
8. $Apply\ binomial\ crossover\ and\ mutation\ operator\ to\ P^{t+1}\ and\ obtain\ new\ offspring\ population\ Q^{t+1}\ with\ size\ N$
9. $Set\ t = t + 1\ and\ go\ to\ Step\ 3.$

Fig. 8 CGA and CGA_DE algorithm

It means that lower (better) rank is preferred between two individuals with different non-domination ranks. On the other hand, if both individuals have the same rank, it prefers individual with lesser crowded region.

Regarding the constrained multi-objective optimization, when comparing two individuals, the situation is somewhat different. Three cases can be observed: (1) one is feasible, the other is not; (2) both are infeasible; and (3) both solutions are feasible. For the constrained multi-objective optimization, NSGA-II modifies the definition of *domination* between two solutions as follows:

An individual i is considered to constrained-dominate an individual j under the following conditions:

(1) Individual i is feasible and individual j is infeasible.
(2) Both individuals are infeasible, but individual i has a smaller constraint violation.
(3) Both individuals, i and j are feasible and individual i dominates individual j with crowded-comparison rule as follows:

$$if\ (i_{rank} < j_{rank})\ then\ i \prec j$$
$$or\ ((i_{rank} = j_{rank})\ and\ (i_{distance} > j_{distance}))$$

Now we are ready to outline the CGA and CGA_DE algorithms in Fig. 8.

As mentioned before, the difference between two algorithms comes from the different mutation operators used in step 2 and step 8 in Fig. 8. In the CGA, we immigrate some random new individuals with an amount of $MR_i \times \left|P^t\right|$ into the parent population. If the mutation rate MR_i is equal to 0.02 with the population size $\left|P^t\right| = 100$, two individuals chosen from the population. Then, these individuals are regenerated randomly and uniformly within the boundaries of each dimension. In

case of CGA_DE algorithm, we employ the traditional mutation operator (rand/1/bin) of DE algorithms. In other words, after generating the offspring population, a certain percent of the child population randomly goes under a mutation operator. Again, if the mutation rate, MR_i is equal to 0.02 with the population size $\left|P^t\right| = 100$, two individuals will be immigrated into the parent population. For each one, we randomly select three individuals from the parent population. The difference of two individuals is taken, multiplied by a uniform random number. r between 0 and 1 in order to add to the third individual as in Eq. (28).

4 Computational Results

In this study, we proposed CGA and CGA_DE algorithms to deal with a multi-objective problem of self-sufficient floating neighborhood design. These algorithms are run for five independent replications on a computer, which has 2.6 GHz Intel core i7-6700HQ processor, 8 GBx2 DDR3 memory and 256 GB SSD. The population size is taken as $\left|P^t\right| = 100$. As a termination criterion, each algorithm was run for 100 generations. Initial values of the parameters are assigned to $CR_i = 0.9$ and $MR_i = 0.5$. With the self-adaptive procedure, we update these values with Eqs. (29) and (30) at each generation.

In Fig. 9, the standard deviations of five replications of the non-dominated solutions for each algorithm are shown. The standard deviation refers to the number of non-dominated solutions for each of 25 generations until 100th generation. From this graph below, we observe that there is no significant change after 75th generation until 100th generation. We observe zero standard deviation in this generation range for both algorithms. Thus, we determine the termination criteria as 100th generation for both algorithms.

Regarding the problem objectives, we analyzed minimum, average and maximum values of each objective function for each algorithm as can be found in Fig. 10, 11, 12, 13, 14, and 15. As it is seen from those graphs, each objective function has different characteristics in terms of their minimum, maximum, and average values generated by each algorithm.

Behaviors of cost-effectiveness are presented in Figs. 10 and 11. Until 50th generation, CGA and CGA_DE algorithms have presented different behavior in terms of average values. However, after 50th generation, both algorithms have generated similar results for the cost-effectiveness objective function.

Related to scenery objective function's results presented by both algorithms, maximum and average values are very similar as shown in Figs. 12 and 13. However, CGA_DE investigates a larger range of solutions, i.e., the results that are more diverse.

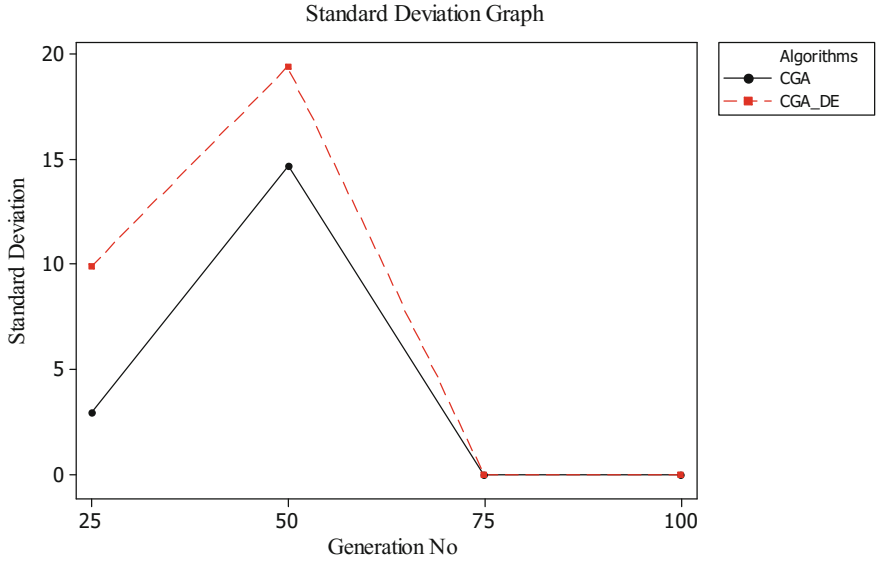


Fig. 9 Standard deviation graph for CGA and CGA_DE

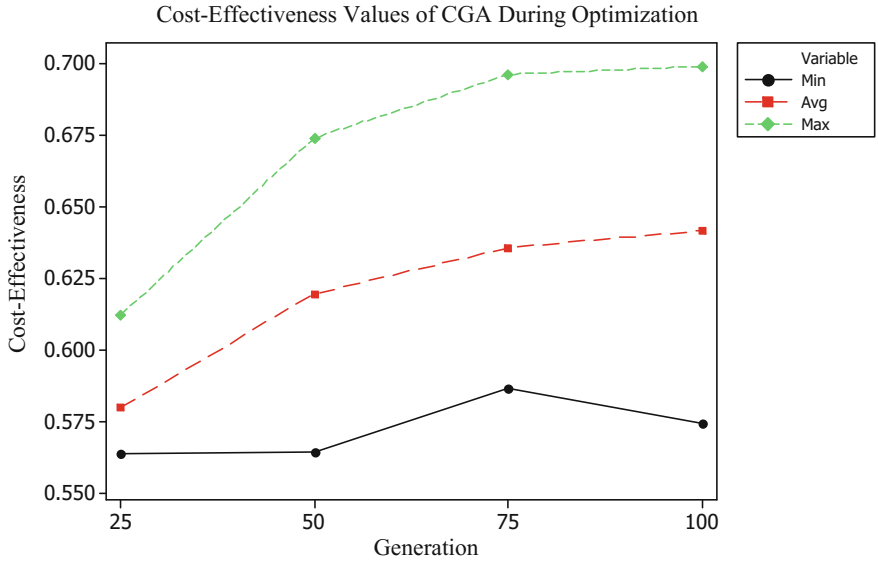


Fig. 10 Cost-effectiveness values of CGA during the 100th generation

In terms of walkability objective, CGA_DE again presents larger range of results as shown in Fig. 14 and 15. CGA is able to achieve 0.25% in maximum value at 100th generation. In addition, the maximum values presented by CGA were almost

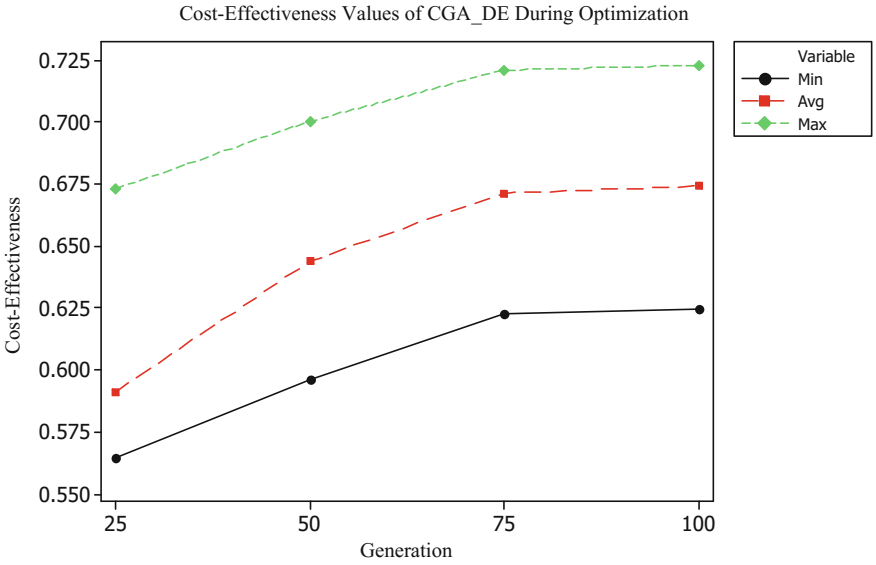


Fig. 11 Cost-effectiveness values of CGA_DE during the 100th generation

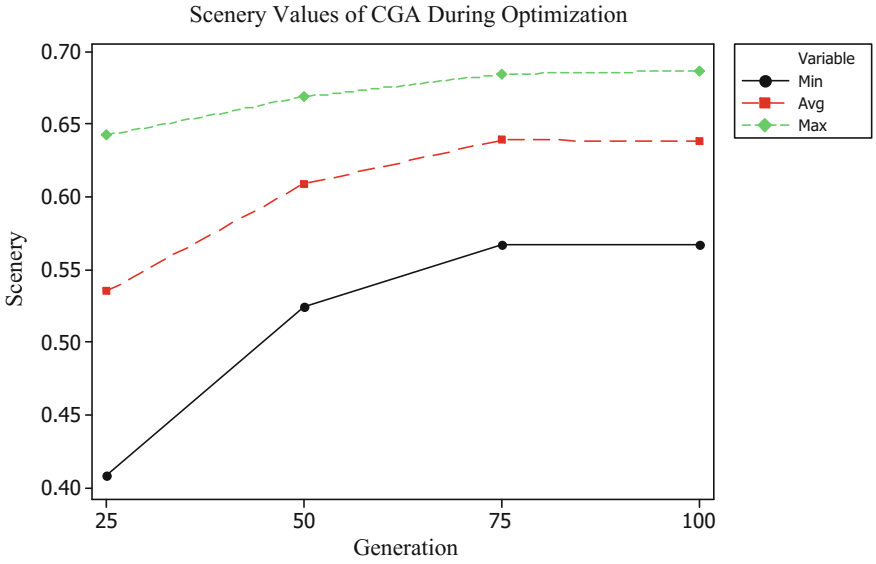


Fig. 12 Scenery values of CGA during the 100th generation

the same in each generation. On the other hand, CGA_DE reaches 0.40% maximum value and presents changing results, which is an advantage for the design optimization problems.

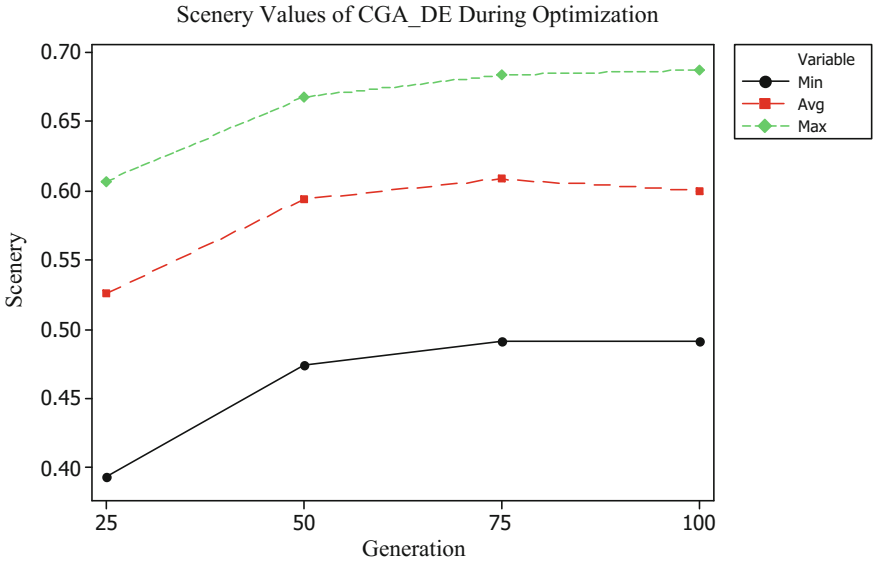


Fig. 13 Scenery values of CGA_DE during the 100th generation

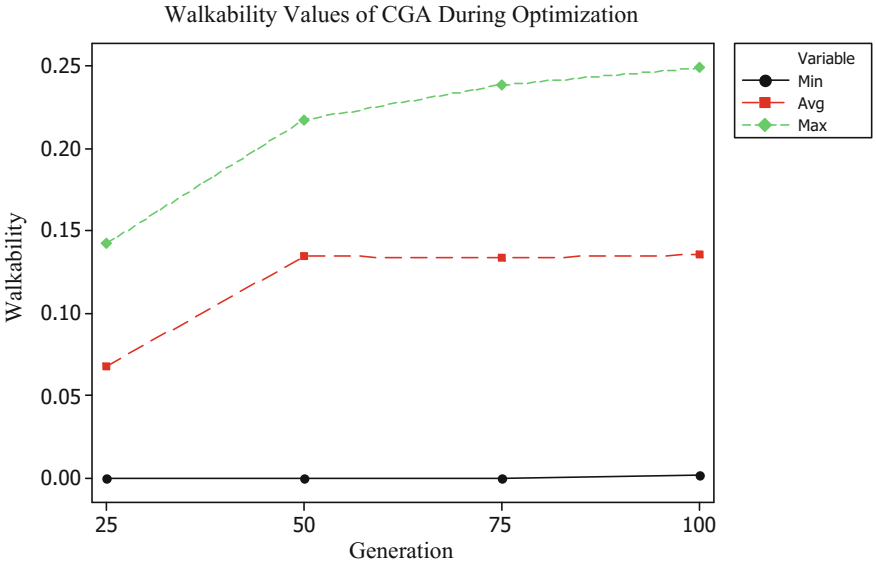


Fig. 14 Walkability values of CGA during the 100th generation

Finally, the non-dominated solutions of CGA and CGA_DE after the 100th generation are given in Fig. 16. In the 3D scatter plot below, red dots correspond to the non-dominated individuals gathered from the CGA_DE; blue dots refer to the

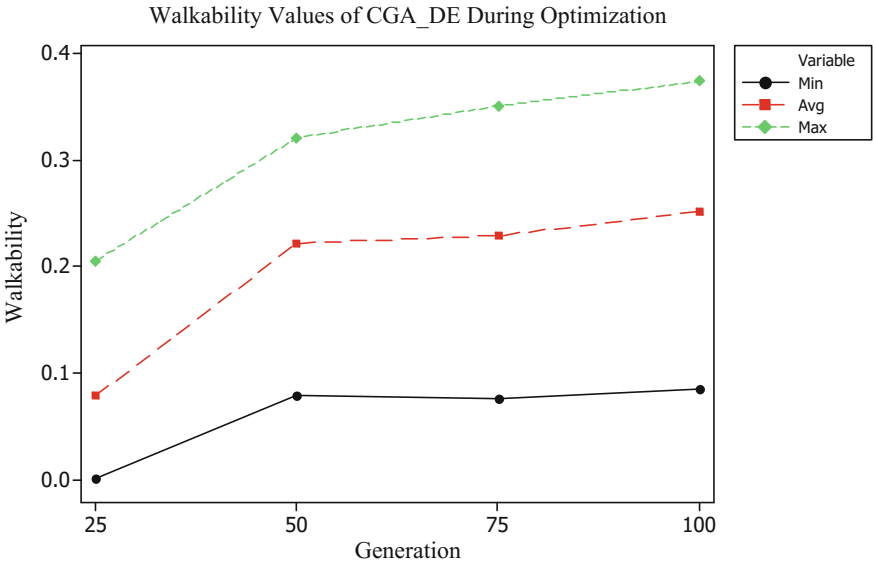


Fig. 15 Walkability values of CGA_DE during the 100th generation

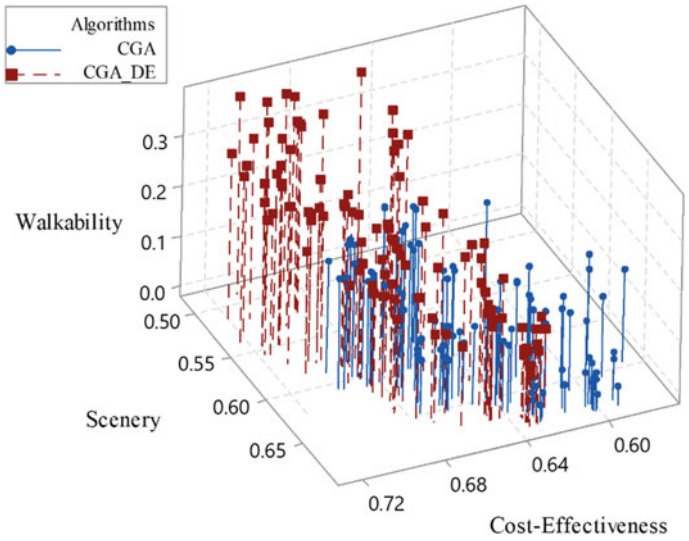


Fig. 16 Non-dominated solutions in the 100th generation for CGA and CGA_DE

results obtained by CGA algorithm. The Pareto front approximation in both cases is investigated for all three objectives, supporting the claim of the objectives being conflicting with each other.

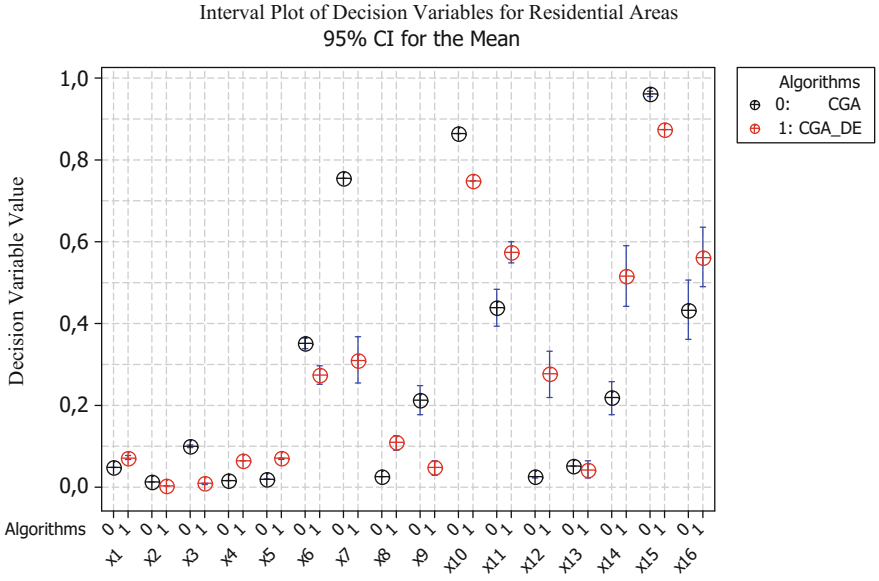


Fig. 17 Interval plot of decision variables for residential areas

From the scatter plot above, it can be seen that solutions present very similar performances but different compositions of 64 design variables. With regard to the architectural qualities of the solutions, we need to analyze the influence of those 64 decision variables on each objective function for each algorithm. In this respect, we generated interval plots for each decision variable category considered in the optimization problem. By this way, we present behavior of each design variable for both CGA and CGA_DE algorithms. Thus, architectural specifications can be better discussed. All interval plots for different decision variable combinations are given in Fig. 17, 18, 19, and 20. In these figures, x1–x16 represents the densities of the residential areas in each cluster from one to sixteen. The percentages of the agricultural areas are represented by y1–y16. We used the notations of z1–z16 for public areas and t1–t16 for the percentages related to green areas.

As can be seen in Fig. 17, the percentages related to residential areas are almost the same in the clusters that are close to the coastline. In addition, the corresponding decision variables x1, x2, x3, x4, and x5 present lower results for CGA_DE algorithm than CGA algorithm. The percentages of residential areas in the clusters that are close to the offshore, in contrast, are higher in both algorithms. In few cases of decision variables with regard to the offshore clusters, the percentages are not as high as expected. This situation can be explained by the limited cluster capacities or the limitations caused by walkability objective.

The decision variables related to the agricultural areas present the highest percentages compared to the densities of other neighborhood functions. This can be explained by the high impact of the agricultural areas on both scenery objective and

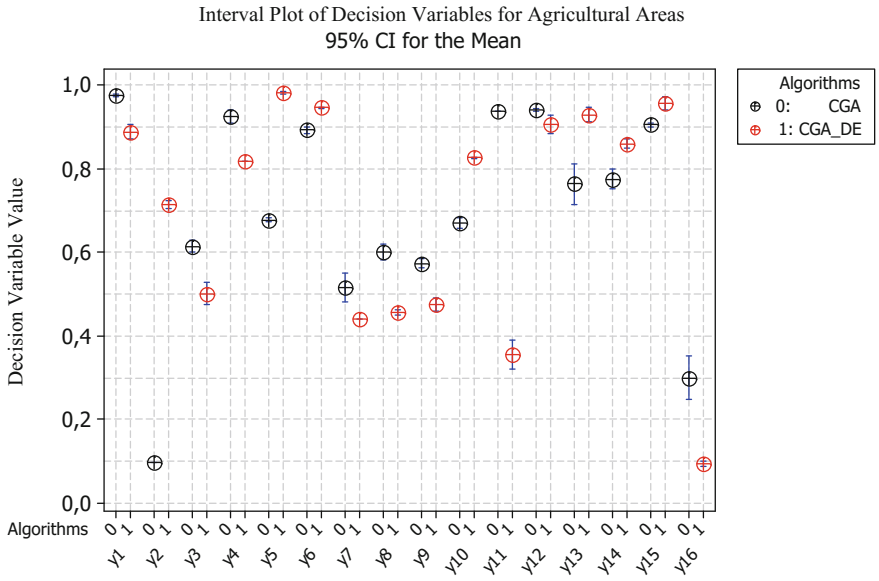


Fig. 18 Interval plot of decision variables for agricultural areas

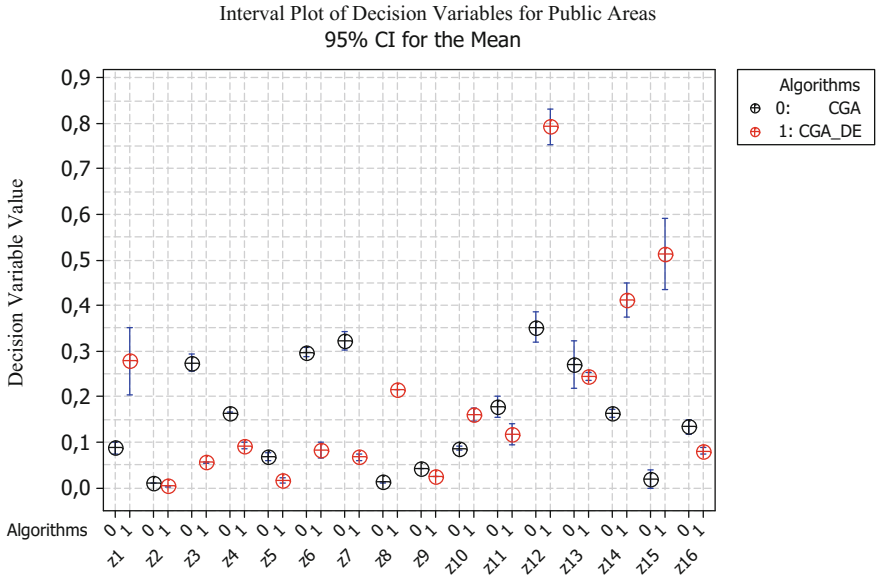


Fig. 19 Interval plot of decision variables for public areas

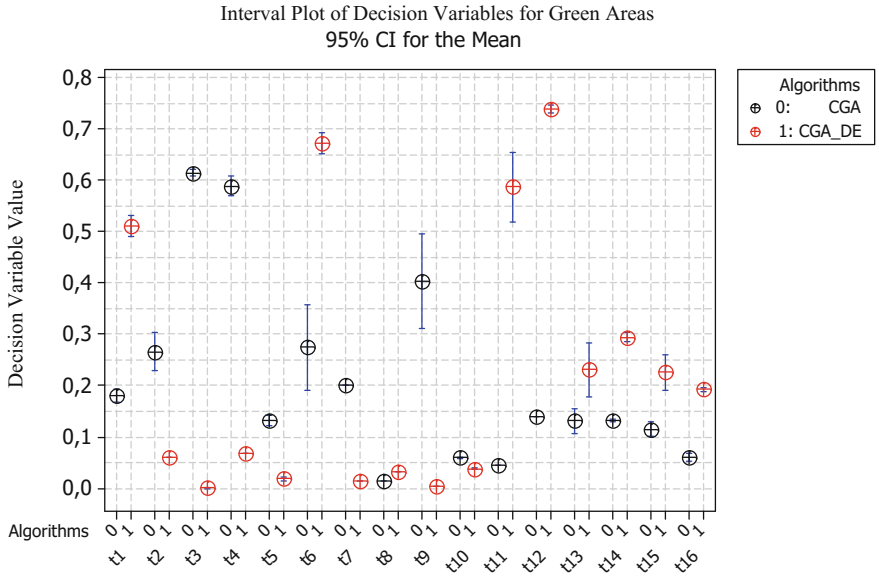


Fig. 20 Interval plot of decision variables for green areas

problem constraints. From the point of algorithm comparison, CGA_DE presents more uniform results than CGA. As can be seen in Fig. 18, CGA picks the percentages of the agricultural areas in some cases.

The percentages of public areas consist just in walkability objective as controllable variables. In contrast to the agricultural areas, public areas have the lowest percentages in total compared to the percentages of other areas as shown in Fig. 19.

The occupancy rates of green areas in clusters that are close to the coastline are higher than in the cluster that are close to the offshore. It means that the effect of green areas on scenery is relatively high. For the decision variables related to green areas, CGA presents more changing results than CGA_DE as shown in Fig. 20.

5 Generative Model and Form-Finding

Generative model of the floating neighborhood problem has been created in the Grasshopper algorithmic modeling environment [14]. Grasshopper is a plug-in for Rhinoceros, which is a well-known CAD program. This section consists of two parts, which are parametric definition of functions' distributions based on rules and form generation of the chosen result.

Fig. 21 Representation of info graphics for each cluster

5.1 Parametric Definition

In the parametric definition of floating neighborhoods, the percentages of each neighborhood function in each cluster are generated based on rule-based decision-making. In order to represent densities, we use info graphics. The info graphics are illustrated in the Rhino model with four different colors in Fig. 21. This figure belongs to one design alternative gathered after optimization, which presents both walkable and scenic characteristic. In so-called infographics, pink, purple, yellow, and green colors correspond to residential areas, public areas, agricultural areas, and green areas, respectively.

5.2 Form Generation

In this section, we explained how the form of floating neighborhood is generated in relation with infographics. Voronoi diagram algorithm is used for further elaborations of the floating neighborhood. As can be seen in Fig. 22, each cluster has different sizes based on their different capacities.

Fig. 22 16 clusters with different sizes [7]

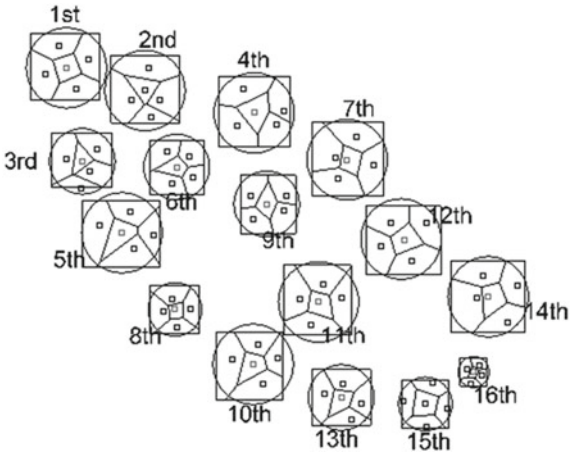


Fig. 23 Voronoi diagram for each cluster [7]

After determination of clusters' sizes, Voronoi diagram algorithm [15] divides each cluster into parts that each one corresponds to each neighborhood function. Figure 23 shows the Voronoi diagrams within the floating neighborhood.

After the final adjustments on the generative model, computer rendering are get prepared. As can be seen in Figs. 24, 25 and 26, final self-sufficient floating neighborhood design is indeed plausible.

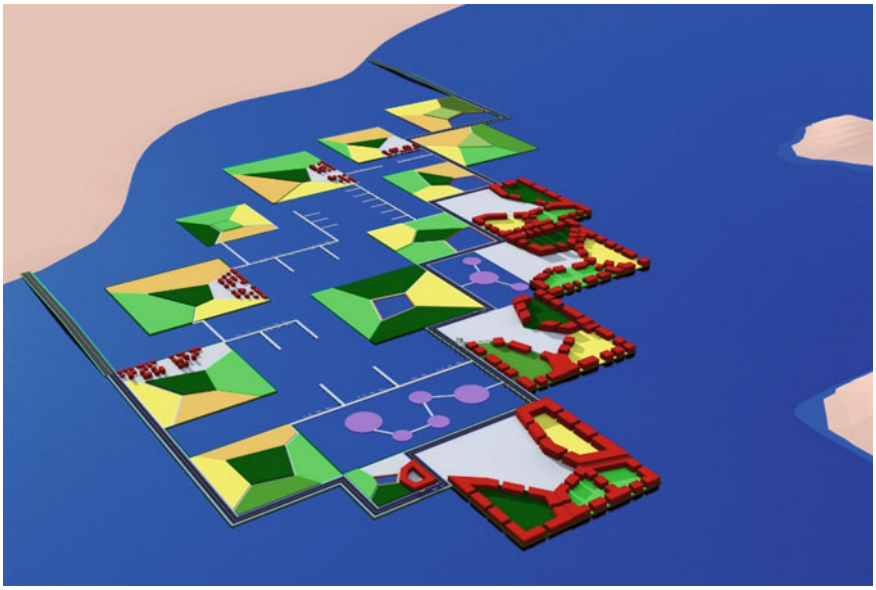


Fig. 24 Computer rendering from general view to floating neighborhoods [7]

Fig. 25 Computer rendering from agricultural areas

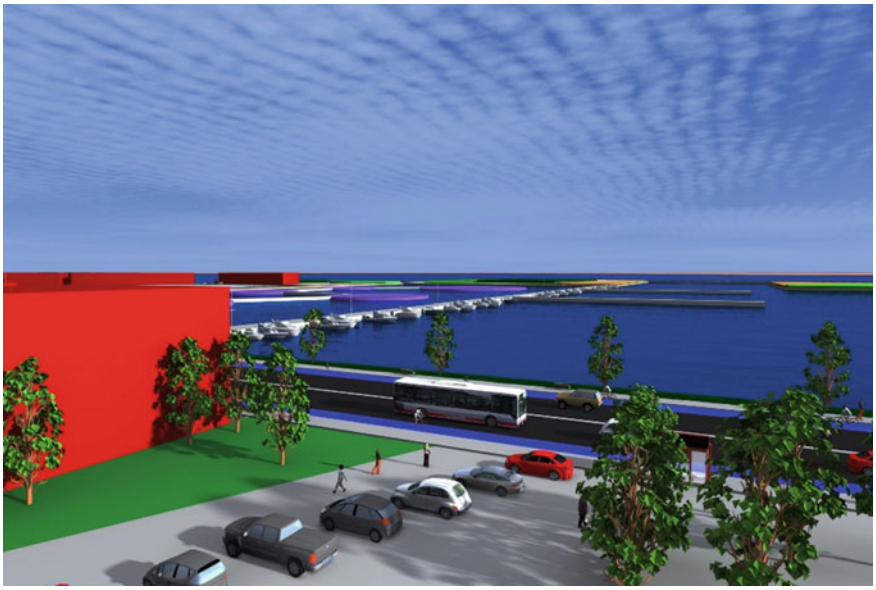


Fig. 26 Computer rendering from residential and public areas

6 Conclusion

In this study, two different evolutionary algorithms, namely CGA and CGA_DE are implemented to solve the complex design problem of floating neighborhoods. The design goals consist of both architectural and engineering aspects, which are cost-efficiency, scenery and walkability. These objectives are conflicting. Thus, the design methodology combines the floating structures design practices with computational intelligence and utilizes computational design strategies. The comparison of algorithms is performed through graphical representations. Both algorithms have presented competitive results. However, CGA_DE found more spread-out solutions than CGA for this multi-objective constrained real-parameter floating neighborhood design problem. In addition to objective function analysis, we also discussed the results of decision variables to analyze architectural qualities of the solutions. As a result, we achieved Pareto front approximations for both algorithms with non-dominated individuals, and both feasible and indeed plausible architectural design solutions.

References

1. Pachauri, R. K., Meyer, L., & Intergovernmental Panel on Climate Change (Eds.). (2015). *Climate change 2014: Synthesis report*. Geneva, Switzerland: Intergovernmental Panel on Climate Change.
2. Field, C. B., & Intergovernmental Panel on Climate Change (Eds.). (2012). *Managing the risks of extreme events and disasters to advance climate change adaption: Special report of the Intergovernmental Panel on Climate Change*. New York, NY: Cambridge University Press.
3. DeltaSync and The Seasteading Report: Design input, location design. Retrieved December 3, 2017, from https://www.seasteading.org/floating-city-project/.
4. Watanabe, E., Wang, M., Utsunomiya, T., & Moan, T. (2017). Very Large Floating Structures: Application, Analysis and Design; Technical Report No. 2004-02. Retrieved December 3, 2017, from http://www.eng.nus.edu.sg/core/Report%20200402.pdf.
5. Floating City concept by AT Design Office features underwater roads and submarines, Dezeen Magazine. Retrieved December 3, 2017, from https://www.dezeen.com/2014/05/13/floating-city-at-design-office/.
6. Floating settlements proposal by Baca Architects. Retrieved December 3, 2017, from https://www.dezeen.com/2017/06/09/video-baca-architects-floating-architecture-homes-movie/.
7. Kirimtat, A., Chatzikonstantinou, I., Sariyildiz, S., & Tartar, A. (2015). Designing self-sufficient floating neighborhoods using computational decision support. In *CEC 2015*, Sendai, Japan.
8. Walk Score Website. https://www.walkscore.com/.
9. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6,* 182–197.
10. Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. In *Eurogen, 2001* (pp. 95–100).
11. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
12. Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Žumer, V. (2006). Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*(6), 646–657.
13. Brest, J. (2009). Constrained real-parameter optimization with e-self-adaptive differential evolution. In E. Mezura-Montes (Ed.), *Constraint-handling in evolutionary optimization*. *Studies in computational intelligence series* (Vol. 198). Springer.
14. Grasshopper, Algorithmic Modeling for Rhino. http://www.grasshopper3d.com/.
15. Brandt, J. W., & Algazi, V. R. (1992). Continuous skeleton computation by Voronoi diagram. *CVGIP: Image understanding, 55*(3), 329–338.

Driving Assistance for Optimal Trip Planning of Electric Vehicle Using Multi-objective Evolutionary Algorithms

M. Khanra, A. K. Nandi and W. Vaz

Abstract This chapter proposes a novel strategy (the so-called Driving Assistance system for Optimal Trip Planning, DAOTP) for electric vehicles (EV), which works based on a multi-objective optimization approach. DAOTP provides an optimal driving strategy (ODS) corresponding to minimum energy consumption, travel time, and discomfort. Since these objectives are conflicting with each other, a multi-objective optimization tool is adopted to solve the problem. Based on the current trip information, first a set of Pareto-optimal solutions (ODSs) is obtained, and then a preferred ODS is selected from them corresponding to a higher level information or using problem-specific multi-criterion aspects for implementation. DAOTP works using route information obtained through GPS (Global Positioning System), Internet, etc. Route information includes the road surface type, weather conditions, trip start and end points, etc. In this chapter, the DAOTP system architecture, concerned MOOP, and the related EV models are presented in details. A brief explanation of multi-objective genetic algorithm that solves the present MOOP is given. The operation of DAOTP is elaborately presented with an application in a simple urban micro-trip planning. After that, the application of DAOTP for highway and complex trip planning are presented. The DAOTP results found in high-speed driving cycles are analyzed. The effectiveness of DAOTP is presented considering some sample trips, and its results are analyzed with varied route characteristics and trip complexity.

Keywords Electric vehicle · Trip planning · Driving assistance
Multi-objective optimization · Evolutionary algorithm · Energy consumption
Trip time · Driving comfort

M. Khanra · A. K. Nandi (✉)
CSIR-Central Mechanical Engineering Research Institute (CSIR-CMERI), MG Avenue, Durgapur, West Bengal, India
e-mail: nandiarup@yahoo.com

W. Vaz
Department of Engineering, University of Wisconsin Colleges, UW-Fox Valley, Menasha, USA

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_7

1 Introduction

As the time passes, research efforts are bringing more and more improvements to the vehicular sector. These improvements are broadly classified into two categories. One involves the hardware of the vehicle. Other improvements, which belong to the Intelligent Transportation System domain, are for driver assistance. The primary goal is to ease the driving experience and properly manage available resources. Some aspects are automatic fixed-distance following navigation and route guidance, operational cost reduction, obstacle avoidance, maintaining the lane, parking assistance, improving comfort and safety, and so on. Some of the various strategies and initiatives that were adopted in the past are described as follows.

In the Automated Highway Systems (AHS), multiple vehicles are coupled electronically for controlling them automatically toward driving at the same speed and safe following distances [81, 91]. AHS forms a semi-autonomous platoon by controlling the vehicles longitudinally as well as tangentially [7]. In a semi-autonomous platoon, one vehicle (preferably which is in the front) is driven actively, and the other vehicles are followed automatically. Under the project KONVOI, funded by German's Federal Ministry of Economics and Technology, an Advanced Driver Assistance System (ADAS) was developed to automatically control the longitudinal and lateral movement of the vehicles behind the actively driven preceding vehicle [57]. ADAS includes six primary components: GPS (Global Positioning System) [41], UMTS (Universal Mobile Telecommunications System) for vehicle–infrastructure communication [91], DIS (Driver Information System) [93], AG (Automated Guidance) [75], WLAN (Wireless Local Area Network) for vehicle–vehicle communication [45], and ACC (Adaptive Cruise Control) [99]. All these modules are connected to a central server. Details of various AHS and ADAS forms and related undertaken projects can be found in [5]. Such systems are primarily suitable for traffic control and safety. AHS and ADAS do not focus on any direct energy management of the vehicle, and can be adopted irrespective of the vehicle type.

There are various navigation systems that help drivers by providing an eco-driving or eco-routing functionality. Vexia's ecoNav solution [94] assists the driver by proving the fuel consumption rate that is determined based on the characteristics of vehicle and driving behavior. Garmin's ecoRoute software [38] finds a fuel-efficient route on the basis of road information and vehicle characteristics. The Freightliner Predictive Cruise Control [86] minimizes the fuel consumption rate by fine-tuning the vehicle speed corresponding to the route driving cycle. Like AHS, the above navigation systems can be adopted irrespective of vehicle type.

Traditionally, routing has focused on exploring shortest paths in a network based on the costs of positive and static edges that represent the distance between two nodes. Contrary to other vehicles, solving routing problems for EV is a challenging work because of the negative edge cost due to regenerative braking, vehicle weight, and limited battery capacity (resulting in the cost of a path being no longer just the sum of its edge costs). Considering these challenges, an attempt was made in [30, 76] to find a solution for energy-optimal routing for EVs. Recently, a few industrial initiatives

were undertaken to deal with the routing issues of EV [13]. NAVTEQ ADAS [69] deals with finding routes that can minimize ICEV fuel consumption and EFV energy consumption by considering additional route information such as altitude, slopes, and curves.

A European Commission-sponsored research project called "Intelligent Dynamics for fully Electric Vehicles" (ID4EV) [77] was initiated to establish an intelligent system which is most energy-efficient and HMI (Human–Machine Interface) capability, and safe braking and chassis systems as well as intelligent functionalities and new Human–Machine Interface (HMI) concepts for the needs of EVs, as for such system it is important for FEVs to have wide acceptability [48]. Various aspects of ID4EV projects can be found in [106].

Another way of driving assistance is the adoption of cooperative driving (a type of AHS), where a flexible platooning is formed among the vehicles which are available within a short distance over a couple of lanes [53]. The cooperative driving technology is based on two studies: Super Smart Vehicle System (SSVS) [87] and inter-vehicle communications with infrared rays from [35]. This driving assistance is primarily focused on the compatibility of safety and efficiency of road traffic.

Similar to other vehicles, EVs possess an inadequate energy storage that results in a short driving range. Moreover, refueling of EV is time consuming due to the long charging time of the battery. In addition, inadequate battery charging stations in contrast to fueling stations of Internal Combustion Engine (ICE)-based vehicles is a critical issue for EVs. As a consequence, efficient usage of battery energy is crucial for EVs. On the other hand, that is not so important for ICE vehicles. Thus, in order to extend the EV range by minimizing energy consumptions, it is required to adopt an optimal driving strategy (ODS) during driving. Additionally, due to the above-stated shortcomings, an appropriate trip plan corresponding to the present battery state-of-charge (SOC) is required prior to the journey start. Many times, due to unanticipated circumstances, deviation from the previous trip plan is happening. The driver needs to deviate from the preplanned trip during the journey. In such conditions, an online assistance is helpful to the drivers toward replanning the rest of the journey. Various issues are required to consider during trip planning such as whether with the remaining battery charge can reach to destination or not [96], what driving strategy to be followed corresponding to the road type for comfortable journey and minimum battery charge depletion [40, 42] and shortest trip duration [60], and what extent the trip expenditure (e.g., cost of charging) can be reduced [82].

A driving strategy refers to the value(s) of all or any of the parameters, namely, speed, acceleration, acceleration duration, deceleration, and deceleration to be applied by the driver during running a vehicle. In order to maintain the above issues, drivers are required to assist during trip performing by suggesting the optimal driving strategy to be followed and corresponding EV range and trip end battery SOC to be expected. An optimal driving strategy can be determined by taking into consideration concurrently three conflicting objectives such as minimum energy consumption, minimum trip time, and maximum driving comfort.

In this regard, the above-stated driving assistance systems are useful but not sufficient for an EV to compete or become comparable to the other vehicle types. In addition to knowing that an EV is a zero emission vehicle and comparatively more efficient such a driving assistance for optimal trip planning is extremely necessary to instill more confidence in EV drivers and to increase the interest of the general public in EVs. From a psychological standpoint, a reliable trip plan with accurate range estimation provided to drivers may be more significant than actually increasing the range of an electric mobility system [34]. Realizing such motivation, researchers propose various new methods and intelligent strategies. For example, due to the limited battery capacity, the driver was assisted by providing a locality-based optimal charging station planning. In [32], the number of charging stations was estimated using a weighted Voronoi diagram compliant with a conventional standard charging station capacity. An optimum charging method was proposed in [84], which includes an individual charging plan for each vehicle while reducing the electricity cost, avoiding distribution grid congestion, and fulfilling the individual vehicle owner's necessities.

With the intention of increasing the EV energy efficiency and range autonomy, Demestichas et al. [25] introduce an autonomous route planning method based on advanced machine learning techniques which monitors the energy consumption continuously. An EV suitability ADAS was designed for traffic estimation and selection of optimal route in [25, 26] suitable for EVs was designed and implemented with traffic estimation and optimal route selection capabilities. It helps the driver to make an appropriate routing decision so as to energy saving and residual range enhancement.

In order to find the best route, a fleet management planning for EV was proposed in [63]. It considers energy consumption based on the road architecture, environmental conditions, vehicle characteristics, driving behavior, traffic situations, and locations of electric charging stations. An advanced EV fleet management architecture was presented in [63], where the best route was decided based on electric power consumption considering information about the road topology (elevation variations, source, destination, etc.), weather conditions, vehicle characteristics, driver profile, traffic conditions, and electric charging station locations. A generalized multi-commodity network flow (GMCNF) model analogous to space exploration logistics [50] was proposed in [17] for finding the best route on the basis of energy consumption and battery charging time.

However, from the previous studies in the literature, it was found that majority of driving assistance systems deal with energy consumption. In most of the systems, energy consumption and trip time are treated separately, and a correlation between them was tried to establish [27]. An energy-driven and context-aware route planning framework for EV is found in [95]. An optimal route is decided here by minimizing the cost function which comprises the trip time and energy consumption. However, there is no such driving assisting systems that consider energy consumption, trip time, and driving comfort simultaneously. In [28], the authors made an attempt to find a driving strategy for ICE vehicles by simultaneous consideration of total travel time, and fuel consumption, and driving discomfort. Indeed, consideration of these three factors is important. The present chapter explores the need for and the requirements of

considering these three aspects simultaneously for optimal trip planning. It describes the related problem and a possible way of implementing in practical application for an EV. A novel framework, Driving Assistance for Optimal Trip Planning (DAOTP), is proposed to guide the driver for trip planning and to drive the EV in an efficient manner. The framework of DAOTP is context-aware as well as energy-efficient since it incorporates real-time traffic and route data, and accounts for multiple driving aspects including trip time, comfort driving, and energy efficiency.

The present DAOTP system first finds a set of Pareto-optimal solutions (ODSs) to the driver after solving a multi-objective optimization problem based on the current route and environmental data. The trip end battery SOC and total trip time corresponding to those optimal driving strategies for a specified trip length are also shown.

A preferred ODS is selected from them corresponding to a higher level information or using problem-specific multi-criterion aspects for implementation.

2 Proposed System of Driving Assistance for Optimal Trip Planning (DAOTP) for an EV

An EV driving assistance system is proposed here. The system presents a number of optimal driving strategies to the driver along with the corresponding driving range, trip end SOC, and total trip time [52]. It works based on a multi-objective concept, where multiple conflicting objectives are considered simultaneously to fix the optimal decision variable(s). Figure 1 demonstrates the architecture of DAOTP system. The system consists of two modules namely HMI (Human–machine interface) and DAOTP. HMI is used to interface between the driver and the DAOTP system with the help of GPS/Internet or other media.

DAOTP assists the driver by endowing with the following information regarding ODS at the time of executing and prior to a trip.

1. Reference acceleration(s)/deceleration(s)
2. Reference trip speed
3. Predicted range corresponding to the present battery SOC
4. Predicted Battery SOC and time at trip completion

The EV trip has three modes: acceleration, constant speed, and deceleration. DAOTP system deals with the first two zones. Since during deceleration, there is a gain of energy through the regenerative system finding an optimal driving strategy for deceleration is not considered here. There is an availability of many studies [15, 16, 100] suggesting how to regenerate maximum energy during deceleration of an EV. Such strategies can be easily incorporated into DAOTP system. In the present study, the electric motor brakes the EV at a constant rate that has a weak dependence on the vehicle speed. In order to estimate trip time prior to the trip, it was required to pick a constant deceleration value for different speed zones.

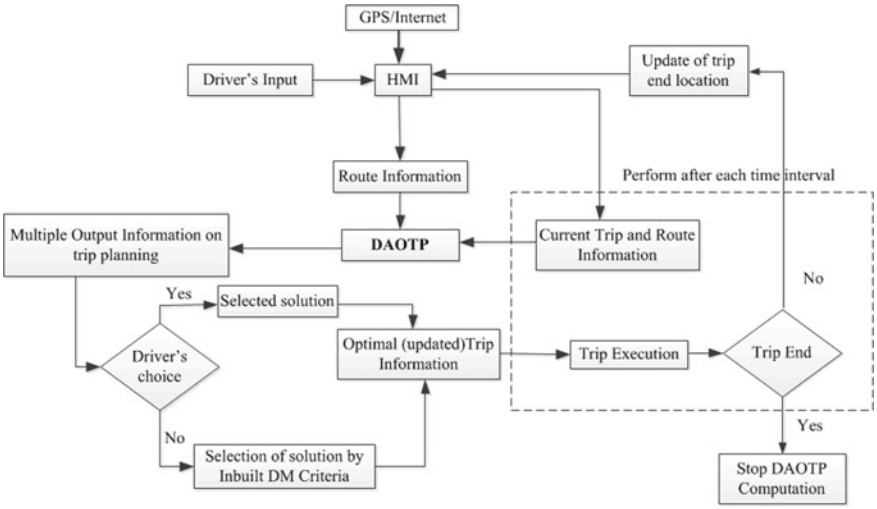


Fig. 1 Schematic layout of DAOTP system architecture

2.1 Working Methodology of DAOTP System

According to the driver's trip start and end locations, the HMI in association with GPS and Internet finds different routes if they exist. The HMI can be attached to the dashboard or mobile through a wireless link like a smartphone. Unless there is a specific choice by the driver, an optimum route is generally selected based on less number of stops, short route distance, good road conditions, less traffic at the time of journey, etc., and such information may be obtained based on past statistical data of route traffic and present traffic conditions. There are separate transportation studies (scheduling and routing problems) [10] that consider how to select an optimal route corresponding to multiple stops, time scheduling, multiple vehicles, etc. The present chapter does not include such optimization problems. The objective of the proposed DAOTP is different from that of scheduling and routing problems. Once a route is specified, DAOTP receives various route data through GPS and/or other resources using HMI. Additionally, the system receives the current battery SOC, EV auxiliary load, etc., with the help of appropriate measuring devices [73] and data acquisition system. Execution of DAOTP is carried out by solving the associated MOOP based on the received data for the chosen route. Once the DATOP execution is completed, a set of solutions (ODS) and corresponding various predicted information (such as range, trip end SOC, and trip time) is presented to the driver for making a suitable trip planning. The driver may choose an appropriate ODS from the solution set based on the present circumstances or his desire. Otherwise, any decision-making technique may be adapted to select a preferred one. The driver is recommended to follow the preferred ODS for driving to the further ODS as evaluated in the subsequent DAOTP execution. The subsequent execution of DAOTP is performed on the basis

of current route data corresponding to the original or updated trip start/end location (destination) and battery SOC.

Various higher level information may be considered by driver for making a decision to choose a preferred solution, such as remaining trip length, trip time, road condition(s), speed limit(s), traffic clogging, etc. Prior knowledge of the range and total trip time for multiple ODSs provides enough flexibility to the drivers to adopt a most suitable ODS that consumes minimum energy resulting in a better range. It is particularly beneficial in a circumstance of low battery change where the driver wants to get to a charging station. Therefore, the key motivation behind DAOTP development is to predict the EV range and trip time for optimal speeds.

During the running of the EV, DAOTP collects various data from different sources after a certain time step. It does so repeatedly considering that trip conditions may change quickly and unpredictably. Based on these updated data, DAOTP executes again and presents a new set of solutions for the rest of the trip. If there are no considerable changes, these solutions would be almost identical to that suggested in the previous time step. That new solutions assist the driver to drive the vehicle accordingly so that he/she can complete the entire trip in an efficient manner while accounting for continuously changing trip conditions. This online functionality of DAOTP accounts for any unanticipated driving circumstances and any variation from the current ODS.

Various inputs to be provided by the driver to DAOTP are as follows:

- Immediate destination.
- Either select a route after getting a feedback from GPS for different possible routes depending on his/her choice or allow the system to choose the route having a shorter distance, less traffic congestion, and fewer stops.
- Maximum allowable trip time.
- Information on auxiliary energy consumption.

  It may be noted that certain data, such as auxiliary energy consumption, are relayed directly to DAOTP and do not require active input from the driver. The driver may choose to increase or decrease the auxiliary energy consumption before DAOTP, like switching the air conditioning off, in order to see what effect it has on the solutions, and he/she may increase or decrease the auxiliary energy consumption during the trip.

  Various step-by-step roles of HMI through GPS/Internet in DAOTP are presented as follows:
- Identify the present location of EV, and then suggests different possible routes to reach the destination.
- Calculate the number of stops in the entire trip for the chosen route and other related route information such as trip distance, traffic congestion, weather condition (wind velocity and direction, rainfall or others), and road conditions (road gradient and quality).
- Based on route information, divide the entire trip into smaller “micro-trips”. Identify different parts of each micro-trip based on the upper and lower speed limits associated with each part. The upper and lower speed limits already take into con-

sideration the road type and profile (e.g., sharp bending, steep gradient, etc.). The magnitude of upper and lower speed limits can be used to define the driving cycle type.

For ease of understanding, the working method of DAOTP is described for a micro-trip. The range and trip time calculations on a trip are described in Sect. 5.

3 Micro-trip and Route Characteristics

A micro-trip may be defined as an excursion between two successive locations of travel route at which the vehicle is definitely stopped [3, 44]. In general, an entire trip comprises several micro-trips. The length (range) of a micro-trip is defined by the distance covered the two stop points of vehicle, and may be called the start and end points of that micro-trip, respectively. A typical micro-trip consists of the following modes or phases:

(i) *Initial acceleration phase*—acceleration(s) to change speed from $v_{\text{ref\_previous\_trip}}$ (end reference speed of the previous micro-trip) to $v_{\text{ref1\_micro - trip}}$ (reference speed of the current (first part) micro-trip) if $v_{\text{ref\_previous\_trip}} < v_{\text{ref\_micro - trip}}$.

(ii) *Constant speed change*—keep running with $v_{\text{ref\_micro - trip}}$, which must be within the speed limit.

(iii) *Speed limit change*—if there is a change in the speed limit during the micro-trip (in case of multiple parts, as shown in Fig. 2), then deceleration(s) to change speed from $v_{\text{ref1\_micro - trip}}$ (reference speed of the first micro-trip part) to $v_{\text{ref2\_micro - trip}}$ (reference speed of the second micro-trip part) before the vehicle reaches the new speed limit zone if $v_{\text{ref2\_micro - trip}} < v_{\text{ref1\_micro - trip}}$, and continue to drive vehicle at $v_{\text{ref2\_micro - trip}}$ during the speed limit zone or acceleration(s) to change speed from $v_{\text{ref1\_micro - trip}}$ to $v_{\text{ref2\_micro - trip}}$ after the vehicle reaches the new speed limit zone if $v_{\text{ref2\_micro - trip}} > v_{\text{ref1\_micro - trip}}$.

(iv) *Final deceleration phase*—deceleration from $v_{\text{ref2\_micro - trip}}$ (reference speed of the current (end part) micro-trip) to $v_{\text{ref\_next trip}}$ (reference speed at the starting of next micro-trip) if $v_{\text{ref\_micro - trip}} > v_{\text{ref\_next\_trip}}$.

According to safety and differentiation of the type of trip, micro-trip comprises one or multiple types of road. Based on the lower and upper speed limits, roads are classified into four different types [8, 74] namely neighborhood (8–40 km/h), urban (40–56 km/h), highway (56–88.5 km/h), and interstate (88.5–112.6 km/h). The minimum and maximum speed limits of each driving cycle need to be strictly followed by the driver. Thus, the proposed DAOTP system must provide valid solutions for each driving cycle. Accordingly, its performance for each driving cycle was studied independently.

Figure 2 demonstrates the velocity profile of a typical complex micro-trip that consists of three parts. The first and third parts of the micro-trip correspond to a low-speed driving cycle and the second part corresponds to a high-speed driving cycle.

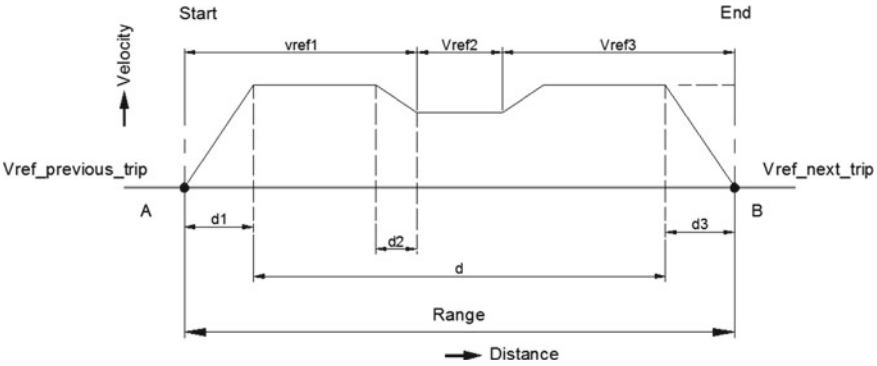


Fig. 2 Velocity versus distance plot of a typical micro-trip

The vehicle reference speeds, $v_{\text{ref1\_micro - trip}}$ and $v_{\text{ref3\_micro - trip}}$, belong to first and third parts of the micro-trip, respectively, whereas $v_{\text{ref2\_micro - trip}}$ is the reference speed associated with the part of the micro-trip. According to the definition of a micro-trip, $v_{\text{ref\_previous\_trip}}$ and $v_{\text{ref\_next trip}}$ in Fig. 2 are zero. According to the rule of traffic lights, at each light, the vehicle has to stop at a red signal light until the signal becomes green. Points A and B (in Fig. 2) are considered as two successive red signals. They could also represent a stop sign. Besides the driving strategy characteristics (DSC) such as speed, acceleration(s), and the respective durations, other environmental and physical factors (information that may be gleaned through various HMI inputs) associated with micro-trip route are the route characteristic parameters (such as road gradient, quality of road surface, air density, wind velocity and its direction, etc.), passenger weight, auxiliary load, start and end location of the micro-trip, road speed limits, initial SOC, etc. The route characteristic parameters are normally defined based on the distance covered and can be obtained through GPS or Internet sources. The other parameters are case-based depending on the driver's desire. For example, a road gradient varies with road length. To determine its value at a particular location on the road, one has to refer to how much distance is covered by the vehicle from a reference point, in general, the start point of micro-trip. Other parameters are dependent on the driver's desires. For example, the driver decides when the air conditioning is running and when it is not.

4 Control System

The control system is the interconnection of components that gives the desired output. Driving a car with the desired speed is the example of closed-loop control system. Here the system (presented in Fig. 3) compares the speed of the car with the desired speed. If any deviation in speed from desired speed, then the controller may increase or decrease the speed so that the deviation becomes zero. The sensor is used to

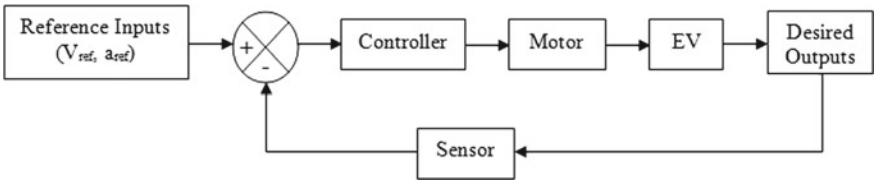


Fig. 3 Basic block diagram of EV speed control system

measure the controlled variables of the system and fed to the input. Then the control system compares the reference inputs (speed/acceleration) with a present output, which is fed by the sensor from the output.

The PI (proportional and integral) controller is used to control the EV acceleration and also maintains a constant speed of EV. The PI controller works during acceleration of EV for changing the speed and when the desired speed is reached, the integral action is switched off. Then only proportional action is activated and maintains a constant speed. The reference speed (v_{ref}) and the reference acceleration (a_{ref}) are used as the input of the controller. The output speed (v) and acceleration (a) are fed back to the input of controller in opposite phase to get desired output and stabilize the system.

5 Range and Trip Time Calculation

As mentioned previously, knowing the range, trip time, and final SOC value is essential for the driver to properly plan a trip. Existing methods for range calculation are summarized as follows. Battery State-of-Charge methods primarily concentrate on precisely calculating the battery SOC (which is similar to the fuel gage on a ICE vehicle) so as to find an estimation of how much battery charge remains. The range to be completed by the residual battery energy can be approximately determined based on the previous knowledge of what distance was covered by depleting how much battery charge. Various studies on battery SOC method based range estimation [6, 14, 29, 46, 80, 83, 85, 101] are found in the literature. This information, while important, is inadequate by itself since the battery's residual energy can be utilized in many different ways according to the driver's preferences. However, it should be optimally utilized to accomplish the driver's purpose. No such appropriate driving strategy is available to the driver to properly utilize the residual battery energy. In addition, since road conditions may vary in course of trip performing, the existing methods are unable to capture such changeable effects as they are intrinsically averaging methods. Energy-based methods calculate the energy utilization based on the current or recent trip and vehicle data Based on the amount of energy consumption rate, the EV range is predicted for the remaining battery charge. An approach to predict the residual driving range and driving time of ICE vehicles using Artificial

Neural Network (ANN) was found in [20]. In that approach, fuel capacity (remaining fuel), engine speed, vehicle velocity and weight, and road slope were considered as the inputs. Though this approach offers valuable information in the course of trip performing based on instantaneous driving parameters, the anticipated range is not known to the driver prior to trip started. Moreover, that approach does not suggest any optimum trip parameters. In [79], an ANN based technique was proposed to calculate the energy utilization for both pure and plug-in hybrid EVs in real-world driving situation. This technique considered various inputs such as average speed, average acceleration, total distance traveled, total duration, etc., to envisage the road category and traffic congestion and, finally the predicted EV energy consumption rate was found to deviate from the measured value y in the range of 20–30 to 70–80%. The authors recommend that as the energy consumption rate and total available energy are known, an accurate EV range prediction can be made using this technique. Once more, the driver does not have knowledge of the anticipated range for the specified conditions stated in the article to devise a driving strategy. An approach to envisage the power for the instant future requirement in EV was proposed in [54]. This approach considers the data from previous power consumption history, acceleration and speed, and road information obtained from a pre-downloaded map. Though the primary motivation of this study was the safety of battery, value of immediate power requirement could be used to calculate the range as well. But this approach is a passive method as the power requirement prediction depends on the driver's actions, and it does not suggest how to formulate a driving strategy.

There are three driving modes in which a vehicle can travel a distance. Two are during the vehicle's acceleration and deceleration and the other is when the vehicle is running at a constant speed. The methodology of finding the estimated distance traveled by a vehicle in these driving modes is presented in the following.

The range (distance) that can be traveled by an EV during constant speed mode is

$$R_g = v_{ref} \times T_v \tag{1}$$

where v_{ref} is the (uniform) speed at which the vehicle is moving and T_v is the corresponding duration. But, in practice, it is very difficult to maintain the vehicle's real speed exactly at v_{ref}, always. Thus v_{ref} is also considered as the commanded/desired speed. Once the accelerator is pressed and its position maintained, it takes some time (acceleration period) for the real vehicle speed, v, to get to v_{ref}. Thus, the instantaneous speed error is calculated as

$$e_v = v_{ref} - v \tag{2}$$

In steady state, v_{ref} and v are close enough to be considered equal from a vehicular standpoint. These two parameters were inputs to the speed controller. Now to cover an R_g with less stored battery energy consumption per unit time, the value v_{ref} becomes low. But, it results in a high T to obtain the same range. On the other hand, a decrease of T means consuming more energy. Thus to find an optimal v_{ref} to cover R_g efficiently,

two contradictory objectives namely minimization of total energy consumption and driving time are required to be solved simultaneously [89].

On the other hand, the range that can be covered by an EV during acceleration is

$$R_g = v_{init} T_a + \frac{1}{2} a_{ref} \times T_a^2 \tag{3}$$

where a_{ref} is the (uniform) acceleration (also called commanded or desired acceleration) at which the vehicle speed increasing to v_{ref} and T_a is the driving time during which a_{ref} can be plausibly retained by the battery. The instantaneous acceleration error is defined as the difference between the commanded acceleration and the real vehicle acceleration, a

$$f = a_{ref} - a \tag{4}$$

Both a_{ref} and a were inputs to the acceleration controller. Like optimal velocity, to find optimal acceleration(s), both the acceleration time and energy consumed for the acceleration are required to be minimized concurrently [68], and after solving such MOOP, it was revealed that a better optimization result is noticed using multiple accelerations compared to a single acceleration. Such analogous results were also observed in [49, 62]. Besides the above objectives, it is necessary to have sufficient comfort during driving. A discomfort journey may result in various health problems [42].

The range covered by an EV during deceleration is

$$R_g = v_{ref} T_d - \frac{1}{2} a_{dec} \times T_d^2. \tag{5}$$

where T_d is the duration for v_{ref} to reach a termination value (normally a low value, say 1 m/s). The EV deceleration was controlled by the speed controller rather than the acceleration controller.

In practice, a trip consists of several such driving modes. The total range (R) overcome by an EV in a typical trip is the summation of all R_g in these three driving modes and the total trip time is the summation of all corresponding T. In order to maximize the EV range and minimize the trip time by means of efficiently using the stored battery energy, the primary concern is to identify the optimal values of v_{ref}, a_{ref}, and a_{dec}. Various extraneous driving-specific parameters that affect v_{ref}, a_{ref}, and a_{dec} can be categorized into three groups: dynamic parameters, static parameters, and navigation control parameters [106]. For the sake of simplicity and reducing the model complexity, inherent model-specific parameters such as driveline dynamics, etc., were not considered, whereas route specific parameters that tend to change during a trip (gradient, elevation, wind, and road surface) were considered in this study. Neglecting these latter parameters can lead to large error in the results due to their more significant contribution to the overall energy consumption [40, 59, 97, 103].

6 Multi-objective Optimization

Optimization is one of the most frequent and persistent problems in real-world systems including engineering. Through this technique, one can reach to an extreme solution corresponding to either maximum or minimum of any objective subject to certain constraints. Contrary to single objective optimization, MOO plays an important role in decision-making toward making a preference among several options related to multiple contradictory objectives. Multiple contradictory objectives are normally found in most of the real-world problems. If all objectives are equally important, extreme value principle (as used in single objective optimization) cannot be adopted to arrive a solution. In such case, a number of solutions may be created based on a negotiation among the objectives. Such negotiation does not allow to consider a solution that is optimal corresponding to simply one objective. Thus, it is required to obtain a set of solutions, among them the designer is allowed to choose one that will fulfill the original intention. Selection of a solution among multiple availabilities is also known as multiple criterion decision-making (MCDM). Thus, the primary motivation for solving truly MOOPs is to find a set of non-dominated solutions. The front formed by the optimal non-dominated solutions is called Pareto-optimal front. The Pareto-front is formed by the solutions in which any change in any of the decision variables aimed at improving a particular performance index will produce deterioration in some of the other performance indices.

In general, a multi-objective problem (MOP) comprises of n number of input parameters (called decision variables, $x_{1,\ldots,n}$), k number of objective functions ($y_{1,\ldots,k}$), m inequality constraints, and j equality constraint. Objective functions and constraints are functions of the decision variables. The optimization goal is as follows:

$$\text{Maximize/minimize} \quad y = f\left(x_{1,\ldots,n}\right) = \left(f_2\left(x_{1,\ldots,n}\right), f_1\left(x_{1,\ldots,n}\right), \ldots, f_k\left(x_{1,\ldots,n}\right)\right) \tag{6}$$

$$\begin{aligned} \text{subject to} \quad & e\left(x_{1,\ldots,n}\right) = \left(e_1\left(x_{1,\ldots,n}\right), e_1\left(x_{1,\ldots,n}\right), \ldots, e_m\left(x_{1,\ldots,n}\right)\right) \leq 0, \\ & h\left(x_{1,\ldots,n}\right) = \left(h_1\left(x_{1,\ldots,n}\right), h_1\left(x_{1,\ldots,n}\right), \ldots, h_j\left(x_{1,\ldots,n}\right)\right) = 0, \end{aligned}$$

$$\begin{aligned} \text{where} \quad & x_1, x_2, \ldots, x_n \in X \\ & y = (y_1, y_2, \ldots, y_k) \in Y \\ & x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \ldots, n \end{aligned}$$

The decision space and objective space are denoted by X and Y, respectively. x_i^L and x_i^U are the lower and upper bounds of each decision variable, x_i which form X. The feasible solutions must satisfy the variable's upper and lower limits and the constraints, $e(x)$ and $h(x)$.

Even though it is sufficient to know the objective ranges and Pareto-optimal shape in order to make a decision, it is essential to select a distinct preferred solution since it will be ultimately used for execution in a driving condition, and the corresponding task is very crucial. According to the categorization made by Veldhuizen and Lamont [92], the articulation of preferences may be carried out either before (a priori) or after (a posteriori) or during (progressive) the optimization search. In this work, two a posteriori-based MCDM techniques were adopted. Reference point based technique [64] identifies a preferred solution based on a reference point. Whereas in the second technique, first a knee-zone [11] on the Pareto-front is identified on the basis of trade-off among the objectives, then based on the higher level information, a preferred solution is selected from the knee-zone.

Various characteristics relating to the problem domain may be considered to decide the reference point. Ideal point is widely used as a reference point in many multi-objective problems. The set of optimal values obtained through optimization process considering each objective independently is referred to the ideal point which is considered here as a reference point in decision-making process. In the present work, a preferred solution among the Pareto-optimal set is identified on the basis of the minimum (Euclidean) distance from the reference point. Sometimes it is noticed that there is a typical portion of the Pareto-optimal front where a small improvement in one objective would lead to a large deterioration in any of the other objectives is treated as the knee-zone. The knee-value of the ith solution in the knee-zone is mathematically defined by Eq. (7) for a MOOP having two conflicting objectives.

$$\kappa_i = \frac{\frac{f_1^{(i-1)} - f_1^{(i)}}{f_2^{(i-1)} - f_2^{(i)}} + \frac{f_2^{(i+1)} - f_2^{(i)}}{f_1^{(i+1)} - f_1^{(i)}}}{2} \tag{7}$$

where the objectives f_1 and f_2 are considered to be maximized and minimized, respectively.

A solution is said to a stronger knee point if its knee-value is higher than that of the others and vice versa. It is obvious that the knee-zone is most likely to be interesting to the decision maker exclusive of any user's preferences knowledge.

6.1 Non-dominated Sorting Multi-objective Genetic Algorithm (NSGA-II)

Nowadays interest in using of EAs in solving MOOP is increasing by realizing that classical methods possess number drawbacks. Some classical methods are unable to generate the Pareto-front with all possible solutions, particularly in nonconvex-type problems. Sometimes in-depth problem information is needed which is difficult to acquire. Moreover longer computational time for repeated simulation run to identify the Pareto-optimal solution independently is an inherent difficulty of classical methods.

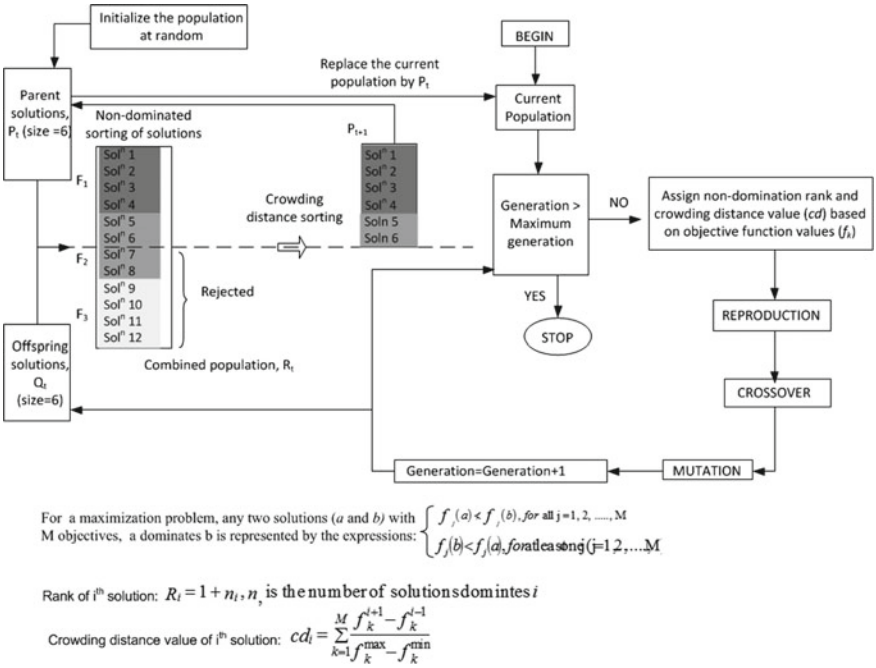


Fig. 4 A schematic presentation of NSGA-II [24]

In contrary, Evolutionary methods are population-based algorithms and have been successfully implemented in many real-world problems having intrinsic complexities in calculating the analytical Pareto-optimal fronts. Earlier efforts of EA implementation in the field of present application [12, 39, 72] confirm its efficacy. In various other real-world problems, EAs are also found to be successfully implemented [22, 28–32, 65, 67, 71, 78]. By realizing that, in the present work, a well-established evolutionary algorithm (EA), NSGA-II (a non-dominated sorting multi-objective genetic algorithm) [24] is considered and its working principle is presented as follows.

Figure 4 demonstrates the working principle of NSGA-II considering binary-coded genetic algorithm (GA). The present demonstration assumes a population size equals six. Present NSGA-II architecture considers crowded tournament selection, polynomial mutation scheme, and simulated binary crossover (SBX). After each GA-generation, six new non-dominated solutions are created. A solution treated as the winner in the tournament should have either lowest rank or larger crowding distance in case of multiple solutions of the same rank.

A generation is started by creating an offspring population Q_t based on the parent population P_t. After that, a collective population, R_t of size 2N is created by combing P_t and Q_t. R_t is then allowed for non-dominated sorting for solution classification based on their ranks. A front is formed by the solutions having the same rank. Figure 4 demonstrates such kind of three fronts (F_1, F_2, and F_3). As the constant population

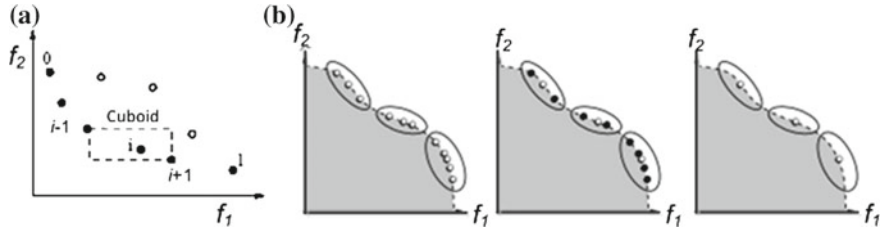


Fig. 5 A schematic representation of **a** crowding distance sorting approach [24], **b** clustering approach [105]

size (= 5) is required to be maintained, forming of the new population (P_{t+1}) is carried out by picking solutions from the different fronts one-by-one starting with the lower rank. The fronts from which no solution is picked are just removed. It may happen that there is a possibility of having more solutions in the final front than the number of solutions just required to reach the population size (= 5) of P_{t+1}. Such a condition happens with F_2 as shown in Fig. 4. In that situation, a niche-preserving strategy (namely crowding distance sorting and clustering approach) is applied to select the required number of solutions from the last front. Considering P_{t+1} as the parent population, P_t, next Q_t was created using the genetic operators, and the same iteration (generation) is allowed to continue until the generation number reaches to a designer specified value.

The crowding distance of an individual is calculated by determining the cuboid length which is equivalent to the summation of the distances of two neighboring solutions from the individual in each objective as demonstrated in Fig. 5a. The mathematical expression to calculate crowding distance (cd_i) is presented in Fig. 4, where M is the number of objectives ($f_{i,\ldots,M}$). Preference of an individual is made with higher crowding distance value.

A schematic representation of the clustering technique is presented in Fig. 5b. In this technique, initially, each solution of the front is treated as the center of the individual cluster. Then two clusters whose distance (which is the Euclidean distance between their centroid) is minimum among that of all cluster pairs are merged to form a single larger cluster, and thereby reducing a cluster from the previous ones. The same procedure of merging two clusters is continued until the number of clusters reaches to the desired number of solutions. For a larger cluster, the solution closest to its centroid is considered and all others are removed.

7 Formulation of Multi-objective Problem for Optimal Trip Planning

Different issues that come to driver's mind while trip planning using EV are mentioned in Sect. 1. In order to know whether EV can reach the destination, an accu-

rate range prediction is necessary. Without considering other reasons (such as EV mechanical structure, road condition, weather, etc.), jerk is responsible for a comfortable journey, and it is required to be minimum during the course of the journey. To reduce the trip time, speed and rates acceleration/deceleration are to be kept high. Reduction of journey cost is achieved by minimizing the overall energy consumption.

Previous studies [9, 18, 55] reveal that deriving harshness have a significant effect on fuel consumptions. In [33], the authors suggest that appropriate changing of driving behavior can reduce the energy consumption considerably. In another study [37], it was suggested that improvement of regenerative braking energy can also be possible by adopting an appropriate deceleration rate. Moreover, the deceleration rate also influences the regenerative braking energy [37]. In some recent works [48, 51, 62, 89], it was revealed that energy requirement during speed changes with constant acceleration rate is found to be more compared to that with multiple acceleration rates. But, the use of multiple acceleration rates increases the jerk that leads to the discomfort [28].

From the above discussions, it is understood that speed, acceleration/deceleration rates, and their durations are the controllable parameters involved to design a trip planning. Whereas, the output parameters (objectives) which are depended on the controllable parameters are cost (energy consumption), range (distance traveled), trip time, and journey comfort. Various works were carried out considering the minimization of trip time as the main objective [60]. A method for prediction of short-term travel duration was proposed based on sensor data from the road in [104]. Moreover, an appropriate driving strategy which highly depends on the road condition is also imperative and it is realized from various studies [9, 18, 33, 55, 88]. Moreover, in order to properly utilize the EV battery's stored energy, it is important that the formulated driving strategy negotiates the predicted range in an optimal manner.

Thus, the driving notion of the driver would be to accelerate the vehicle to reach a speed in the shortest time with adequate comfort while expending the minimum amount of energy possible. In other sense, it indicates that driver would like to accelerate the EV comfortably to a chosen speed with both minimum energy and minimum time. But, these objectives are contradictory to each other, meaning that the improvement of one objective deteriorates the other and vice versa [27, 28].

7.1 Problem Definition

The corresponding MOOP that arises is defined as follows:

$$\text{Minimization of trip time } \left(T_{Trip}\right) = f_T\left(v_{ref}, a_{ref}, k_p, k_i\right) \tag{8}$$

$$\text{Minimization of Energy consumsion } (E) = f_B\left(v_{ref}, a_{ref}, k_p, k_i\right) \tag{9}$$

$$\text{Minimization of average jerk } \left(J_{Avg}\right) = f_{J_{Total}}\left(a_{ref}^{1,2,\ldots,K}, t_{1,2,3,\ldots K-1}, k_p, k_i\right) \tag{10}$$

subject to

$$v_{\min}^{dc} \leq v_{ref} \leq v_{\max}^{dc} \tag{11}$$

$$a_{\min} \leq a_{ref}^{1,2,\ldots,K} \leq a_{\max} \tag{12}$$

$$0.01 \leq k_{pa} \leq 0.3 \tag{13}$$

$$0.01 \leq k_{ia} \leq 3.0 \tag{14}$$

$$0.01 \leq k_{pv} \leq 0.3 \tag{15}$$

$v_{\min}^{dc}$ and $v_{\min}^{dc}$ are the minimum and maximum EV speed (v) limits, respectively. Speed limits depend on the driving cycle (dc) that is currently followed [74] and various safety issues concerning traffic congestion, road condition, weather, location importance such as school, hospital, etc., and so on [2, 56, 66]. $a_{\min}$ and $a_{\max}$ are the minimum and maximum allowable EV accelerations, a related to the EV specification, traffic congestion, road condition, weather, etc. Depending on the EV model type considered, $a_{\min}$ and $a_{\max}$ are taken as 0.1 and 3.0 m/s^2, respectively. k_{pa} and k_{ia} are the proportional and integral gains, respectively, of acceleration controller. k_{pa} was considered to vary in the range 0.01–0.3 and k_{fa} is in the range 0.01, 3.0. The lower and upper values of k_{pv}, velocity controller proportional gain are taken as 0.01 and 0.3, respectively.

In this chapter, the present optimization problem is solved using NSGA-II considering crowding distance approach as a niching strategy discussed in Sect. 6 in finding optimal driving strategies for EV. Since the present application is a constrained optimization problem, the non-dominated solutions are identified based on the superiority approach of the feasible individuals [23].

8 Formulation of Objectives

The proposed DAOTP system is formulated here considering EV model topology introduced in [36, 58, 100]. For the sake of reducing complexity, a simplified model was used. The EV model comprises three major related components: electric motor, battery, and the vehicle dynamics. The acceleration and speed of electric motor were controlled using a proportional–integral (PI) controller and a proportional controller, respectively. The EV model takes the inputs, reference acceleration, a_{ref}, and speed, v_{ref}. The model outputs the objective values after performing several iterations with a certain time step in a simulation process. The simulation was terminated when the reference speed, v_{ref}, was zero indicating the completion of trip, corresponding to n loop iterations. Various model parameters considered for the present study are enlisted Table 1.

Table 1 EV model parameters

Type	DC brushed
Electric motor	
Motor moment of inertia coefficient, I	0.05
Copper losses, kc	0.3
Iron losses, ki	0.01
Windage losses, kw	0.000005
constant electronics losses, $Conl$	600
Proportional controller gain for speed, KP	2.0
Critical motor speed, w_c (rpm)	733
Maximum motor speed, $w_{\max}$ (rpm)	1326
Battery	
Capacity, Cap (A h)	53
Initial state-of-charge, SOC_{init}	1.0
Number of cells in parallel, NP	1
Number of cells in series, NS	76
Type	Lithium-Ion
Voltage, V_P (V)	394
Battery efficiency, $bateff$	0.99
Battery long transient capacitance, $C_{Transient\_L}$ (MF)	0.22375
Battery long transient resistance, $R_{Transient\_L}$ (mΩ)	0.9968
Battery short transient capacitance, $C_{Transient\_S}$ (MF)	0.03518
Battery short transient resistance, $R_{Transient\_S}$ (mΩ)	0.9338
Battery series resistance, R_{Series} (mΩ)	1.4932
Vehicle	
Air density, ρ_{air} (kgm^3)	1.143
Frontal Drag coefficient, C_D	0.19
Back Drag coefficient, C_{D_back}	0.3
Frontal area, A_f (m^2)	1.8
Back area, A_b (m^2)	1.4
Gravitational acceleration, g (ms^2)	9.81
Mass including passengers and drivers, m (kg)	1460
Overall gear ratio/tire radius, G (m^{-1})	37
Rolling resistance coefficient, μ	0.014
Transmission	Single-speed
transmission efficiency, $geff$	0.95
Regenerative braking factor, $Rgen$	0.5

8.1 Electric Motor Model

The motor model's inputs Various parameters such as battery voltage, V_P, reference acceleration, a_{ref}, vehicle real acceleration, a, reference speed, v_{ref}, vehicle real speed, v, and rotational speed, w, are considered as the inputs to the motor model. The outputs are the battery current, I_P, and electric motor torque, τ.

The speed error defined in Eq. (2) is rewritten for the ith loop iteration as

$$e(i) = v_{ref} - v(i) \tag{16}$$

The acceleration error defined in Eq. (4) is rewritten for the ith loop as

$$f(i) = a_{ref} - a(i) \tag{17}$$

The switching function, SF of the motor is determined as follows:

$$\begin{aligned} SF(i) &= k_p f(i) + k_i\, f_{\text{int}}(i), \ \text{if e(i)} < f_{sa} \cdot v_{ref} \\ SF(i) &= KP \cdot e(i) \text{ and } f_{\text{int}}(i) = 0, \quad \text{otherwise} \end{aligned} \tag{18}$$

where f_{sa}, switching factor from speed to acceleration controller was taken to be 0.02. $f_{\text{int}}(i)$ is the integral of acceleration error which is defined by $f_{\text{int}}(i-1) + f(i) * dt$ and $f_{\text{int}}(0) = 0$.

The switching function value found using Eq. (18) is saturated as follows to calculate the motor torque.

$$SF(i) = \left.\begin{matrix} 1 \\ -1 \\ SF(i) \end{matrix}\right\} \begin{matrix} \text{if } SF(i) > 1 \\ \text{if } SF(i) < -1 \\ \text{otherwise} \end{matrix} \tag{19}$$

The torque needed from the motor is

$$\tau(i) = SF(i) \cdot \tau_{\max}(i) \tag{20}$$

where τ_{max} is the maximum torque that can be safely developed by the motor and its value is decided according to Eq. (21).

$$\left.\begin{matrix} \tau_{\max}(i) = 140 \\ \tau_{\max}(i) = 9.1274 \\ \tau_{\max}(i) = \frac{102000}{w(i-1)} \end{matrix}\right\} \text{if} \begin{matrix} w(i-1) \le w_c \\ w(i-1) \ge w_{\max} \\ w_c < w(i-1) < w_{\max} \end{matrix} \tag{21}$$

The units are N-m. The motor current, $I_M(i)$ is

$$I_M(i) = \frac{\tau(i) \cdot w(i)}{V_P(i-1)} \tag{22}$$

Now, the motor efficiency, *effmotor*, is calculated by Eq. (23).

$$effmotor\,(i) = \frac{\tau(i) \cdot w(i)}{\tau(i) \cdot w(i) + \tau(i)^2 \cdot kc + w(i) \cdot ki + w(i)^3 \cdot kw + Conl} \tag{23}$$

The above-calculated motor efficiency was saturated as follows:

$$effmotor(\mathrm{i}) = \left.\begin{array}{c} 1 \\ -1 \\ effmotor(\mathrm{i}) \end{array}\right\} \begin{array}{c} \text{if } effmotor(\mathrm{i}) > 1 \\ \text{if } effmotor(\mathrm{i}) < -1 \\ \text{otherwise} \end{array} \tag{24}$$

The effective motor current $I_{M\_eff}(i)$ is derived from the motor current, $I_M(i)$ and considering the *effmotor* and *conveff* (converter efficiency), as follows:

$$\left.\begin{array}{l} I_{M\_eff}(i) = \frac{I_M(i)}{conveff \cdot effmotor(i-1)} \\ I_{M\_eff}(i) = I_M(i) \cdot conveff \cdot effmotor(i-1) \cdot Rgen \end{array}\right\} \text{if } \begin{array}{c} SF(i) > 0 \text{ and } v_{ref} \neq 0 \\ SF(i) < 0 \end{array} \tag{25}$$

The regenerative braking factor, *Rgen*, is the fraction of the total available regenerative energy that is converted to battery energy.

8.2 Battery Model

The lithium-ion battery model presented in [19] is adopted here. Battery current, I_P, is the input of battery model. The model's outputs are V_P, and battery state-of-charge, *SOC*.

Considering the maximum and minimum limits of battery current, the battery current to be calculated using the revised motor current is as follows:

$$I_P(i) = \left.\begin{array}{c} I_{M\_\max} \\ I_{M\_\min} \\ I_{M\_eff}(\mathrm{i}) \end{array}\right\} \begin{array}{c} \text{if } I_{M\_eff}(i) > I_{M\_\max} \\ \text{if } I_{M\_eff}(i) < I_{M\_\min} \\ \text{otherwise} \end{array} \tag{26}$$

Here, $I_{M\_\max}$ and $I_{M\_\min}$ are considered as 400 and −400 A. Considering the battery efficiency and mode of EV speed change (acceleration/deceleration), the effective battery current, $I_{P\_eff}(i)$ can be derived from battery current, $I_P(i)$ defined in Eq. (26) as follows:

$$\left.\begin{array}{l} I_{P\_eff}(i) = \frac{I_P(i)}{bateff} \\ I_{P\_eff}(i) = I_P(i) \cdot bateff \end{array}\right\} \text{if } \begin{array}{c} SF(i) > 0 \\ SF(i) \leq 0 \end{array} \tag{27}$$

The current flowing through an individual cell is

$$I_{cell}(i) = \frac{I_{P\_eff}(i)}{N_P} \tag{28}$$

The battery SOC is calculated as

$$SOC(i) = SOC(i-1) - \int \frac{I_{cell}(i)}{Cap}\, dt \tag{29}$$

The initial value of $SOC(i)$ is the so-called initial state of charge of the battery, SOC_{init}. The battery voltage is

$$V_P(i) = N_S V_{cell}(i) \tag{30}$$

where V_{cell} is the voltage of an individual cell. Based on the Kirchhoff's Current Law, the voltage of $C_{Transient\_L}$ is calculated as

$$\frac{d}{dt}\big(V_{C\_L}(i)\big) = \frac{I_{cell}(i)}{C_{Transient\_L}} - \frac{V_{C\_L}(i)}{C_{Transient\_L} R_{Transient\_L}} \tag{31}$$

and the voltage of $C_{Transient\_S}$ is

$$\frac{d}{dt}\big(V_{C\_S}(i)\big) = \frac{I_{cell}(i)}{C_{Transient\_S}} - \frac{V_{C\_S}(i)}{C_{Transient\_S} R_{Transient\_S}} \tag{32}$$

The open-circuit voltage is calculated as

$$V_{OC}(i) = -1.031e^{(-35SOC(i))} + 3.685 + 0.2156SOC(i) - 0.1178SOC(i)^2 + 0.3201SOC(i)^3 \tag{33}$$

The voltage of an individual cell is determined as

$$V_{cell}(i) = V_{OC}(i) - V_{C\_S}(i) - V_{C\_L}(i) - I_{cell}(i)R_{Series} \tag{34}$$

8.3 Vehicle Dynamics Model

The electric motor torque, τ, the model input, and the model's outputs are a, v, and R, and the rotational speed, w. The aerodynamic drag force acting on the EV is

$$F_D(i) = \frac{1}{2}\rho_{air} A_f C_D v(i)^2 \tag{35}$$

The frictional force between the road and wheel is

$$F_{rr}(i) = \mu mg \tag{36}$$

The traction force supplied by the motor is

$$F_t(i) = \tau(i)G \cdot geff \tag{37}$$

Force caused by road gradient is

$$F_{gra}(i) = \sin\left(\frac{grad \cdot \Pi}{180}\right) \cdot mg \tag{38}$$

Force caused by the vehicle inertia is

$$F_{inertia}(i) = I \cdot m \cdot a(i) \tag{39}$$

Considering the wind velocity and its direction (with respect to vehicle movement) effect into the drag force [40], and assuming all the braking force come from the electric motor, the acceleration is

$$a(i) = \frac{\tau(i) \cdot G \cdot geff + \frac{1}{2}\rho_{air} A_b C_{D\_back} \left(v(i) - v_{wind \cdot \cos(\theta_{wind})}\right)^2 - \mu mg - \sin\left(\frac{grad \cdot \Pi}{180}\right) \cdot mg - I \cdot m \cdot a(i)}{m},$$
$$\text{if } v_{wind} > v(i) \text{ and } 270° < \theta_{\text{wind}} < 90° \tag{40}$$

$$a(i) = \frac{\tau(i) \cdot G \cdot geff - \frac{1}{2}\rho_{air} A_f C_D \left(v(i) - v_{wind \cdot \cos(\theta_{wind})}\right)^2 - \mu mg - \sin\left(\frac{grad \cdot \Pi}{180}\right) \cdot mg - I \cdot m \cdot a(i)}{m},$$
$$Otherwise \tag{41}$$

where θ_{wind} (in degree) ($0° < \theta_{\text{wind}} < 360°$) is the wind velocity angle measured with respect to the EV movement direction, grad (in degree) is the road gradient (it becomes negative if the road is downhill, and positive if the road is uphill), ρ_{air} is the air density at the EV's current location.

The EV speed can be calculated using.

$$v(i) = v_{i-1} + \int a(i)\, dt \tag{42}$$

ω is given by

$$\omega(i) = v(i)G \tag{43}$$

The distance traveled by EV, $x(i)$ is given by

$$R(i) = R(i-1) + v(i)\, dt \tag{44}$$

The objective functions are defined as follows:

$$T_{Trip} = n \cdot dt \tag{45}$$

$$E = \sum_{i=1}^{n} I_P(i) V_P(i) \tag{46}$$

$$j_i = \frac{a(i+1) - a(i)}{dt} \tag{47}$$

$$J_{Avg} = \frac{\sum_{i=1}^{n_d} j_i}{n_d} \tag{48}$$

where j_i is the jerk experienced for a time step, dt due to change of acceleration. J_{Avg} is the average jerk value calculated based on jerks (j_i) that are found to be greater than the desired value, J_{desire} (i.e., $j_i > j_{desire}$). This automatically excludes very low values of jerk due to controller or simulation conditions.

8.4 EV Simulation

The EV simulation is carried out starting from its present speed (v_{init}) with multiple accelerations for a time period (D_{acc}) to reach to the desired reference speed (v_{ref}). Once the speed reaches close to v_{ref} (say 99%), switching from acceleration controller to speed controller is made, and vice versa depending on the driving mode. The EV jerk was recorded at the end of each time interval. The EV energy consumption (E_{acc}), total jerk (J_{Total}), trip time, T_{Trip} and range (R_{a}) were calculated at the simulation end. By plotting the values of different objectives, a non-dominated front (i.e., Pareto front) was obtained. The termination criteria of the EV simulation was to satisfy any one of the following:

$$\left.\begin{aligned} &SOC \leq 0.2 \\ &x \geq Range_{Trip} - 10 \\ &v_{\text{End of trip}} \leq 5 \text{ kph} \end{aligned}\right\} \tag{49}$$

According to the Road Safety Authority (RSA), to stop or slow down a vehicle, braking should be applied while accounting for a minimum stopping distance from the stopping point or from the location of the start of a speed limit lower than the present one. The total stopping distance (TSD) is normally the summation of the driver's reaction distance and the braking distance, and it also depends on the dryness or wetness of the road surface [31]. TSD increases exponentially with the current vehicle speed. For the sake of simplicity, a linear relationship is considered here. In the present context, when the residual trip distance is less than a minimum TSD (Trip End Safety Distance for braking (*TESDB*) approached to stop the vehicle), the current EV speed reference becomes $v_{\text{End of trip}}$. *TESDB* is calculated using Eq. (50).

$$TESDB = dec\_factor\_safety \frac{v^2_{\text{ref\_trip end}}}{2 \cdot a_{\text{dec}}} \tag{50}$$

where a_{dec} (m/s^2) is calculated using Eq. (51) based on v_{diff_dec} (m/s), the difference between the desired speed after deceleration and the current EV speed. *dec_factor_safety* is the safety parameter used during deceleration mode to ensure that the vehicle slows down or stops within the allowable distance. The deceleration (a_{dec}) was assumed depending on $v_{\text{diff\_dec}}$ as follows:

$$\begin{matrix} a_{dec} = 2.3 \\ a_{dec} = 2.25 \\ a_{dec} = 2.35 \end{matrix} \quad \text{if} \quad \begin{cases} 100 \leq v_{diff\_dec} \leq 129 \\ 50 \leq v_{diff\_dec} \leq 100 \\ 0 \leq v_{diff\_dec} \leq 50 \end{cases} \tag{51}$$

According to the micro-trip presented in Fig. 2, $v_{\text{ref\_trip end}}$ in Eq. (50) refers to v_{ref3}.

9 Results and Discussions

9.1 Neighborhood Micro-trip Planning

The results of the DAOTP system in the present work are presented based on a few assumptions as follows. The speed limits are chosen only based on the corresponding driving cycle followed by EV. The safety factors such as safe following distance, etc., related to traffic congestion, road condition, weather, location importance, etc., are not considered while fixing the EV speed limits for the sake of simplicity. Other assumptions concerning accelerations limits, model parameters, etc., have been mentioned in the previous sections. In this study, other trip parameter values considered are depicted in Table 2.

Moreover, the applied micro-trip is simple in the sense that it follows only one driving cycle type for the entire trip length. The velocity profile versus distance of a simple micro-trip consisting of one driving cycle type is presented in Fig. 6. The simple micro-trip consists of one acceleration mode followed by a constant speed model within the speed limits of the driving cycle, and after that, the EV comes to a

Table 2 Values of different parameters considered during simulation process

Parameters	Value
dt (in, s)	0.01
Auxiliary load (in, J/s)	10500
SOC_{init}	1.0
J_{desire}	1.0
K	2
dec_factor_safety	1.25

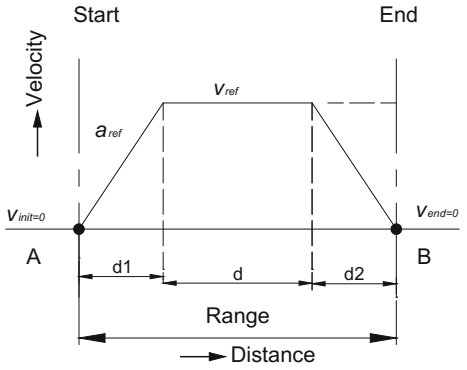


Fig. 6 Velocity versus distance plot of a simple micro-trip consisting of one driving cycle

Table 3 Values of GA parameters

GA Parameters	Value
N	250
Mutation probability	1/length of chromosome
Crossover probability	0.985
No. of bits used for each variable	20
Random seed	0.4
Number of generation	100

stop during the deceleration mode. The values of GA parameters presented in Table 3 are used for MOO process throughout this chapter unless otherwise stated.

In an urban area, depending on the vehicle speed limits, two driving cycle types, neighborhood (v_{erf} is varying from 8 to 40 km/h), and urban (v_{ref} is varying from 40 to 56 km/h) are applicable.

Figure 7 presents the Pareto fronts achieved by DAOTP after solving the MOOP (defined in Sect. 7) for minimization of energy consumption, trip time, and average jerk in the neighborhood driving cycle are presented in Fig. 7. In this figure, the Pareto fronts are shown for six different trip lengths (ranging from 0.5 to 20 km) keeping the other parameters related to the route (presented in Table 4) fixed. The wind angle is measured based on the same coordinate system as followed by the vehicle. That is, if the direction of the wind and the vehicle are the same, the wind angle equals zero.

In Fig. 7, Pareto front spans in all three objectives (energy, trip time, and average jerk) were observed to be dissimilar for dissimilar trip length, as anticipated. With increasing the trip length, the Pareto fronts are found to be migrated gradually move away from the origin. Among the six chosen trip lengths, some optimal solutions of trip lengths especially, 5, 8, and 10 km are found to be of a very low average jerk (less than 4 m/s^3), longer trip time and more energy consumption. On the other hand, few solutions of trip length (in particular, 0.5 and 5 km) possess a very high average

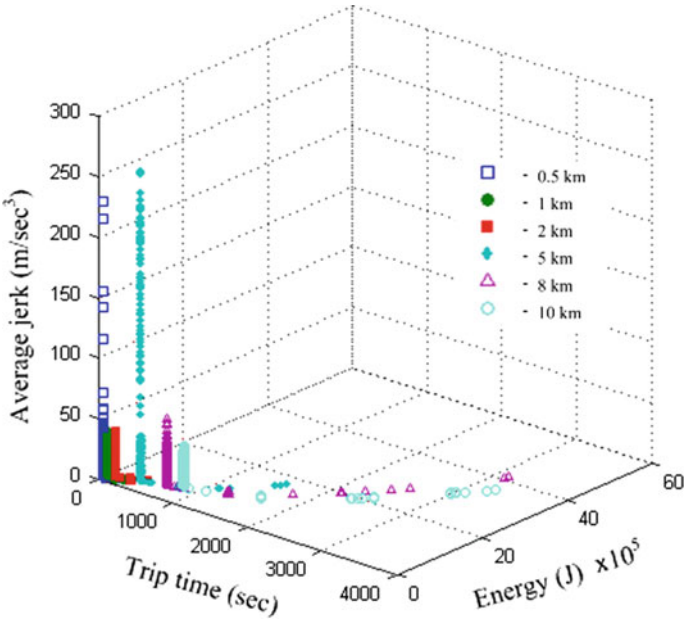


Fig. 7 Pareto-optimal fronts obtained in solving MOOP defined in Sect. 7 in neighborhood driving

Table 4 Route characteristic data

Sr. No.	Distance		Route characteristic parameters	
			Road gradient	
	From	To	Value	Angle (°)
1	0	Trip end	0	–
			Rolling frictional coefficient of road surface	
2	0	Trip end	0.015	–
			Density of air	
3	0	Trip end	1.143	–
			Wind velocity (km/h)	
4	0	Trip end	0	0

jerk value (more than 100 m/s^3) with short trip time and low energy consumption. It is obvious that such kind of solutions having extreme objective values is not to be attractive to the driver. By realizing this, Pareto-optimal solutions are screened by limiting the average jerk values in the range, 4–100 m/s^3, and the revised fronts are shown in Fig. 8. It is interesting to observe that as the trip length increases, less number of solutions outside the above mentioned average jerk limit belong to the Pareto front. This is to be expected since the definition of average jerk involves

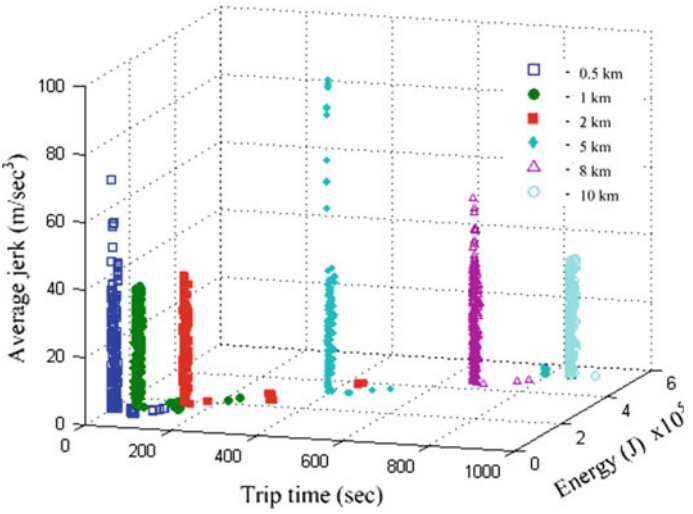


Fig. 8 Revised Pareto-optimal fronts based on average jerk limits (4 m/s^3 < average jerk < 100 m/s^3) in neighborhood driving cycle for different simple micro-trip length

dividing by the number of times (n_d) the jerk exceeds a certain limit (J_{desire}) according to Eq. (48). Since there is only one acceleration mode for all the micro-trips in Fig. 8, the average jerk is expected to generally decrease as the trip length increases. n_d increases, but j_i is found to be low when the EV runs at a constant speed.

The amount of energy stored in a battery is realized by knowing its SOC, and it can easily measure and presented to the driver through a suitable device based on based on multi-modal electrochemical impedance spectroscopy [73] and also easily interpreted by the driver. Thus, instead of actual energy utilization, it is more expedient for a driver to know the associated residual battery SOC. For that reason, the optimization results are discussed here in terms of SOC value at the end of the trip (*SOC*) and the trip time.

In Fig. 9, plot of *SOC* versus trip time for the 0.5 km micro-trip is presented. It is observed that after a certain trip time, *SOC* is found to be decreasing with increasing trip time because of a low jerk. It is evident that though the average jerk is low, such kind of solutions is not attractive at all. Instead solutions with a high *SOC* or a low trip time will be more interesting. A dotted ellipse in Fig. 9 shows the most attractive portion of the *SOC*-trip time plot and the corresponding solutions are replotted in Fig. 10. Though these solutions are non-dominated with respect to the original (three) objectives, are not non-dominated with respect to *SOC* and trip time. As a result, a one-to-one *SOC*-trip time plot is not anticipated. Figure 10 shows that *SOC* decreases with increasing time because the energy consumption is inversely varying with time. The variation of energy consumption with *SOC* is also demonstrated in Fig. 10. The energy consumption is found to be linearly varying with

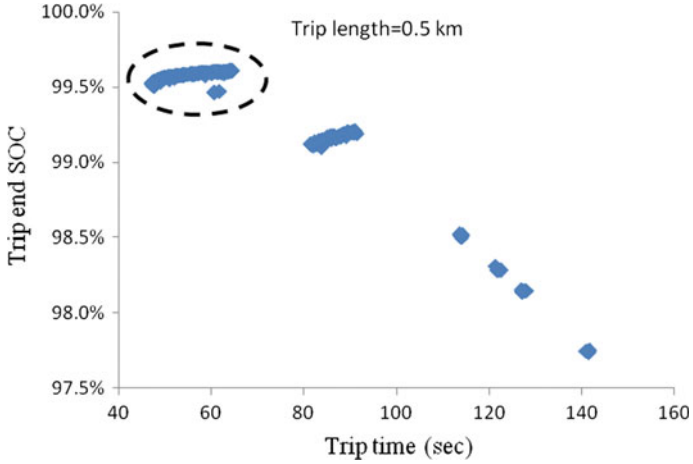


Fig. 9 Variation of end SOC with trip time in 0.5 km range

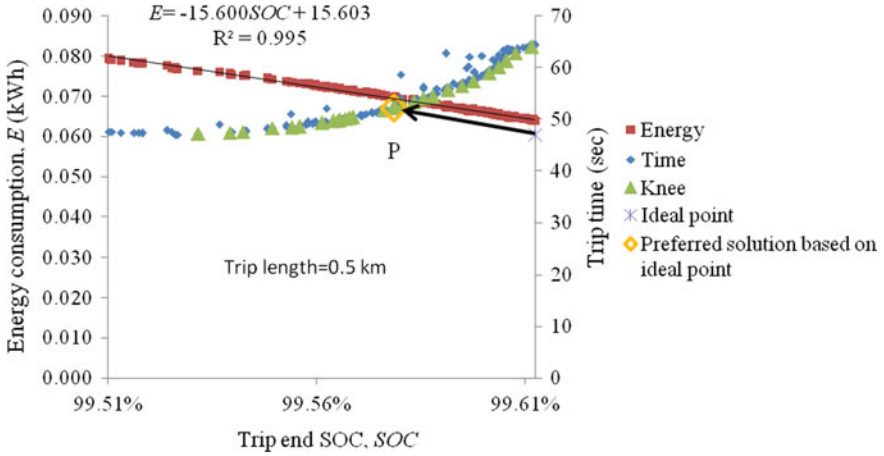


Fig. 10 Range-SOC-time plot corresponding to a micro-tip of 0.5 km

SOC with a gradient of −15.6. The empirical linear equation with a corresponding the coefficient of determination (R^2) value is depicted in the Fig. 10.

Similar studies on the variation of *SOC* with trip time were conducted for the other trip lengths. Figure 11 is similar to Fig. 9 except that the solutions correspond to the 5 km micro-trip. Considering the interesting part of the plot based on the relationship between SOC and time, the trip end SOC (*SOC*) versus trip time plot is presented in Fig. 12. It also shows the energy consumption of the interesting solutions. The linear relationship of energy with *SOC* is depicted in the Fig. 12 and has a high R^2 value, just like Fig. 10. Moreover, after analyzing the energy and trip end SOC relationships of trip lengths 0.5 and 5 km as shown in Figs. 10 and 12, as well as that found in

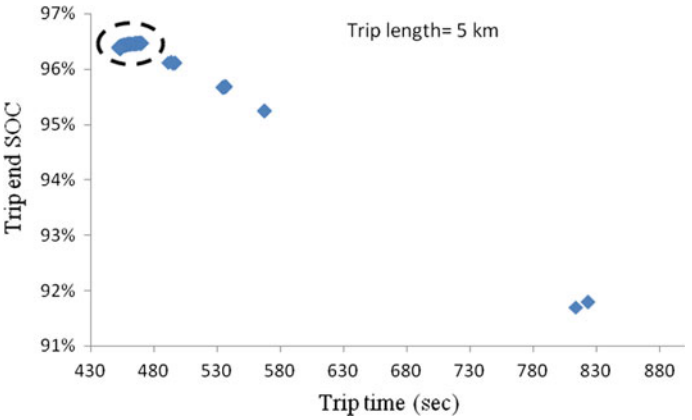


Fig. 11 Variation of trip end SOC with trip time for 5 km trip

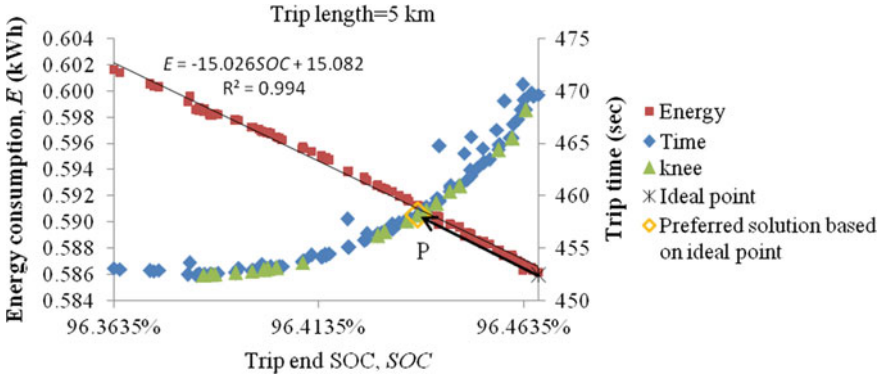


Fig. 12 Energy–*SOC*–time plot corresponding to a micro-tip of 5 km

other trip lengths, it was found that the slopes and intercepts are very close up to a trip length of 10 km with standard deviations, 0.22 and 0.20, respectively. After taking a mean value of both slope and intercept, a generalized linear relationship among (kWh) and *SOC* (0–100%) the optimal solutions defined in Eq. (52) may be applicable up to 10 km trip length for the present EV model.

$$E = -15.4245SOC + 15.464 \tag{52}$$

After 10 km trip length, the slope and intercept were found to be different and less than that in Eq. (52).

Figure 10 and Fig. 12 present the initial twenty knee-points evaluated using Eq. (7) for the trip lengths 0.5 km and 5 km, respectively. The reference point based preferred solutions for these trip lengths are also depicted in Fig. 10 and Fig. 12, respectively. The ideal (reference) point based preferred solution is found to be one of the twenty

knee points, and it is observed in both the trip lengths. However, it is not always true. An example is the case of 8 km trip length (not shown in the chapter).

Values of the three objectives, decision variables, and *SOC* are shown in Table 5. As mentioned in Sect. 8, a deceleration safety factor (presented in Table 2) was considered during braking the EV to ensure that the vehicle stopped before or close to the trip end location. Now, depending on the EV speed the *TESDB* is varied. Thus, instead of the same trip length (assumed prior to the trip), each solution possesses a different distance but always close to the desired trip length. The actual range covered corresponding to the optimal solution is also enlisted in Table 5. Prior to the initiation of the journey, DAOTP evaluates the optimal driving strategies in different driving modes for the entire trip in an offline mode based on the route information and battery SOC. According to the DAOTP predictions, the driver makes a decision for his/her trip planning. After starting the trip, DAOTP begins evaluating the new (revised) optimal driving strategy in every (certain) time interval according to the current EV state and route characteristic data in online mode till the trip ends. This online suggestion of DAOTP accounts for any deviation from the initial optimal driving strategy due to unforeseen circumstances or a lack of concentration by the driver. In such circumstances, the driver will be notified right after a time interval by comparing the previous and present DAOTP predictions. The subsequent driving strategies will be very similar to the previous strategy if the deviation applied driving strategy from the predicted is minor.

In both the trip lengths (0.5 and 5 km), it was found that the optimal speed is close to the maximum speed limit of neighborhood driving cycle (40 km/h). The variation of EV speed with trip end SOC of the Pareto-optimal solutions found in 0.5 km trip is presented in Fig. 13. About 94% of the solutions possess the speed around 40 km/h, which is the speed limit. From Table 5, it was noticed that in both trip lengths, the first acceleration value is found to be higher than the second acceleration value. On the other hand, the first acceleration's duration was found to be lower than that of the second acceleration. The variations of EV accelerations and acceleration durations with trip end SOC found in the 0.5 km trip length are presented Fig. 14. In Fig. 14, the first accelerations of nearly all Pareto-optimal solutions are observed to be similar, and possessed a high value (close to the upper limit of the acceleration range presented in Eq. (12)). On the other hand, the second acceleration is increasing with increasing the trip end SOC with a second-order polynomial relationship (with a coefficient of determination, $R^2 = 0.9773$) defined by Eq. (53). Similarly, the duration of first acceleration of the Pareto-optimal solutions was found to be a constant value close to 1.65 s and that of the second acceleration to be varying with trip end SOC according to Eq. (54) with a coefficient of determination, $R^2 = 0.95$.

$$a_2 = 5.279 \times 10^6 \mathrm{SOC}^2 - 1.051 \times 10^7 \mathrm{SOC} + 5.236 \times 10^6 \tag{53}$$

$$t_2 = 1.01 \times 10^8 \mathrm{SOC}^2 - 2.01 \times 10^8 \mathrm{SOC} + 9.99 \times 10^7 \tag{54}$$

After conducting the similar innovation study for trip length 5 km, it was observed that the first acceleration value and corresponding duration of the Pareto-optimal

Table 5 Preferred optimal solutions obtained using reference point based method in neighborhood micro-trip

Trip length (km)	Ideal point according to SOC–Time plot	Optimal values of objectives	Optimal values of decision variables					Actual range covered, R	Preferred Pareto-optimal solution
			V	a_k	t_k	k_p	k_i		
0.5	SOC = 0.996084; Time = 47.038	T_{Trip} = 52.1573; E = 251006; J_{Avg} = 22.614	39.994	a_1 = 2.99542; a_2 = 0.321796	t_1 = 1.65767; t_2 = 19.1627	0.222576	0.707864	485.981	Denoted as "P" in Fig. 10
5	SOC = 0.964669; Time = 452.430	T_{Trip} = 458.166; E = 2128720; J_{Avg} = 34.8546	39.9992	a_1 = 2.94564; a_2 = 0.289447	t_1 = 1.68578; t_2 = 21.2146	0.245648	0.730512	4985.99	Denoted as "P" in Fig. 12

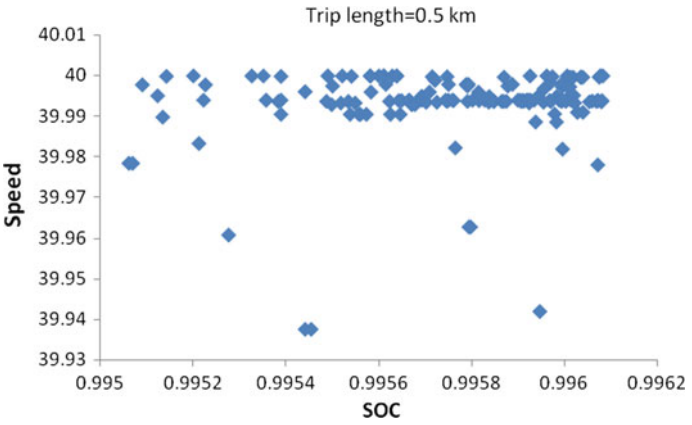


Fig. 13 Variation of EV speed with SOC in 0.5 km range

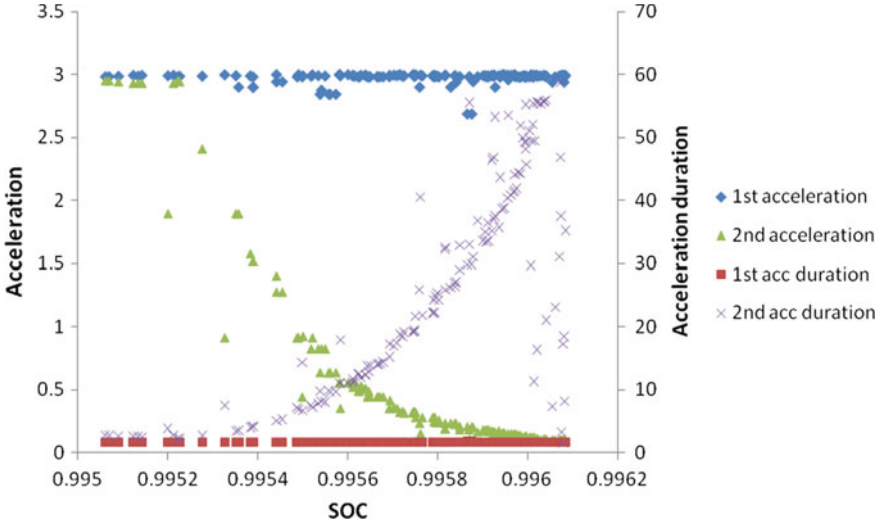


Fig. 14 Variations of EV accelerations and acceleration durations with SOC in 0.5 km range

solutions are found to be the same as observed in the case of trip length, 0.5 km. The relationships of the second acceleration and its duration with trip end SOC are also found to be a second-order polynomial defined by Eqs. (55) and (56) corresponding to the coefficients of determination (R^2), 0.9865 and 0.9826, respectively. The difference between the coefficient values of the corresponding relationships of trip length 0.5 and 5 km are found to be negligible. This suggests that the acceleration strategies are found to be almost similar nature irrespective of the trip length in neighborhood micro-trips. This implies that the migration of the Pareto fronts in Fig. 9 that was

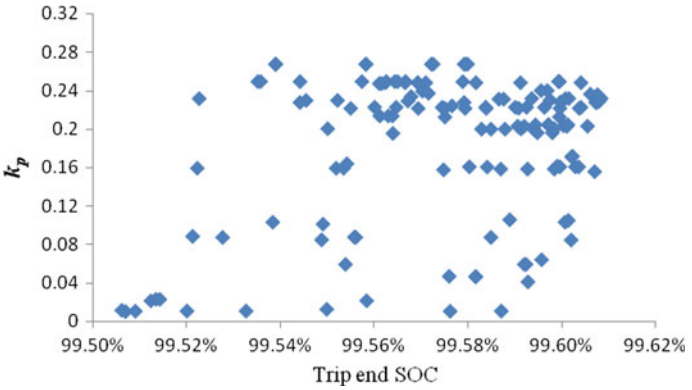


Fig. 15 Variation of k_p with trip end SOC in 0.5 km range

alluded to previously in this section is solely due to the duration of constant speed mode dictated by the trip length.

$$a_2 = 6.458 \times 10^6 \mathrm{SOC}^2 - 1.246 \times 10^7 \mathrm{SOC} + 6.007 \times 10^6 \tag{55}$$

$$t_2 = 1.046 \times 10^8 \mathrm{SOC}^2 - 2.016 \times 10^8 \mathrm{SOC} + 9.715 \times 10^7 \tag{56}$$

After analyzing the variations of k_p and k_i with trip end SOC, as presented in Figs. 15 and 16, respectively, it was noticed that most of solutions are lie within a particular region. In the case of trip length 0.5 km, k_p is varying from 0.2 to 0.25 and k_i, from 0.3 to 0.8; 67% and 62%, respectively, of the entire solutions are within these ranges. Similarly, k_p and k_i values of most of the Pareto-optimal solutions obtained in higher trip lengths vary with certain ranges. However, from Table 5, a comparatively high value of both k_p and k_i was found in the preferred solution with higher trip length.

Table 6 shows the energy savings that are achieved on various trips. The % energy savings are evaluated based on the minimum and maximum energy consumptions of the interesting solutions as found after optimization process. Such energy savings can be obtained by sacrificing the corresponding trip duration. The maximum and minimum energy consumption, the lowest value of trip end *SOC*, and its effective range found among the optimal solutions corresponding to a trip length are also enlisted in Table 6. Like energy saving, the % trip time in the fourth column of Table 6 are also calculated corresponding to the minimum and maximum time required to complete the trips. From Table 6, it was observed that the energy saving reduces with increasing the trip length. However, the energy saving per unit trip time lost is getting more as the trip length increases (e.g., 0.46 and 1.06, in trip length 0.5 and 10 km, respectively). The following describes a scenario that highlights the versatility of the proposed DAOTP system and possible energy savings among the obtained Pareto front solutions in exchange for sacrificing trip time. For the sake of demonstration,

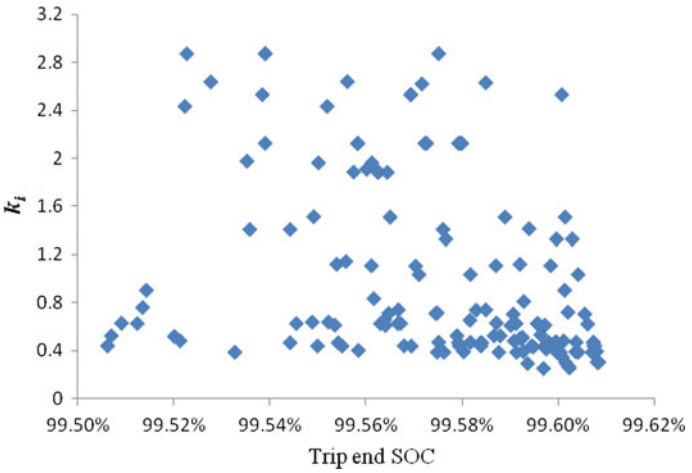


Fig. 16 Variation of k_i with trip end SOC in 0.5 km trip length

Table 6 Energy saving corresponding to time lost for different neighborhood micro-trip length

Trip length (km)	Maximum energy consumption (kwh)	Minimum energy consumption (kwh)	Energy saving (%)	Trip time lost (%)	Lowest SOC (%)	Effective SOC range (%)
0.5	0.07326	0.06507	12.581	27.14	99.5062	0.1022
1	0.13717	0.12221	12.237	21.14	99.1587	0.0986
2	0.25265	0.23817	6.078	9.76	98.464	0.096
5	0.60166	0.58616	2.644	40.22	96.3635	0.1034
8	0.96582	0.93420	3.385	2.67	94.169	0.1985
10	1.23428	1.16499	5.947	5.59	92.5536	0.4233

the solutions having maximum energy consumption are supposed to be suboptimal, considered to be obtained without DAOTP. On the other hand, the solutions having minimum energy consumption are considered to be the best DAOTP solutions. This demonstration should not be confused with selection criteria (decision-making techniques/driver preference) of a DAOTP solution for implementation.

In a downtown area where the neighborhood driving cycle is normally followed, the vehicle is required to stop frequently due to more traffic congestion and traffic lights. As a result, to cover a trip length, the vehicle breaks the entire trip length into multiple micro-trips, the beginning and end of which correspond to two successive stops. After analyzing the data presented in Table 6, it was found that there will be 6.82% $(=(0.07326 \times 2 - 0.13717) \times 100/0.13717)$ more energy consumption if the EV runs 1 km with a stop after every 0.5 km compared to traveling the same distance at a stretch without adopting the optimized driving strategy. This energy consumption can be reduced to 5.12% $((0.06507 \times 2 - 0.13717) \times 100/0.13717)$ if

DAOTP results are followed during the journey with breaks. So, there will be a saving of up to 1.70%. (=6.82–5.12). For more number of stops in a trip length, the saving is more. For instance, the extra energy loss to travel 5 km distance with a stop after every 0.5 km traveled instead of a continuous journey and without adopting DAOTP results is 21.76%. This excess energy consumption comes down to 8.15% if DAOTP results are adopted and the saving of energy will be up to 13.61%. corresponding to 2.722% per km. If in both cases (continuous and journey with stops) DAOTP results are utilized, the amount of energy loss will be reduced from 21.76% to 11%, and 6.82% to 6.48%, for trip lengths 5 km and 1 km, respectively, with a stop after every 0.5 km. The above study reveals that the benefits of adopting DAOTP system during performing a trip with EV.

9.2 Urban Micro-trip Planning

Similar studies were conducted on DAOTP application in urban micro-trips (speed range 40–56 km/h) for the same six trip lengths as considered in neighborhood driving cycle. The nature of Pareto fronts found was found similar to that of the neighborhood driving cycle. The interesting solutions in a Pareto front were identified following the concept based on the problem-specific information that was adopted in the neighborhood driving cycle. The relationship between E and SOC was found to also be linear. The generalized linear relationship derived after averaging the slope and intercept found in six trip lengths is presented in Eq. (57). The standard deviations of the slope and intercept in different trip lengths were found at 0.25 and 0.24, respectively. These are slightly lower than the slope and intercept values for neighborhood trips.

$$E = -14.871SOC + 14.995 \tag{57}$$

The preferred solution was found out using the reference point technique based on the ideal point. In Table 7, values of optimum decision variable of the preferred solutions found in urban micro-trips of lengths 0.5 and 5 km, along with the corresponding objectives, actual range covered and the respective ideal point used are listed. After comparing the data of Tables 5 and 7, a similarity in the optimization results was for the optimal speed, both the acceleration values and the first acceleration duration for both the trip lengths. However, in urban driving cycles, it was found that for the 0.5 km trip length, the optimal value of the second acceleration duration is low compared to that in the neighborhood driving cycle. This happens because since the trip length is short so that before reaching the optimal speed, the EV covers the trip length. Due to this reason, a smooth plot of the second acceleration duration with trip end SOC (as seen in the neighborhood driving cycle, Fig. 14 corresponding to trip length, 0.5 km) may not be expected through a similar nature of the relationship of the second acceleration value with trip end SOC was observed. As the trip length increases, similar results as observed in the neighborhood driving

cycle can be anticipated in the urban driving cycle as well. Moreover, a significant difference in the k_p and k_i was found. In the higher speed driving cycle, the optimum values of those variables reduce with increasing trip length.

The above DAOTP results are found based on the route characteristic data presented in Table 4 and considering an initial battery SOC of 100%. Changes of route characteristic parameters and a different initial battery SOC may affect the DAOTP results. The effectiveness of route characteristic parameters, such as road gradient, road surface condition, wind velocity, elevation, and battery initial SOC on DAOTP results are investigated in highway micro-trips planning.

9.3 Highway and Interstate Micro-trip Planning

Results for high-speed driving cycles are presented here for seven trip lengths ranging from 1 to 50 km. The trips are considered based on the structure of a simple micro-trip as shown in Fig. 6. The same EV model and related model parameter values as mentioned above are adopted here. By solving the associated MOOP (presented in Sect. 7 of the first part), a smooth Pareto front surface was found for each trip length in both high-speed driving cycles, highway and interstate, as shown in Figs. 17 and 18, respectively. This is unlike the results found in Figs. 7 and 8 for low-speed driving cycles. During the optimization process, the same GA parameters presented in Table 2 along with the other mentioned assumptions are also considered. From these figures, it was observed that the span of each objective (the difference between the minimum and maximum values) such as T_{Trip}, E, and J_{Avg} in the Pareto fronts are increasing with increasing the trip length.

In the Pareto fronts found in low driving cycles, it was observed that there exist many solutions having a low average jerk value (less than 4 m/s^3) that corresponds to a high trip time and energy consumption or a very high average jerk value (more than 100 m/s^3) with a low trip time and energy consumption. Moreover, by plotting the trip end SOC (SOC) with the T_{Trip} of optimal solutions, even after sorting based on the restricted average jerk values (4 m/s^3 $< J_{\text{Avg}} <$ 100 m/s^3), it was noticed that after a certain trip time there were some solutions in the Pareto-optimal set that possess an uninteresting feature, i.e., the battery SOC is decreasing with increasing trip time. The presence of such uninteresting features was observed for even higher trip lengths, till 10 km. However, for higher trip lengths, the number of such solutions reduces significantly. Such adverse characteristics were not found in the optimal solutions obtained after MOO (multi-objective optimization) for high-speed driving cycles, according to Figs. 17 and 18. Figure 17 and Fig. 18 present the Pareto fronts obtained by the minimization of trip time, energy, and average jerk for a simple micro-tip with different lengths for highway and interstate driving cycles, respectively. Therefore, for high-speed driving cycles, Pareto-optimal solutions after MOO do not require any such sorting based on problem-specific higher level information. The SOC versus T_{Trip} plots are drawn directly after optimization as shown in Figs. 19 and 20 corresponding to the highway and interstate driving cycles, respectively. From

Table 7 Preferred optimal solutions obtained using reference point based method in urban micro-trip

Trip length (km)	Ideal point according to SOC–Time plot	Optimal values of objectives	Optimal values of decision variables					Actual range covered, R
			v	a_k	t_k	k_p	k_i	
0.5	SOC = 0.996379; Time = 35.0901	T_{Trip} = 46.4382; E = 241263; J_{Avg} = 18.4386	55.9811	a_1 = 2.90455; a_2 = 0.196576	t_1 = 2.38019; t_2 = 3.98986	0.206065	1.05836	468.613
5	SOC = 0.968969; Time = 324.565	T_{Trip} = 341.042; E = 1820110; J_{Avg} = 11.9862	55.9898	a_1 = 2.98857; a_2 = 0.145721	t_1 = 2.32508; t_2 = 58.4407	0.156088	0.381912	4972.67

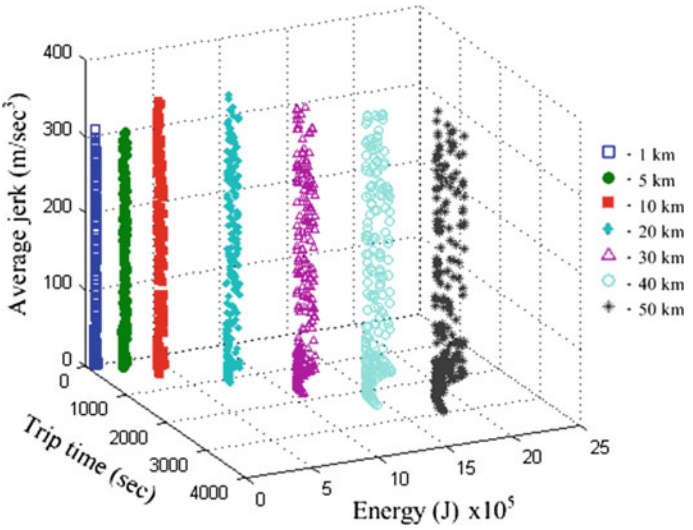


Fig. 17 Pareto-optimal fronts obtained in solving MOOP defined in Sect. 7 in highway driving

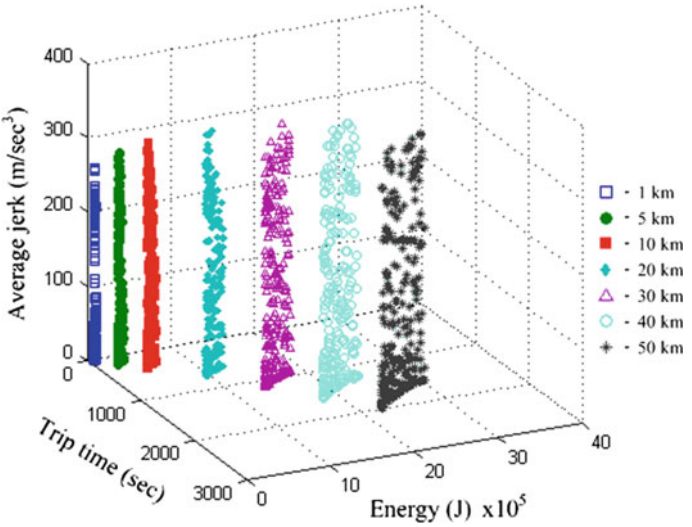


Fig. 18 Pareto-optimal fronts obtained in solving MOOP defined in Sect. 7 in interstate driving

Figs. 19 and 20, it was noticed that the lowest Pareto front *SOC* of any driving cycle is always found to be higher with higher trip length. Contrary to that, the lowest Pareto front trip time of a driving cycle is always noticed to be lower with higher trip length. Such finding is evident since the time requirement and energy consumption are required to be more for a longer trip. This observation suggests the obtained optimization results are meaningful and realistic. In order to demonstrate the optimized

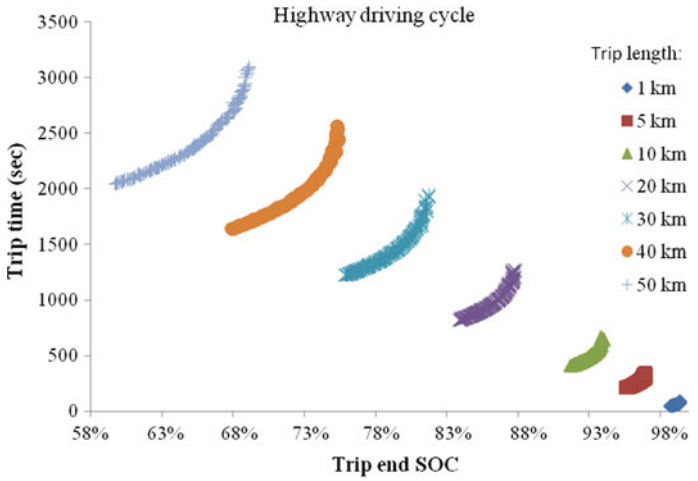


Fig. 19 Variation of trip end SOC with trip time for a highway driving cycle

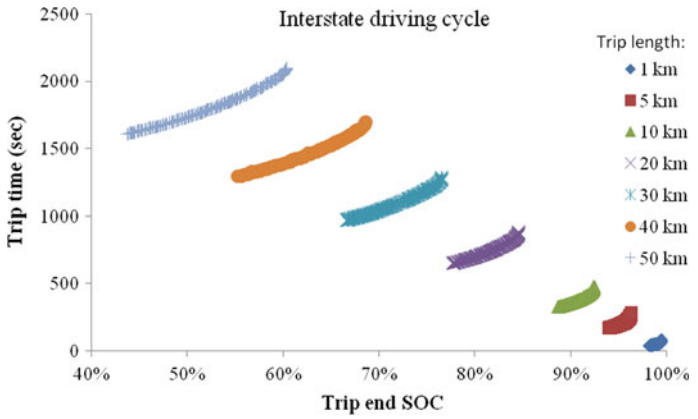


Fig. 20 Variation of trip end SOC with trip time in interstate driving cycle

driving strategy characteristics (*DSC*), a simple micro-trip consisting of one driving cycle (whose velocity profile versus distance is demonstrated in Fig. 6 is considered here. The plots of trip end SOC and trip time for the simple micro-tip of 20 km for highway and interstate driving cycles are shown in Figs. 21 and 22, respectively. Both the figures also demonstrate the corresponding energy consumption, E. The first few best knee-points and the preferred solution obtained using reference point based DM technique are pointed out in the SOC-T_{Trip} plot.

As in low-speed driving cycles, a linear relationship between E and SOC for both highway and interstate driving cycles is found and these are defined in Eqs. (58) and (59), respectively.

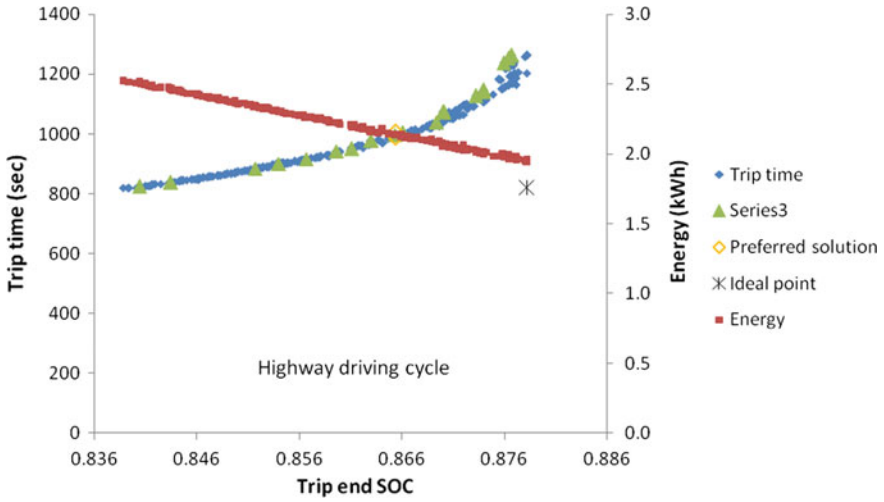


Fig. 21 Energy-trip end SOC–Trip time plot corresponding to minimization of trip time, energy and average jerk in a simple highway micro-tip of 20 km length

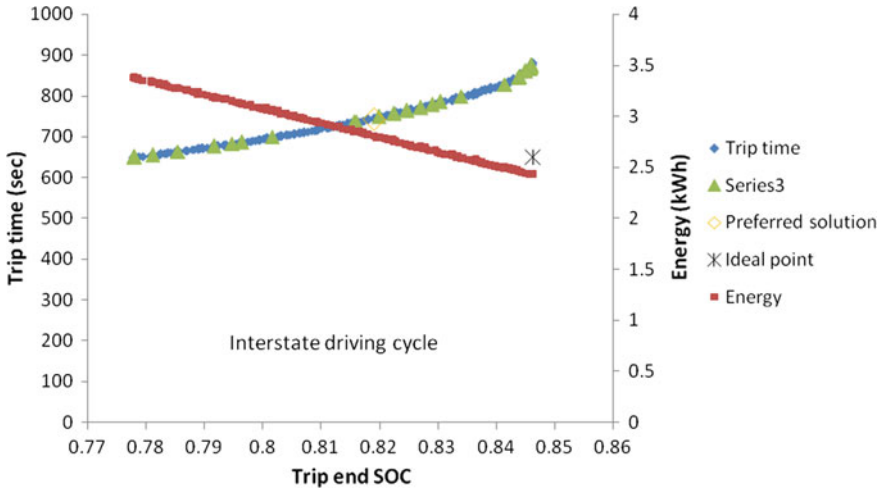


Fig. 22 Energy-trip end SOC–Trip time plot corresponding to minimization of trip time, energy and average jerk in a simple interstate micro-tip of 20 km length

$$E = -14.35SOC + 14.594 \tag{58}$$

$$E = -13.782SOC + 14.164 \tag{59}$$

The slope and intercepts of Eq. (58) are calculated by taking the mean of the slope and intercept values found for different trip lengths in Fig. 17. The corresponding STDs (standard deviations) are 0.198 and 0.215, respectively. Similarly, the slope

and intercepts of Eq. (59) are determined for Fig. 18 and the corresponding STDs are 0.220 and 0.172, respectively. Comparing the relationships between E and SOC for all four driving cycles, it was found that both the slope and intercepts of the linear (average) empirical relation are decreasing with increasing EV speed (v_{ref}). Such findings suggest that in low-speed driving cycles, the SOC decreases with respect to energy consumption at a lower rate compared to that in high-speed driving cycles, which follows typical battery discharge characteristics [4, 61]. This confirms the reliability and meaningful nature of optimization results expected from the proposed DAOTP system.

In Table 8, the values of objective functions (energy consumption, trip time, and average jerk) corresponding to the preferred solutions (representing the optimal driving strategy) obtained through the DAOTP system for high-speed driving cycles for a trip length of 20 km are presented. Table 8 also shows the values of related decision variables (driving characteristics namely EV speed, acceleration(s) and corresponding duration(s), and controller gains), and SOC. Here, two acceleration rates ($k=2$) are assumed during acceleration mode. In contrast to low-speed driving cycles, the optimal speeds were not found to be very close to the upper driving cycle speed limit. Here, it is interesting to note that the optimal speed has a tendency to lie close to the upper driving cycle speed limit as it is varying from high to low. The motivation of such findings can be enlightened by taking into consideration the dependency of motor power and efficiency on speed is explained in [58, 90]. At a low speed, the electric motor efficiency is very low. At medium speeds, it is almost as a constant. At high speeds, the efficiency starts to deteriorate. In order to minimize the trip time, the EV speed is required to be high. Coincidentally, the upper speed limit of a low-speed driving cycle is positioned in the EV speed range that yields the maximum motor efficiency. On the other hand, since power consumption increases in a quadratic manner with EV speed (v) and the maximum speed of a low-speed driving cycle is not too high, MOGA selects the optimal speed closer to the upper limit of the low-speed driving cycle. On the other hand, at high EV speeds, though the trip time reduces, the motor shows low efficiency in addition to the power consumption being extremely high as it is a function of v^2. Therefore, solutions close to the upper limit of the high-speed driving cycle are not selected by MOGA. For these reasons, it is observed that the optimal solution is found further and further away from the upper speed limit of a driving cycle when it changes from low to high speed. As in low-speed driving cycles, a high value of the first acceleration (a_1) is also observed in high-speed driving cycles. On the other hand, for low-speed driving cycles the value of the second acceleration (a_2) was found to be significantly lower than a_1. Though the same result was observed for a highway driving cycles, the a_2 value in the interstate driving cycle was found to be comparatively higher than that in low-speed driving cycles (sometimes, close to a_1). But in those cases, t_2 was found to be opposite in nature. The above findings on optimal DSC are only based on the route characteristic parameters depicted in Table 4. Such findings may be different according to the changes of route characteristic parameters and initial battery SOC. In the following, variations of DAOTP results with these parameters are analyzed for the same trip and driving cycle.

Table 8 List preferred optimal solutions found in different driving cycles for trip length, 20 km

Driving cycle	Ideal point	Optimal values of objectives			Optimal values of decision variables					Trip end SOC	Actual range
		T_{Trip} (s)	E (kWh)	J_{Avg} (m/s^3)	v (km/hr)	a_k (m/s^2)	t_k (s)	k_p	k_i		
Highway	$SOC = 0.878268$; $T_{\text{Trip}} = 818.918$	996.891	2.132308	14.9524	74.0454	$a_1 = 2.49943$; $a_2 = 0.185805$	$t_1 = 3.23687$; $t_2 = 65.8194$	0.201308	0.987817	0.865352	19952.3
Interstate	$SOC = 0.846342$; $T_{\text{Trip}} = 647.571$	743.324	2.807139	21.0881	97.6888	$a_1 = 2.98934$; $a_2 = 2.95014$	$t_1 = 4.05312$; $t_2 = 5.70701$	0.224253	2.91371	0.819008	19926.4

10 Effect of Initial SOC on DAOTP Results

Trip planning is carried out based on the initial battery SOC = 1.0. The EV can attain its maximum range with an optimal driving strategy since the battery energy is maximum. Therefore, a driver can plan a trip with greater confidence, especially if the trip length is below the maximum range. In certain situations, a full battery SOC before starting a trip may not be possible. Besides this, the study of the effect of initial SOC on DAOTP results [52] is important in the sense that if the vehicle is unexpectedly required to stop or to modify the original trip plan or due to some unforeseen reason, the original DAOTP results may no longer be valid. In that case, the remainder of the trip must be planned once more based on the current SOC, which would be different from the initial SOC before the trip was initiated. This scenario is not to be confused with the general operation of DAOTP, which continuously updates itself. That is, once the trip is planned, DAOTP updates itself after each time interval, but the optimal driving strategy remains the same provided the route characteristics also remain the same.

The battery energy depleted in terms of SOC difference between before and after the trip performed is presented in Figs. 23 and 24 in highway and interstate driving cycles, respectively. The results are taken considering a trip length of 20 km. From both the figures, it was observed that the rate of decreasing *SOC* is more when the initial battery SOC is low compare to high initial SOC irrespective of driving cycle type. For instance, to complete the 20 km interstate trip in 800 s, the SOC drops are found to be 44.565%, 29.288%, 21.356%, and 16.480% corresponding to the initial battery SOCs 0.4, 0.6, 0.8 and 1.0, respectively as found in Fig. 24. Considering the same trip length and initial SOC values (0.4, 0.6, 0.8 and 1.0), the SOC drops are found as 39.47%, 25.825%, 18.96, and 14.54%, respectively, for a highway for a trip time of 900 s, as shown in Fig. 23. These results offer the conclusion that *SOC* significantly depends on the initial battery SOC. The amount of *SOC* dropped with initial battery SOC follows an intricate relationship which is not studied here. Moreover, a unique relationship does not exist for all driving cycles. The width of the Pareto-optimal front corresponding to both *SOC* and trip time in a driving cycle does vary with the initial SOC value. Figures 23 and 24 also show the distance (range) covered by EV with initial SOC = 0.4. It is observed that all the solutions in the Pareto front are found to cover the 20 km trip length for a highway driving cycle. However, for the interstate driving cycle, some solutions are found that are not able to complete the entire trip. The SOC values of those solutions with trip time less than 600 min decrease below the limiting SOC value of 0.2 (a typical value to protect the battery and it offers a safety factor) before completing the desired trip length of 20 km. The above findings argued that an averaging method (a naive predictor) on the basis of previous experiences of past trip may suggest incorrect results where SOC consumption in concerned, specifically if it is a different driving cycle type. For most of the cases, the initial SOC of the present trip may not be the same as adopted in the previous trip. Moreover, changing the route characteristic information increases the

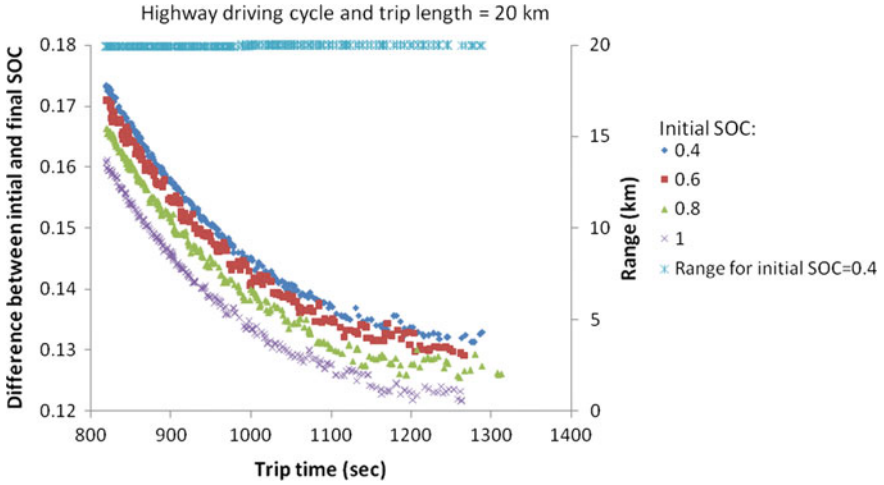


Fig. 23 SOC change with trip time for a highway driving cycle

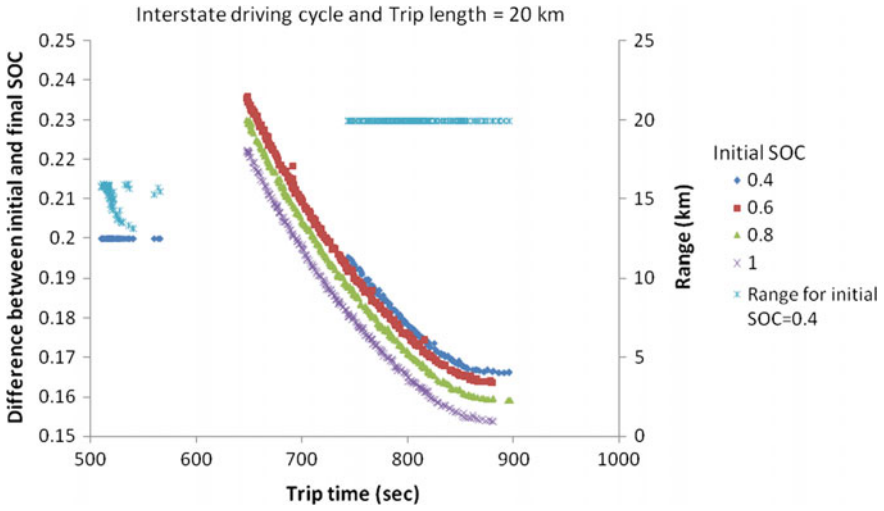


Fig. 24 SOC change with trip time in interstate driving cycle

probability of yielding inaccurate trip planning results. These issues are discussed in the subsequent section.

Table 9 Road gradient distribution of the micro-trip route

Sr. No.	Distance (km)		Gradient
	From	To	
1	0	2	0
2	2	8	Gradient value
3	8	12	0
4	12	18	−Gradient value
5	18	20	0

11 Influence of Route Characteristics on DAOTP Results

The influences of various route characteristic parameters on the DAOTP results are presented here [52]. Among various route characteristic parameters, the present study considers the most influential and common ones. These are the road gradient, wind velocity and direction, road surface, and air density due to change of elevation.

11.1 Road Gradient

In order to investigate the road gradient influence on DAOTP results, eight different road gradient values ranging from 0 to 3° are taken. The route architecture corresponding to each micro-trip follows the road gradient distribution presented in Table 9. Each route consists of both road gradient effects that may be encountered during a trip, uphill and downhill, denoted by positive and negative signs, respectively, in Table 9. Each gradient zone extends for an equal length, 6 km, in the trip. This is representative of a round trip, wherein the driver ends up at the starting point after trip, meaning there is no net change in elevation due to the road gradient.

Figure 25a, b demonstrate the influence of road gradients on DAOTP outcomes for highway and interstate driving cycles, respectively. It is quite obvious that more energy is consumed to maintain a constant vehicle speed in a road with positive gradient. In contrast, braking may be required in a road with negative gradient to keep a constant speed. However, during braking, there is a scope to recover the EV kinetic energy to some extent through regeneration techniques. The energy recuperation efficiency of a regenerative braking system depends on many factors such as vehicle speed, battery SOC, battery temperature, motor available braking torque, braking force, braking behavior [37], etc. and has a complex nature [43]. In the present study, the regenerative braking factor, *Regen* (defined by the fraction of the total available regenerative energy that is converted to battery energy), is considered as 0.5, which is a typical value for EVs. The deceleration rate is considered according to the vehicle speed defined in Eq. (51).

Although the positive and negative road gradient exist for the same fraction of the trip, the DAOTP results with the road gradient effect are found to be different

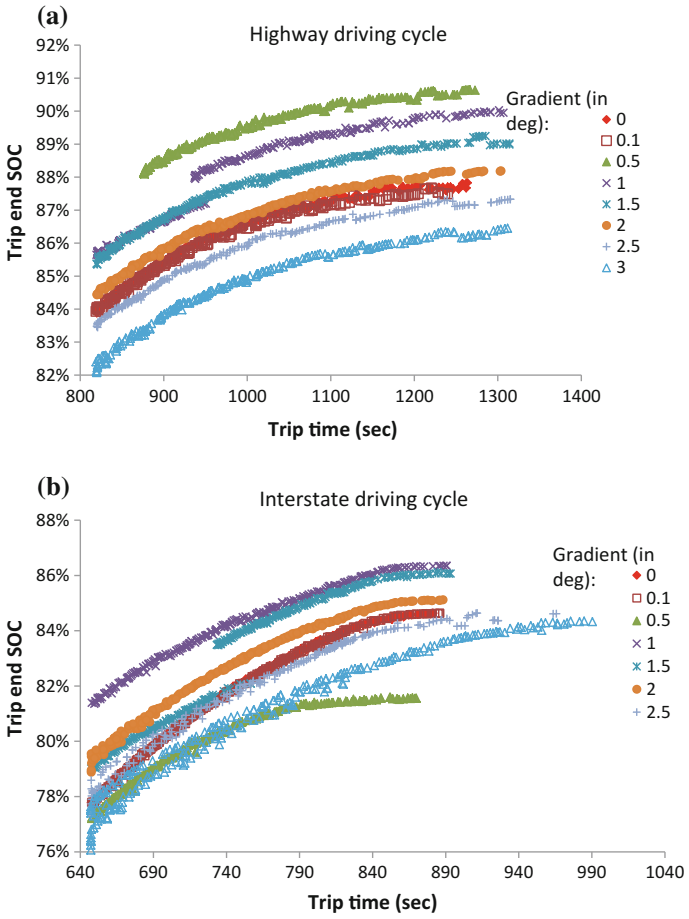


Fig. 25 Variation of trip end SOC with trip time for different road gradients to cover a trip length of 20 km in **a** Highway and **b** Interstate driving cycles

from that with zero gradient for both the driving cycles. For the highway driving cycle, a small gradient ($\leq 2°$) improves the trip end SOC, except very small gradient (0.1) or without any gradient. For high gradients ($>2°$), the trip end SOC deteriorates significantly, meaning more energy consumption during positive gradient than energy recovery through regeneration during negative gradient. These results highly depend on the regenerative braking system used and road gradient distribution of the rout. The influence of road gradient on network-wide vehicle energy consumption can be found in [59]. The same effects on the DAOTP results with road gradient were found for the interstate driving cycle. However, the Pareto front characteristic corresponding to different road gradient with respect to that of zero road gradient is not similar for highway and interstate driving cycles.

11.2 Wind Speed and Its Direction

In order to demonstrate the wind effect on DAOTP results, two wind speeds (10 and 30 km/h) blowing in two different directions (0° and 135° with respect to the EV driving direction) are considered. The results obtained with wind are compared with those obtained without wind. The Pareto fronts are presented in Fig. 26a, b. Figure 26a, b show the comparative results for highway and interstate driving cycles, respectively.

Generally, an energy saving is anticipated when the wind blowing in the direction of vehicle speed. In contrast, high energy consumption is needed while the wind flow is against the vehicle speed. A similar notion is also observed in the results presented in Fig. 26. However, the effect of wind on energy consumption variations follows a complex relationship. Various parameters such as vehicle speed, wind speed, and its direction, vehicle driving modes, etc. have an effect on the energy consumption. After analyzing the results, it was observed improvement rate of energy consumption is comparatively less than the deterioration rate of that. The reason behind such kind of phenomenon is that drag force is proportional to the squared of vehicle effective speed, as shown in Eq. (35). However, an unexpected observation for highway driving cycles was noticed in the case when the wind is in the same direction as the vehicle, as shown in Fig. 26a. It was noticed that some solutions in the Pareto front, particularly those having a high trip time, exhibit a trip end SOC that is less than that found without any wind. Moreover, as the wind speed increases, more solutions are found to follow this behavior. Through further simulations (not presented in this chapter), it was found that when the wind speed is very high (>50 km/h), the solutions follow the behavior as observed in Fig. 26b. Similar findings were noticed in low-speed driving cycles. The reason may be due to the fact that wind in the same direction as the vehicle speed reduces the net drag force. As a result, it accelerates the vehicle more per unit time step which yields more jerk. In the present MOOP, one of the objectives is to minimize the average jerk (calculated using Eq. (48)), so MOGA does not select those solutions experiencing more jerk. In low-speed driving cycles (low v_{ref}), the acceleration duration is shorter than that in high-speed driving cycles and a short acceleration duration yields a comparatively a high average jerk. For this reason, such results are observed in the optimal solutions when applying the DAOTP system to low-speed driving cycles.

Practically, wind follows a turbulent behavior in on-road environment. Due to that the wind effect on energy consumption becomes complicated in nature [97]. Thus, the nature of the results considering wind turbulence effects is difficult to predict [21, 98].

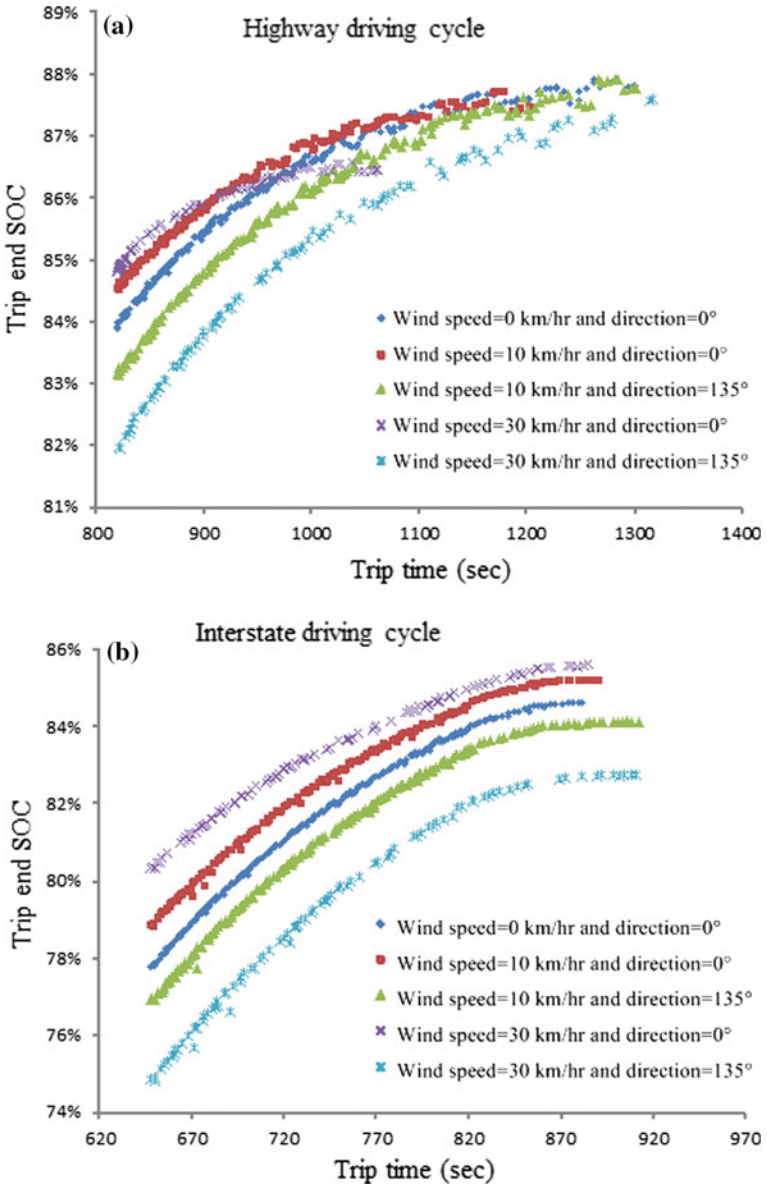


Fig. 26 Effect of wind on DAOTP results to cover a trip length of 20 km in **a** Highway and **b** Interstate driving cycles

Table 10 Road surface rolling frictional coefficient distribution of the micro-trip route

Sr. No.	Distance (km)		Friction coefficient of road surface
	From	To	
1	0	10	0.015
2	10	20	*M*

11.3 Road Surface

The properties of the road surface that affect the objectives, trip time, energy consumption, and jerk, are surface irregularity and rolling friction coefficient, μ [48]. So far, the above studies were carried out for a constant road surface with $\mu = 0.015$. The factor, irregularity, particularly restricts the speed limit of a route with the goal of avoiding jerk and accidents. This results in affecting the vehicle energy consumption. The effect of this factor can be simply tackled by constraining the speed limit of a route before finding optimum driving strategy. Thus, the influence of road surface irregularity on the DAOTP results is not investigated here. The effect of the rolling friction coefficient on the DAOTP results is investigated by comparing the *SOC* with respect to trip time plot found for six different friction coefficients. This study is carried out with a micro-trip of length 20 km having the first 10 km length with $\mu = 0.015$ and the remaining 10 km length with a different μ value, as shown in Table 10. Figure 27a, b demonstrate *SOC* with the trip end time for highway and interstate driving cycles, respectively, for different μ values. From these figures, it was noticed that *SOC* was found to be reduced as rolling frictional coefficient increases. Furthermore, it was noticed that for low value of μ (0.015), a little increase in μ results to a high loss of *SOC*. But, when μ is 0.02 or above, a uniform loss of *SOC* with increasing μ was found.

The friction coefficient between the tire and the roadway depends on the material used to make the road (such as asphalt, basalt, concrete, epoxy, etc.), the road surface condition (such as dry, wet, etc.), and the natural deterioration of the road surface roughness due to traffic and climate [47]. There are well-established methods to measure the road surface friction coefficients [1]. A detailed study on vehicle fuel consumption due to friction resistance can be found in [70]. These studies are applicable for any vehicle type.

11.4 Elevation

The elevation primarily affects the temperature and pressure of air at that location. This results in changing the air density thereby affecting the drag force acting on the EV. A high elevation with respect to sea level decreases the air density. The above studies were carried out for a constant air density, $\rho_{air} = 1.143$ kg/m^3. In

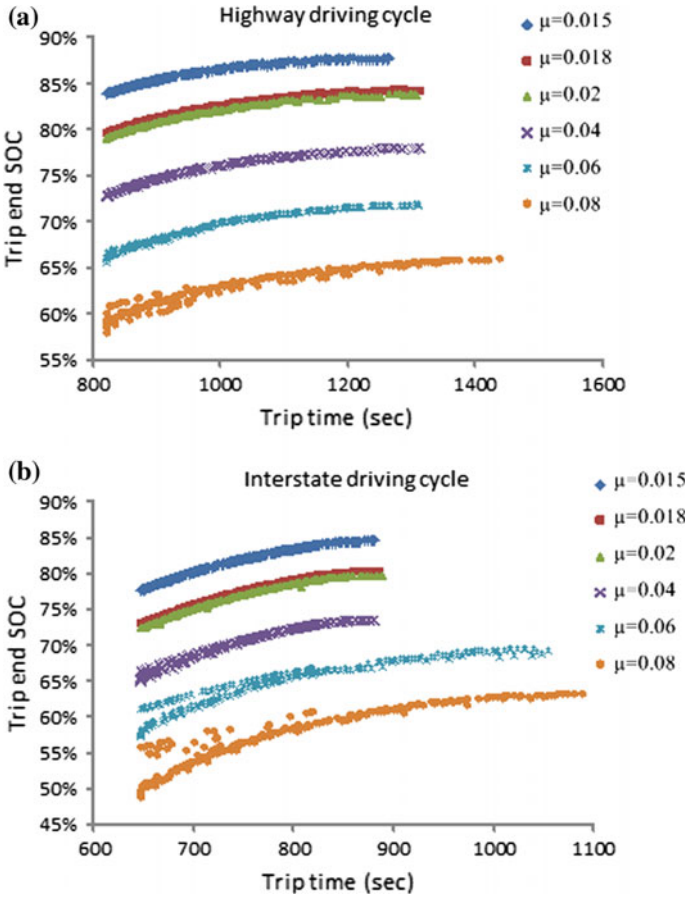


Fig. 27 Effect of road rolling friction coefficient on DAOTP results to cover a trip length of 20 km for **a** Highway and **b** Interstate driving cycles

order to analyze the effect of air density on the DAOTP results, a micro-trip is considered where the ρ_{air} of a part of the route is varied, as shown in Table 11. In order to understand the effect of road elevation only, the road gradient is assumed here constant, though normally in a trip, the elevation effect is always present in association with road gradient. The results for three air density values are presented in Fig. 28a, b for highway and interstate driving cycles, respectively. In both high-speed driving cycles, an improvement in trip end SOC was observed by reducing the air density. The effect of air density was not found to be more significant comparing to other route parameters.

As the elevation increases, the air becomes thinner. Since air is needed for combustion, and due to the unavailability of air at higher altitudes, diesel engine produces a lesser amount of power. For, In high altitude in theory gasoline engines requires less

Table 11 Air density distribution of the micro-trip route

Sr. No.	Distance (km)		Density of air (due to elevation)
	From	To	
1	0	2	1.143
2	2	18	ρ_{air}
3	18	20	1.143

fuel due to lower throttle friction from wider throttle opening. In [103], a study on fuel consumption due to the impact of elevation of a gasoline passenger car was carried out. It was found that the reduction of energy consumption was different in different driving cycles and different elevations. Moreover, the impact of higher elevation was not apparent and there is not always a decrease in fuel consumption, as also observed in Fig. 28a. Interestingly enough, EVs (powered by batteries and electric motors) are nearly immune to power losses due to elevation in this respect. Moreover, as air gets thinner at higher altitude, the drag force reduces, and corresponding the energy consumption.

12 Influence of Micro-trip Complexity on DAOTP Results

A complex micro-trip is defined as a micro-trip consisting of two or more driving cycles. Here, the DAOTP results are investigated for a complex micro-trip that consists of two neighborhood portions and one highway portion. The influence of micro-trip complexity on DAOTP results are analyzed by comparing the trip end SOC versus trip time of optimal solutions from a complex micro-trip with those found in simple micro-trip consisting of either a neighborhood or a highway driving cycle. The positions and lengths of the different portions are shown in Table 12, keeping the total trip length at 20 km. Figure 29 presents the demonstrative DAOTP results for the different complex micro-trips. These are formulated by considering different lengths of the highway portion in order to know how complex micro-trip results vary with composition. In the Fig. 29 legend, the first and last entries correspond to purely neighborhood and purely highway micro-trips, respectively, based on the formulas used in Table 12. From Fig. 29, it was found that as the length of the highway portion increases, the Pareto front width in terms of both objectives is expanded. The trip end SOC is found to be increasing in both directions whereas the trip time always decreasing as the highway portion's length increases. Using the ideal point based reference point technique of a finding preferred solution, the optimal trip end SOC are 0.855346, 0.856298, 0.858101, 0.863259, and 0.865352 for highway driving cycle lengths 0, 5, 10, 15, and 20 km, respectively. There is a polynomial relationship observed between the trip end SOC and the highway portion length. Such a relationship is not applicable to any kind of complex micro-trip. Practically,

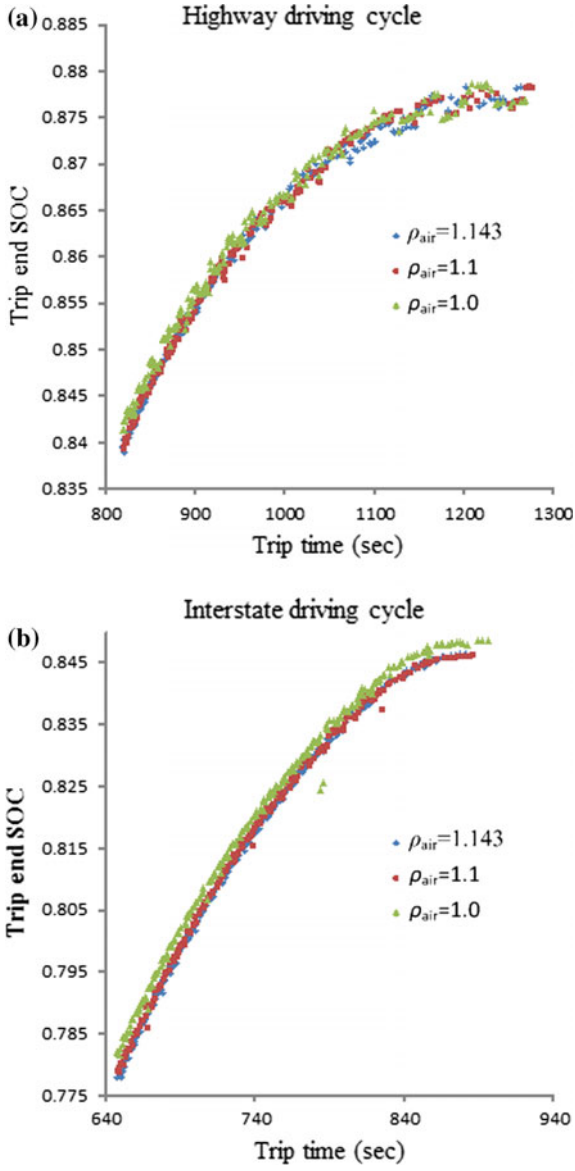


Fig. 28 Effect of air density on DAOTP results to cover a trip length of 20 km in **a** Highway and **b** Interstate driving cycles

depending on the complexity, the types of driving cycle involved, their lengths, and their positions in the micro-trip, this relationship changes.

The DAOTP system presented in this chapter directs the driver not only for proper driving but also it allows the EV to function each components efficiently with their highest efficiency level. Consequently, it lowers the energy consumption and enhances the service life of EV components and safety as well. Thereby, EV

Table 12 Architecture of complex trip

Sr. No.	Micro-trip distance (km)		Driving cycle
	From	To	
1	0	(Total trip length − Length of highway)/2	Neighborhood
2	(Total trip length − Length of highway)/2	Length of highway	Highway
3	(Total trip length − Length of highway)/2	Total trip length	Neighborhood

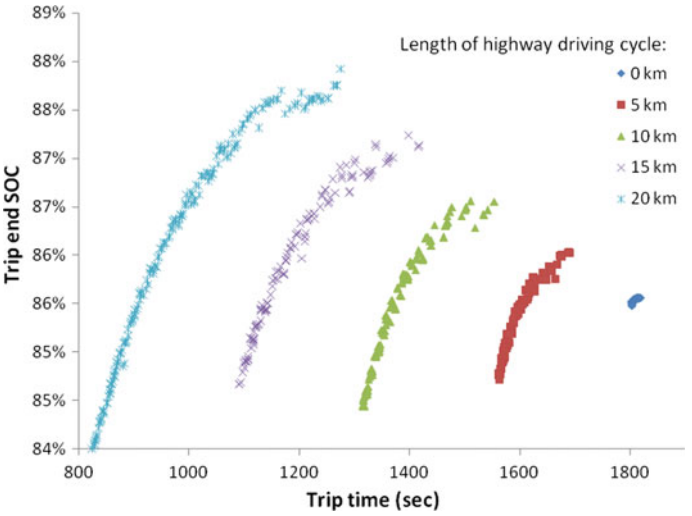


Fig. 29 Variation of trip end SOC with trip time for different complex micro-trips

operating costs becomes reduced. Moreover, the overall driving experience is also improved by adopting the assistances offered by DATOP system. Present driver assistance systems (DAS) may offer useful energy utilization for EVs through finding a low cost path to the destination but they do not assist the driver how to operate the EV components competently. The most significant benefit of DAOTP compared to existing ADAS is that it assists the driver by supporting an optimal driving strategy to follow throughout the entire trip. By doing so, it can guarantee the trip completion without of the possibility of exhausting the battery energy unexpectedly before the trip is completed. Otherwise, it gives a caution prior to the trip or well before the complete battery charge depleted to a take an alternative decision such as to find a charging station. Consequently, a high level of assurance towards successfully performing the trip is built up in driver's mind. Thus, use of DAOTP certainly promotes the popularization of environmentally friendly EVs by growing the reliability of EVs in the public's perception, which is not found so far.

In the present article, jerk that hinders the comfortable journey is assumed to be generated due to changes of acceleration/deceleration rate. Besides this, there are other causes of experiencing jerk, such as driveline dynamics, indiscretion of road, variation of wind speed, etc. that are not considered here. Jerk generated due to such causes may be minimized by taking precautions such as restricting the speed limit, etc. The proposed DAOTP can be integrated with existing DAS such as AHS (Automated Highway Systems), ADAS (Advanced Driver Assistance System), Navigation and Routing devices [102], etc.

13 Summary and Conclusions

A novel and easy to implement driving assistance strategy for EVs, DAOTP is presented here. The proposed DAOTP approach is unique and is superior to existing methods because of its multi-faceted approach. It is model-specific, which means it considers all the characteristics of the vehicle in question and ensures that the hardware such as the electric motor is operated in the most efficient regime. If there are conflicts amongst different hardware regimes, the algorithm chooses a globally optimal solution on the basis of efficiency trade-offs amongst all the hardware involved. This means that the stored energy is used in an efficient manner making DAOTP an energy management strategy as well. The key feature of DAOTP is trip planning: providing the driver with several optimal driving strategies to negotiate a trip so as to optimize various conflicting objectives such as minimum discomfort, energy consumption, and time. Not only does this provides an efficient and comfortable ride, but also instills some measure of confidence in the driver that the EV will be able to reach the destination given the current SOC status. The multi-objective approach gives the driver the ultimate freedom to decide which objectives are more important. This choice is bound to vary from driver to driver and DAOTP takes this into consideration. Finally, DAOTP is an online system, which means not only does it give the driver prior knowledge about the driving strategy for a given trip, it also updates itself based on current road and trip conditions, some of which may be unforeseen. The result is that the driver always has knowledge of the optimal driving strategy for the remainder of the trip, which may be different from the first predicted driving strategy prior to the commencement of the trip.

In the present chapter, the proposed DAOTP architecture for an EV and the role of each component associated with this system are described. The DAOTP executes by solving a multi-objective optimization problem involving minimizations of trip time, energy consumption, and average jerk. The models of these objectives are presented considering the three EV components, battery, electric motor, and vehicle dynamics. A multi-objective genetic algorithm is utilized to solve the multi-objective problem. The DAOTP system is applied first in low-speed driving cycles considering a simple micro-trip. Trip planning, starting from identifying a route based on trip starting point and destination, how to receive the route characteristic data through HMI using GPS and Internet, utilizing these data, producing a set of Pareto-optimal solution after

solving the multi-objective problem, then identifying the set of interesting solutions based on a sorting process using problem-specific information, and finally selecting a preferred solution based on a decision-making technique or driver's choice for implementation, is systematically presented. It has been observed that an optimized driving strategy may substantially reduce EV energy consumption. Analyzing DAOTP results for low-speed driving cycles in different trip lengths clearly indicates its importance for implementation in EV.

The results of the DAOTP system for a micro-trip in high-speed driving cycles were presented later. The optimal driving strategies were determined through a multi-objective optimization of minimizing total trip time, energy consumption, and the average jerk during speed changing. The influence of various route parameters and the initial SOC on the DAOTP results were investigated through the analysis of the trip end SOC with respect to the trip time of the optimal solutions obtained after the optimization process.

Effects of *initial battery SOC*, wind, road gradient and elevation on energy consumption, trip time and jerk are carried out here, and the research findings suggest that based on the naïve predictors that use previous trip performing experience planning of current trips may not be an optimum one. Moreover, naïve predictors are unable to provide energy-driving parameters. The present DAOTP systems considers the route characteristics and driver's preference while providing an optimal driving strategy that leads to an energy-efficient trip with minimum trip time and discomfort. Extensive analysis of the results suggests that the proposed approach is reliable as well as efficient. It is concluded that approaches based on previous experience is not sufficient for EV trip planning, and the present DAOTP system is a substitute to address this necessitate.

Acknowledgements This book chapter is written based on the partial research work carried out under Indo-US fellowship program (Fellowship Letter Reference: IUSSTF Fellowships 2013/14) supported by IUSSTF, India.

References

1. Ahn, C., Peng, H., & Tseng, H. E. (2013). Robust estimation of road frictional coefficient. *IEEE Transactions on Control Systems Technology, 21*(1), 1–13.
2. Ameratunga, S. (2009). Traffic speed zones and road injuries: Speed management is key. *BMJ, 339,* b4743. https://doi.org/10.1136/bmj.b4743.
3. Andre, M. (2004). *Real-world driving cycles for measuring cars pollutant emissions—Part A: The ARTEMIS European driving cycles*. Institut National de Recherche sur les Transports et leur Securite (INRETS), Report INRETS-LTE 0411.
4. Andrea, D. (2010). *Battery management systems for large lithium ion battery packs*. Artech House Publishers.
5. Alfraheed, M., Dröge, A., Klingender, M., Schilberg, D., & Jeschke, S. (2011). Longitudinal and lateral control in automated highway systems: Their Past, present and future. In S. Jeschke, H. Liu, & D. Schilberg (Eds.), *ICIRA 2011, Part II, LNAI 7102* (pp. 589–598). Berlin, Heidelberg: Springer.

6. Baronti, F., Zamboni, W., Femia, N., Saletti, R., & Chow, M. Y. (2013). Parameter identification of Li-Po batteries in electric vehicles: A comparative study. In *IEEE International Symposium on Industrial Electronics*, Taipei, Taiwan, May 28–31, 2013 (pp. 1–7). https://doi.org/10.1109/isie.2013.6563887.
7. Bergenhem, C., Hedin, E., & Skarin, D. (2012). Vehicle-to-vehicle communication for a platooning system. *Procedia—Social and Behavioral Sciences, 48,* 1222–1233.
8. Berry, M. I. (2010). *The effects of driving style and vehicle performance on the real-world fuel consumption of US light-duty vehicles*. Master's Thesis, Massachusetts Institute of Technology. http://web.mit.edu/sloan-auto-lab/research/beforeh2/files/IreneBerry_Thesis_February2010.pdf.
9. Bingham, C., Walsh, C., & Carroll, S. (2012). Impact of driving characteristics on electric vehicle energy consumption and range. *IET Intelligent Transport Systems, 6*(1), 29–35.
10. Bodin, L., & Levy, L. (1994). Visualization in vehicle routing and scheduling problems. *ORSA (Operation Research Society of America) Journal on Computing, 6*(3), 281–288.
11. Branke, J., Deb, K., Dierolf, H., & Osswald, M. (2004). Finding knees in multi-objective optimization. In X. Yao et al. (Eds.), *PPSN 2004*. LNCS 3242 (pp. 722–731). Heidelberg: Springer.
12. Cai, L., Rad, A. B., & Chan, W. L. (2007). A genetic fuzzy controller for vehicle automatic steering control. *IEEE Transactions on Vehicular Technology, 56*(2), 529–543.
13. Carminat Tom Tom Navigation System [CTTNS]. (2012). Retrieved December, 2014, from http://www.tomtom.com/en_gb/products/built-in-car-navigation/carminat-tomtom/.
14. Ceraolo, M., & Pede, G. (2001). Techniques for estimating the residual range of an electric vehicle. *IEEE Transactions on Vehicular Technology, 50*(1), 109–115.
15. Chakraborty, D., & Nandi, A. K. (2016). Finding optimal deceleration with serial regenerative braking of electric vehicle using a multi-objective genetic algorithm. In *IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (IEEE ICPEICES 2016)*, Delhi, India, July 4–6, 2016. https://doi.org/10.1109/icpeices.2016.7853333.
16. Chakraborty, D., & Nandi, A. K. (2016). Finding optimal driving strategy zone of electric vehicle during deceleration with serial regenerative braking. In *7th Power India International Conference (IEEE PIICON 2016)*, Rajasthan, India, November 25–27, 2016.
17. Chale-Gongora, H. G., de Weck, O., Doufene, A., Ishimatsu, T., & Krob, D. (2014). Planning an itinerary for an electric vehicle. In *Proceedings of IEEE Energy Conference (ENERGYCON 2014)*, Cavtat, Dubrovnik, Croatia, May 13–16, 2014 (pp. 1385–1391).
18. Cheng, C., McGordon, A., Poxon, J., Jones, R., & Jennings, P. (2010). A model to investigate the effects of driver behaviour on hybrid vehicle control. In *25th World Battery, Hybrid, and Fuel Cell Electric Vehicle Symposium and Exhibition*, Shenzhen, China, November 5–9, 2010.
19. Chen, M., & Rincón-Mora, G. A. (2006). Accurate electrical battery model capable of predicting runtime and I–V performance. *IEEE Transactions on Energy Conversion, 21*(2), 504–511.
20. Chen, Y. L., Huang, C. H., Kuo, Y. W., & Wang, S. S. (2012). Artificial neural network for predictions of vehicle drivable range and period. In *Proceeding of IEEE International Conference on Vehicular Electronics and Safety*, Istanbul, Turkey, July 24–27, 2012 (pp. 329–333).
21. Cooper, K. R., & Watkins, S. (2007). *The unsteady wind environment of road vehicles, Part One: A review of the on-road turbulent wind environment*. SAE Technical Paper 2007-01-1236.
22. Cortes, P., Larraneta, J., Onieva, L., Garcia, J. M., & Caraballo, M. S. (2001). Genetic algorithm for planning cable telecommunication networks. *Applied Soft Computing, 1*(1), 21–23.
23. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*(2/4), 311–338.
24. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.
25. Demestichas, K., Adamopoulou, E., Kipp, W., Kaminski, T., & Benz, T. (2011). EcoGem: An intelligent advanced driver assistance system for fully electric vehicles. In *18th World Congress on Intelligent Transport Systems*, Orlando, Florida, U.S.A., October 16–20, 2011.

26. Demestichas, K., Adamopoulou, E., Masikos, M., Kipp, W., & Benz, T. (2011). Intelligent advanced driver assistance system for electric vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 5–9, 2011 (pp. 78–82).
27. Dib, W., Serrao, L., & Sciarretta, A. (2011). Optimal control to minimize trip time and energy consumption in electric vehicles. In IEEE *Conference on Vehicle Power and Propulsion Conference (VPPC)*, Chicago, IL, September 6–9, 2011 (pp. 1–8). https://doi.org/10.1109/vppc.2011.6043133.
28. Dovgan, E., Tušar, T., Javorski, M., & Filipič, B. (2012). Discovering comfortable driving strategies using simulation-based multiobjective optimization. *Informatica (Slovenia), 36*(3), 319–326.
29. Du, L. L., Li, B., & Zhang, H. D. (2013). Estimation on state of charge of power battery based on the grey neural network model. *Applied Mechanics and Materials, 427–429,* 1158–1162.
30. Eisner, J., Funke, S., & Storandt, S. (2011). Optimal route planning for electric vehicles in large networks. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, Hyatt Regency San Francisco, August 7–11, 2011 (pp. 1108–1113).
31. Fambro, D. B., Fitzpatrick, K., & Koppa, R. J. (1997). *Determination of stopping sight distance*. NCHRP (National Cooperative Highway Research Program) Report 400, National Academy Press, Washington D.C.
32. Feng, L., Ge, S., & Liu, H. (2012). Electric vehicle charging station planning based on weighted Voronoi diagram. In *Proceedings of Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific*, March 27–29, 2012 (pp. 1–5). IEEE.
33. Fiat Eco: Drive, Eco-driving uncovered: the benefits and challenges of eco-driving based on the first study using real journey data (2010). Retrieved February 10, 2015, from http://www.lowcvp.org.uk/assets/reports/Fiat_Eco-Driving%20Uncovered.pdf.
34. Franke, T., Neumann, I., Bühler, F., Cocron, P., & Krems, J. F. (2012). Experiencing range in an electric vehicle: Understanding psychological barriers. *Applied Psychology: An International Review, 61*(3), 368–391.
35. Fujii, H., Hayashi, O., & Nakagata, N. (1996). Experimental research on intervehicle communication using infrared rays. In *Proceedings of IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, September 19–20, 1996 (pp. 266–271).
36. Gantt, L. R., Alley, R. J., & Nelson, D. J. (2011). Battery sizing as a function of powertrain component efficiencies for various drive cycles. In *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2011)*, August 28–31, 2011, Washington, DC.
37. Gao, Y., Chen, L., & Ehsani, M. (1999). *Investigation of the effectiveness of regenerative braking for EV and HEV*. SAE Technical Paper 1999-01-2910. https://doi.org/10.4271/1999-01-2910.
38. Garmin ecoRoute HD software [GEHDS]. (2010). Retrieved January, 2015, from http://www.garmin.com.
39. Ghoseiri, K., & Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing, 10*(4), 1096–1107.
40. Gillespie, T. D. (2003). *Fundamentals of vehicle dynamics*. Society of Automotive Engineers, Inc. 400 Commonwealth Drive, Warrendale, PA 15096-0001.
41. Global Positioning System, Volume 1—Theory and Applications [GPSTA]. (1996). In B. Parkinson, J. Spilker, P. Axelrad & P. Enge (Eds.), Published by the American Institute of Aeronautics and Astronautics, Inc. 370 L'Enfant Promenade, SW, Washington, DC 20024-2518. ISBN 978-1-56347-106-3.
42. Grant, P. R., & Haycock, B. (2008). Effect of jerk and acceleration on the perception of motion strength. *Journal of Aircraft, 45*(4), 1190–1197.
43. Guo, J., Wang, J., & Cao, B. (2009). Regenerative braking strategy for electric vehicles. In *IEEE Conference on Intelligent Vehicles Symposium*, June 3–5, 2009, Xi'an, China (pp. 864–868).

44. Haan, P. D., & Keller, M. (2001). *Real-world driving cycles for emission measurement: ARTEMIS and Swiss cycles*. Final Report.
45. Harding, J., Powell, R. G., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., et al. (2014). *Vehicle-to-vehicle communications: Readiness of V2V technology for application*. Washington, DC: National Highway Traffic Safety Administration, Report No. DOT HS 812 014.
46. Hansen, T., & Wang, C.-J. (2005). Support vector based battery state of charge estimator. *Journal of Power Sources, 141*(2), 351–358.
47. Heissing, B., & Ersoy, M. (2011). *Chassis handbook fundamentals, driving dynamics, components, mechatronics, perspectives*. Wiesbaden: Vieweg + Teubner Verlag | Springer Fachmedien Wiesbaden GmbH.
48. ID4EV project [ID4EVP]. (2014). Retrieved December, 2014, from http://www.id4ev.eu.
49. Imanishi, H., Takada, Y., & Wakisaka, T. (2002). An acceleration control algorithm for an electric motor driven vehicle in consideration of the reduction of energy consumption. *Nippon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C, 68*(5), 1512–1517.
50. Ishimatsu, T., de Weck, O., Hoffman, J., Ohkami, Y., & Shishko, R. (2013). A generalized multi-commodity network flow model for space exploration logistics. In *AIAA SPACE 2013 Conference & Exposition*, San Diego, California, USA, September 10–12, 2013.
51. Karmakar, M., & Nandi, A. K. (2016). Trip planning for electric vehicle through optimal driving using genetic algorithm. In *International Conference on Power Electronics, Intelligent Control and Energy Systems (IEEE ICPEICES 2016)*, July 4–6, 2016, Delhi, India.
52. Karmakar, M., & Nandi, A. K. (2016). Driving assistance for energy management in electric vehicle. In *International Conference on Power Electronics, Drives and Energy Systems (IEEE PEDES 2016)*, December 14–17, 2016, Trivandrum, India.
53. Kato, S., Tsugawa, S., Tokuda, K., Matsui, T., & Fujii, H. (2002). Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *IEEE Transactions on Intelligent Transportation Systems, 3*(3), 155–161.
54. Kim, E., Lee, J., & Shin, K. G. (2013). Real-time prediction of battery power requirements for electric vehicles. In *Proceeding of ACM/IEEE International Conference on Cyber-Physical Systems*, Philadelphia, PA, USA, April 8–11, 2013 (pp. 11–20).
55. Knowles, M., Scott, H., & Baglee, D. (2012). The effect of driving style on electric vehicle performance, economy, and perception. *Int. J. of Hybrid Vehicles, 4*(3), 228–247.
56. Kockelman, K. (2006). *Safety impacts and other implications of raised speed limits on high-speed roads*. NCHRP (National Cooperative Highway Research Program) Web-Only Document 90 (Project 17-23): Contractor's Final Report, Transportation Research Board of the National Academies.
57. Kunze, R., Ramakers, R., Henning, K., & Jeschke, S. (2009). Organization and operation of electronically coupled truck platoons on German motorways. In M. Xie, Y. Xiong, C. Xiong, H. Liu, & Z. Hu (Eds.), *ICIRA 2009*. LNCS-5928. Heidelberg: Springer (pp. 135–146).
58. Larminie, J., & Lowry, J. (2012). *Electric vehicle technology explained* (2nd ed.). Chichester, West Sussex, UK: Wiley.
59. Levin, M. W., Duell, M., & Waller, S. T. (2014). *Effect of road grade on networkwide vehicle energy consumption and ecorouting*. Transportation Research Record: Journal of the Transportation Research Board, No. 2427, Transportation Research Board of the National Academies, Washington, pp. 26–33.
60. Lipp, T., & Boyd, S. (2014). Minimum-time speed optimisation over a fixed path. *International Journal of Control, 87*(6), 1297–1311.
61. Liu, G., Ouyang, M., Lu, L., Li, J., & Han, X. (2015). Online estimation of lithium-ion battery remaining discharge capacity through differential voltage analysis. *Journal of Power Sources, 274,* 971–989.
62. Lu, D., & Ouyang, M. (2014). Torque-based optimal acceleration control for electric vehicle. *Chinese Journal of Mechanical Engineering (English Edition), 27*(2), 319–330.

63. Mehar S., & Rémy, G. (2013). EV-planning: Electric vehicle itinerary planning. In *Proceeding of IEEE International Conference on Smart Communications in Network Technologies (SaCoNeT)*, June 17–19, 2013, Paris (pp. 1–5). https://doi.org/10.1109/saconet.2013.6654583.
64. Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Boston: Kluwer.
65. Moura, A., Rijo, R., Silva, P., & Crespo, S. (2010). A multi-objective genetic algorithm applied to autonomous underwater vehicles for sewage outfall plume dispersion observations. *Applied Soft Computing, 10*(4), 1119–1126.
66. Nan, Z., & Gouzhong, M. A. (2007). Research on the theory and application of road speed limit. In *Proceedings of the first international Conference on Transportation Engineering*, Southwest Jiaotong University, Chengdu, China, July 22–24, 2007 (pp. 2699–2704).
67. Nandi, A. K., Deb, K., Ganguly, S., & Datta, S. (2012). Investigating the role of metallic fillers in particulate reinforced flexible mould material composites using evolutionary algorithms. *Applied Soft Computing, 12,* 28–39.
68. Nandi, A. K., Chakraborty, D., & Vaz, W. (2015). Design of a comfortable optimal driving strategy for electric vehicles using multi-objective optimization. *Journal of Power Sources, 283*(2015), 1–18.
69. NAVTEQ ADAS for Hyundai [NAH]. (2012). Retrieved December, 2014, from http://press.navteq.com/NAVTEQ-Map-Powers-New-Navigation-System-For-Hyundai-Solaris.
70. NRC. (2006). *Tires and passenger vehicle fuel economy—Informing consumers, improving performance*. Transportation Research Board. Washington D.C., USA: National Research Council of the National Academies (Special report 286).
71. Oduguwa, V., Tiwari, A., & Roy, R. (2005). Evolutionary computing in manufacturing industry: An overview of recent applications. *Applied Soft Computing, 5*(3), 281–299.
72. Onieva, E., Naranjo, J. E., Milanes, V., Alonso, J., Garcia, R., & Perez, J. (2011). Automatic lateral control for unmanned vehicles via genetic algorithms. *Applied Soft Computing, 11*(1), 1303–1309.
73. Orazem, M. E., & Tribollet, B. (2011). *Electrochemical impedance spectroscopy*. Wiley, Hoboken.
74. Ostensen, A. G. (1997). Effects of raising and lowering speed limits on selected roadway sections. U.S. Department of Transportation, Federal Highway Administration, Publication No. FHWA-RD-97-084.
75. Ramakers, R., Henning, K., Gies, S., Abel, D., & Max, H. (2009). Electronically coupled truck platoons on German highways. In *IEEE International Conference on Systems, Man and Cybernetics (SMC 2009)* (pp. S.2409–S.2414).
76. Sachenbacher, M., Leucker, M., Artmeier, A., & Haselmayr, J. (2011). Efficient energy-optimal routing for electric vehicles. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, Hyatt Regency San Francisco, August 7–11, 2011 (pp. 1402–1407).
77. Sandro, B. (2011). ID4EV—Intelligent dynamics for fully electric vehicles Barcelona motor show. Barcelona, May 18, 2011.
78. Sengupta, A., & Upadhyay, A. (2009). Locating the critical failure surface in a slope stability analysis by genetic algorithm. *Applied Soft Computing, 9*(1), 387–392.
79. Shankar, R., & Marco, J. (2013). Method for estimating the energy consumption of electric vehicles and plug-in hybrid electric vehicles under real-world driving conditions. *IET Intelligent Transport Systems, 7*(1), 138–150.
80. Shen, W. X., Chau, K. T., Chan, C. C., & Lo, E. W. C. (2005). Neural network-based residual capacity indicator for nickel-metal hydride batteries in electric vehicles. *IEEE Transactions on Vehicular Technology, 54*(5), 1705–1712.
81. Shladover, S. (2008). AHS research at the California PATH program and future AHS research needs. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES 2008)* (pp. S.4–S.5).
82. Shortt, A., & O'Malley, M. (2009). Impact of optimal charging of electric vehicles on future generation portfolios. In *Proceeding of IEEE PES/IAS Conference on Sustainable Alternative Energy (SAE)*, Valencia, Spain, September 28–30, 2009. https://doi.org/10.1109/sae.2009.5534861.

83. Smith, K. A., Rahn, C. D., & Wang, C. Y. (2010). Model-based electrochemical estimation and constraint management for pulse operation of lithium ion batteries. *IEEE Transactions on Control Systems Technology, 18*(3), 654–663.
84. Sundstrom, O., & Binding, C. (2010). Planning electric-drive vehicle charging under constrained grid conditions. In *Proceeding of 2010 International Conference on Power System Technology "POWERCON2010"*, Hangzhou, Zhejiang Province, China, October 24–28, 2010, published by IEEE (pp. 1–6).
85. Szumanowski, A., & Chang, Y. (2008). Battery management system based on battery nonlinear dynamics modeling. *IEEE Transactions on Vehicular Technology, 57*(3), 1425–1432.
86. The Freightliner Predictive Cruise Control [FPCC]. (2015). Retrieved January, 2015, from http://www.freightlinertrucks.com/Multimedia/MediaLibrary/Trucks/?AssetId=15990456-b11e-4966-a5be-c70e22cafcca.
87. Tsugawa, S. (1993). Super smart vehicle system: Future intelligent driving and the measures for the materialization. In *Proceeding of Surface Transportation: Mobility, Technology, and Society. Proceedings of the IVHS AMERICA 1993 Annual Meeting*, Washington, D.C., April 14–17, 1993 (pp 192–198).
88. Van Mierlo, J., Maggetto, G., Van De Burgwal, E., & Gense, R. (2004). Driving style and traffic measures—Influence on vehicle emissions and fuel consumption. *Proceedings of Institute of Mechanical Engineers Part D: Journal of Automobile Engineering, 218*(1), 43–50.
89. Vaz, W., Nandi, A. K., Landers, R. G., & Koylu, U. O. (2015). Electric vehicle range prediction for constant speed trip using multiobjective optimization. *Journal of Power Sources, 275,* 435–446.
90. Vaz, W., Nandi, A. K., & Koylu, U. O. (2016). A multi-objective approach to find optimal electric vehicle acceleration: simultaneous minimization of acceleration duration and energy consumption. *IEEE Transactions on Vehicular Technology, 65*(6), 4633–4644.
91. Vehicle Platooning and Automated Highways [VPAH]. (2013). Retrieved December 05, 2014, from http://www.path.berkeley.edu/PATH/Publications/Media/FactSheet/VPlatooning.pdf.
92. Veldhuizen, D. A. V., & Lamont, G. B. (1998). Evolutionary computation and convergence to a pareto front. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, & R. Riolo (Eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference* (pp. 22–25), San Francisco, CA.
93. Venhovens, P. J. Th., Bernasch, J. H., Löwenau, J. P., Rieker, H. G., & Schraut, M. (1999). *The application of advanced vehicle navigation in BMW driver assistance systems*. SAE Technical Paper Series, 1999-01-0490.
94. Vexia's ecoNav solution [VES] (2012). Retrieved December, 2014, from http://www.vexia.es/en/econav_technology.htm.
95. Wang, Y., Jiang, J., & Mu, T. (2013). Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing. *IEEE Transaction on Intelligent Transportation Systems, 14*(3), 1331–1345.
96. Wareham-Norfolk, K. (2011). Range limits of electric vehicles. *Engineering & Technology, 6*(3), 22–23.
97. Watkins, S., 7 Saunders, J. (1998). *A review of the wind conditions experienced by a moving vehicle*. SAE Technical Paper 981182.
98. Watkins, S., & Cooper, K. (2007). *The unsteady wind environment of road vehicles, Part Two: Effects on vehicle development and simulation of turbulence*. SAE Technical Paper 2007-01-1237.
99. Winner, H., Witte, S., Uhler, W., & Lichtenberg, B. (2006). *Adaptive cruise control system aspects and development trends*. SAE Technical Paper Series 961010.
100. Xu, G., Li, W., Xu, K., & Song, Z. (2011). An intelligent regenerative braking strategy for electric vehicles. *Energies, 4,* 1461–1477.
101. Yao, E. J., Wang, M. Y., Song, Y. Y., & Zuo, T. (2013). Estimating the cruising range of electric vehicle based on instantaneous speed and acceleration. *Applied Mechanics and Materials, 361–363,* 2104–2108.

102. Ygnace, J. L., Al-Deek, H. M, & Lavallee, P. (1990). *Vehicle navigation and route guidance technologies: Push and pull factors assessment*. Path (Program on Advanced Technology for the Highway) Research Report, Institute of Transportation Studies, University of California Berkeley, UCB-ITS-PRR-90-2.
103. Zervas, E. (2011). Impact of altitude on fuel consumption of a gasoline passenger car. *Fuel, 90,* 2340–2342.
104. Zhang, X., & Rice, J. A. (2003). Short-term travel time prediction. *Transportation Research Part C: Emerging Technologies, 11*(3–4), 187–210.
105. [Switzerland. TIK-Schriftenreihe Nr. 30, Diss ETH No. 13398, Shaker Verlag, Aachen, Germany].
106. Zlocki, A., Huang, Q., Ghaouty-El, M., Eckstein, L., & Grundmann, H. G. (2013). Intelligent functionalities for fully electric vehicles. In SAE-China and FISITA (Eds.), *Proceedings of the FISITA 2012 World Automotive Congress*. Lecture Notes in Electrical Engineering 200 (pp. 267–274). Berlin, Heidelberg: Springer.

Evolutionary Computation in Blast Furnace Iron Making

Bashista Kumar Mahanta and Nirupam Chakraboti

Abstract Various methods and models are used to optimize the different aspects of blast furnace iron-making process involving the quality of hot metal, productivity, and the cost of production. Analytical models which are used to solve such problems are often not adequate to obtain the optimized results. Nowadays data-driven models are used effectively for this purpose. In this chapter, various soft computing techniques, with a special emphasis on evolutionary computation methods are presented to elaborate their implementation in different areas of blast furnace iron making. The input–output data models and simulation results are discussed with respect to various aspects of ferrous production metallurgy like sintering process, gas scheduling, prediction and process control of hot metal, temperature prediction of hot metal, silicon content in the hot metal, burden and gas distribution, carbon dioxide emission, performance of rotary kiln, coal injection, biomass injection, and many objective optimization problems, etc. The results and comparison between various models are thoroughly analyzed and a discussion regarding these data-driven models and their influence in the blast furnace iron-making process is presented in a comprehensive manner.

Keywords Blast furnace · Evolutionary computation · Genetic algorithm
Genetic programming · Neural network · Burden distribution
Industrial blast furnace · Evolutionary computation · Optimization · Sintering
Gas scheduling · Viscosity · Pulverized coal · Oxygen enrichment · Rotary kiln

B. K. Mahanta · N. Chakraboti (✉)
Department of Metallurgical and Materials Engineering,
Indian Institute of Technology, Kharagpur, India
e-mail: nchakrab@gmail.com

B. K. Mahanta
e-mail: bashista_mahanta@yahoo.co.in

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management
and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_8

1 Background

The iron and steel industry brought revolution in world's economy after the Second World War. The industrial revolution started in the mid of the nineteenth century and the per capita consumption of steel is one of the major indexes of the industrial achievement till date. About 5.63% of earth crust is actually Fe [1]. The most of it are the two oxides, hematite (Fe_2O_3), and magnetite (Fe_3O_4). Iron also exists as hydroxide in the crust. Pure iron is very soft and very prone to oxidation; therefore, it needs to be alloyed with other elements to enhance its properties as per the desired requirements. Traditionally iron making is conducted in the iron blast furnaces, where the carbon monoxide is formed through a gasification process [2] and is used as a reducing agent for the extraction of metallic iron from these iron. The carbon-saturated iron, which contains several other alloying elements that come out of the blast furnace is known as the hot metal. By controlling alloying elements, carbon being the primary one, through an oxidation process, the steel making is done subsequently [2]. C, Mn, Si, P, and S are common elements present in steel. For special engineering applications many other alloying elements, e.g., B, Cr, Ni, Pb, Mo, Al, Ti, Cu, Co, V, and W are added. World crude steel production reached 1630 million tons (Mt) for the year 2016 [3]. World average steel use per capita has been increasing steadily. In the last decade, it has been increased from 184.2 kg in 2005 to 246.8 kg in 2014 [3]. In India, 2014, per capita consumption of steel was only 59.4 kg as against the world average of 216.6 kg [4]. Using of iron and steel in various applications throughout the world is increasing significantly. The steel market is highly competitive at the moment, efficient production and lower price remain the key to success for any steel company. On both counts, rigorous optimization of the existing processes is absolutely necessary. Currently, iron making is an extremely competitive industrial process and in order to stay afloat, the steel companies require improvements in areas like quality increment, cost reduction, efficient productivity, and process optimization. These lead to some challenging problems requiring advanced optimization knowledge. To carry out such optimization tasks in the traditional blast furnaces (BF) is difficult as the reactor is both cumbersome and complicated. The chemical processes and transport phenomena which occur inside the furnace also warrant some very advanced domain knowledge. Using analytical models, justice often cannot be done to such complexity [5]. As an effective alternate strategy, data-driven models using the actual plant data are now quite successfully used to study and optimize blast furnaces [5]. Through this approach, it is actually possible to solve some blast furnace optimization problems that are intractable through the conventional strategies based upon thermodynamic or kinetic models. Now, the advent of the nature-inspired computing [6] has contributed significantly in this area. In recent times, nonlinear and noisy data from the blast furnace [7] are modeled and optimized through different types of evolutionary algorithms. Few examples include, genetic programming (GP), artificial neural networks (ANN) , support vector method (SVM), etc. This chapter aims at providing a comprehensive critical review of such work.

This chapter is organized as follows. Iron-making process is explained in the next section. After that a brief description of blast furnace with various units in the iron-making processes are discussed. Next, various optimization processes associated with iron making are explained. Subsequently, it is followed by a brief application of evolutionary computation methods contributing to different components of blast furnace iron making, such as burden distribution, production rate, gas flow and gas flow rate, tuyere velocity, heat loss, coke rate, carbon rate, chemical composition, chemical reaction, etc. The prime intention of this chapter is to show the real picture of data-driven models applied in these areas and to examine how it is actually helping to solve the real-world problems related to iron and steel making industry. Finally, conclusions are drawn with respect to the studies of various aspects of iron making and some future aspects are highlighted at the end.

2 Iron-Making Process

2.1 Development in Iron Making

The iron and steel industry started to grow in the late nineteenth century. There were significant developments that took place in the field of design and engineering. In this period, there was no change in the process technology, the reasons behind were as follows [2]:

- Knowledge regarding physiochemical and metallurgical aspects of blast furnace reaction was limited. The application of thermodynamics toward blast furnace started around 1930. The kinetics behavior and reaction involved in transport phenomenon were in elementary stages. It behaves like a black box in which internal state is not known; only input and output information carried out the operation.
- The physical and chemical study took place around 1950. In Japan, a running blast furnace was rapidly solidified by the help of liquid nitrogen. Then, the furnace was cut into various sections and samples were collected from various zones for physical and chemical analysis.

The basic changes that took place up to 1950 led to various stages which are as follows [2]:

- More powerful blowing machine,
- Higher blast temperature during operation,
- Bigger size furnaces,
- Better charging equipment and necessary machinery,
- Improve raw material storage,
- Use better refractory and improve the design of the furnace lining, and
- Quick measurements of input and output temperatures.

Since 1950, significant changes have taken place in the field of blast furnace iron making, producing tremendous effect in the field of world economy. Major improvements happened due to better understanding of the science and inner working process of the furnace, which was initially difficult for the metallurgists to figure out.

The major developments after 1950 are as follows [2]:

- Burdens are used in the form of sinter and pallets,
- Better quality coke with proper size was maintained,
- Injection of liquid fuel, gas, and pulverized solid hydrocarbons through the tuyeres,
- Large furnace volume,
- Higher blast furnace temperature inside furnace,
- High top pressure,
- Oxygen enrichment of the air blast,
- Better burden distribution, and
- Computer added process control.

2.2 Modern Iron Making

The reduction of iron ore is carried out through two alternative routes. First one is smelting route in which, first, the ore is reduced to Fe, and then melted in the furnace and tapped. Subsequently, the hot metal is delivered to the steel-making shop. Alternately, without using the blast furnace, it is possible to carry out the iron reduction directly at a low temperature using reducing gases. This is known as the direct reduction process and the iron produced this way is called directly reduced iron (DRI). Even 500 years after its initiation, 70% of the iron steel industries follow the blast furnace route. Blast furnace has grown considerably in size during the twentieth century. In earlier days', furnace had a hearth diameter of 3–4 m and were producing 100,000 tons hot metal per year but nowadays the hearth diameter of the furnace is kept between 14 and 15 m and are producing 4–5 MT per year [8]. The size of the blast furnace is often expressed as the working volume. The working volume is the volume of the blast furnace that is available for the process. The production of steel in 2003 was 669 million tons. This was increased to 1630 million tons [3] in 2016. The most important reason for the success of increase in production of steel is due to the improvement in the process of iron making. The efficiency, productivity, and scalability are improving continuously in the iron-making field. There are about 21 furnaces in the world having volume more than 5000 mt^3. Some larger scale production units in the world are Baosteel (China, producing 37 million metric tons per year), POSCO (South Korea, producing 35.4 million metric tons per year), Nippon steel (Japan, producing 35 million metric tons per year), and TATA steel (India, producing 23.2 million metric tons per year) [8]. To increase the production rate new alternative processes like COREX, FINEX, and Hismelt are being used to produce liquid iron from the ore. Remaining 30% of the steel production mostly uses recycle steel scrap or DRI as feed material. Midrex, HYL, and rotary kiln are

the most important DRI process to produce steel. Globally, the DRI plants produced around 7257 million tons of iron as reported by Midrex Technology for the year 2015 [9].

3 Process Units

3.1 Blast Furnace

Iron blast furnace is a vertical shaft, which is used to melt the iron ore and to produce hot metal by heat exchange and chemical reaction. The burden charge consisting of iron oxide, flux and coke and it provides through the throat from the top of the furnace. The iron oxide is reduced to hot metal by receiving heat from the ascending hot blast (oxygen, nitrogen) and injection (oil/pulverized coal) supply from the bosh region. Below the throat, there is an increase in diameter to some extent known as the shaft after which the furnace maintains a constant diameter cylindrical section known as the belly. Just below it, there is a decrease in diameter known as bosh region, where the blast enters the furnace through the surrounding tuyeres. A tuyere is a cooled copper conical nozzle having 12 numbers of tuyeres in small furnaces and up to 42 in bigger furnaces. The preheated air (1000–1300 °C) is blown through it. The oxygen of the blast burns coke to CO and several combustion zones, one in front of each tuyere, exist in the tuyere zone. The hearth zone is the bottom-most region, where the hot metal and slag are stored. The inside volume of the blast furnace is divided into different zones according to the physical significance and the chemical reactions occurring inside the furnace The top portion of the furnace is named as lumpy zone; burden enters and stacks layer by layer as solid. The reduction of the iron oxide takes place in the cohesive zone. Hematite (Fe_2O_3) is converted into magnetite (Fe_3O_4), subsequently to wüstite (Fe_xO) by the ascending reducing gas containing carbon monoxide, producing carbon dioxide as reaction product. The iron ore begins to soften and melt in the cohesive zone and at the lower end iron melts and disperses through the solid coke layers. Below the cohesive zone the raceway starts: here, the coke is burnt to carbon monoxide, which consists of oxygen and some inert nitrogen. At the center of the bosh lies closely a packed stack of unreacted coke known as the dead-man zone. The last zone is known as hearth zone where slag floats on the surface of the hot metal and is separated away. A schematic diagram of a blast furnace is shown in Fig. 1.

3.2 Process Description

In the upper part of a hearth about half meter below the edge of the bosh, opening spaces are provided throughout the periphery of the hearth through which heated air

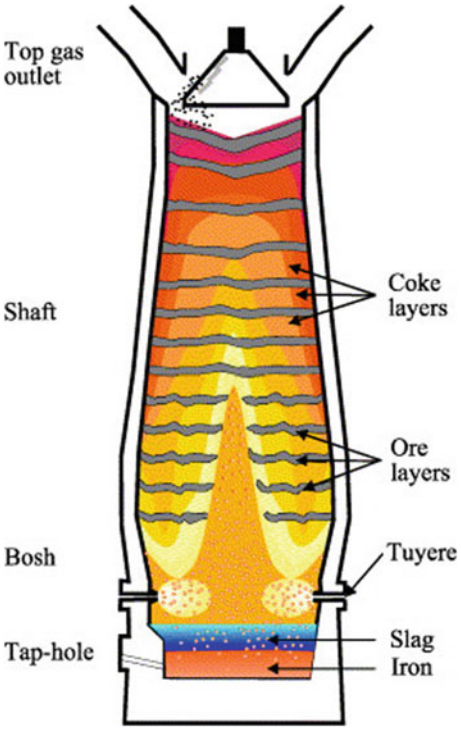


Fig. 1 A schematic representation of the various regions of an iron blast furnace [7]

is blown by means of the tuyeres. The hot blast gasifies coke and other carbon-based materials injected through the tuyere, which are usually coal, natural gas, and oil. In this process, the oxygen in the blast is transferred into gaseous carbon monoxide. Coke (C_k) which together with other raw materials descends from the throat into the hearth and comes in contact with the hot blast in the oxidation zone in front of individual tuyeres, where intensive combustion of the coke carbon takes place.

$$C_k + O_2 = CO_2$$

As there is surplus carbon in the hearth, carbon dioxide is reduced to carbon monoxide. Carbon monoxide, thus, always escapes from the oxidation zones as a product of coke combustion.

$$C_k + CO_2 = 2CO$$

In areas with lack of oxygen, the carbon of coke is burnt imperfectly to CO.

$$C_k + 0.5O_2 = CO$$

The raw materials iron ore, coke, and limestone are charged into the furnace from the top through the throat. The iron oxide (ore) Fe_2O_3 is reduced mainly by the ascending CO. The reduction of the iron ore takes place and is converted into final hot metal.

$$Fe_2O_3 \Rightarrow Fe_3O_4 \Rightarrow Fe_xO \Rightarrow Fe$$

The layered structure of raw materials forms inside the shaft, which consists of ore, flux, and coke, which are melted, in the lower part of the furnace and burden layers gradually move downward. The burden distribution plays an important role for smooth operation of the furnace [10]. The charging equipment manipulates the burden distribution. Modern equipment gives better option for the distribution of burden effectively.

4 DRI

Direct reduction consists of family of processes where iron ore reduction with negligible melting takes place using solid or gaseous reducing agents. Natural gas or noncoking coal is used as reductant as well as primary source of energy. The product of this process is a solid one and is used in the steel making along with scrap.

4.1 Rotary Kiln

In this method, the reduction process requires iron ore and coal, which are transported through a moving bed inside a rotary kiln. It is an efficient heat exchanger with a complex structure, usually 1.5–3 m in radius and between 50 and 225 m in length. It can handle a charge up to 3000 tons. The kiln acts as a solid and fluid transporter, where it effectively mixes the reacting to facilitate the reduction of iron oxides. Combustion of fixed carbon and the release of the volatile matter from the noncoking coal also takes place in this reactor [11].

4.2 Process Description

In all coal-based DR processes using rotary kilns, the inlet end is used to feed sized lump iron ore and relatively coarse fraction of noncoking coal. Coal not only acts as a reducing agent but it also supplies heat required for maintaining the temperature profile of the charge inside a kiln. Inside the rotary kiln, iron oxide is reduced to metallic iron at a temperature close to 1000 °C, the reducing gases are generated to control the combustion of coal. The product discharges into a rotary cooler, where

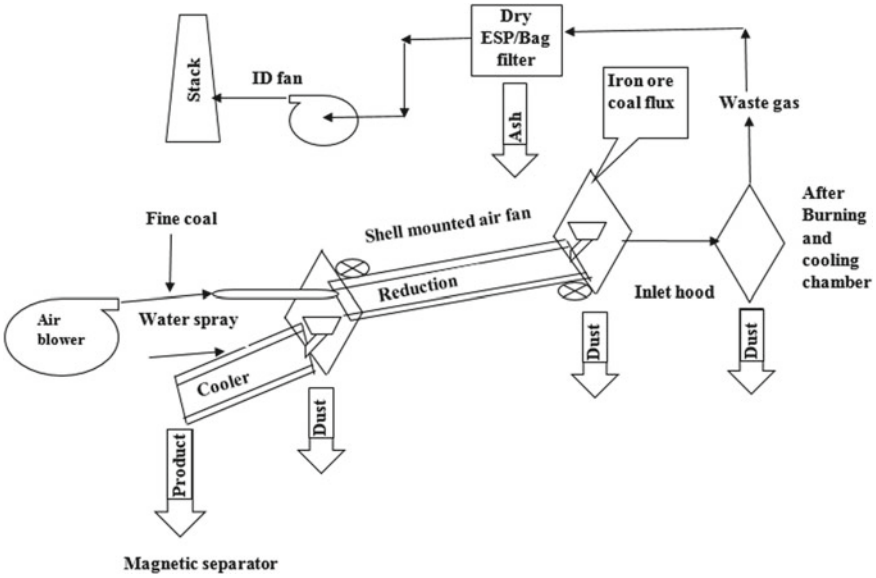


Fig. 2 Rotary kiln line diagram

continuous spraying of water over the rotary shell is provided to maintain the temperature of the DRI less than 100 °C. The discharged product from the rotary cooler is passed through a mechanical screen and magnetic separators to ensure the product specifications and then stored in the storage bins. The Fe_2O_3 used in this process undergoes a stage-wise reduction and gets converted into Fe_3O_4, followed by Fe_xO, and finally the metallic iron forms. The process is schematically shown in Fig. 2.

5 Process Optimization

The structure of the blast furnace is quite cumbersome and the process is very complicated and complex due to existing chemical reaction and transport phenomena. This complexity of the system is a major bottleneck that often hindered further improvements of the process. The potential improvements of the process and their effect on overall performance can be analyzed by the help of mathematical and data-driven modeling. Today, the global demand of high-quality steel leads to increase in the material cost and brings in various challenges pertinent to steel quality. To address it, optimization now plays an important role in the iron-making process, which helps to achieve an efficient production of good quality steel. In earlier stage, linear programming was used for optimizing the production sequences in an integrated steel plant [12] and several other improvised strategies were followed [13]. Until 1950, very little was known about the condition inside a blast furnace during an operation.

The major breakthrough took place in 1950 when a group of Japanese researchers did dissection of a running furnace after it has been solidified with the help of liquid nitrogen passing through the tuyeres, which led to the deeper understanding regarding physical and chemical processes inside the furnace [9]. Initially, this helped to develop a large number of mathematical models [14] and later on the focus shifted toward the data-driven models [5].

The blast furnace process was treated as an entity in the early days and the initial mathematical models are known as the zero-dimensional models. There the blast furnace was split into different regions based upon the distinct reactions and phenomena taking place in them. The thermodynamic relations were used in the calculation of thermal and chemical reactions [15]. One-dimensional models were introduced subsequently to predict the variation of gas and solid temperatures and also the chemical composition of the charge along the height of the furnace [16]. In 1-D models, the furnace was discretized into finite sections. For the calculation of temperature and composition in these sections, the heat and mass transfer equations and the standard equations for the rates of chemical reactions were used in the model. This concept was later extended in two-dimensional [17–20] and three-dimensional models using computational fluid dynamics (CFD) and the associated strategies [21–23]. As indicated before, the process chemistry and transport phenomena of the blast furnace are often very complex, therefore, using such analytical models, the result one obtained is often just approximate and the interpretation of data is seldom at the expected level. In the current approaches, determining the accurate values of the model parameters using the data and mimicking the physics of the process, also using it in tandem with their thermodynamic and transport equations lead to a more reliable strategy.

In early stage, numerous soft computing techniques such as fuzzy logics and artificial neural network were used to describe the iron-making process. These models provided a useful way to find out the decent results and were computationally fit to satisfy the desired requirements, and furthermore they could address the nonlinear problems [24]. An accurate model is, however, always crucial for understanding the furnace operation. In the latter part of the twentieth century, the computational hardware grew tremendously and became readily available in the research and development domain, and computationally demanding optimization techniques like evolutionary algorithms started being increasingly used to solve various industrial problems. Evolutionary algorithms provide a distinct advantage over many other optimization techniques due to their simplicity and versatility. The problem behind gradient-based method [25, 26] is that by contrast to evolutionary algorithms they do not work, where non-differentiable objective functions exist. Evolutionary approaches may also be used very effectively to solve multi-objective optimization problems related to the domain of iron making. Recently, genetic algorithms (GAs) have been applied in different areas related to the various activities of blast furnace [5]. GAs are flexible enough to accommodate any process model or data available a priori. One need not change the basic algorithm to deal with a different process, even though the basic physics may be entirely different. These models can be rapidly executed and such optimization strategies can also be used to simultaneously change the input variables,

in order to find out a desired state where an objective is minimized or maximized, satisfying certain goals [10].

6 Soft Computing Techniques

Many researchers have developed computationally expensive analytical models based on the theoretical approach to solve optimization problems belonging to blast furnace iron making. Due to highly noisy and nonlinear nature of the blast furnace data, it was difficult to resolve the blast furnace problems through these analytical models. These models depend upon large number of parameters which are difficult to know exactly and often there is a complex interaction between them, which cannot be precisely determined in the tough working situation of the blast furnace, where the operating temperature is more than 1500 °C. To overcome this problem, the soft computing technique comes into picture which is flexible in nature to incorporate imprecision, noise, and uncertainty and can generate results close to the practical requirements. The idea of soft computing was developed in 1981 by Zadeh [27]. He defined soft computing as an integration of fields like neural networks, fuzzy logics, probabilistic computing, and evolutionary computing. These methodologies are used to develop models, allowing solution to the real-world problems which are not modeled or too difficult to model analytically. The major role of soft computing is to exploit the tolerance of uncertainty, approximate reasoning, imprecision, and partial truth which resemble with human decision-making. Now, this technique is used in numerous fields like quality improvement, maximization of productivity, process optimization, and cost minimization, all of which need some complicated optimization process.

6.1 Fuzzy Logic

Fuzzy logic is an expansion of Boolean logic that consists of fuzzy sets, where classical set theory plays an important role and explains details. Modern fuzzy logic was introduced by Zadeh [28]. In fuzzy logic, which is also named as diffuse logic, continuum truth value and alternative solutions are taken as logical proposition. For example, a proposition (A) has the truth value 0.3 and its complement (A^c) has the truth value 0.6. The negation operator is applied, so that the addition of both the truth values must not be necessarily one. Fuzzy logic is used as an interpretation process to explain the properties of neural networks, in which performance is described precisely. The fuzzy logic provides to build an inference process in which actions are often flexible, nonlinear, and discontinuities which are close to the human activity. It can accommodate a certain degree of uncertainty which is generated due to large number of factors such as randomness, ignorance, complexity, and imprecision. Such problems are easily and effectively handled by Fuzzy systems.

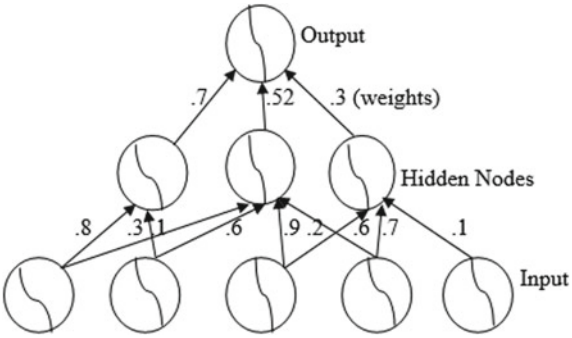


Fig. 3 The schematic arrangement of a typical feed-forward neural network

6.2 Artificial Neural Network

Artificial Neural Networks [29] are computational techniques based on the biological nervous system, where neurons are acting as decision-makers. ANN can be used as function approximation, data processing, pattern recognition, and a very good tool for creating data-driven computational models of the complex system. This network consists of units or nodes (the circle) and connection (the arrow). Each connection is stated by a number (weight), which indicates the strength of the connection. Connection with the +ve weights are known as excitatory, the ones with –ve weights are known as inhibitory. The architecture generates by nodes and connection with suitable weights form the topology of the network. In a feed-forward network connection, the information is directed only one way from top to bottom without loops. On the other hand, in a layered structure, the nodes are arranged in several layers to carry out the information. The schematic arrangement of a typical feed-forward neural network is presented in Fig. 3.

In a natural process, neurons produce electrical pulses along with their connections and transfer information to the neighboring neurons. It expresses as neuron (i) transmit a signal (Xi), which is a function of accumulated incoming signal (Yi). This is mathematically represented by

$$\mathrm{Yi} = \mathrm{f}(\mathrm{Xi})$$

where input is the waited sum of external input or output from the nodes.

$$\mathrm{Xi} = \sum_{\mathrm{j}=0}^{\mathrm{n}} \mathrm{WijZj}$$

Weights of an input Zj to node i are denoted by Wij. The activation function fj may be any function, i.e., linear, piecewise linear, sigmoidal, etc. All these information are represented mathematically in Fig. 4.

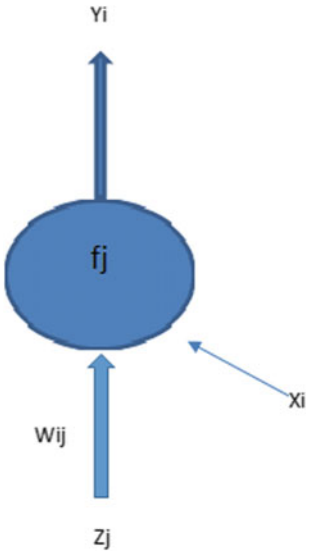


Fig. 4 Mathematical representation of signal flow in a network

The output from a node can serve as output from the system to the environment or as an input to a node in the next layer. The input and output layers correspond to the set of nodes which receive the input signals and generate the output signal for the system. The layers between these two layers communicating with the environment are called "hidden layers." Each such layer receives the signal as input which is the output of the previous layer and transmits the output to the next layer. The number of nodes in the hidden layers, as well as the number of hidden layers, are internal model parameters which affect the performance of the network. The weights of all the connection functions and biases of all nodes together define the artificial neural network.

6.3 Genetic Algorithms

Genetic algorithms (GA) were one of the pioneering evolutionary algorithms developed by Holland [30]. Holland has explained the idea of adaption behavior of the genetic algorithm. GA is an efficient evolutionary computing inspired by Darwin's theory "Survival of the Fittest."

Genetic algorithm treats a variable of the problem as gene, and the juxtaposition of such genes for all the variables leads to an individual. A group of individuals or candidate solution is known as the population of the problem. The initial population is randomly generated. The genetic information evolves in a bit string of a fixed length. A fitness value is used to calculate the strength of individual. The success of the solution and the survival depends on the fitness of individual. Three operators:

selection, crossover, and mutation are used in GA search process for each step called generation. In each generation, individual is evaluated by the fitness value, which defines the individual strength in the solution resolved through these three operators. The fitness members of the population are selected and produce offsprings for the next generation. The individuals having minimum fitness values are not selected in next generation and replaced by new offspring.

Crossover is the major genetic operator in GA, where a position along the bit string is randomly selected and the parent chromosome is divided into two sections, then each section is swapped with the same kind of another parent. The new offspring consists of a different part from each parent and there by inheriting bits from both. The crossover is carried out at a higher probability rate. The second major GA operator is the mutation. Mutation can alter the bit at lower probability rate. Crossover extends new areas in search process, i.e., excel optimum, whereas mutation provides existing area toward a near optimal solution. The newly generated offsprings are then placed back into the population and the process is repeated for many generations until a user-defined termination criteria are satisfied. After defined generation, the individuals who have the best fitness value can survive and evaluate the optimal solution. GA is an important tool to handle complex problems, where conventional methods have failed.

6.4 Evolutionary Neural Network Algorithm

Evolutionary neural network algorithm (EvoNN) [7, 31, 32] solves complex problem having nonlinearity and discontinuity in the objective functions, which is impossible to solve by other algorithms. Starting from a noisy dataset, it is able to create a set of distinct and optimized networks serving as the surrogate models of the system, where the objective functions are subsequently optimized. The corrected Akaike information criterion (AICc) [31] is used to identify a suitable model from this optimum set. Evolution is carried out in the lower part of the network by which flexibility is provided through numbers of weights and hidden nodes. A predator–prey multi-objective genetic algorithm [7] is applied in the lower part of the network to produce training data, which makes it separate from conventional neural network. In EvoNN, the available dataset is subdivided into a number of overlapping sub-datasets, in order to help in better training and testing of the data. The evolution starts with a random population of networks each having separate architecture and set of weights. The crossover operation in the algorithm involves swapping of two similarly positioned hidden nodes of two randomly selected individuals with a certain probability. It is shown in Fig. 5. The mutation of any weight $\mathrm{Wij}^{\mathrm{m}}$ of a population member result in

$$\mathrm{Wij}^{\mathrm{m}\mu} = \mathrm{Wij}^{\mathrm{m}} + \lambda\left(\mathrm{Wij}^{\mathrm{k}} - \mathrm{Wij}^{\mathrm{l}}\right)$$

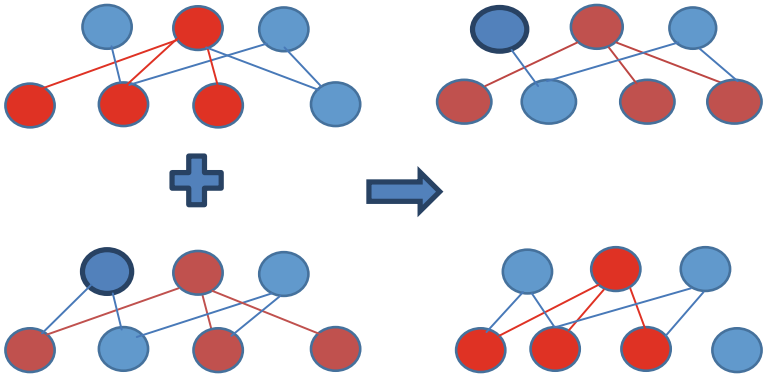


Fig. 5 Crossover operation in an evolutionary neural network

where $Wij^{m\mu}$ is the mutated weight and k and l are randomly chosen population members.

The number of prey that survives after given number of generations of predators' attack is ranked by applying the Fonseca scheme [7]. Here, individual (i) is noted as rank (Ri) and computed as

$$Ri = 1 + \theta i$$

where θi shows the number of individual dominating the individual (i), as per the weak dominance condition [32]. The set of best individual that emerges from this procedure, with a rank of one, approximate the Pareto frontier.

A fitness function (∅p, q) is computed for predator q evaluating a prey p to decide on its survival in the next generation.

$$\varnothing p, q = \alpha q\, Ep + (1 - \alpha q)Np \tag{1}$$

where Ep is the training error of the ANN and Np is the number of active links between input layers and the hidden layers. In Eq. (1), αq is a weight value attributed randomly to the predator q such that αq = [0, 1].

EvoNN uses linear transfer function for output nodes. The top portion above the hidden layer is trained by a linear least square (LLSQ) algorithm [32]. The optimized input from the lower portion significantly improves the output by gradient-based solver procedure. Using LLSQ, mathematical convergence is possible at the output stage.

6.5 Genetic Programming

Genetic programming (GP) [33, 34] refers to the class of genetic algorithm where the binary string is replaced by a tree structure encoding. The concept of using a population, fitness, selection, and crossover exists here but used in a different way. The total population is represented by a tree structure where recombination occurs by exchanging the sub-trees and mutation by a random change in the tree structure. The most important feature of GP is that it is capable of providing mathematical expression for a given dataset connecting the input and the output. The function set which defines the total structure consists of mathematical operators like addition, subtraction, multiplication, square root, and division is used along with a terminal set of the variables and constants. In tree encoding, the operators are set at the nodes and terminals and are set as leaves. Crossover is applied between two randomly picked parent trees and they produce the offsprings, which are then muted. Often, the recombination is carried out at a prescribed depth of the participating parents. At the same time, various types of mutations are also used in the process. In some cases, a sub-tree can be eliminated and grown again randomly (standard mutation), or an operator in the terminal set can be replaced by another of the same arity or two sub-trees of a unique parent could be swapped (monoparental exchange). Often, the nodes are derived from the given functional set and they reach the maximum depth by adding terminal nodes. This is known as the full process. On the other hand, in the growing process it is possible that both functional and terminal nodes are selected and added till the tree reaches the prescribed maximum depth. The third alternate, the so-called ramped half and half process is achieved by mixing both the strategies. Nowadays, ramped half and half type of initial population is used everywhere in GP as it maintains maximum diversity.

GP provides a suitable data-driven model just like a neural network. The GP is a heuristic search technique successfully applied to solve many real-world problems from different fields. While solving the real-world problem, a number of observations are to be considered, if the *i*th tree corresponds to a function $f_i(x)$ and tested against a known output y_i, then its fitness is calculated by the root mean square error (RMSE) [33]. Figure 6 presents a simple mathematical function $f(x_1, x_2) = (1 + x_2)/x_1$ evolved through GP. It requires only addition and division to define the function set in this simple example.

6.6 Bi-objective Genetic Programming (BioGP)

A traditional GP algorithm is used to lower the RSME values for the population members and usually at the end of the prescribed number of generation, it would pick up a member with lowest error. This lowest error offspring may often generate overfitting problem in the data while selecting another with larger error would under fit it [33]. In BioGP an evolutionary method is used to solve complicated problems

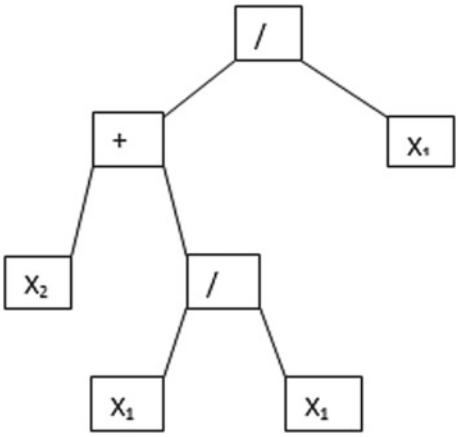


Fig. 6 A simple GP tree

by constructing a Pareto [34] trade-off between complexity and prediction accuracy of a family of GP trees. Initially, for a few generations fixed by the user, BioGP gets trained like the conventional GP through the minimization of the training error, before it transfers to the bi-objective mode. It is more efficient than the regular GP on a number of aspects. Here, one can get a number of Pareto optimal models in their unique tree structures, each showing the best trade-offs between the accuracy and complexity. In BioGP, a linear node is placed from which the other nodes are extended [35]. The user defines the number of roots during the operation. Individual root represents a tree according to a nonlinear problem. The linear part then generates a weighted sum of the outputs, coming from the various roots. In BioGP, a bias value is also applied to a linear node, which has a typical neural net structure. Here, the linear least square technique is used to evaluate the weights and the bias values, same as upper portion of an EvoNN model. The fitness is also calculated from a root mean square error. Here, also a predator–prey [36] genetic algorithm is used to optimize the problem. A common problem that occurs in the conventional GP is known as bloat [37], where the tree structure grows to be very large and after that any extra growth or a small reduction in it does not improve the fitness value. BioGP handled such difficulty by growing a number of sub-tree separately, before integrating them through a bias. To detect and cure any rogue branches, an error reduction ratio is used. At the end, BioGP generates a Pareto frontier and each individual of the frontier defines a separate GP-based optimized model.

7 Commercial Software Kimeme

Kimeme is a commercial software to solve multi-objective and multidisciplinary problems that exist in the optimization field. It has been developed and marketed by the Italian company, Cyber Dyne Srl [38]. It is used to design data-driven optimization

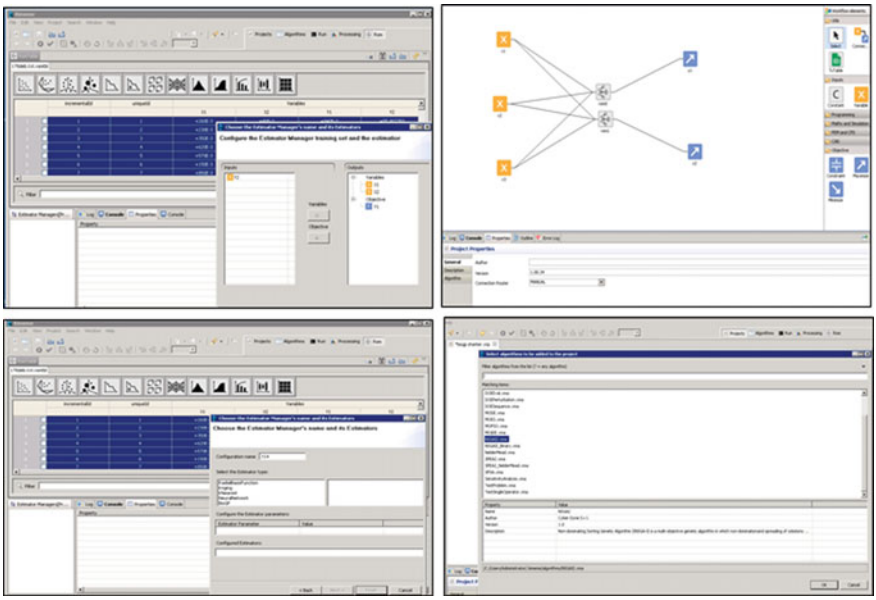


Fig. 7 A typical user interface of KIMEME

problems related to various fields and analysis is provided through a large repertory of algorithms, which are incorporated inside the Kimeme interface. Here, a network infrastructure is used to share the computational load. Different types of critical evaluation can be achieved using this software. The mutually conflicting problems are easily tackled using the built-in estimate managers and the existing algorithms. Figure 7 shows a typical user interface of the Kimeme software.

8 Soft Computing Techniques Applied to Iron and Steel Making

The soft computing technique applied to different components of blast furnace iron making such as burden distribution, production rate, gas flow and gas flow rate, tuyere velocity, heat loss, coke rate, carbon rate, chemical composition, chemical reaction, etc. Many models have been developed to solve problems of these critical areas belonging to blast furnace iron making. These are discussed below briefly and highlighted the research work in present modeling processes.

8.1 Optimization of Sintering Process

In many integrated steel plants, iron ore is sintered before it is used in the blast furnace. To fulfill the requirement of the sintering process, chemical and physical parameters, e.g., basicity and product diameter need to be within preset target values. The blending yard and dosing plants carry out mixing of the raw materials required to produce sinters of acceptable quality. Advanced control models and systems are used for the production of sinters of optimum size and quality.

An ore blending optimization model was developed by Wu et al. [39] for the iron ore sintering process. This was based upon the characteristics of iron ore. Both two-step method and simplex method have been used to build the model. This allowed distinguishing the calculation of optimized blending proportion of iron ores from that of other sintering materials, leading to improved calculation efficiency. This model has been successfully applied in the practical field. Umadevi et al. [40] worked on optimization of wüstite in the sinter plant at JSW steel limited in India. The productivity, strength reduction degradation index (RDI), reducibility, and the microstructural phases were examined in their evaluation process to produce sinter. Their analyses have indicated that the sinter with wüstite content between 8.5 and 10% shows higher productivity, higher strength, and lower RDI. Zhang et al. [41] developed a multi-objective optimization and analysis model of sintering process based on back propagation (BP) neural network. The model has been used to optimize sintering process and the production cost, the energy consumption, the quality, and the output are considered at the same time. Using a genetic algorithm and BP neural network, it has been found that simulation is consistent with the production of the sintering process. Nath et al. [42] developed a model for optimization of the suction pressure for iron ore sintering where GA was used. For the smooth operation and higher productivity of blast furnace, the quality of sinter is important as it improves the permeability and reducibility of burden materials. The operating condition of suction pressure has been evaluated by applying GA at different bed positions, optimum quality, productivity, and thermal efficiency of the process. The optimum input parameters and control parameters are evaluated through this tool effectively. In another study, Nath et al. [43] also presented mathematical modeling and optimization of a two-layer sintering process. The purpose was to improve the quality of the sinter and also to increase fuel efficiency. Both genetic algorithms and computation fluid dynamics were used in tandem to achieve this. The two-layer sintering process has been analyzed by CFD-based model and optimum coke rate is evaluated by GA optimization technique. The two-layer sintering process led to more acceptable results in all aspects, including the quality of sinter, reduction of return fines, and also for the reduction of coke rate. Mitra [44] proposed an evolutionary surrogate optimization model based upon ANN to solve complex sintering process during burden charge distribution. In this process along with the computationally expensive original model, a relatively less expensive metamodel was built using the data provided by the original model. During the optimization running the metamodel significantly reduced the overall computation burden. These models were used in two-layer sintering process, in which the

optimization decides the individual thicknesses of the layers along with the coke consumption in the process. Using the metamodels, the execution time apparently was brought down to the 60 percent of the original and it improved the decision support system utilization without compromising the quality of the optimized solution.

8.2 Gas Scheduling Optimization

A multilayer encoding GA-based granular fuzzy inference for blast furnace gas scheduling was developed by Wang et al. [45]. In this approach, the fluctuation characteristics of flow of the by-product gas user are considered for varying length of data separation. Also, the time warping normalization is used to equalize the partition data into the granules with the uniform length.

A sophisticated multi-input and multi-output blast furnace gas system (BFG) consists of blast furnace transportation network and a series of gas users. The minimization of production cost, resources waste, and the environment pollution control can be possible by this system automatically. The fuzzy rules established on the basis of data granulations of gas users and the system adjustment amount by clustering the granular-based simple set of the adjust moments. Then, fuzzy reference model is established to evaluate system adjustment. In BFG system, performance of the granular process depends on various gas users, who produce different influences on the gas-scheduling procedure and the numbers of clustering required in the system. A multilayer genetic algorithm based parameter selection was used by these researchers to figure out an optimal subset of gas users and also to optimize the parameters related to the granulation process. Here, the front level acts as a layer of feature selection and cluster number of each gas user is optimized through the optimization process. Single point crossover was adopted to the population of the primary level, while a multipoint crossover scheme was employed in the secondary level where a real encoding was used. This search procedure improved the strength of the algorithm.

Wang et al. [45] have checked the efficacy of their approach using data from actual plant operation. They have randomly analyzed information available for a period of 25 days. Out of total of 250 entries in the dataset, 200 were used for training, 30 were used for validation, and the rest were set aside as test set, where the situations of gas supply surplus and shortage were both considered. This model was capable of selecting the adjustable users and also could calculate the scheduling timing with an accuracy above 85%. In the scheduling process, adjustable gas users play an important role in order to make the energy system stable and assuring safety during the plant operation. The results indicate that the model still needs to improve its prediction accuracy, in order to be applied with a minimum human interaction.

8.3 Prediction and Process Control of Hot Metal

In this modeling processes, the basic things which are focused in the blast furnace are melting rate (tons of hot metal produced per hour), the carbon content in the hot metal, and the specific amount of oxygen per ton of hot metal. The melting rate is largely determined by the qualities of the hot blast (in particular, the amount of oxygen in the hot blast) that directly influence melting rate in the blast furnace.

The blast furnace hot metal production and the related processes are simulated with various mathematical models [46–48]. Analyses have been used for the prediction and control of the hot quality metal production in the blast furnace through various data-driven approaches like artificial neural network, fuzzy logic, Bayesian network, and genetic programming and support vector method [49–53].

The genetic programming has been used to model and optimize various aspects of the blast furnace iron making and the results were pitted against other modeling strategies like linear regression and support vector regression [54]. In this analysis, three different models are constructed to obtain results. These models address to various issues of the blast furnace process: the required specific amount of oxygen, the metal carbon content in the hot metal, and the melting rate. The dataset used for all numerical experiments consists of hour-wise evaluated values of the blast furnace variables for a particular time schedule. The data of injected hot blast strongly related to the prediction of the melting rate of the blast furnace. The result is that the melting rate is significantly influenced by the hot blast and the amount of oxygen injected into the blast furnace. It was significantly difficult to predict the amount of oxygen. The model analysis ensured that cooling losses and the weight of the coke dump in the lower area of the blast furnace have a greater impact on the specific amount of oxygen. The prediction of the carbon content is a prime factor of the hot metal because it directly influences the quality of hot metal produced. The model shows that the carbon content is strongly linked with the other alloys (Si, Mn, and Ti) in the hot metal which is confirmed from a variable impact analysis. Consequently, the inter-relationship between different variables can be properly explored through such models to obtain a better insight of the process.

A dynamic regional viscosity prediction model of blast furnace slag was developed by Guo et al. [55]. Their model was based upon various metallurgical properties. For example, the slag viscosity prediction model was based upon discrete viscosity point data. It leads to the following problems. In the initial case, there is a systematic error among the sets of the measured viscosity data and in the second case, the prediction accuracy is affected by applying the traditional fitting algorithms to all known viscosity data. The establishment of the numerical viscosity model depends on the selection of exact amount of reference points obtained from the dataset and the optimization was carried out between the slag components and the slag viscosity.

This study had aimed on the prediction of viscosity of blast furnace slag and successful implementation of the dynamic regional blast furnace viscosity prediction model. The information of the independent system was produced through the partial least square regression method. It also described the dependents and eliminated noise

interference in the system. Here, the model results were validated using viscosity data of both the actual and synthesized blast furnace slags from a Chinese steel plant. In this work, the average prediction error was calculated as 6.82% and the mean value deviation was quite low. This corroborated the model predictions.

8.4 Predicting Hot Metal Temperature Using Neural Networks

Blast furnace hot metal temperature prediction plays an important role in this field. It controls and helps the operators to make the system faster and accurately run. In this field, lots of development took place and various models are developed to compute the temperature inside the furnace, which are also capable of calculating the hot metal temperature.

Some inner regions of the blast furnace remain unreachable by the furnace probes but an approximate information of their behavior is possible to obtain by mathematical models. Hot metal temperature is one of the major parameters, which needs to be handled properly during the operation. It should be maintained at a set point in order to provide a stability to the furnace, needed to produce a homogeneous quality of pig iron. Parametric models are developed to achieve this goal. In this methodology, few selected parameters will able to determine the relationship between the system inputs and outputs [56]. The artificial neural network (ANN) is robust against the noise and easily adapt to the changing condition. A model based on fuzzy logic tools to compute the temperature of hot metal in a blast furnace was developed by Martin et al. [57]. Here, the input variables in the model use the control variables of a running blast furnace, for example, the amount of pulverized coal injection, moisture, the level of oxygen addition, blast volume, and the mineral/coke ratio, and using these parameters the model predicted the hot metal temperature. Jeminez et al. [58] discussed the neural network based methods which enable to help temperature forecasting of the hot metal.

In BF operation, numerous process variables are routinely recorded. The process variables which influence the hot metal temperature are blast temperature, blast moisture, blast volume, oxygen enrichment, ore to coke rate, pulverized coal injection rate, etc., which are selected as input parameters. Hot metal temperature measure at the iron runner, i.e., at the outlet of the tap hole is considered as output variable. The structure chosen for the model belongs to a class of NARX model (nonlinear autoregression model with exogenous input) [59]. The model is as follows:

$$\dot{Y} = F(y(k-1), y(k-2) \ldots, y(k-p), u(k), u(k-1) \ldots, u(k-q)) \quad (2)$$

Here, forecast value of the variable is y considered for sampling schedule k, p is the previous sampling schedule variable, and u is the input variable at previous sampling schedule q.

The model structure is represented by two steps. Firstly, input variable time average (t_a) is used which is evaluated by taking the values of each variable from schedule t_a at which hot metal temperature is considered and examined backward until a period of two times to the estimated response (2τ) of each variable has been covered. Secondly, the average value with this procedure ($\tilde{u}_i$) is employed as input for the model.

$$\tilde{u}_i = \sum_{t=ta-2\tau}^{ta} e^{-2\left(\frac{[\tau-(ta-t)]}{\tau}\right)^2} ui(t) \Big/ \sum_{t=ta-2\tau}^{ta} e^{-2\left(\frac{[\tau-(ta-t)]}{\tau}\right)^2} \quad (3)$$

Here, t represents the schedule at which ui was collected.

Equation (2) is rewritten with respect to the specific requirement of the model just described.

$$\tilde{T}(k) = F(T(k-tta), tta(k), thc(k), tic(k), ui(k), ui(k-1) \ldots . ui(k-2\tau) \quad (4)$$

Here, thc (elapsed time between tapings), tic (elapsed time from the beginning of the taping until hot metal temperature was measured), tta (elapsed time since the previous temperature measured), and T (hot metal temperature was taken one step backward).

In the next step, the function is defined in Eq. (2) from which an artificial neural network was chosen. Here, a multilayer perception was introduced with a single hidden layer.

Once the structure of the neural network was established and its training concluded, the input variable sets were used to compare their predictions with the performance of the actual operational blast furnace. In the first step, the neural network was introduced to exhibit a NARX model described in Eq. (2). This model should be able to generate the values for the next hot metal with regards to the given input data and the best hot metal temperature evaluated at the blast furnace. There are many possible paths to generate models that are capable of predicting the temperature of hot metal. In current scenario, data-driven model plays an important role to find out the result very close to the physical interference.

8.5 Modeling of Silicon Content in Hot Metal

The product quality and thermal states of the blast furnace in iron-making processes are directly influenced by the silicon content [Si]. It acts as an important index of the quality of the liquid metal and is directly related to its thermal state. Silicon enters the blast furnace as an impurity in iron ore and coke, mainly in the form of silicates. The percentage of silicon in the sinter is up to 3.5% and in pellet it is 2%, and in coke it is about 5–6% as a main component of ash is silicon [5]. High silicon content leads to a higher amount of slag and this would eliminate the P and S in the hot metal. Generally, Si content should be controlled in the range of 0.5–0.7%. Therefore, optimum operation requires low hot metal Si content. The hot metal Si

control is a very difficult task, as no direct measurements of the conditions inside the hearth are actually possible.

Predicting the Si content of the hot metal is one of the major concerns of the researchers in this area. Castore et al. [60] discussed various facts related to control of hot metal Si content and indicated few challenging steps encountered in automatic control. Phadke and Wu [61] established a closed loop strategy for the prediction and control of hot metal Si content. In their method, a feedback system leads to a time lag of a minimum of one-time step. Their first study [62–65] had considered a liner problem with time series analysis and subsequently it was followed by *vector autoregressive moving average with exogenous inputs* (VARMAX) and state-space modeling [66, 66]. For further details of this strategy, the readers are referred to the original articles. Östermark, and Saxén [66] have also used a VARMAX modeling approach for predicting the hot metal Si content. Chung et al. [68] used an *autoregressive with exogenous input* (ARX) model, detailed in their paper. There a dynamic data system method was applied to predict the Si content in the hot metal. The inputs consist of temperature and moisture content, blast volume, ore to coke ratio, oil injection, burden decent rate, average coke temperature, adiabatic flame temperature, permeability of the bed, and the initial level of Si content. Some old analyses were used along with these information, and it gave rise to an efficient model. Zeng and Gao [69] expressed the importance of attribution and detection of large fluctuation in the blast furnace data to predict the hot metal Si content. This strategy could determine the missing values of attribution and led to a satisfactory prediction of the Si level. In the time series approach model, [53] also consider the input values of previous time steps. This is very important in silicon prediction as the system consists of large number of complex processes. In this case, silicon level is given by

$$\mathrm{Si}(t) = \mathrm{f}(\mathrm{x}(t), \mathrm{x}(t-1) \ldots \mathrm{x}(t-n)) \tag{5}$$

where $t-1, t-2, t-n$ are the processing time steps.

Sheel and Deo [70] created a linear fuzzy regression model on blast furnace data, applying a genetic algorithm to determine the fuzzy coefficients of the linear equation describing the silicon content. Lue et al. [71] carried out optimization of Si content in the hot metal using data of an operational plant using a fuzzy C-means clustering algorithm. The clustering model was successfully utilized to predict the conditions of higher production and minimum cost. It could produce a good selection of Si content around the mean value of Si content calculated from the operational history of the furnace. Nonlinear methodology has been utilized in the process as well and achieved only a limited success despite continued efforts for over 20 years. ANN [29], SVM [72], and other related data-driven approaches, thus, were not fully effective to predict the Si content in the hot metal. Chen [73] presented a model to predict the Si content in the hot metal using ANN and qualitative reasoning. Here, he could identify ten principal variables which included pressure losses, temperature, burden variables, and also the blast parameters. A sampling time of 48 minutes was selected in view of the average tapping time. This analysis produced accurate prediction for a period of

600 taps. Chen et al. [74] introduced a combined strategy, where an ANN based GA was used to predict the Si content in the hot metal. They used a total of 11 independent variables. Previously, tapping result of Si was also included in the list of inputs. Saxén et al. [75] developed a method, where a bi-objective ANN based GA model was constructed for the calculation of Si content. Saxén and Pettersson [76] developed an ANN pruning technique to evaluate the major input parameters and their significant time lags. By the help of this methodology, the inputs were successfully ranked and the proper time lags were determined. These researchers found that the specific blast volume, tuyere heat loss, and a gas resistance index with a time lag between two to three hours were the most important input parameters. Nurkkala et al. [77] generated a data-driven nonlinear model for selecting the best pruning results and reported a Pareto front between modeling error and the complexity of the model [7]. These models typically correlate the hot metal silicon with inputs such as blast volume, blast temperature, pressure, coal injection, ore to coke ratio, overall pressure loss, hydrocarbon injection rate, etc. The problem of selecting model complexity and input variables were thus handled properly and results were satisfactory. A self-organizing ANN was used by Saxén et al. [78], which also utilized a linear switching strategy for the prediction of Si content in the hot metal of a commercial blast furnace. This strategy was efficiently used to determine the exact process parameters.

8.6 Modeling of Burden and Gas Distribution

The burden and gas distribution play an important part in the improvements in a blast furnace process and it was studied by many researchers [79–92]. The productivity and effectiveness of blast furnace solely depend upon the way of iron ore reduced and drop through the upper part of the furnace. In most of the industrial blast furnaces, hopper systems are utilized for the storage of the granular raw materials. First, the materials are weighed in the surge hopper, and then a conveyor belt is used to carry them to top bunker. The iron bearing materials containing sinter, pellets, and coke are dumped layer-wise through the throat. The radial distribution of the charge plays an important role as it directly affects the reducing process in the furnace [79, 80]. The blast furnaces utilize the movable armor position, bell-type charging devices, various tilting angles, and the rotating chute in furnaces with bell-less top to directly adjust the charge distribution. The other parameters like dump size, charging sequence, gas flow, and the stock level are also known to influence the distribution [80]. A typical bell charging equipment and movable armor are shown in Fig. 8.

Charging in bell-top furnaces: In this type of furnaces, bell and hopper are provided at the top. By raising the bell up, the hopper is blocked and by lowering the bell down materials are charged inside the furnace. In many cases, a sealed double bell technique is used instead of a single bell to eliminate the gas loss. This arrangement is very efficient in nature but on the other hand, it is not much flexible for the operators in order to come up with the required burden distribution. Most of the times the

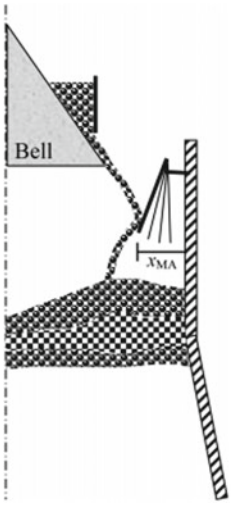


Fig. 8 A schematic representation of the bell charging equipment and the movable armor [90]

raw materials are dumped close to the wall. To move the charge toward the center, movable armor positions are set by the blast furnace operators.

Bell-less top: Bell-less top charging system is relatively recently used in the blast furnace raw material charging process. This was developed by Paul Wurth and was successfully applied in the industrial blast furnaces in the year 1972. In this technique, a gated hopper along with a rotating chute, revolving around the axis of symmetry, controls the charge distribution as per requirement. This arrangement provides higher flexibility for the blast furnace operators, as they can control the inclination angle of the chute in order to decide on the size and position of the charged material. Due to higher flexibility, bell-less charging is considered better than the bell-top charging.

A number of techniques are used to measure the burden distribution. For example, mechanical profile meters are applied to locate the charge stock line in the furnace [81] noncontact techniques utilizing radar microwaves [82] or laser [83] infrared, or gamma radiation are also used routinely and the depth of the burden layer is often determined using magnetometers. Most of the techniques, which are mentioned are expensive in installation and maintenance of the equipment. A number of models for indirect measurement of the charge distribution have also been proposed to circumvent the problem of direct measurements inside the furnace owing to its adverse environment and wear that often takes place there. The flow and the burden distribution which needs to be considered is quite expectedly complex.

The various methodologies and models have been developed to improve the process and increase the productivity effectively. Mitra and Saxén [84] used an evolutionary approach for computing the faster charging programs in the blast furnace. The model describes the burden formation and descent procedure in the blast furnace and can be used for charging design. Many recent models are based on discrete element modeling or computational fluid dynamics. Zhoe et al. [85] developed both 2-D and 3-D discrete element methods (DEM) and applied them to calculate the stress

distribution between the solids. They also successfully simulated the solid flow, computed the particle velocities, and determined the streamlines for the flow. Porosity of the blast furnace was also estimated. Natsui et al. [86] examined the variation of descending velocity in the packed bed configuration existing inside the blast furnace and compared the results with the help of a discrete elemental method, as well as a cold model satisfying the similarity criteria. Xu et al. [87] explained the cylindrical fluctuation and formation of gas flow by the help of continuum model and also a 2-D discrete element strategy. These models provide very elaborate results but have limited application to control due to prohibitive computational requirements.

Pettersson, et al. [88] proposed a GA-based charging program by which the required radial ore and coke distributions can be determined. In their method, they were able to handle complex and non-differentiable optimization problems, offering additional flexibility. Their algorithms have evolved various charging programs satisfying the practical constraints. They were able to significantly reduce the differences between the target and the computed charge distributions. The charging program consists of two coke dumps followed by a pallet dump and coke mixture. The dump size and the armor position were stated using a simple genetic algorithm. A significance problem has occurred during above approach because during charging not only burden distribution but also gas distribution plays an important role. As the gas temperature is measured radially in or above the burden surface their measurements are controlled by the operators. Mitra and Saxén [89] developed a gas distribution model based on the burden distribution in the furnace. Here, they have attempted to determine the radial temperature profile at the burden probe level by simulating a charging program. This is followed by applying a GA that makes an effective search to suggest the proper dumping sequence and the corresponding chute setting. It has demonstrated that the algorithm effectively could produce a charging program and meet the target. The gas distribution along with burden distribution have also given significant results and brought additional advantages as compared to the burden distribution model.

In another approach for modeling burden layer formation, Hinnela [90] used an ANN model for the thickness of burden layers in the iron-making blast furnace process. Feed-forward and recurrent neural networks were used in his studies. A cross-validation procedure is used for the selection of network size. The evolved data-driven model was initially studied and different inputs were predicted. This model was used to analyze the formation of different layers in the furnace, and also to study the stock level. Also, it could ascertain the impact of changing the movable armor settings; the burden descend rate was also evaluated and compared with the actual plant practice. The comparison result is shown in Fig. 9.

Pettersson et al. [91] developed an evolutionary algorithm to estimate the layer thickness of the charged layer in a furnace. This eliminates the dependence on mathematical models and these researchers have used data-driven model successfully for these processes. Mitra et al. [92] developed a charging optimization process using multi-objective evolutionary and genetic algorithms. The charging program produces the required charge and gas distribution in the iron-making blast furnace which are evolved using a multi-objective GA-based optimization strategy. Here, the concept

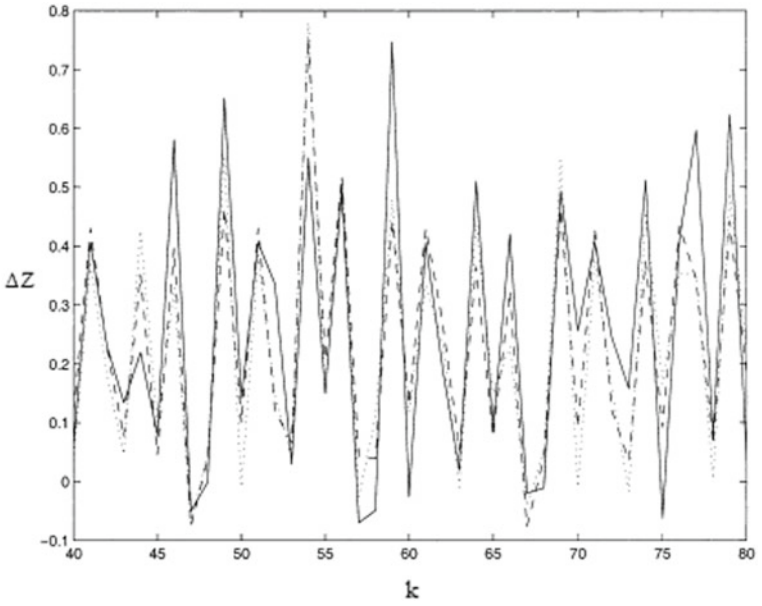


Fig. 9 Operational layer thickness (——) compared with the predictions of a feed-forward (– – – – –) and recurrent (·····) network. The burden level and the movable armor settings were the external outputs [90]

of *k*-optimality [93] criteria have been used to solve this type of complex problem and developed a significant improvement over earlier strategies. In this paper, nine objectives are considered with 15 input variables, out of which four pallet dump masses are continuous, while the rest are discrete. For optimization work, surrogate models have been developed using EvoNN [7]. Through this model, a total of 10055 input vectors are generated of which 2264 are found to be valid and are used for training the network. The model accuracy is expressed in terms of root mean squire errors (RSME), which varies with the magnitude of the parameter *k* used in the formulation. The model is used in this burden distribution program to optimize desired gas flow distribution characterized by temperature profiles at in burden probe of the shaft. The result indicated that using k-optimality, selection criteria moved the population toward the optimal, which is efficient and evolve interesting solutions. The final Pareto front is analyzed and discussed and concluded that the charging program evolves for the targeted temperature profiles which are reasonable and partly novel. The result is shown in Fig. 10.

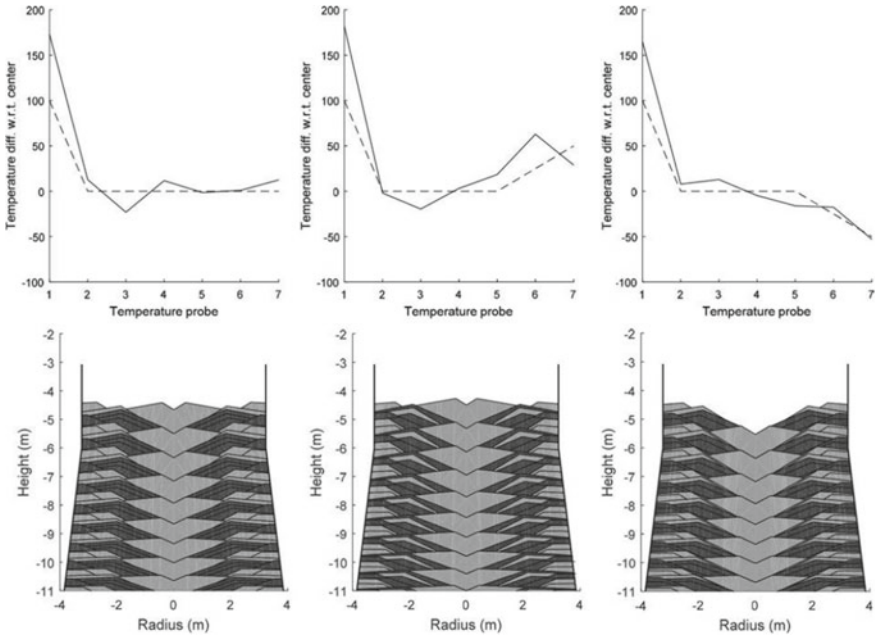


Fig. 10 Top: Target gas temperature distribution with respect to a central temperature (dashed line) and resulting gas temperature (solid line). Bottom: Cross section of the blast furnace shaft with burden distribution [92]

8.7 Optimization of Carbon Dioxide Emission

Carbon dioxide emission in the blast furnace is a serious issue in every steel industry. Global warming is a common threat to all the countries and the steel industry is a known polluter. It is found that around 1.7 tons of CO_2 are emitted during every ton of hot metal production [5]. In the steel industries, a large amount of fossil fuel is absorbed as reducing agents. Gases emitted by the blast furnace are transferred to the downstream process as supplementary energy source. The absorption of reducing elements in the upstream process is a principal source of the C input. Reduction of CO_2 emission is an important part of the energy balance in the steel works and it is directly connected with the energy saving. The environmental restriction on the emission through fossil fuel is one of the important concerns pertinent to CO_2 emission in steel industries. The replacement of the blast furnaces is a major challenge to the world steel industry, which is impractical and not likely in the present scenario. A much more sensible option is to modify the operation and the system as and when possible. In this scenario, the feasibility of recycling the top gas after CO_2 stripping needs to be thoroughly examined to determine some more efficient operational strategies.

Japanese researchers [94, 95] have recently conducted simulation studies of blast furnace top gas recycling process. Austin et al. [95] developed a simple recycling method of using cold gas with O_2 addition and reduction of CO_2 stripped hot gas after heating it along with an oxygen blast. Nagami et al. [96] studied plastic injection inside the furnace along with top gas recycling and an injection of O_2, while fixing the main conditions at the high-temperature zone. Murai et al. [97] proposed the option for minimization of CO_2 emission through an efficient recycling of blast furnace top gas. They have recommended tuyere and shaft injections along with a very large amount of plastic injection to effectively minimize the amount of CO_2 emission.

Researchers mentioned in this discussion have proposed the requirement of a thorough analysis of the material and energy transfer inside the furnace. This leads to finding the economic advantages of the proposed novel process of emission control. Recently, Helle et al. [98] examined the condition in which a large amount of oxygen enrichment of blast and top gas recycling would be economically possible through an optimization of the recycle process. Wang et al. [99] considered basically a simulation of the entire plant. His study considered the sinter plant, coke oven, blast furnace along with hot stoves and the tripping unit, basic oxygen furnace (BOF) for steel making and also the associated power plant, employing an elaborate model. Ghanbari et al. [100] used an optimization strategy to recycle and gas utilization process to reduce the carbon dioxide emission. In this study, the prime emphasis is given to the steel producing industry for their future existence and sustainability, for which the application of numerical methods was deemed to be crucial. In this work, blast furnace process methodology was reexamined through carbon computing by utilizing its various units, which were subsequently recombined with a polygeneration process to produce distinct heat, methanol, and electric energy in addition to the ferrous products. Their model used a super structure, where optimization is carried out to minimize the residual gases, the internal energy required under a target operating cost as well as the investment cost for the new process units. Using an optimal design along with a flexible operating system, these researchers were able to bring down the level of CO_2 emission by approximately 30%.

To study such processes related to blast furnace, Pettersson et al. [101] have used a linear model developed from the first principle of thermodynamics model [102] to get a cost function F_1 and emission function F_2

$$F_1 = 1/\dot{m}_{hm}(\dot{m}_i c_i + \text{Max}\{\dot{m}_{hm}F_2 - A, 0\})$$
$$F_2 = 1/\dot{m}_{hm} \sum_i \dot{m}_i e_i$$

Here $\dot{m}_i$ is the mass flow rates for resource i, c_i, and e_i are the corresponding cost and emission terms. $\dot{m}_{hm}$ is the hot metal production rate, and A is the legal emission allowance. The authors used the predator–prey genetic algorithm [36] to obtain a Pareto front describing the best combination for the cost and emission for different production rates.

The linear model has described the normal blast furnace process but the top gas recycling process introduces severe nonlinearity in the system. Therefore, solving the problem in a linear way generates an error at the end point of the objective space, where the optimum lies. To get accurate result, Mitra et al. [103] handled this nonlinearity using a genetic algorithm and obtained the Pareto frontier solutions for different top gas recycling conditions. Agrawal et al. [104] used the evolutionary neural networks to create the model of the carbon dioxide emission and productivity using actual data from the blast furnace. The neural networks used are Bayesian, Akaike [31], and corrected Akaike criteria [31] along with predator–prey algorithm used earlier by Petterson et al. [7]. This is used to compute the various Pareto fronts at the different silicon levels. Jha et al. [105] performed a useful comparison between different algorithms. BioGP, EvoNN, and the NSGAII algorithms [106] were extensively studied by them, along with two commercial optimization softwares, Kimeme [38] and modeFrontier [107]. Data-driven models described by these investigators were applied to CO_2 emission, silicon content in the hot metal and also the productivity for an operational blast furnace. The idea was to compute the optimal trade-off between the extent of CO_2 emission and productivity at different silicon levels in the hot metal. They have used a predator–prey genetic algorithm [36] for computing the Pareto optimality. Mohanty et al. [93] used a multi-criteria approach to study the top gas recycling process in a blast furnace. Their method involved the notion of k-optimality [103] implemented in a multi-objective GA, which allowed simultaneous optimization of a large number of objectives following an evolutionary way. The results appeared to be quite important considering the CO_2 reduction requirements that are currently implemented in the steel industry.

The top gas recycling practice in the furnace reduces the emission of CO_2, and reduces the production cost, which is a very difficult task. Various authors describe many processes technologies to control and optimize the system to get the best result. The advantage of GA-based Pareto optimization is now fully established in the metallurgical domain, and is being used efficiently for solving many nonlinear problems in the field.

8.8 Optimization of the Performance of a Sponge Iron Rotary Kiln

Coal is used in the rotary kiln operations for producing sponge iron through a gas—solid reaction. This is one of the principal sectors in the current practice of iron making. A large number of concepts have been developed till date for this process which are playing an important role in the ferrous production industry. The quality of directly reduced iron produced by rotary kiln is generally expressed through its metallic iron content. The iron content and the rate of sponge iron production are the functions of different operating parameters like iron ore quality, particle size and feed rate, coal quality and kiln rotation speed, etc. Various models have been developed

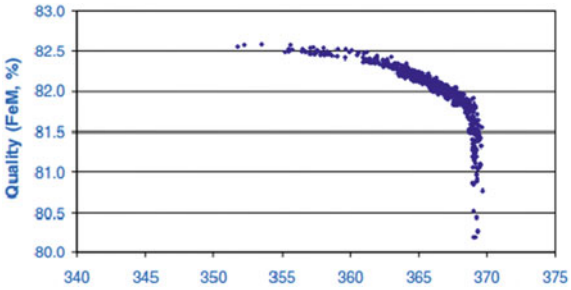


Fig. 11 Rotary kiln Pareto trade-off curve for an air input range of 2000–2200 Nm^3 (ton of iron ore)$^{-1}$ [112]

and processes are studied as well as designed to produce the optimum quality of iron metal at the output.

Haywood et al. [108] developed one-dimensional kiln model to evaluate design process. This model has been used for a number of pyro-metallurgical processes. The one-dimensional kiln model was based on a FORTRAN program. It provided computed data on the solid and gas temperature profiles in the kiln, the extent of reduction as a function of time and also the gas composition. Gasevic et al. [109] introduced a model that explains the characteristic of a rotary kiln system with respect to its variables. Stander and Craig [110] introduced a domain reduction model which could correlate the performance of the combined blower, the outlet gas temperature, flow rate, and the maximum temperature of the solid bed inside the kiln. Croft et al. [111] introduced a three-dimensional CFD model to evaluate various design options to make the flow control inside the rotary kiln more efficient.

Mohanty et al. [112] studied and explained how an ANN is used to model the production rate and metallic iron content data from an actual industrial rotary kiln operation. They have used predator–prey genetic algorithm [36] to perform a constrained optimization to examine the quality and quantity of the product. A typical result of the optimization study is presented in Fig. 11, which shows the Pareto front corresponding to an air input range between 2000 and 2200 Nm^3 . (ton of iron ore)$^{-1}$. The second Pareto front shown in Fig. 12 corresponds to an air input range between 1800 and 2000 Nm^3 . (ton of iron ore)$^{-1}$. The second Pareto front is produced to examine the benefits of reduction of air input on the objectives considered in this study.

In the rotary kiln, the direct reduction process is utilized in less number of areas which are covered by this modeling process for optimization work. This is a key area to allow data-driven modeling process to optimize the various processes as well as the process parameters.

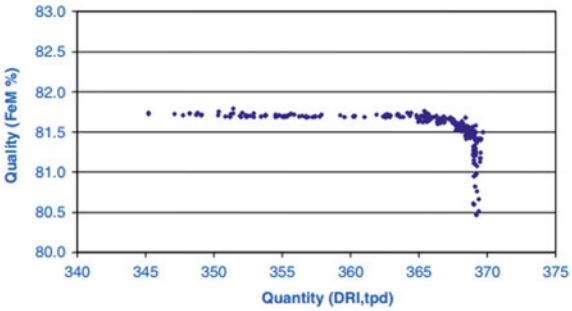


Fig. 12 Rotary kiln Pareto trade-off curve for an air input range of 1800–2000 Nm^3 (ton of iron ore)$^{-1}$ [112]

8.9 Modeling of Coal Injection

Coal is injected into the blast furnace as a cheap fuel which produces heat as well as used as a reductant inside the blast furnace. The coal injection is economically more feasible process than those using 100 percent coke. It increases the production rate by reducing the production cost. After the injection of coal into the raceway through the tuyeres, it combusts first, then produces heat and carbon monoxide in the reduction process. Therefore, it is desirable to optimize the coal injection to increase the reduction rate and the evolution of heat in an economic way. Various models have been developed to configure the coal injection system which are suitable for the iron and steel making.

Hunty et al. [113] explained the combustion condition and also the mechanism of combustion of coal in the blast furnace and established various attributes influencing coal combustion, for example, type of the coal used, rate of injection, location of the injector, and also the blast parameters. They had presented some mathematical description of coal/coke combustion in the blast furnace. Chen [114] presented a 3-D transport model based upon the CFD software PHOENICS. They attempted to simulate the trajectory of coal injection in the tuyere with two types of lance design for the pulverized coal injection (PCI) process. Shen et al. [115] produced a 3-D mathematical model of the combustion of coke and pulverized coal. The model is utilized in the region of lance–blow pipe–tuyere–raceway–coke bed to simulate furnace phenomena of pulverized coal. In another study, Shen et al. [116] developed a CFD model of the flow and combustion of powdery coal in a packed bed of coke. It has been applied to simulate the impact of PCI in the raceway and the surrounding coke bed. Using such models, often it is very difficult to predict or simulate the blast furnace phenomena realistically, as the process is very complex and nonlinear in nature. Giri et al. [35] handled this problem and found acceptable results using both EvoNN and BioGP algorithms. Figures 13 and 14 show the Pareto frontiers for the solutions maximizing coal injection and minimizing solution loss.

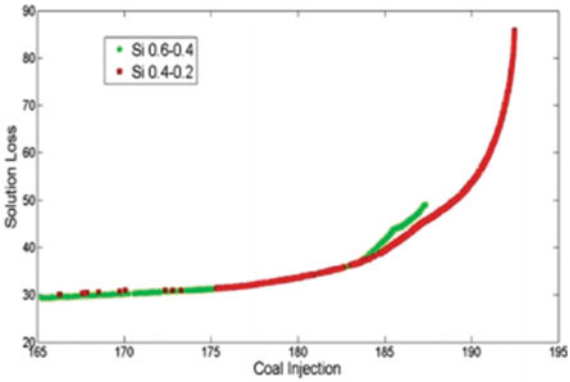


Fig. 13 Pareto frontier between solution loss and coal injection computed using BioGP [35]

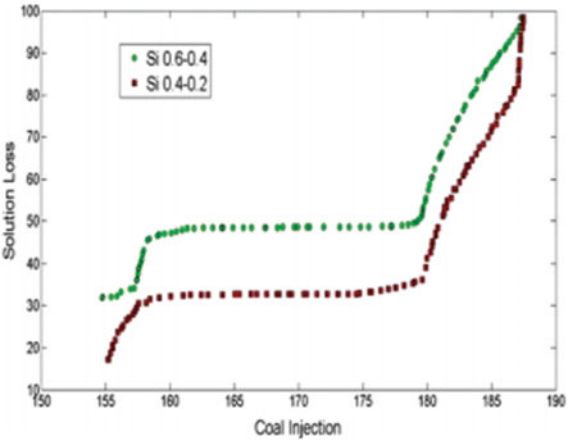


Fig. 14 Pareto frontier between solution loss and coal injection computed using EvoNN [35]

8.10 Optimization of Operating Condition of an Integrated Steel Plant Using Biomass Injection

In Steel industry, iron ore reduction is a very intensive energy utilization process, which largely depends upon the fossil fuels which produces carbon dioxide emission. The CO_2 emission can be reduced by replacing a part of the fuel or reductants with biomass. Now biomass is used as a secondary ingredient of fuel charge harvested in a sustainable way. It contains renewability and partial CO_2 neutrality. Now, quite a few steel plants have shown interest to minimize production cost by applying biomass injection and the suitable state of the blast furnace for its implementation is determined through optimization.

Lacroix et al. [117] given a significant flexibility to economics and emission by adding varying amount of biofuel and the charging ratio of coke and coal. Norgate and Langberg [118] investigated the optimal balance between charcoal and coal, which is directly influenced by emission caps and carbon tax. As the caps decrease

and the taxes increase and thus flexibility is now key to the competitive steel making. Wiklund et al. [119] attempted to optimize a hypothetical steel plant process under biomass injection. Their optimization considered energy, raw material, and emission costs and a likely credit for the sold power and heat. A piecewise linearized model was used by them to decrease the computational requirement for determining the minimum production cost of a steel plant. Helle et al. [120] have also developed a mathematical model to optimize iron production with biomass injection in the blast furnace. The injection of pulverized coal has been studied for long using experiments and efficient transport models based upon computational fluid dynamics technique [21–23]. However, the potential improvement in this field is limited. Wiklund et al. [121] studied the resource allocation problem in an integrated steel plant using the blast furnace. The nonlinear optimization problem formulated by them has been solved using GA. Their findings suggested that applying a nonuniform distribution of the resources may lead to some economic advantages. Final convergence of their model could not be guaranteed by solving it in a linear way, and therefore the evolutionary approach was necessary. Nowadays, many researchers have adopted the biofuel injection as an alternative way to make the process utility economic and try to solve it by the help of data-driven modeling. Till date, only a handful of research work has been processed for the biofuel injection modeling and optimization that have used the evolutionary algorithm.

8.11 Many Objectives Optimization Problem Related to Blast Furnace

Chugh et al. [122] developed a metamodel using an operational blast furnace data to address many objective problems pertinent to that reactor. An inference vector guided evolutionary strategy has been successfully introduced in this study to construct surrogate models to optimize multi-objective problem formulated by them. Their formulation consists of eight objectives and a total of 12 decision variables as shown in Table 1. Handling such large number of objectives was previously not possible by other methods. The result produced is more precise and efficiently optimized the hot metal production process.

The challenge in the optimization of the blast furnace is that it involves multiple conflicting objectives, leading to multi-objective optimization problems with many objectives [104]. The reference vector based evolutionary algorithm is very suitable for solving such problems. Previously, it was difficult to solve the multi-objective problems of more than three objectives, such handicap no longer exists in an evolutionary approach. Recently, Wang et al. [123] have categorized data-driven optimization into two types; online and off-line. In online optimization, a small amount of new data is required and in off-line optimization, no extra data other than those are available in hand are required. Data-driven evolutionary algorithm conducted off-line with a small amount of information must address two challenges: firstly,

Table 1 Process variables and objectives used in the surrogate model

Process variables	
Pellet (%)	–
Specific flux consumption (Kg/thm)	–
Limestone (Kg/thm)	–
Dolomite (Kg/thm)	–
LD slag (Kg/thm)	–
Quartz (Kg/thm)	–
Manganese (%)	–
Alkali additives (Kg/thm)	–
Alumina additives (Kg/thm)	–
wüstite (%)	–
SiO_2 (%)	–
CaO (%)	–
Cost functions	*Task*
Tuyere cooling heat loss (GJ/h)	Minimization
Total BF gas flow (NM^3/hr)	Maximization
Tuyere velocity (m/s)	Maximization
Heat loss (GJ/hr)	Minimization
Corrected productivity (T/mt^3/day)	Maximization
Coke rate (Kg/thm)	Minimization
Plate cooling heat loss (GJ/hr)	Minimization
Carbon Rate (Kg/thm)	Minimization

how to construct a reliable surrogate model based on the limited amount of data in hand and how to emerge the surrogate without a true objective function. Secondly, how to handle the several objectives simultaneously and efficiently to obtain a set of Pareto optimal solutions. Recently, an algorithm called K-RVEA [124] has been used to advent the scope of the surrogate-assisted optimization in many objective problems to solve computationally expensive objective function. The RVEA algorithms have adopted to optimize problems where objective functions are built using kriging method.

In this study, a running blast furnace was regularly monitored and information was collected on numerous process parameters. It is not readily known which parameters significantly influence the objectives. Therefore, initially significant process variables are identified. There is a conflicting nature between the various objectives having different ranges of values. It is visualized from the objectives, when productivity increases and coke rate decreases, in the same time there is an increase in the blast furnace gas flow. Here, surface plots are shown in Figs. 15 and 16, which define the relationship between various objectives.

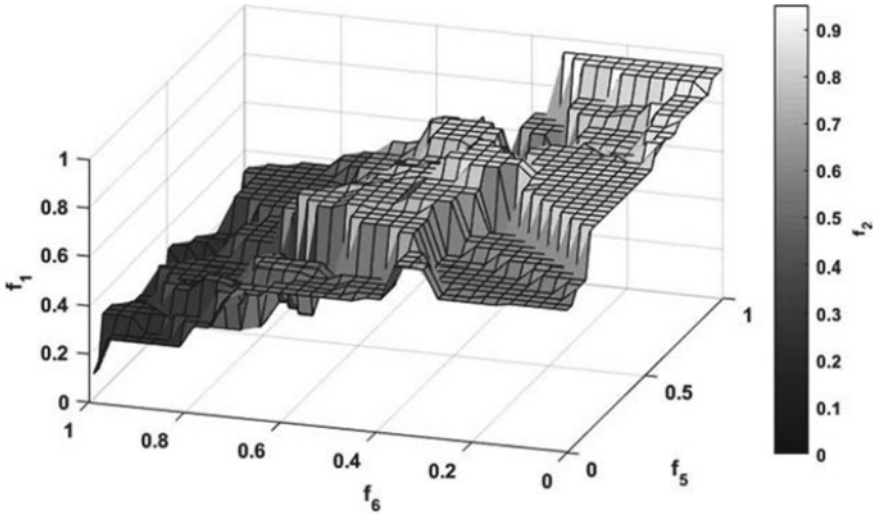


Fig. 15 A surface plot showing relationship between total blast furnace gas flow (f_2) and there three objectives, namely, corrected productivity (f_5), coke rate (f_6), and tuyere heat loss (f_1) [122]

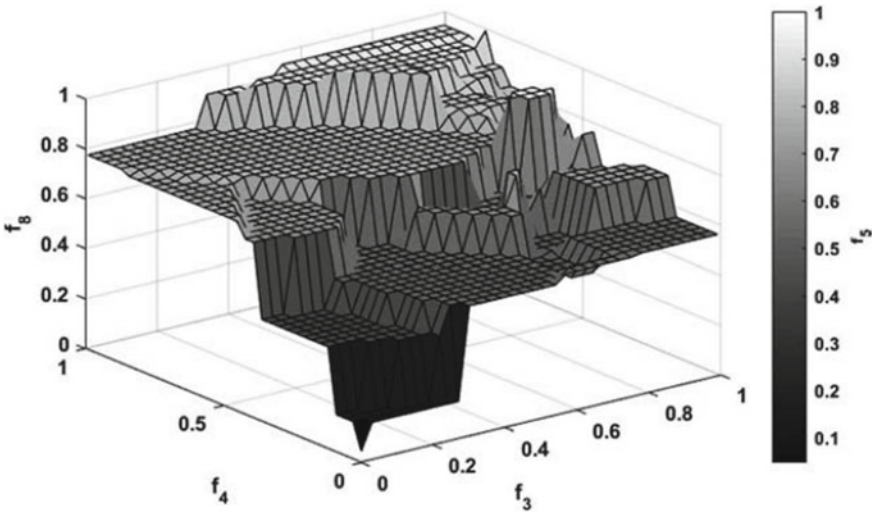


Fig. 16 A surface plot showing relationship between, namely, corrected productivity (f_5) and there three objectives, namely, tuyere velocity (f_3), heat loss (f_4), and carbon rate (f_8) [122]

The present approach uses simple surrogate model with 12 most significant process variables in the same time a very large possible of alternatives that most relevant to minimization or maximization with eight objectives. Surrogate approach makes the optimization very fast and quite acceptable and accurate.

9 Conclusion and Future Prospects

The chapter has reviewed papers on optimization of various processes and input parameters in the field of iron making. Till the date, very less number of papers are there, which are related to this principal area but it can explain magnificently many aspects of the topic pertinent to complex modeling and optimization problem and this can be considered as a very important emerging area in the present-day iron making. Evolutionary algorithms have clear advantages in nonlinear optimization and these methods are used to solve multi-objective problems. This chapter has highlighted the quality scientific research employing a wide range of evolutionary computational methods contributing to different aspects of blast furnace iron making as well as the DRI process. However, real-world applications are few, probably due to the complexity of the process phenomena.

In the future, attention should be focused on the use of evolutionary algorithms in each and every aspect of iron-making field to evaluate result in a faster rate. So, faster algorithms need to be developed, which would require less calculation to reach the optimum. It should be helpful in system control and decision-making of iron-making process.

References

1. The White Book of Steel, *World Steel Association*, 2012. ISBN 978-2-930069-67-8.
2. Ghosh, A., & Chatterjee, A. (2008). *Iron and Steel Making Theory and Practice*. Delhi: PHI Learning Private Limited.
3. Worldsteel Association. Resource Efficiency, http://www.worldsteel.org/en/dam/jct:1568363d.
4. Worldsteel Association. Resource Efficiency, http://www.worldsteel.org/en/dam/jcr:37ad1117.
5. Gugurathi, A. M., & Babu, B. V. *Evolutionary computation techniques and application*.
6. Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Berlin: Springer.
7. Pettersson, F., Chakraborti, N., & Saxén, H. (2007). A genetic algorithm based multi objective neural net applied to noisy blast furnace data. *Applied Soft Computing, 70,* 387–397.
8. Geerdes, M., Chaigneau, R., Kurunov, J., Lingiardi, O., & Ricketti, J. *Modern blast furnace iron making an introduction*. Delft University: IOS Press.
9. Midrex Technology, www.Midrex.com.
10. Hinnela, J., Saxén, H., & Pettersson, F. (2003). Neural network model of burden layer formation dynamic in blast furnace. *Industrial and Engineering Chemistry Research, 42*(11), 2314–2323.
11. Kritzinger, H. P., & Kingsley, T. C. (2015). Modeling and optimization of a rotary kiln direct reduction process. *Journal of the South African Institute of Mining and Metallurgy, 115,* 419–424.
12. Fabian, B. T. (1958). A linear programming model of integrated iron and steel production. *Management Science, 4,* 415–441.
13. Omori, Y. (1987). *Blast furnace phenomenon and modeling*. London: Elsevier.
14. Muchi, I. (1967). Mathematical model of blast furnace. *Transaction of Iron and Steel Institute of Japan, 7,* 223.

15. Peacey, J. G., & Davesport, W. G. The iron blast furnace theory and practice. *Pergamon International Library of Science*. Robort Maxwell, M.C.
16. Rist, A., & Meysson, N. (1967). A dual representation of the blast furnace mass and heat balance. *Journal of Metals, 19,* 50–59.
17. Kilpinen, A. (1988). An on line model for estimating the melting zone in a blast furnace. *Chemical Engineering Science, 43,* 1813–1818.
18. Nath, N. K. (2002). Simulation of gas flow in blast furnace for different burden distribution and cohesive zone shape. *Material and Manufacturing Processes, 17,* 671–681.
19. Dong, X. F., Pinson, D., Zhang, S. J., Yu, A. B., & Zulli, P. (2006). Gas-powder flow in blast furnace with different shape of cohesive zone. *Applied Mathematical Modeling, 30,* 1293–1309.
20. Hatano, M., & Kurita, K. A. (1992). Mathematical model of blast furnace with radial distribution of gas flow, heat transfer and reaction considered. *Transaction of the Iron and Steel Institute of Japan, 22,* 448–456.
21. Zhou, Z., Zhu, H., Yu, A., Wright, B., Pinson, D., & Zulli, P. (2005). Discrete particle simulation of solid flow in a model blast furnace. *ISIJ International, 45,* 1828–1837.
22. Decastro, J. A., Nogami, H., & Yagi, J. (2002). Three dimensional multiphase mathematical modelling of the based on multi-fluid model. *ISIJ International, 42,* 44–52.
23. Adema, A. (2014). DEM. CFD modelling of the iron making blast furnace. *TU Delft*, Delft University of Technology.
24. Hykin, S. S. (2009). *Neural networks and learning machines*. Prentice-Hall; Holland, J. H (1975). *Adaption in nature and artificial systems*. Ann Abbor, MI: The University of Michigan Press.
25. Coello-Coello, C. A., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithm for solving multi objective problems*. New York: Kluwer Academic Publishers.
26. Miettinen, K. M. (1998). *Non linear multi objective optimization*. Buston, Massachusetts: Klwer Academic Publishers.
27. Zadeh, A. L. (1987). *Fuzzy logic and fuzzy systems: Selected paper*. Wiley.
28. Lu, Y. Z. (1996). *Industrial intelligent control*. New York: Wiley.
29. Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. Cambridge, MA: MIT Press.
30. Holland, H. J. (1992). *Adaption in natural and artificial systems*. Cambridge, Massachusetts, London: MIT press.
31. Pettersson, F., Biswas, A., Sen, P., Saxén, H., & Chakraborti, N. (2009). analyzing leaching data for low grade manganese ore using neural nets and multi objective genetic algorithms. *Materials and Manufacturing Processes, 24,* 320–330.
32. Monal, D. N., Sarangi, K., Petterson, F., Sen, P. K., Saxén, H., & Chakraborti, N. (2011). Cu-Zn separation by supported liquid membrane analyzed through multi objective genetic algorithms. *Hydrometallurgy, 107,* 112–123.
33. Collets, P. (2007). Genetic programming in hand book of research on nature inspired computing for economics and management. In: J.-P. Rennrd (Ed.), *Idea: Hershey* (pp. 59–73).
34. Poli, R., Langdon, W. B., & Mcphee, N. F. (2008). *A field guide to genetic programming*. Published via http://lulu.com (With Contributing by Koza, J. R.).
35. Giri, B. K., Pettersson, F., Saxen, H., & Chakraborti, N. (2013). Genetic programming evolved through bi-objective algorithms applied to a blast furnace. *Materials and Manufacturing Processes, 28*, 776–882.
36. Li, X. (2003). A real-coded predator-prey genetic algorithm for multi objective optimization: In: C. M. Fonseca, P. J. Fleming, E. Zitztler, K. Dev, & L. Thiele, (Eds.), EMO lecture notes in computer science, 2632 (pp. 207–221). Berlin: Springer.
37. Wilson, B. (2014). In J. Valadi, P. Siarry (Eds.), *Application of Metaheuristics in Process Engineering*, Switzerland: Springer International Publishing.
38. Kimeme^{TM} (2014). www.kimeme.com.
39. Wu, S., Oliviera, D., Dat, Y., & Xu, J. (2012). Ore blending optimization model for sintering process based on characteristic of iron ores. *International Journals of Minerals, Metallurgy and Metals, 19,* 217–224.

40. Umadevi, T., Karthik, P., Mohapatra, P. C., Prabhu, M., & Ranjan, M. (2013). Optimization of FeO in iron ore sintering at JSW steel limited. *Iron Making and Steel Making, 30*(3), 180–189.
41. Zhang, J., Xie, A., & Shen, F. (2007). Multi objective optimization and analysis model of sintering process based of BP network. *Journals of Iron and Steel Research, 14,* 1–5.
42. Nath, N. K., & Mitra, K. (2013). Optimization of suction pressure for iron sintering by genetic algorithm. *Iron Making and Steel Making, 31*(3), 199–206.
43. Nath, N. K., & Mitra, K. (2007). Mathematical modeling and optimization of two layer sintering process for sinter quality and fuel efficiency using genetic algorithm. *Materials and Manufacturing Processes, 20*(3), 335–349.
44. Mitra, K. (2013). Evolutionary surrogate optimization of an industrial sintering process. *Material and manufacturing process, 28*(7), 768–775.
45. Wang, T., Zhao, J., Sheng, C., Wang, W., & Wang, L. (2016). Multilayer encoding genetic algorithm based granular fuzzy interference for blast furnace gas scheduling. *IFAC Paper Online, 49,* 132–137.
46. Andahazy, D., Loffler, G., Winter, F., Feilmayr, C., & Burgler, T. (2005). Theoritical analysis on the injection of H_2, CO_2, CH_4, rich gases into the blast furnace. *ISIJ International, 45,* 166–174.
47. Andahazy, D., Shaby, S., Loffler, G., Winter, F., Feilmayr, C., & Burgler, T. (2006). Governing process of gas and oil injection into the blast furnace. *ISIJ International, 46,* 486–502.
48. Shaby, S., Andahazy, D., Winter, F., Feilmayr, C., & Burgler, T. (2006). Reducing ability of CO_2 and H_2 of Gases formed in the lower part of the blast furnace by gas and oil injection. *ISIJ International, 46*(7), 1006–1013.
49. Radhakrishanan, V. R., & Mohamod, A. R. (2001). Neural networks for the identification and control of blast furnace hot metal quality. *Journal of Process Control, 10,* 77–85.
50. Chen, J. (2001). A predictive system for blast furnace by integrating a neural network with qualitative analysis. *Engineering Application of Artificial Intelligence, 19,* 77–85.
51. Wang, W. (2008). Application of BP network and principal component analysis to forecasting the Si content in blast furnace hot metal. *Workshop of Intelligent Information Technology Application, 3,* 42–46.
52. Vapnic, V. N. (1995). *The nature of statistical learning theory*. New York, NY USA: Springer.
53. Chang, C. C., Lin, & C. J. Libsum (2001). *A library for support vector machines*.
54. Kronberger, G., Feilmayr, C., Kommenda, M., Winkler, S., Affezeller, M., & Bargler, T. (2009). System identification of blast furnace process with genetic programming. In *IEEE*, 5258751.
55. Guo, H., Zhu, M., Yan, B., Deng, S., Li, X., & Liu, F. (2017). Dynamic regional viscosity prediction model of blast furnace slag based on the partial least squire regression. *The Minerals Metals and Materials Society, 69,* 395–401.
56. Ljung, L. (1999). *System identification* (p. 140). New Jercy: Prentice Hall.
57. Martin, R. D., Obeso, F., Machon, J., Barea, R., & Jeminez, J. (2013). Hot meatal temperature prediction in blast furnace using advance model based on fuzzy logic tools. *Iron and Steel Making, 34*(3), 241–247.
58. Jeminez, J., Machon, J., deAyla, J. S., & Obeso, F. (2004). Blast furnace hot metal temperature prediction through neural networks based models. *ISIJ International, 44,* 573–580.
59. Michael, A., Henson, D. E., & Seborg (1997). *Nonlinear process control*. New Jercy: Prentice Hall (p. 26).
60. Castore, M., Gandolfi, G., Palella, S., & Taspedini, G. (1972). Dynamic model for hot metal Si prediction in blast furnace control. *In process development of iron practice* (pp. 152–159). London, UK.
61. Phadke, M. S., & Wu, S. M. (1974). Identification of multi input-multi output transfer function and noise model of a blast furnace from close loop data. *IEEE Transactions on Automatic Control, 19,* 944–951.
62. Phadke, M. S., & Wu, S. M. (1974). Identification of multi input-multi output transfer function and noise model of a blast furnace from close loop data. *IEEE Transactions on Automatic Control, 19,* 944.

63. Unbehauen, H., & Diekmann, K. (1982). Application of MIMO identification to a blast furnace in IFSC identification and system parameter estimation. In *IFSC* (p. 235), WA.
64. Chao, Y. C., Su, C. W., & Hueug, H. P. (1986). The adaptive auto remain models for the system dynamics and prediction of blast furnace. *Chemical Engineering Communications, 44,* 309.
65. Saxen, H. (1994). Short term prediction of silicon content in pig iron. *Canadian Metallurgical Quarterly, 33,* 319.
66. Östermark, R., & Saxén, H. (1996). Vermax. Modeling of blast furnace process variable. *European Journal of Operational Research*, *90*, 85.
67. Saxén, H., & Östermark, R. (1996). State realization with exogenous variables test on blast furnace data. *European Journal of Operational Research, 89,* 24.
68. Chung, Y. H., Chou, R. J., Chang-Chang, G. M., & Chen, G. L. (1986). An auto regressive model for the system dynamics of blast furnace. *Journal of the Chinese Institute of Chemical Engineers, 17,* 107–111.
69. Zeng, J. S., & Gao, C. H. (2009). Improvement of identification of furnace iron making process by outlier detection and missing value imputation. *Journal of Process Control, 19,* 1519–1528.
70. Sheel, P.,& Deo, B. (1999). Production of silicon content of hot metal using fuzzy-GA regression. GECCO. In *Ist Annual Conference on Genetic and Evolutionary optimization* (Vol. 2, pp. 1685–1690).
71. Lue, S. H., Huang, J., Zeng, J. S., & Zhanyng, Q. S. (2011). Identification of the optimal control center for ballast furnace thermal state based on the Fuzzy–C-means clustering. *ISIJ International, 51,* 1668–1673.
72. Jian, L., Gao, C. H., Li, L., & Zeng, S. (2008). Application of Least squire support vector machine to predict the Si content in blast furnace hot metal. *ISIJ International, 48,* 1659.
73. Chen, J. (2001). A predictive system for blast furnace by integrating a neural network with qualitative analysis. *Engineering Applications of Artificial Intelligence, 14,* 77–85.
74. Chen, W., Wang, B. X., & Han, H. L. (2010). Prediction and control for silicon content in pig iron of blast furnace by integrating artificial neural network with genetic algorithm. *Ironmaking & Steelmaking, 37,* 458–463.
75. Saxen, H., Pettersson, F., & Gunturu, K. (2007). Evolving nonlinear time series model of the hot metal silicon content in the blast furnace. *Material and Manufacturing Processes, 225,* 577–584.
76. Saxén, H., & Pettersson, F. (2007). Nonlinear prediction of the hot metal silicon content in the blast furnace. *ISIJ International, 47,* 1732–1737.
77. Nurkhala, A., Pettersson, F., & Saxén, H. (2011). Nonlinear modeling method applied to prediction of hot metal silicon in the iron making blast furnace. *Industrial & Engineering Chemistry Research, 50,* 9236–9248.
78. Saxén, J., Saxén, H., & Toivonen, H. T. (2016). Identification of switching linear system using self-organizing models with application to silicon prediction in hot metal. *Applied Soft Computing, 47,* 271–280.
79. Biswas, A. (1981). *Principle of blast furnace iron making*. Brisbane, Australia: Cootha Publishing House.
80. Omori, Y. (Ed.). (1987), *Blast furnace phenomena and modeling*. London: Elsevier.
81. Lizuka, M., Kajikawa, S., Nakatani, G., Wakmoto, K., & Matsumura, K. (1980). Nippon Kokan Tech. *Rep, Overseas, 30,* 13.
82. Ohno, J., Yashino, H., Shirakawa, Y., Tsuda, A., Watanabe, S., Hirata, T., et al. (1986). In: *Proceedings of IFAC Conference in Automation in Mining, Mineral, and Metal Processing* (p. 353), IFSC, Laxenberg.
83. Iwamura, T., Sahimura, H., Maki, Y., Kawai, T., & Asano, Y. (1982). *Transactions of the Iron and Steel Institute of Japan, 22,* 764.
84. Mitra, T., & Saxén, H. (2014). Evolution of charging programs for achieving required gas temperature profile blast furnace. *Materials and Manufacturing Processes, 30*(4), 474–487.

85. Zhou, Z., Zhu, H., Yu, A., Write, B., Pinson, D., & Zull, P. (2005). Discrete particle simulation of solid flow in model blast furnace. *ISIJ International, 45,* 1828–1837.
86. Natsui, S., Nogami, H., Veda, S., Kano, J., Inoue, R., & Ariyama, T. (2011). Simulation three dimensional analysis of gas solid flow in blast furnace by combining discrete element method and computational fluid dynamics. *ISIJ International, 51,* 41–50.
87. Xu, J., Wu, S., Kou, M., Zhang, L., & Yu, X. (2011). Circumferential burden distribution generation at bell less top blast furnace with parallel type hoppers. *Applied Mathematics Modelling, 35,* 1439–1455.
88. Pettersson, F., Saxén, H., & Hanneia, J. (2005). A genetic algorithm evolving charging program in the iron making blast furnace. *Materials and Manufacturing Processes, 20*(3), 351–361.
89. Mitra, T., & Saxén, H. (2014). Model for fast evolution of charging programs in the blast furnace. *Metallurgical and Materials Transactions, 45*(6), 2382–2394.
90. Hinnela, J. (2003). *Data driven modeling of burden distribution in the blast furnace*. Ph.D. Thesis. Heat Engineering Laboratory, Abo Academy University.
91. Pettersson, F., Hinnela, J., & Saxén, H. (2003). Evolutionary neural networks modeling of blast furnace burden distribution. *Materials and Manufacturing Processes, 18*(3), 385–399.
92. Mitra, T., Pettersson, F., Saxén, H., & Chakraborti, N. (2016). Blast furnace charging optimization using multi objective evolutionary and genetic algorithms. *Materials and Manufacturing Processes, 32*(10), 1179–1188.
93. Mohanty, K., Mitra, T., Saxén, H., & Chakraborti, N. (2016). Multicriteria in a top gas releasing blast furnace optimize through a k-optimality based algorithm. *Steel Research International, 87,* 1284–1294.
94. Hu, C., Han, X., Li, Z., & Zhang, C. (2009). Comparison of CO_2 emission between COREX and blast furnace iron making system. *Journal of Environmental Sciences, 21,* 116–120.
95. Austin, P. R., Nogami, H., & Yagi, J. (1998). Prediction of blast furnace performance with top gas recycling. *ISIJ International, 38,* 239–245.
96. Nogami, H., Chu, M., & Yagi, J. (2005). Numerical analysis of blast furnace operation with top gas. *Rev. Material-Paris*, 102–189.
97. Murai, R., Sato, M., & Ariyama, T. (2004). Design of innovative blast furnace for minimizing CO_2 emission based on optimization of solid fuel injection and top gas recycling. *ISIJ International, 44,* 2168–2177.
98. Helle, H., Helle, M., Saxén, H., & Pettersson, F. (2010). Optimization of top gas recycling condition under high oxygen enrichment in the blast furnace. *ISIJ International, 50,* 931–938.
99. Wang, C., Rayman, C., & Dahl, J. J. (2009). Potential CO_2 emission reduction for BF-BOF steel making based on optimize use of ferrous burden material. *Green House Gas Control, 3,* 29–36.
100. Ghanbari, H., Pettersson, F., & Saxén, H. (2015). Optimal operation strategy and gas utilization in a future integrated steel plant. *Chemical Engineering Research and Design, 102,* 322–336.
101. Pettersson, F., Saxén, H., & Deb, K. (2009). Genetic algorithm based multicriteria optimization of iron making in blast furnace. *Materials and Manufacturing Processes, 24,* 343–349.
102. Poli, R., Langdon, W. B., & Mephee, N. F. (2008). *A field guide to genetic programming*. Published via http://lulu.com (With Contributing Koza, J. R.).
103. Mitra, T., Helle, M., Pettersson, F., Saxén, H., & Chakraborti, N. (2011). Multi objective optimization of top gas recycling condition in the blast furnace by genetic algorithm. *Materials and Manufacturing Processes, 26*(8), 475–480.
104. Agrawal, A., Tiwari, U., Pettersson, F., Das, S., Saxén, H., & Chakraborti, N. (2010). Analyzing blast furnace data using evolutionary neural network and multi objective genetic algorithm. *Iron Making and Steel Making, 37,* 353–359.
105. Jha, R., Sen, P. K., & Chakraborti, N. (2014). Multi objective genetic algorithm and genetic programming models for minimizing input carbon rates in a blast furnace compared with a conventional analytic approach. *Steel Research International, 85*(2), 219–232.
106. Deb, K. (2001). *Multi objective optimization using evolutionary algorithms*. Chi Chester: Wiley.

107. Poles, S., Vassileva, M., & Sasaki, D. (2008). Multi objective optimization. In: J. Branke, K. Dev, K. Miettinen, & R. Slowinski, (Eds.), *Multi objective optimization lecture notes in computer science* (Vol. 5252, pp. 329–348). Springer.
108. Haywood, R., Tayler, W., Plikas, N., & Warnica, D. (2009). Enhanced problem solving, the integration of CFD and other engineering application. In *7th International Conference on CFD in the Minerals and the Process Industries*, CSIRO, Melbourne, Australia.
109. Gasevic, D., Kaviani, N., & Hatala, M. (2007). Metamodeling in mega models, model driven engineering language and systems. In *10th International Conference* (pp. 91–1005), Nashville, Tennessee.
110. Stander, N., & Craig, K. (2002). On the robustness of a simple domain reduction scheme for simulation based optimization. *Engineering computation, 19,* 431–450.
111. Croft, T. N., Cross, M., Slone, A. K., Williams, A. F., Bennett, C. R., Blot, P., et al. (2009). CFD analysis of an induration cooler on an iron are grate kiln pelletizing process. *Minerals Engineering, 22,* 859–873.
112. Mohanty, D., Chandra, A., & Chakraborti, N. (2009). Genetic algorithm based multi objective optimization of an iron making rotary kiln. *Computational Materials Science, 45,* 181–188.
113. Hunty, W. P., Lee, G. K., & Price, J. T. (1991). Fundamentals of coal combustion during injection into a blast furnace. *Progress in Energy and Combustion Science, 17,* 373–395.
114. Chen, C. W. (2005). Numerical analysis for the multi phase flow of pulverized coal injection inside blast furnace tuyere. *Applied Mathematical Modeling, 29,* 871–884.
115. Shen, Y. S., Gua, B. Y., Yu, A. B., Austin, P. R., & Zulli, P. (2011). Three dimensional modelling of in furnace coal/coke combustion in a blast furnace. *Fuel, 90,* 728–738.
116. Shen, Y. S., Yu, A. B., Austin, P. R., & Zulli, P. (2012). CFD study of the furnace effects of operating condition. *Powder Technology, 223,* 27–38.
117. Lacroix, P., Dauwels, G., Dufresene, P., Godijn, R., Perini, P. G., & Stricker, K. P. (2001). High blast furnace productivity operations with low coke rates in the European Union. *Metallurgical Research & Technology, 3*(98), 259–268.
118. Norgate, J., & Langberg, P. (2009). Environmental and economic aspects of charcoal use in steel making. *ISIJ International, 49*(4), 587–595.
119. Wiklund, C. M., Helle, M., Saxén, H., & Pettersson, F. (2016). Economic assessment of options for biomass pretreatment and use in the blast furnace. *Biomass and Bioenergy, 91,* 259–270.
120. Helle, H., Helle, M., Saxén, H., & Pettersson, F. (2009). Mathematical optimization of iron making with bio mass as auxiliary reductant in the blast furnace. *ISIJ International, 49,* 1316–1324.
121. Wiklund, C. M., Helle, M., Kohl, T., Järvinen, M., & Saxen, H. (2017). Feasibility study of woody bio mass use in a steel plant through process integration. *Journal of Cleaner Production, 142,* 4127–4141.
122. Chugh, T., Chakraborti, N., Sindhya, K., & Jin, Y. (2017). A data driven surrogate assisted multi objective evolutionary algorithm applied to a many objective blast furnace optimization problems. *Materials and Manufacturing Processes, 32*(10), 1172–1178.
123. Wang, H., Jin, Y., & Jansen, J. O. (2016). Data driven surrogate assisted multi objective optimization of a trauma system. *IEEE Transaction on Evolutionary Computation*.
124. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., & Sindhya, K. (2018). A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation, 22*(1), 129–142.

Data-Driven Design Optimization for Industrial Products

Swati Dey, N. Gupta, S. Pathak, D. H. Kela and Shubhabrata Datta

Abstract Prescriptive data analytics along with optimization techniques in complex industrial processes lead to extraction of useful knowledge from the data, mapping of the relation between the inputs and outputs of the products and/or processes, product quality improvement, cost reduction, process simplification and designing new product with improved performance. There are several statistical methods and soft computing techniques, which are useful to generate the objective functions for optimization from industrial data. In this chapter, a case study is discussed where ANN model with desirability function forms the objective function. GA-based search is employed to get the optimized solutions, and to find the behaviour of the alloying elements in steel with desired performance.

Keywords Data-driven design · Statistical methods · Artificial neural network Desirability function · Genetic algorithm · Industrial application

1 Introduction

Applications of optimization techniques in complex industrial processes and product qualities are often restricted due to unavailability of suitable mathematical model correlating the independent variables with the dependent variables, which can be utilized as the objective function. In case of a highly multivariate situation involving non-linear relationship among the inputs and outputs, the analytical models in many

S. Dey (✉)
Indian Institute of Engineering Science and Technology, Shibpur 711103, India
e-mail: swatidey@yahoo.com

N. Gupta
Birla Institute of Technology and Science, Pilani 33303, India

S. Pathak · D. H. Kela
Texmaco Rail and Engineering Ltd., Belgharia 700056, India

S. Datta
SRM Institute of Science and Technology, Kattankulathur, Chennai 603203, India

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_9

cases are not adequate to serve a purpose, as these type models involve many assumptions. Such assumptions lead to poor predictions, not acceptable for many industrial applications. In such cases, data-driven models may provide the ideal solution. Huge improvement in the field of data mining or data analytics in the recent years has developed several tools to extract useful knowledge from the data, as well as to map the relation between the inputs and outputs of the products and/or processes. There are several statistical methods, e.g. regression analyses, which are used for this purpose long time for different industrial processes, particularly in metallurgical industries [1]. Later newer techniques, grouped in the name of machine learning, soft computing, artificial intelligence or computational intelligence are developed [2]. These techniques have better capability to handle complex, non-linear multivariate systems. In the industrial case study cited in this chapter, one of most widely used tool in the above group, viz. artificial neural network (ANN) [3–5], is used to develop models using industrial data for predicting the mechanical properties of steel used for some special purposes. These ANN models constitute the basic components for the formation of the objective functions, leading to an amalgamation of data-driven modelling and design optimization concepts. Industrial requirements for the materials properties are most of the time quite stringent and must be within a narrow allowable limit. To achieve the combination of properties with such stringent limits in the most viable way, optimization techniques could be employed. In the present work genetic algorithm [6–8], the most widely used global optimization technique, is used for the optimization. But to find suitable combinations of composition and processing parameters, in this case, the heat treatment parameters, for which all the properties will fit within the desired range, the problem could not be formulated as a simple optimization process in a single or multi-objective mode. Because none of the properties here are being minimized or maximized in this work. So, in this unique approach of design optimization of the composition of the steel to achieve certain performance level, where the properties of the steel are mapped with the input variables through ANN models, the ANN outputs are converted between 0 and 1, depending on its closeness to the desired value, through desirability function [9]. If the output is exactly the desired value then its desirability is 1, and the desirability is reduced on either side of the desired value. To get the overall desirability of the performance of the steel, all the desirability values of the individual properties are converted to composite desirability, which is the geometric mean of all the desirability values. Finally, the composite desirability is optimized using GA.

2 Data-Driven Design and Prescriptive Analytics

The concept of data analytics (DA) leads to a method called data-driven design (D3) of products. DA is the process of investigating data sets in order to conclude regarding the information contained in it. Continuous improvement in the computational techniques particularly in the domain of artificial intelligence and machine learning has made DA an extremely effective tool. DA techniques are widely used

nowadays in commercial industries to enable the organization make more-informed business decisions. Four types of DA are commonly used, viz. descriptive, diagnostic, predictive and prescriptive data analytics [10]. Descriptive analytics is a method for answering 'what has happened'. Several basic statistical analyses, along with decision tree, cluster analysis, support vector machine can be used in this level of data analytics. In the next stage of analysis, called diagnostic analytics, the question 'why' is addressed. In industrial situations, analysis of the cause of defects in any product may be an ideal example. Principal component analysis and rough set analysis are some of the tools capable of executing such type of analysis. Predictive analytics is nothing but the well-known approach of developing correlations among the dependent (output) and independent (input) variables. All techniques capable of generating data-driven models, e.g. multiple linear/non-linear regression, artificial neural network, neuro-fuzzy inference system, etc. The last stage of data analytics is the prescriptive analytics, through which the solution of the problem may be suggested. Optimization techniques like genetic algorithm, and methods of multi-criteria decision making may be used for searching such solutions.

In the case of industrial products, the concept of prescriptive analytics and the tools used in the process can lead to design optimization of the product. The industrial data used for such designing of the products with improved performance can be called as the data-driven design (D3).

2.1 Sources of Data

In case of data-driven design for industrial products, the best source of data should be the data generated from the particular setup itself. In that case, the plant constraints will be reflected in the data and incorporating such constraints and limitations in the design process will be obvious. The availability of data in the industrial scenario depends on the level of automation of the plant. It can vary from manual record keeping to comprehensive digital data warehousing. A systematic and dependable record keeping system for the whole production process will be handy for the above data analytics. It can help improving the product quality, reducing the defects, decreasing the production cost and even designing a new product. Sometimes designing a whole new product may need a research and development set-up of the industry, which can use this data analytics method for designing and in the process reduce the number of trials, and thus the expenditures in term of time and money. In such product design even data published by others, including academic researches may also be incorporated.

2.2 Statistical Methods—Examples in Materials Design

In case of materials design, statistical models are used by researchers for many decades. These models are mostly multiple linear (pseudo-linear) regression (MLR) models. The Hall–Petch relation describing the effect of grain size on the strength of an alloy is the most popular example. Other than that, there are many other models describing the transformation temperatures of alloys, mostly steel, which are useful for designing the composition and the heat treatment schedule [11]. These models include transformation temperatures during continuous cooling (and heating) in steel and isothermal transformation start temperatures in steel. There are models of carbon equivalent in steel and molybdenum equivalent in titanium alloys also. Several researchers have even developed MLR models for strength (and hardness) prediction for alloys.

2.3 Soft Computing Methods—Examples in Materials Design

In case of designing materials, soft computing techniques are being used frequently during the last few decades [12, 13]. The tools used most frequently are ANN [14, 15] and GA [16, 17]. Other than those two, fuzzy inference systems [18], rough set analysis [19, 20] and genetic programming [21] are also used quite frequently. The purpose of using such tools, considering the outcome, is different in most of the cases. The requirement from the data and approach for designing the product decides the selection of the tools. Developing data-driven model using ANN and using those models for GA-based optimization is an effective approach for handling industrial data for designing. ANN models can capture the non-linearity of the system quite easily and map the input–output relationship of a highly complex materials system, and thus gaining popularity in the domain of materials engineering. GA being an established optimization tool can handle the ANN models effectively as the objective functions in case of materials design optimization, in single-objective or multi-objective optimization problems.

3 Case Study

In this book chapter, a case study of the method of materials' designing is reported in the following sections, where ANN and GA has been used in tandem, as mentioned above, to design steel with specific performance level for specific applications. Before the results are discussed the methods of ANN and GA are briefly discussed.

Table 1 The range of values of the composition and process variables along with the applications and desired values of the properties

Grade	Steel A	Steel B	Steel C	Steel D
%C	0.28–0.33	0.2–0.3	0.32 max	0.26–0.32
%Mn	0.8–0.10	0.6–1.1	1.85 max	1.4–1.7
%Si	0.40–0.60	0.3–0.5	1.5 max	0.35–0.8
%S	<0.03	<0.03	<0.03	<0.03
%P	<0.03	<0.03	<0.03	<0.03
%Ni	0.5–0.6	–	–	–
%Cr	0.5–0.8	–	–	–
%Mo	0.15–0.25	–	–	–
%Al	0.03–0.05	0.03–0.05	0.03–0.05	0.03–0.06
Heat treatment	Normalizing, quenching and tempering	Normalizing	Normalizing and tempering	Normalizing, quenching and tempering
Component	Coupler component	Side frame, bolster	Pivot	Draft gear
Physical properties	UTS: 828 MPa YS: 690 MPa %El: 14 %Ra: 30 BHN: 241–311	UTS: 505 MPa YS: 303 MPa %El: 20 %Ra: 30	UTS: 621 MPa YS: 414 MPa %El: 22 %Ra: 45 BHN: 179–241	UTS: 655 MPa YS: 448 MPa %El: 10 %Ra: 25 BHN: 217–289

3.1 Materials and Objectives

In this work, four grades of steel were considered. They are Steel A, B, C and D, with a total number of data for these four grades were 1152, which are completely generated in the producing plant in their testing facility. Table 1 gives the range of compositional and processing parameters for the steel grades. It also includes the application of the steel with their target properties.

The primary objective of the present work is to develop ANN models for the mechanical properties of the steel (all four grades), viz. yield strength (YS), ultimate tensile strength (UTS), percent elongation (%El), percent reduction in area (%Ra) and hardness (expressed as BHN). These models will be able to predict the strength (YS and UTS), ductility (%El and %Ra) and hardness of the steel, when the composition and processing parameters are provided. The user will get a prior assessment of the properties before the heat treatment of steel, or even plan to modify the heat treatment schedule a priori. Then GA is used to search the composition of the steel to achieve a target property. Here, ANN is combined with desirability functions to make the objective functions for achieving targeted level of properties. Finally, in this chapter, some arbitrary performance level is set to extract knowledge from the database and how the composition and other variables are behaving in such situation is studied, purely from an academic perspective. The database includes nine compositional and two processing parameters as inputs and five properties as outputs.

3.2 Artificial Neural Network

ANN used here is feed forward multilayered perceptron backpropagated with scale conjugate gradient algorithm. The normalization of the inputs and outputs is done using the relation shown in Eq. (1).

$$x^N = \frac{2(x - x_{\min})}{x_{\max} - x_{\min}} - 1 \tag{1}$$

where the normalized value of a variable x is x^N. The maximum and minimum values of x are x_{max} and x_{min}.

Hyperbolic tangent transfer function is operated on the weighted combination of the inputs $\left(x_j^N\right)$ as shown in Eq. (2) to create the output of the hidden neurons (h_i). The suffixes 'i' and 'j' are used to denote the input and hidden neuron numbers, respectively.

$$\mathrm{h_i} = \tanh\left(\sum_{\mathrm{j}} w_{\mathrm{ij}}^{(1)} x_{\mathrm{j}}^{\mathrm{N}} + \theta_i^{(1)}\right) \tag{2}$$

The output is then calculated through a linear weighted summation of the outputs of the hidden units, as given in Eq. (3):

$$\mathrm{y} = \sum_{\mathrm{i}} \mathrm{w_i^{(2)} h_i} + \theta^{(2)} \tag{3}$$

In the above equations, y = output, x_j^N = normalized inputs, h_i = output from hidden units, w_{ij}, w_i = weights and θ = bias. Different outputs can be obtained by changing the weights w_{ij} in Eqs. (2) and (3). The weights and biases are optimized by backpropagation of the error.

3.3 Genetic Algorithm

Genetic algorithm (GA) is a nature-inspired computing technique, where the Darwinian concepts of natural selection are followed to find the optimum solution through population-based searching. It is basically stochastic global search method, which produces better solutions using the principle of natural selection and survival of the fittest. A new set of solutions is created by this process at each generation, where the individuals are selected according to their level of fitness. Then breeding of the fit solutions is done using operators named selection, crossover and mutation. This process leads to the evolution of populations of individuals that are better suited to their environment than their predecessors, just as in natural adaptation. The

process starts with initializing random population consisting of potential solutions called chromosomes representing individuals. The fitness is obtained from the evaluation of the solution through objective function. Once the fitness value is evaluated, the population goes through the selection process. Then the crossover operator is employed on selected solutions to generate new solutions or individuals. It is basically a process of reproduction where two solutions swap their partial structure. As a result, a pair of offspring solution is created. The mutation operator is applied to supply diversity in the solution set.

3.4 Desirability Function

Desirability scale converts the actual attribute value to a dimensionless scale, d. The method is used for the optimization of a multiple response system. The desirability function generating any value lesser the highest or desired value indicates possibility of improvement of the product quality. In a two-sided desirability function, the target is bound between the lower and upper limits. Equation (4) shows the formulation.

$$\left.\begin{aligned} d &= \left[\frac{Y - Y_{min}}{T - Y_{min}}\right]^{r1}, \; Y_{min} < Y \leq T \\ &= \left[\frac{Y_{max} - Y}{Y_{max} - T}\right]^{r2}, \quad T < Y \leq Y_{max} \\ &= 0, \qquad\qquad Y \langle Y_{min} \text{ or } Y \rangle Y_{max} \end{aligned}\right\} \tag{4}$$

where d is the desirability ($0 \leq d \leq 1$), Y is the actual performance, Y_{max} and Y_{min} are the maximum and minimum values and T is the most desirable value or the target value ($Y_{min} \leq T \leq Y_{max}$). The general form of functional relationship between 'output' values (y) and desirability scale is shown in Fig. 1. In the figure, the following five points in the x-axis are shown.

y_{LSL}: Lowest specification of y
y_{USL}: Highest specification of y
y_T: Most desirable value (target) of y
$y_{LSL\text{-}SR}$: Lowest specification of y for a given target
$y_{USL\text{-}SR}$: Highest specification of y for a given target

The desirability function for *i*th variable, lying between 0 and 1, is then calculated by the set of Eq. (5).

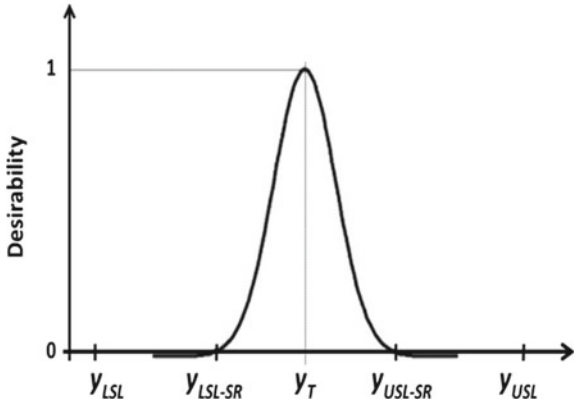


Fig. 1 The desirability function and output variables

$$d_i = \begin{cases} 0 & if\ y_i < y_{LSL_{SR_i}}\ or\ y_i > y_{USL_{SR_i}} \\ \frac{f(y_i) - f\left(y_{LSL_{SR_i}}\right)}{f\left(y_{T_i}\right) - f\left(y_{LSL_{SR_i}}\right)} & if\ y_{LSL_{SR_i}} \le y_i < y_{T_i} \\ \frac{f(y_i) - f\left(y_{USL_{SR_i}}\right)}{f\left(y_{T_i}\right) - f\left(y_{USL_{SR_i}}\right)} & if\ y_{T_i} < y_i \le y_{USL_{SR_i}} \\ 1 & if\ y_i = y_T \end{cases} \tag{5}$$

A composite desirability function was framed taking the geometric mean of the desirability values of each of the required material properties to be achieved as shown in Eq. (6):

$$D_c = \sqrt[n]{D_1 \times D_2 \times \cdots \times D_n} \tag{6}$$

Here, D_c is the value of the composite desirability function while D_1, D_2, …, D_n are the value of the desirability functions of the different properties. The objective function of the optimization process is the maximization of D_c.

3.5 Results and Discussion

Five ANN models were developed considering one output variable at a time. Table 2 presents the total number of data used for modelling and the number of hidden nodes used to get a good fit between target and achieved values of the output variables. The number of hidden nodes in the ANN models with single hidden layer are selected through trial with varying number of nodes in a wide range.

The scatter plots showing the target versus the achieved values are given in Fig. 2. The database having four grades of steel having four different levels of strength

Table 2 Data count and the number of hidden nodes used for modelling

Output variable	YS	UTS	%Elongation	%Reduction in area	Hardness
Total number of data	1147	674	1098	745	304
Number of hidden nodes	6	6	6	16	9

together, and thus the target as well as achieved values form four distinct clusters in the plots for the strength models (Fig. 2a, b). In the case of hardness model, the number of clusters is three (Fig. 2c), as the Steel B did not have the data for hardness. In case of the plots from the ductility models (Fig. 2c, d), the clusters are not that distinct. The ANN models provide good prediction of the properties of all the grades of steel. But in order to design the composition and/or heat treatment parameters towards achieving some target properties, the ANN models are combined with desirability functions and used as objective functions in the optimization process. The methodology adopted is discussed here in detail.

The requirement of the industry may vary from achieving a predefined range of any property to achieving targeted overall performance of the steel consisting of more than one property. In such cases, the requirement is to design the composition of the steel along with the processing. Sometimes, the searching may be more constrained, where the steel is given and only the suitable heat treatment route needs to be designed. All these designs need to be done keeping the mill constraints in consideration. Thus, the work must be such a way that it must be able to decide either ways, i.e. if the inputs are known then the output can be told and vice versa. Here, we bring in the concept of desirability function which is very much useful for industry situations. A desirability value between 0 and 1 gives a chance to improve the product quality. For solving multiple performance characteristics optimization problems when all the objectives are required to attain a specific goal simultaneously, their desirability function approach can be a potent solution. A multiple performance optimization problem can be converted to optimization of a single response problem, as the objective function is the generated through composite desirability. In this work, the outputs of the ANN models describing the properties of the steel are converted to individual desirability values. The limits of the desirability functions are set as per range of properties predefined by the industry, as described in Table 1, GA can be used to optimize the desirability value and the solutions will provide parameters to achieve the desired properties. In case of achieving certain levels in more than one property, the composite desirability is calculated combining the individual desirability values, as mentioned in Sect. 3.4. The desirability concept is incorporated in the SGA code for single objective optimization developed by IIT Kanpur Genetic Algorithm Laboratory, with necessary additions and modifications. Individual desirability values are calculated for the outputs YS, UTS, %elongation, %Ra and hardness. The composite desirability is the geometric mean of the desirability of the individual

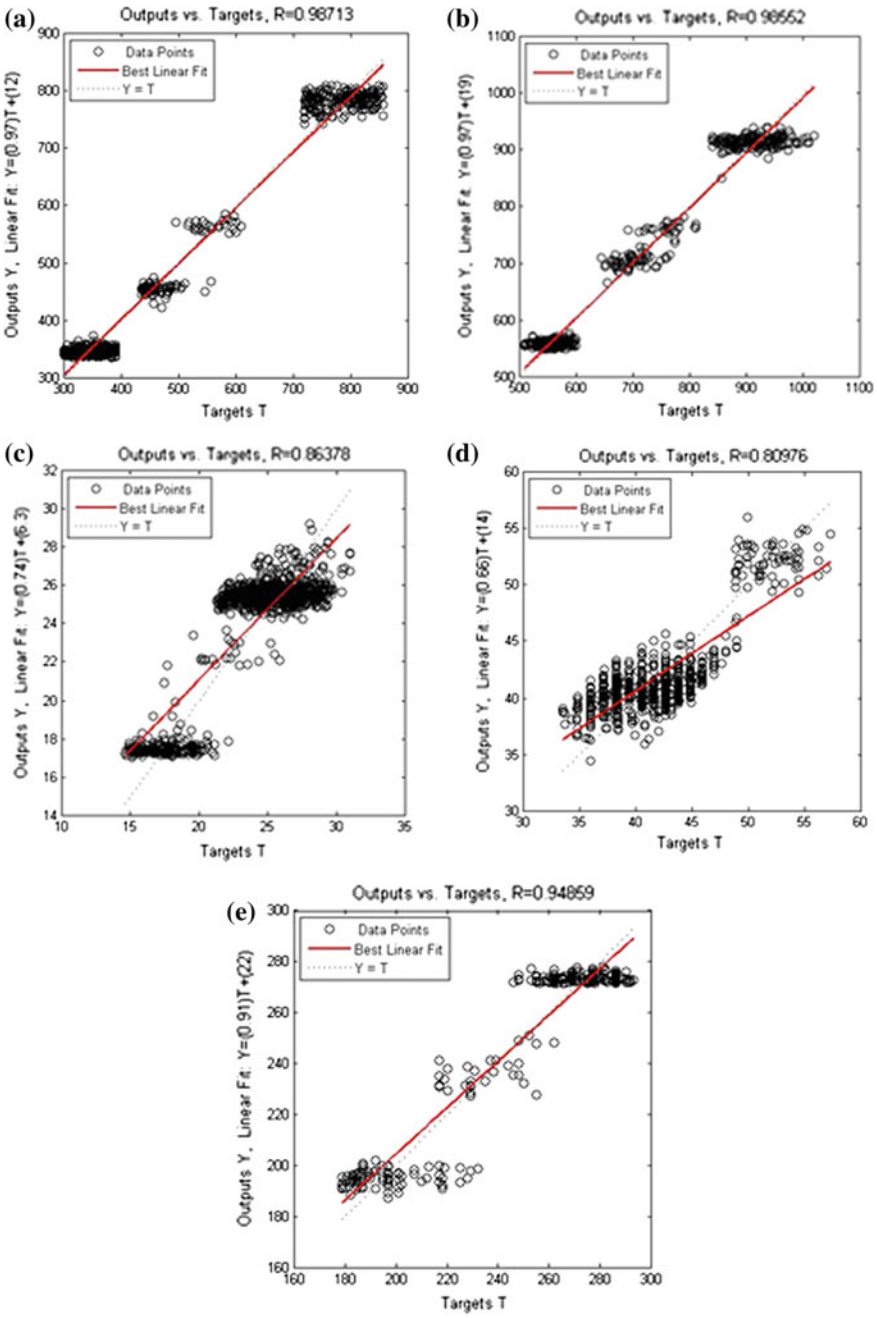


Fig. 2 Scatter plots showing the prediction capability of the ANN models predicting different properties, viz. **a** YS, **b** UTS, **c** %El, **d** %Ra and **e** Hardness

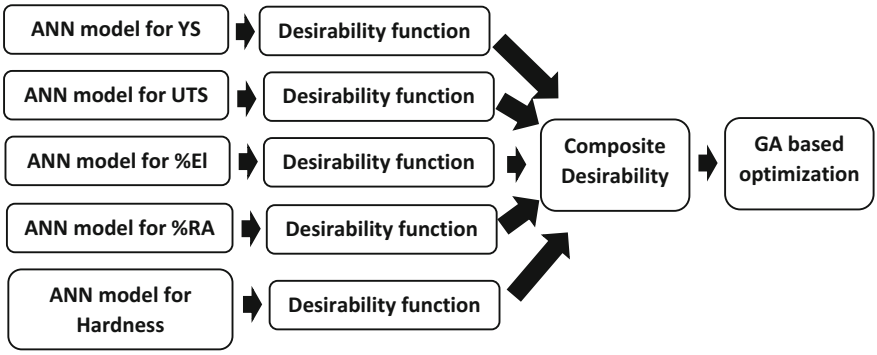


Fig. 3 Scheme of computation using ANN, Desirability function and GA in tandem

attributes. The objective function is the maximization of this composite desirability. The scheme of the computational approach is shown in Fig. 3.

The method adopted for searching the optimum solutions with maximum desirability or composite desirability is found to be efficient enough for computing all types standard requirement of the industry. But to find the efficacy of the method for finding the arbitrary combination of properties, several experimentations on the computer program are executed. In all the studies, the modelling approach are found within acceptable error level, and also found to be following the fundamentals of physical metallurgy principles. These findings gave the opportunity to use the computational approach for extracting further knowledge from the given dataset. In an attempt to get deeper understanding of the four steel grades and their behaviour, the method is applied for finding the desirability values of the steel properties for increasing the desired ductility of the steel keeping the desired YS and UTS fixed (Fig. 4). The procedure is implemented for all four grades of steel separately, and the desired YS and UTS are kept as the corresponding values for those grades, as mentioned in Table 1. Figure 4a shows the variation of desirability values of YS, UTS and %El for varying desired values of %El for Steel A. It is seen that for this grade of steel, it is easier to achieve the desired strength values for all desired ductility values. But lower ductility could not be achieved through the GA-based searching maintaining the strength level same. Presumably the achieved strength level corresponding to such low desired values of ductility is quite high. Thus the compromised solutions the searching process could produce, particularly for the strength properties, had lower desirability values compared to those in higher ductility. In the case of Steel B, the situation is different (Fig. 4b). Here, it could be seen that for the optimum solutions of %El (nearest to its desired values), the desirability values of UTS is quite high. But both YS and %El have moderate desirability values. For Steel C, the desirability values of both strength properties are comparatively high (Fig. 4c). But in Steel D, the desirability value of ductility improves gradually with higher values of desired ductility (Fig. 4d). Here also, it seems that at lower ductility, searched solutions had more strength than their desired values. Thus, this process could search the possible

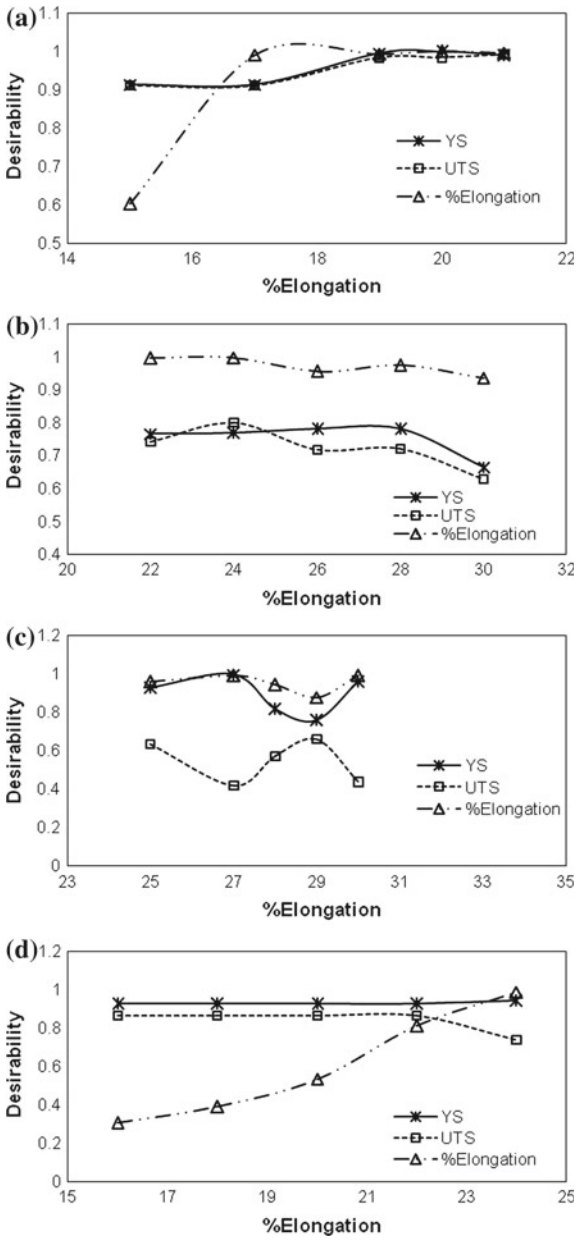


Fig. 4 The optimum desirability values of YS, UTS and %El achieved for varying desired %El in **a** Steel A, **b** Steel B, **c** Steel C and **d** Steel D

solutions which are better than the desired properties set by the industries. These solutions with superior performance could be achieved through some modifications of the compositions and processing.

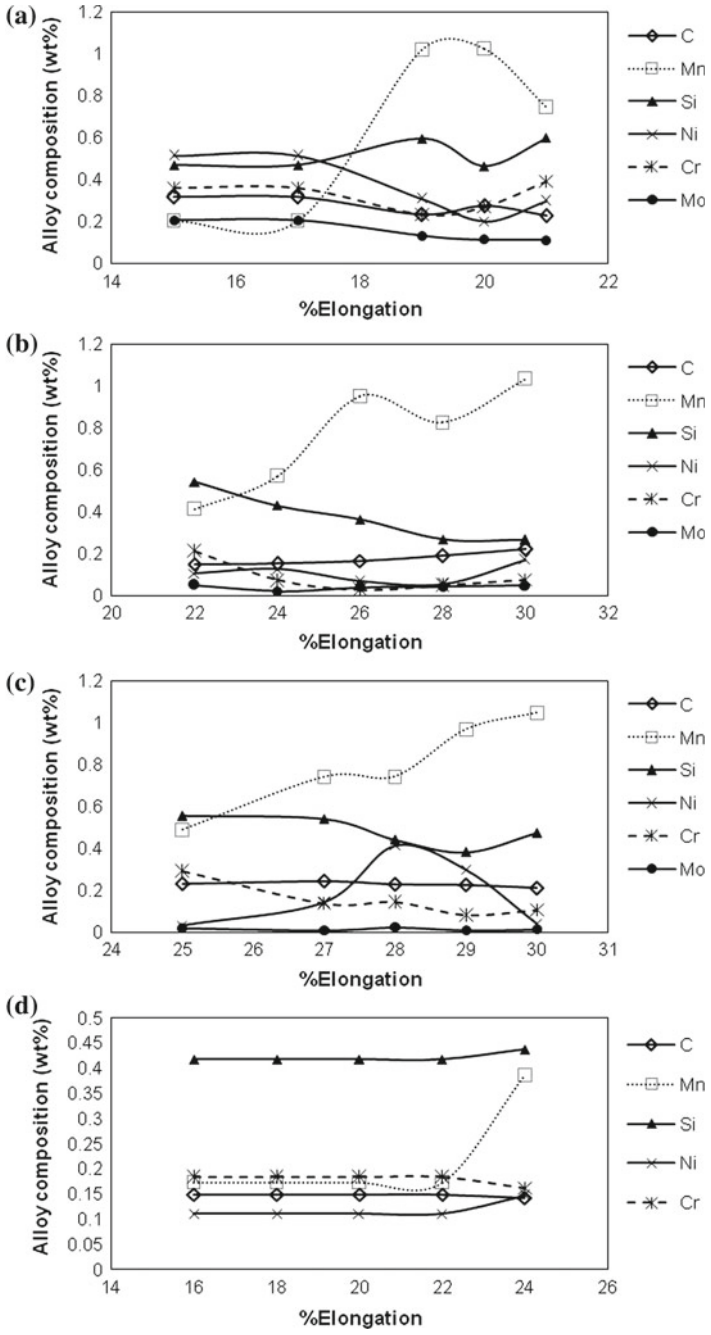


Fig. 5 Variation of the element concentrations in the solution with optimum desirability for varying desired %El in **a** Steel A, **b** Steel B, **c** Steel C and **d** Steel D

In order to see the variation in composition for the optimum solutions with the variation of the desired %El, the elements' weight percentages are plotted against the desired %El in Fig. 5. Here, it is seen that only in Mn content, there is some significant variation. Addition of other elements is almost the same in all cases. The Mn content is reduced significantly for lower level of the desired ductility. This may be due to the necessity of reducing the achieved strength level in the optimum solutions as discussed above. The same trend is visible in all four cases. In some cases, a decrease in Si content is evident at higher desirable values of ductility. This may have been due to the fact that Si is a ferrite strengthener. Decreasing Si will lead to softer ferrite and thus improved ductility.

4 Concluding Remarks

The method of prescriptive analytics using optimization technique can be used effectively in solving diverse types of industrial problems. It can be used in the form of data-driven design, where optimization techniques can play an important role to design products with improved performance. Statistical methods and soft computing techniques are useful in this regard to generate the objective functions. In the case study described here, ANN technique, in conjunction with desirability function, is used as the objective function. GA is used successfully for not only getting the optimized solutions but also to find the behaviour of the compositional variables in searching the closest candidates for desired performance.

References

1. Ramachandran, K. M., & Tsokos, C. P. (2009). *Mathematical statistics with applications*. Gurgaon, India: Elsevier.
2. Datta, S. (2016). *Materials design using computational intelligence techniques*. CRC Press.
3. Kumar, Satish. (2004). *Neural networks—A classroom approach*. New Delhi: Tata McGraw-Hill Publishing Company Limited.
4. Datta, S., & Chattopadhyay, P. P. (2013). Soft computing techniques in advancement of structural metals. *International Materials Reviews, 58*, 475–504.
5. Kundu, M., Ganguly, S., Datta, S., & Chattopadhyay, P. P. (2009). Simulating time temperature transformation diagram of steel using artificial neural network. *Materials and Manufacturing Processes, 24*, 169–173.
6. Goldberg, D. E. (2002). *Genetic algorithms in search, optimization and machine learning*. Pearson-Education: New Delhi.
7. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester: John Wiley & Sons Ltd.
8. Chakraborti, N. (2004). Genetic algorithms in materials design and processing. *International Materials Reviews, 49,* 246–260.
9. Das, P., Mukherjee, S., Ganguly, S., Bhattacharyay, B. K., & Datta, S. (2009). Genetic algorithm based optimization for multi-physical properties of HSLA steel through hybridization of NN and desirability function. *Computational Materials Science*, *45*, 104–110.

10. Banerjee, A., Bandyopadhyay, T., & Acharya, P. (2013). Data analytics: Hyped up aspirations or true potential? *Vikalpa, 38,* 1–11.
11. Gorni, A. A. (2015). Steel forming and heat treating handbook, Version of 20 January 2015. www.gorni.eng.br/e/Gorni_SFHTHandbook.pdf.
12. Bhadeshia, H. K. D. H. (1999). Neural networks in materials science. *ISIJ International, 39,* 966–979.
13. Bhadeshia, H. K. D. H. (2009). Neural networks and information in materials science. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 1,* 296–305.
14. Datta, S., & Banerjee, M. K. (2004). Kohonen network modeling of thermo-mechanically processed HSLA steel. *ISIJ International, 44*, 846–851.
15. Mohanty, I., Datta, S., & Bhattacharjee, D. (2009). Composition-processing-property correlation of cold rolled IF steel sheets using neural network. *Materials and Manufacturing Processes, 24*, 100–105.
16. Sinha, A., Dey, S., Chattopadhyay, P. P., & Datta, S. (2013). Optimization of mechanical property and shape recovery behavior of Ti-(∼49 at%) Ni alloy using artificial neural network and genetic algorithm. *Materials & Desig*n, 46, 227–234.
17. Datta, S., Sultana, N., Zhang, Q., & Mahfouf, M. (2013). Optimal design of titanium alloys for prosthetic applications using a multi-objective evolutionary algorithm. *Materials and Manufacturing Processes, 28*, 741–745.
18. Dey, S., Datta, S., Chattopadhyay, P. P., & Sil, J. (2008). Modeling the properties of TRIP steel using AFIS: A distributed approach. *Computational Materials Science, 43*, 501–511.
19. Dey, S., Dey, P., & Datta, S. (2017). Rough-fuzzy-GA based search for Al alloys for cryogenic application. *Materials and Manufacturing Processes, 32*, 1075–1081.
20. Dey, S., Dey, P., Datta, S., & Sil, J. (2009). A rough set approach to predict strength and ductility of TRIP steel. *Materials and Manufacturing Processes, 24*, 150–154.
21. Dey, S., Dey, P., & Datta, S. (2017). Design of novel age-hardenable aluminium alloy using evolutionary computation. *Journal of Alloys and Compounds, 704*, 373–381.

Intuitionistic Fuzzy Approach Toward Evolutionary Robust Optimization of an Industrial Grinding Operation Under Uncertainty

Nagajyothi Virivinti and Kishalay Mitra

Abstract The existence of uncertainty is inherent and unavoidable in process systems. The sources of it might be linked, though not limited, to the estimation of model parameters through experimental/regression exercises and their related errors apart from the regular variations in the process operations. Thus, the outcomes of an optimization study by assuming uncertain parameters as non-varying one, as opposite to the actual scenario, would prompt to incorrect results and sometimes even infeasible solutions. One of the ways to handle such situations is by using intuitionistic fuzzy numbers (IFNs) to represent the uncertainty which considers both the membership and the nonmembership degree and carry out the uncertainty analysis based on the intuitionistic fuzzy logic. In this study, the above approach has been adopted to a real-life case study which contains highly nonlinear parameters named industrial grinding process considering different moods of a decision-maker under uncertain situations, i.e., optimistic, pessimistic, and mixed. Various sources of uncertainties are categorized as uncertainties related to model parameters (e.g., tuning parameters inside the model) and operational parameters (e.g., feed stream uncertainties), and their individual and amalgamated effects on the grinding process which is a multi-objective optimization problem have been analyzed. A novel way of determining the parametric sensitivity in presence of number of uncertain parameters has also been proposed. Based on the extent of non-determinacy to be resolved, the newly defined membership degree enables conducting comparative study of fuzzy and intuitionistic fuzzy robust optimization under different scenarios. Additionally, comparative analysis has been carried out for the proposed IFRO algorithm with the benchmark robust worst-case formulation (Ben-Tal and Nemirovskii 2001) and it has been observed that IFRO gives better results when compared with the worst-case formulation.

Keywords Intuitionistic fuzzy number · Robust optimization
Optimization under uncertainty · Pareto solutions · Industrial grinding process

N. Virivinti · K. Mitra (✉)
Department of Chemical Engineering, Indian Institute of Technology Hyderabad, Kandi 502285, Telangana, India
e-mail: kishalay@iith.ac.in

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_10

Robust solution · Multi-objective optimization · Evolutionary algorithms
Grinding · NSGA-II · Uncertainty · Pareto · Industrial process
Genetic algorithm

1 Introduction

Though uncertainty is inherently present in almost all systems, a common practice while analyzing the performance of a system is to ignore them for the sake of simplicity. Results obtained through this simplified process might enable the decision-maker to decipher the embedded mechanisms of a complex system; however, these oversimplified results might turn out to be unrealistic, sometimes infeasible and, therefore, not of much use for implementation because the inherent sources of uncertainties and their impacts on the system are ignored while generating the results. A real-life optimization problem can have various sources of uncertainties such as uncertainties related to the data, information, operation, etc., to name a few. The reasons for data-related uncertainty could be attributed to the errors related to measurements while performing experimentation; operation-related uncertainties could be attributed to variation in the quality of raw materials received, whereas the information-related uncertainties could be attributed to the mismatch in model assumptions. To elucidate these facts further, one can imagine the situation when an optimization problem utilizes a model of the process that is built on certain assumptions. In most of the cases, despite sincere efforts, a modeler is not sure about the complete validity of these assumptions, and therefore these models use certain empirical parameters which are eventually tuned with the process data to musk any mismatch of model predictions with the processes. However, as mentioned earlier, the data with which the entire model prediction is matched is having its own uncertainties and one cannot disregard the likelihood that even a little change in the abovementioned parameter values can make the obtained optimal solution infeasible since many times the optimal solution lies on the boundaries of the constraints. Similarly, a model tuned with a set of input operational conditions might perform quite unsatisfactorily with a different set of operational conditions. Under such scenarios, a decision-maker can customarily prefer to have a solution not much sensitive to variations in the model and input parameters. This is especially so important for an industrial scenario where under all practical uncertain situations, the industry has to produce a series of products with specified quality consistently. Therefore, there exists a genuine need for developing optimization techniques, under the broad theme of optimization under uncertainties, which are equipped to generate robust solutions immunized against these sources of uncertainties [1].

The general approach of developing optimization methodologies [1] to handle uncertainties in the parameters is to convert the optimization under uncertainty (OUU) problem (e.g., Min $\mathbf{c}x$, $\mathbf{P}x \leq \mathbf{q}$, where x is the set of decision variable and parameters $\mathbf{P}$, $\mathbf{q}$, $\mathbf{c}$ are uncertain) into a deterministic equivalent formulation (DEF), and that can be solved using standard linear/nonlinear programming techniques. DEF

from OUU problem can be obtained by distinct philosophies existing in the literature. One class of popular approach assumes the uncertain parameters as random and enough information is available to quantify as well as estimate the probability distribution associated with the variation of the parameters. Robust optimization [2], one such methodology, proposes to associate with an uncertain problem its robust counterpart and to use the associated robust optimal solutions. The robust counterpart is developed in such a way that the solutions which satisfy the constraints presented in the original optimization problem are called as robust feasible solutions and worst-case oriented philosophy is applied for objective function (best possible robust value of objective function among all robust feasible solutions) which quantifies the quality of robust feasible solution. The solutions obtained by robust counterpart are immunized against uncertainty called as robust solutions and the optimal value is called as robust optimal value. Along with robust optimization, there are some other methodologies which use probabilistic information of uncertain parameters [3]. Stochastic programming (SP) with two stages [4] is one of the techniques in which decision variables are divided into two stages depending upon the realization of uncertainty: one, after the realization of uncertainty (second stage) and the other, before realization of uncertainty (first stage). As a result, the second stage, unlike the first stage, considers different scenarios of uncertain parameters and the objective function is the sum of expected values of second stage components which are after realization of uncertainty and first stage components. Several instances of uncertain parameters are generated inside the uncertain space and are used to calculate the expected objective function. However, in this approach, as the number of uncertain parameters increases, the problem size increases exponentially which will lead to computationally expensive problem. Not only the problem size explosion but also division of the problem into two stages will be very difficult due to interconnections of the variables present inside the problem.

To overcome the drawbacks of TSSP, chance constrained programming (CCP) is introduced by Charnes and Cooper [5] which is advantageous with respect to problem size explosion and execution time. CCP assumes that the constraint may not be satisfied for all the possible instances of uncertain parameters and it is decided by the confidence level of the decision-maker. Probability of constraint satisfaction is used to represent the partial satisfaction of the constraints. CCP converts the OUU problem to DEF using the probability of constraint satisfaction. As mentioned earlier, the DEF obtained by CCP is advantageous with respect to problem size with increase in the number of uncertain parameters.

Expected value model (EVM) is another technique used to convert OUU problem to DEF by expected values of objective functions and the constraints. Expected value model uses only expected values of objective function and the constraints but it neglects the performance variability of objective function or constraint inside the uncertain parameter space. Performance variability is one of the important factors which decides the deviation of function value from the deterministic value. Performance variability can be decided by standard deviation (SD) of objective function and the constraints. Therefore, the robust optimization (RO) optimizes the expected objective functions along with the minimization of performance variability. In this

way, robust solutions are determined by robust optimization which will be having minimum performance variability inside the uncertain parameter space. Several researchers have applied the abovementioned uncertainty handling techniques *(TSSP, CCP, EVM, and RO)* to several industrial problems like supply chain planning, production scheduling, and bio-energy production [4, 6–8]. *Nevertheless, for all these techniques, the quantification of probability distribution function for a given set of data is not less challenging and, if the quantification is possible, it can consume huge amount of time as well as data which is unrealistic in many cases. Moreover, it is often preferred that the probability distribution should be confined within certain set of well-behaved and known probability distributions for ease in treating the problem mathematically.*

On the contrary, there are other methodologies, such as fuzzy, rough, vague, and interval fuzzy set [9–12] based methods that are free from above shortcomings and have been proposed and applied to solve many real-world problems considering uncertainties. To describe the behavior of complex or imprecise systems, fuzzy set (FS) theory, introduced by Zadeh [9] has been proved to be a perfect mathematical implement. The fuzzy number (FN) was defined by Zimmermann [13] and it is a special kind of fuzzy set used to describe the uncertain value. Fuzzy set based approaches use membership degree or function (MF), whose value lies between 0 and 1, which represents the degree of uncertainty in a parameter. The value of "1" represents the complete MF, whereas the value of 0 represents the complete nonmembership function (NMF). The values in between 0 and 1 are used to represent the intermediate degree of memberships. In between the two bounds of 0 and 1, the MF can vary linearly or nonlinearly. This leads to various fuzzification approaches, e.g., linear, Gaussian, trapezoidal, triangular, etc. MFs along with different fuzzy measures can finally define a DEF to the original OUU problem. Dubios and Prade [14] introduced the concept of possibility space and fuzzy variable and defined how to map the possibility space to a real number enabling one to define DEF efficiently using possibility measure. Based on FS theory, several uncertainties handling techniques are developed, e.g., fuzzy chance constrained programming (FCCP), fuzzy expected value model (FEVM), and fuzzy robust optimization (FRO) to name a few [15–18]. CCP uses probability of constraint satisfaction, whereas probability of constraint satisfaction is replaced with possibility of constraint satisfaction in FCCP. Along with FCCP, FEVM is another technique which is utilized to convert the OUU problem into DEF and it replaces the uncertain objective functions and constraints by expected values of objective function and constraints, and expected values can be calculated using the possibility measures. Virivinti and Mitra [15] have applied fuzzy expected value model technique to a highly nonlinear model of industrial grinding process (IGP). However, only the expected values of the uncertain parameters might not be sufficient to handle the uncertain optimization problem. Hence, the concept of fuzzy robust optimization, another way to handle uncertain optimization problems which consider the variance arising due to uncertainty in parameters in addition to their expected values, has been introduced. *The importance of standard deviation is such that it minimizes performance variability due to existence of uncertain parameters.* Fuzzy based approaches like FCCP and FEVM have been applied by

several researchers in real-life applications [15–18] and the solutions obtained by these approaches address the different aspects of uncertainties. However, as mentioned earlier, the solutions obtained with the abovementioned techniques may not be robust solutions. Therefore, several researchers have used FRO technique to obtain the robust optimal solutions. FRO is gaining prominent importance as it gives the robust solutions in the presence of uncertainty. Applications of FRO are very few in the literature [19–22].

As a generalization of FS, Atanassonav [23] proposed intuitionistic fuzzy set (IFS), which considers both MF and NMF to represent the level of uncertainty. In FS theory, NMF complements the MF as a fuzzy number can be either a member or a nonmember. However, it may not always be the case because there may be degree of hesitation, which should be considered in addition to MF and NMF. Considering the three parameters, IFS came into realization where the sum of MF and NMF might be less than or equal to unity. This can be explained by taking an example of voting process. MF is used to represent acceptance which is the percentage of electorates who have opted or voted for the corresponding government. In FS theory, NMF can be calculated as the part of electorates who have not opted or voted for the government. However, in IFS theory, NMF is used to represent the rejection part, electorates who have not voted or opted for the government. Next, the percentage of electorates who have not participated in electorate process is represented by non-determinacy which is called as hesitation. Non-determinacy or hesitation degree can be defined as complement of the sum of MF and NMF and this does not exist in the FS theory. Hence, IFS is one of the mathematical implements which represents the imprecise information using both MF and NMF that FS alone is not able to represent. IFS-based optimization of engineering processes has been gaining popularity in the recent past [24]. Applications of intuitionistic fuzzy concept in different fields are very few in the literature. Therefore, in this work, intuitionistic fuzzy concept has been applied to FRO of a real-life case study: an industrial grinding process (IGP) that faces extreme uncertainty in terms of model as well as operational parameters.

The role of RO as compared to the EVM and the generic nature of IFS have been discussed earlier. The deterministic IGP has two conflicting objectives (simultaneous maximization of throughput (Th) and midsize fraction (Ms), four control variables (CV's), and three manipulated variables (MV's) that are part of a grinding model consisting of 36 differential algebraic equations representing several states of the process. However, each of these objective functions and constraints are functions of several parameters which are uncertain in nature. These parameters are divided into two categories: (1) Model-related uncertain parameters (e.g., grindability indices (GI), grindability exponents (GE), and sharpness indices (SI)) which are obtained by regression analysis of steady-state experimental data, and thereby subjected to uncertainty involved in the regression or experimental procedures; (2) Operational parameters such as five feed size classes (feed size distribution) are assumed as uncertain parameters, as the IGP can handle varied range of feed sources. The OUU problem has, therefore, been converted into a DEF using intuitionistic FRO technique by assuming uncertain parameters as IFNs. As the OUU problem contains nonlinearly related uncertain parameters, the extension of the DEF was not straightforward and

the simulation-based intuitionistic fuzzy approach has been adopted to propagate the effect of uncertainty in the objective functions and the constraints. The DEF contains four objective functions which are expected values of the objective functions and SD values of objective functions which will minimize the performance variability, apart from the constraints that have also similar modified components. A well-developed evolutionary algorithm, non-dominated sorting genetic algorithm-II (NSGA-II) [25], has been used to observe the effect of uncertainty on the final Pareto solutions. A novel way of performing sensitivity analysis of the uncertain parameters has also been proposed. Finally, comparative analysis of IFRO with FRO and the benchmark robust worst-case formulation has been carried out [2].

This chapter is organized as follows: Sect. 2 contains some introductory concepts of intuitionistic fuzzy sets and robust optimization. In Sect. 3, optimization under uncertainty formulation and deterministic equivalent formulation (DEF) of the IGP have been presented. Section 4 discusses the results of application of IFRO on IGP. Conclusions are presented in Sect. 5.

2 Preliminaries

Robust Optimization:
To explain robust optimization, we start with simple linear programming (LP) example which is used in several applications. The structure of the linear problem is as follows:

$$\min_x \left\{ C^T x : Px \leq q \right\} \tag{1}$$

where C, P, q are the matrices containing numerical values which may not be certain in all the cases. Hence, uncertain linear programming problem can be defined as

$$\min_x \left\{ C^T x : Px \leq q \right\} : (C, P, q) \in U \tag{2}$$

It represents the uncertain LP problem as the collection of LP problems with the (C, P, q) values varying in the uncertainty set U. A solution which satisfies the constraint $Px \leq q$, whatever may be the realization of uncertainty in U, is called as robust feasible solution. At robust feasible solution, the objective function value can be interpreted (uncertain) by *worst-case oriented approach*, which is used to quantify the quality of robust feasible solution. The best robust feasible solution is obtained by the following formulation:

$$\min_x \left\{ \sup_{(C,P,q)\in U} C^T x : Px \leq q \right\} : (C, P, q) \in U \tag{3}$$

The above is called as robust counterpart for Eq. (1). The robust optimal solutions can be determined by the robust counterpart given above. Next, to explain the tackling of uncertainty in nonlinear optimization problems, consider a nonlinear optimization problem which has n objective functions and m constraints

$$\begin{aligned} \text{Minimize} f(x, \tilde{a}) &= (f_1(x, a_1), f_2(x, a_2) \ldots f_n(x, a_n))^T \\ \text{s.t. } g_j(x, b) &\leq 0, \quad j = 1, \ldots, m \\ x_i^L &\leq x_i \leq x_i^U, \quad i = 1, \ldots, k \end{aligned} \tag{4}$$

where a and b are uncertain parameter vectors. RO is one of the uncertainties handling techniques which assumes that the uncertainty information of a and b are available with the decision-maker or can be assumed based on the experience. Robust counterpart of optimization under uncertainty problem is used to calculate the robust optimal solution of the problem. $\varphi_i = \max f_i(x, a)\ \forall a \in U\ i = 1, \ldots, n$

$$\varphi_j = \max g_j(x, b)\ \forall b \in U\ j = 1, \ldots, m \tag{5}$$

For each of the objective function and the constraints, worst case is calculated by the expressions given above. The robust counterpart of the optimization problem is given by

$$\begin{aligned} &\min \phi_i \qquad i = 1, \ldots n \\ &\text{s.t.} \phi_j \leq 0\ j = 1, \ldots, m \end{aligned} \tag{6}$$

The solutions obtained by the abovementioned robust counterpart are called as robust optimal solutions and optimal value is called as robust optimal value.

Fuzzy Set: A fuzzy set F in D is represented by a membership degree (MF) $\mu_F(x)$. Each point in D corresponds to a real number in the membership degree interval [0, 1].

$$F = \{x, \mu_F(x) | x \in D\} \tag{7}$$

The values of $\mu_F(x)$ at x represent the grade of membership of x in F. Thus, the nearer the value of $\mu_F(x)$ to unity, the higher the association of x in F.

Intuitionistic fuzzy set: An intuitionistic fuzzy set IF in D is represented as given below:

$$IF = \{x, \mu_{IF}(x), \nu_{IF}(x) | x \in D\} \tag{8}$$

where the functions $\mu_{IF} : D \rightarrow [0, 1]$ and $\nu_{IF} : D \rightarrow [0, 1]$ are the membership and nonmembership degrees, which satisfy the constraint $0 \leq \mu_{IF}(x) + \nu_{IF}(x) \leq 1$. The non-determinacy can be defined as complement of sum of MF and NMF values.

$$\pi_{IF}(x) = 1 - \mu_{IF}(x) - \nu_{IF}(x) \tag{9}$$

For ordinary fuzzy sets $\pi_{IF}(x) = 0$ for each $x \epsilon D$. These sets have the form $\{x, \mu_{IF}(x), 1 - \mu_{IF}(x) | x \in D\}$.

With this background, Yager [26] initially defined the IFN as (μ_{IF}, ν_{IF}) and later it is modified as triplet by introducing non-determinacy term $(\mu_{IF}, \nu_{IF}, \pi_{IF})$ where $\mu_{IF} + \nu_{IF} + \pi_{IF} = 1$.

2.1 Intuitionistic Fuzzy Optimization (IFO)

Let us consider an OUU problem which contains some uncertain information (a and b) in objective functions or constraints.

$$\begin{aligned} \text{Minimize} f(x, \tilde{a}) &= (f_1(x, a_1), f_2(x, a_2) \ldots f_n(x, a_n))^T \\ \text{Subjected to } g_j(x, b) &\leq 0, \quad j = 1, \ldots, m \\ x_i^L \leq x_i &\leq x_i^U, \quad i = 1, \ldots, p \end{aligned} \tag{10}$$

The OUU problem can be solved by calculating the membership and nonmembership degrees using Bellmann and Zadeh [27] extension principle which is given below:

$$\begin{aligned} \mu_l(x) &= \min(\mu_{f_k}(x), \mu_{g_j}(x)) \\ \nu_l(x) &= \max(\nu_{f_k}(x), \nu_{g_j}(x)) \end{aligned} \tag{11}$$

The OUU problem can be converted into DEF by IFO theory which uses IFN to represent the degree of uncertainty. The IFO formulation maximizes the membership degree which is acceptance degree and minimizes the nonmembership degree which is rejection degree of the objective functions and the constraints [28].

$$\begin{aligned} &\max\alpha, \min\beta \\ &\alpha \leq \mu_l(x) \\ &\beta \geq \nu_l(x) \quad \text{, where } l = 1, 2, \ldots, m+n. \\ &\alpha \geq \beta \\ &\alpha + \beta \leq 1 \end{aligned} \tag{12}$$

where minimum acceptance level of objectives and the constraints is represented by α, and β represents the maximum degree of rejection of objective function and the constraints. The above equivalent deterministic optimization problem can also be written as

$$\begin{aligned} &\max \alpha - \beta \\ &\alpha \leq \mu_l(x) \\ &\beta \geq \nu_l(x) \\ &\alpha \geq \beta \\ &\alpha + \beta \leq 1 \end{aligned} \tag{13}$$

Yager [26] found out some flaws in the above optimization formulation. Let us consider two cases whose membership degree values are 0.4, 0 and nonmembership degree values are 0.8 and 0. The α-β values for these two cases are −0.4, 0, respectively. The IFO formulation given above selects the value which is having maximum α-β values which are nothing but 0. But the MF value of that case is zero which is not the right decision to consider. Hence, Yager [26] made some modifications in the above optimization formulation by considering non-determinacy value which is neglected in the above formulation. Hence, the new membership degree value can be calculated by

$$\mu_{new,k}(x) = \mu_k(x) + \lambda \pi_k(x) \tag{14}$$

The value of λ can be selected based on how much non-determinacy can be resolved which will be in between 0 and 1. λ = 0 uses only membership degree to carry out the analysis and it does not resolve the non-determinacy term. The results obtained with λ = 0, match with the results obtained with FRO which does not consider the non-determinacy term. Hence, it is clear that IFRO is more generic and FRO is a special case of IFRO. λ = 1 considers complement of nonmembership degree as a new membership degree to carry out the analysis. In our further discussion, λ = 0.5 will be used which is in favor of both membership and nonmembership degree values and it resolves half of the non-determinacy value. However, the effect of λ on the optimization results can be observed by changing the values of λ.

2.2 *Intuitionistic Fuzzy Robust Optimization*

Let us consider an OUU problem which has n objective functions and m constraints as given in Eq. (10). Here, a and b are the uncertain variables involved in the objective functions and constraints, respectively.

The OUU problem can be converted into DEF using RO technique which uses expected values and standard deviation values of objective function and constraints. The converted DEF is given below:

$$\begin{aligned} \text{Maximize } Exp\{f(\mathbf{X}, \tilde{a})\} &= (Exp\{f_1(\mathbf{X}, a_1)\}, Exp\{f_2(\mathbf{X}, a_2)\} \ldots Exp\{f_n(\mathbf{X}, a_n)\})^T \\ \text{Minimize } SD\{f(\mathbf{X}, \tilde{a})\} &= (SD\{f_1(\mathbf{X}, a_1)\}, SD\{f_2(\mathbf{X}, a_2)\} \ldots SD\{f_m(\mathbf{X}, a_m)\})^T \\ \text{s.t. } Exp\{g_j(\mathbf{X}, b)\} + \kappa SD\{g_j(\mathbf{X}, b)\} &\leq 0,\ j = 1, \ldots, m \\ x_i^L \leq x_i &\leq x_i^U,\ i = 1, \ldots, p \end{aligned} \tag{15}$$

The expected values of objective function and constraints can be calculated using expected value operator which will use possibility value to calculate the expected value [29].

$$Exp^{pos}(f(\mathbf{X}, a)) = \int_0^{+\infty} pos\{f(\mathbf{X}, a) \geq r'\} - \int_{-\infty}^{0} pos\{f(\mathbf{X}, a) \leq r'\} \tag{16}$$

The possibility value can be calculated using the formula given here

$$pos\{f(\mathbf{X},a) \geq r'\} = \max_{1 \leq L \leq N} \{\mu_L | \{f(\mathbf{X}, a_L) \geq r'\}\} \tag{17}$$

The standard deviation value can be calculated using expected value calculated above. The standard deviation can be calculated as

$$SD\{f(\mathbf{X}, a)\} = \sqrt{Exp[(f(\mathbf{X}, a) - e)^2]} \tag{18}$$

where $e = Exp\{f(\mathbf{X},a)\}$.

As uncertain parameters are related in nonlinear fashion, expectation and standard deviation values can be calculated using fuzzy simulation approaches. Fuzzy simulation approach to calculate the expected value of any function is given below:

Fuzzy Simulation Algorithm for calculation of Expected Value:

$$E = Exp\{f(\mathbf{X},a)\} \text{ or } Exp\{g(\mathbf{X},b)\}$$

Step 1: Let us assume e = 0;
Step 2: Generate N″ random numbers (ζ) and μ_L *and* ν_L(L = 1, 2, …, N) from ε-set of a or b, where ε is a sufficiently small number where μ and ν are the MF and NMF values;
Step 3: Respective non-determinacy π_L and modified membership function μ_{new} are calculated;
Step 4: Set the two numbers $f_{\min} = \min(f(\mathbf{X}, a_1), f(\mathbf{X}, a_2), \ldots, f(\mathbf{X}, a_{N''}))$ and $f_{\max} = \max(f(\mathbf{X}, a_1), f(\mathbf{X}, a_2), \ldots, f(\mathbf{X}, a_{N''}))$;
Step 5: Randomly generate N' values of r' between $f_{\min}$ and $f_{\max}$;
Step 6: If $r' \geq 0$, then $e = e + pos(f(\mathbf{X},\zeta) \geq r'))$;
if $r < 0$, then $e = e - pos(f(\mathbf{X},\zeta) \leq r'))$;

Step 7: Repeat the above two steps for N' times;
Step 8: $Exp = \max(f_{\min}, 0) + \min(f_{\max}, 0) + e \cdot (f_{\max} - f_{\min}) \big/ N'$.

In this manner, expected values and SD values are calculated and can be used in the intuitionistic fuzzy robust optimization formulation. The IFRO formulation can be solved for three different scenarios, namely, optimistic, pessimistic, and mixed scenarios, reflecting three different moods of a decision-maker under different uncertain situations. Membership degree calculation is the same for these three approaches, whereas the calculation differs while defining the nonmembership degrees.

Consider the uncertainty on the lower and upper side of deterministic value a or b. $p > 0$ is the tolerance of membership degree, which means that the value of a or b can be relaxed by an amount p on either side. Then, the MF for upper and lower side uncertainties can be defined as

$$\mu(x) = \begin{cases} 1, & x = a \\ 1 + \frac{(x-a)}{p}, & a - r < x < a \\ 0, & x \le a - p \end{cases}; \quad \mu(x) = \begin{cases} 0, & x \ge a + p \\ 1 + \frac{(a-x)}{p}, & a < x < a + p \\ 1, & x = a \end{cases} \tag{19}$$

Similarly, determine the values of NMF in such a way that it satisfies the constraint that the sum of both MF and NMF is less than or equal to one. The definition of nonmembership degree and the construction of MF, NMF, and modified MF for these three approaches is presented now [30].

Optimistic approach: Let us consider r > 0 is the tolerance for NMF means the value of p is relaxed by certain amount r. The values between $a + p$ and $a + p + r$ for upper side uncertainty, and $a - p$ and $a - p - r$ for lower side uncertainty are not considered as feasible because the value of membership degree is zero and at the same time they are not considered as infeasible because the value of NMF is not one. The expressions for NMF of upper and lower side uncertainties are as given in Virivinti and Mitra (2015). The new membership degree can be calculated using Eq. (14) which does not neglect the non-determinacy term. The MF, NMF, and new MF for optimistic approach is shown in Fig. 1.

Pessimistic approach: Let us consider $s > 0$ is the tolerance for NMF for pessimistic approach. The value of a is relaxed by an amount p to construct membership degree. Similarly, for the construction of nonmembership degree, the value of r is restricted by an amount s. The solutions between $a + p$ and $a + p - s$ for upper side uncertainty, and $a - p$ and $a - p + s$ for lower side uncertainty are not feasible (membership degree value is not one) and not considered as infeasible also (nonmembership degree is also not one). The expressions to calculate the NMF for the lower and upper side parameters is shown in Virivinti and Mitra (2015). The membership and nonmembership degrees for the pessimistic approach are given in Fig. 2.

Mixed approach: Let us consider $u > 0$, $w > 0$ are tolerances for nonmembership degree for mixed approach. The value of a is relaxed by an amount p for the construction of membership degree and the value of p is relaxed by an amount u and

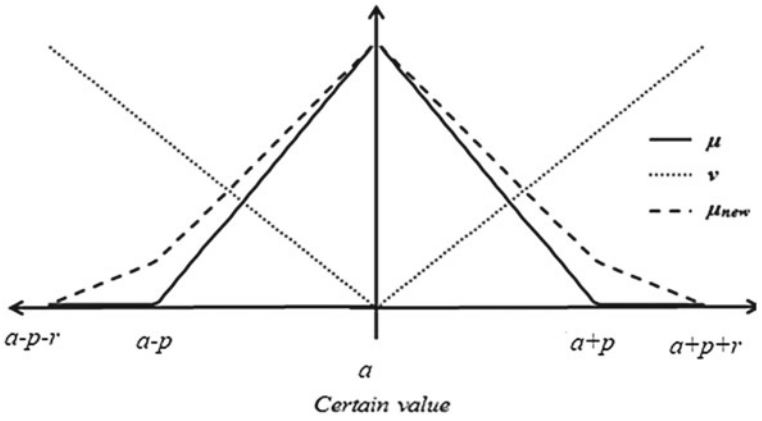


Fig. 1 Optimistic approach: membership degree (μ), non-membership degree (ν), and new membership degree (μ_{new})

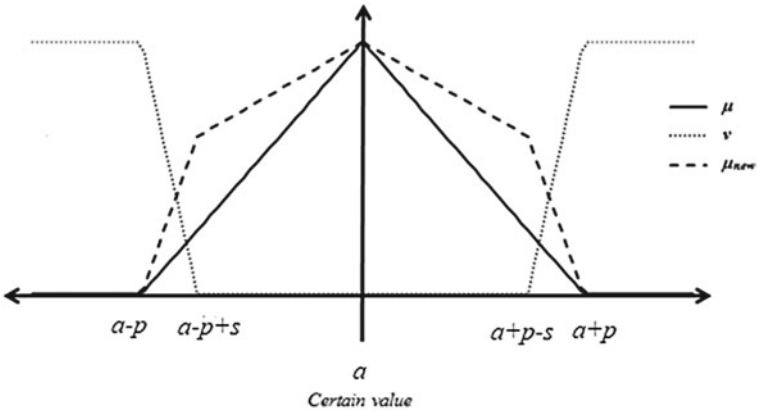


Fig. 2 Pessimistic approach: membership degree (μ), non-membership degree (ν), and new membership degree (μ_{new})

restricted by an amount w. The solutions between $[a + p, a + p + u]$ (Upper side uncertainty) and $[a - p - u, a - p]$ (lower side uncertainty) are not feasible (membership degree is equal to zero) and the solutions between $[a + p + u - w$, a + $p]$ (upper side uncertainty) *and* $[a - p, a - p - u + w]$ (lower side uncertainty) are not infeasible (nonmembership degree is not one). Hence, both the areas are considered to carry out the analysis. The membership and nonmembership degrees for mixed approach are shown in Fig. 3.

In this way, IFRO concept has been applied to an IGP using these three approaches while considering the uncertainty in two sets of uncertain parameters for lower and upper side uncertainties.

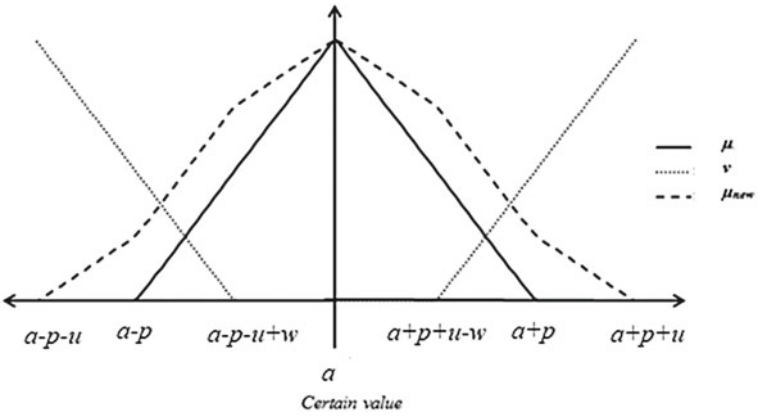


Fig. 3 Mixed approach: membership degree (μ), nonmembership degree (ν), and new membership degree (μ_{new})

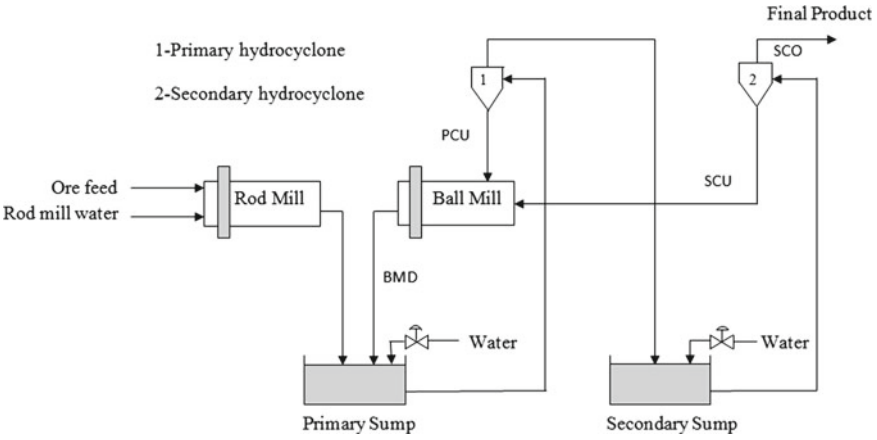


Fig. 4 An industrial grinding process flow diagram

3 Case Study: Industrial Grinding Process

3.1 Process Description

The IGP has been considered here which has the following four units: rod mill (RM), ball mill (BM), numbers of sumps (PS, SS), and hydrocyclones (PC, SC). The fresh feed along with water is fed to the rod mill initially. The slurry generated from rod mill is sent to the series of sumps along with water, and thereby the mixture is fed to a series of cyclones. The detailed description about the IGP can be found in Mitra (2009). The schematic diagram of industrial grinding process is shown in Fig. 4.

The unit operations used in the IGP are modeled using population balance methods and empirical correlations. Material and energy balances are modeled by the population method and the breakage functions and sharpness indices are calculated by empirical correlations The model equations for the grinding circuit goes like this. Six differential equations for the rod mill which contains the overall mass balance equation and equations for five individual size classes. Similarly, six equations will be there for ball mill, primary sump, and secondary sump. In total, it will lead to 24 such differential equations. Next, in hydrocyclones, efficiency factors for five different size classes are calculated by five corrected efficiency equations and will lead to ten algebraic equations. For primary sump and secondary sump pump factor states, two more algebraic equations have been added. The grinding circuit will lead to a set of 36 differential algebraic equations.

The complete set of model equations will lead to a set of differential algebraic equations and they can be solved using public domain software called differential algebraic system solver (DASSL) [31]. The details about the grinding process and the set of model equations can be found in [32] and not attached here for the sake of brevity.

3.2 Deterministic Optimization Formulation

As mentioned earlier, the IGP under consideration has two objective functions and four CVs and three MVs. The objectives of industrial grinding process are maximization of throughput (Th) and midsize fraction (MS). These two objectives are conflicting in nature and do not contain a unique solution. All the control variables (percent passing of coarse (CS) and fine (FS) size classes), percent solids (PS), and recirculation load (RCL) are selected based on the upper bound constraint. The process contains three MVs which are flow rates of water to both the sumps and the fresh ore to the rod mill. With this background, the DEF of IGP is as follows:

Objective functions

$$\underset{S_R, W_{ps}, W_{ps}}{Max} \; TP(\Phi, \Psi)$$

$$\underset{S_R, W_{PS}, W_{SS}}{Max} \; MS(\Phi, \Psi)$$

Subjected to

$$CS(\Phi, \Psi) \leq CS^U$$

$$FS(\Phi, \Psi) \leq FS^U$$

$$PS(\Phi, \Psi) \leq PS^U$$

$$RCL(\Phi, \Psi) \leq RCL^U$$

Bounds on decision variables

$$S_R^L \leq S_R \leq S_R^U$$

$$W_{PS}^L \leq W_{PS} \leq W_{PS}^U$$

$$W_{SS}^L \leq W_{SS} \leq W_{SS}^U \tag{20}$$

The optimization formulation given above contains some parameters inside the brackets which represent that the objective function and the constraints are functions of the parameters (Φ, Ψ) which are uncertain in nature. *Here, Φ is a vector containing six uncertain parameters $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)$ related to model uncertainty. These are grindability index of rod mill and ball mill (RMGI and BMGI), the grindability exponents for the rod mill and the ball mill (RMExpo, BMExpo), and the sharpness indices for the primary and secondary cyclones (PCSI, SCSI), respectively. These parameters are assumed as deterministic or fixed in deterministic case. However, these parameters may undergo experimental and regression errors as these are obtained from regression analysis of experimental results. Hence, considering them as certain would not be the correct approach. In the similar manner, Ψ is a vector which contains four uncertain parameters $(\psi_1, \psi_2, \psi_3, \psi_4)$ which are four size fractions fed to the rod mill and these are called as operational-related uncertain parameters. The finest size fraction out of five size fractions can be calculated by the complement of the sum of four size fractions. As the parameters are uncertain in nature, these parameters are considered as IFNs which consider both MF and NMF to carry out the uncertainty analysis.* The OUU problem can be converted into DEF by IFRO which uses expected values and SD values of objective function and constraints. The results obtained by IFRO approach are compared with the results obtained by robust optimization developed by Ben-Tal and Nemirovskii [2]. Before going to the discussions of intuitionistic fuzzy robust optimization, let us formulate the robust counterpart for industrial grinding optimization problem.

3.3 Robust Counterpart for Industrial Grinding Optimization Problem

As mentioned in the previous section, model and operational parameters are considered as uncertain. These uncertain parameters are assumed to vary 20% on the lower side which defines the uncertainty space for the parameters. By this assumption, robust counterpart for the industrial grinding problem can be written as

Objective functions

$$\underset{S,W_1,W_2,U}{Max} \{Min(TP(\Phi, \Psi))\}, (\Phi, \Psi) \in U$$

$$\underset{S,W_1,W_2}{Max} \{Min(MS(\Phi, \Psi))\}, (\Phi, \Psi) \in U$$

Subjected to

$$Min\left(CS(\Phi, \Psi) - CS^U\right) \leq 0, (\Phi, \Psi) \in U$$

$$Min\left(FS(\Phi, \Psi) - FS^U\right) \leq 0, (\Phi, \Psi) \in U$$

$$Min\left(PS(\Phi, \Psi) - PS^U\right) \leq 0, (\Phi, \Psi) \in U$$

$$Min\left(RCL(\Phi, \Psi) - RCL^U\right) \leq 0, (\Phi, \Psi) \in U$$

Decision variables bounds

$$\begin{aligned} S_R^L \leq S_R \leq S_R^U \\ W_{ps}^L \leq W_{ps} \leq W_{ps}^U \\ W_{ss}^L \leq W_{ss} \leq W_{ss}^U \end{aligned} \tag{21}$$

The solution to the above worst-case robust counterpart can be compared with solutions obtained with intuitionistic fuzzy robust optimization which is explained in the next section.

3.4 Intuitionistic Fuzzy Robust Optimization

As mentioned earlier, these ten uncertain parameters are assumed as IFNs and OUU problem can be converted into DEF which contains four objective functions maxi-

mization of expected values of the objective functions (Th and Ms) and minimization of SD values of objective functions. The resultant DEF is shown below:

$$\text{Max } E(TP(\Phi, \Psi))$$

$$\text{Min } SD(TP(\Phi, \Psi))$$

$$\text{Max } E(MS(\Phi, \Psi))$$

$$\text{Min } SD(MS(\Phi, \Psi))$$

subjected to

$$\text{E}(\text{PS}(\Phi, \Psi)) + \kappa[\text{SD}(\text{PS}(\Phi, \Psi))] \leq \text{PS}^{\text{U}}$$

$$\text{E}(\text{CS}(\Phi, \Psi)) + \kappa[\text{SD}(\text{CS}(\Phi, \Psi))] \leq \text{CS}^{\text{U}}$$

$$\text{E}(\text{FS}(\Phi, \Psi)) + \kappa[\text{SD}(\text{FS}(\Phi, \Psi))] \leq \text{FS}^{\text{U}}$$

$$\text{E}(\text{RCL}(\Phi, \Psi)) + \boldsymbol{\kappa}[\text{SD}(\text{RCL}(\Phi, \Psi))] \leq \text{RCL}^{\text{U}}$$

Decision variable bounds

$$\begin{aligned} S_R^L \leq S_R \leq S_R^U \\ W_{PS}^L \leq W_{PS} \leq W_{PS}^U \\ W_{SS}^L \leq W_{SS} \leq W_{SS}^U \end{aligned} \tag{22}$$

The value of κ represents the relative weightage of expected value and SD components in the constraints. The value of one is used for κ to convert the OUU problem to DEF. The uncertain parameters in the model are related in nonlinear fashion and uncertainty handling technique for such problems with intuitionistic fuzzy logic is very rare in the literature. Fuzzy simulation based approaches are used to solve the above DEF and it is mentioned in Sect. 2.2 and NSGA-II is used to determine the Pareto optimal solutions.

4 Results and Discussions

As mentioned earlier, the case study of the IGP has two objective functions, four CVs, three MVs, and ten uncertain parameters. The aim is to identify the effects of these ten uncertain parameters on the IGP multi-objective optimization problem. As the uncertain parameters belong to two different categories, operational and model, the uncertainty analysis has been started first by considering the uncertainty in model-related uncertain parameters which are followed by the uncertainty in operational-related uncertain parameters. Next, uncertainty analysis has been carried out by considering the uncertainty in both model- and operational-related uncertain parameters.

Next, by considering the uncertainty in ten parameters, Pareto optimal solutions have been determined in three different approaches, namely, optimistic, pessimistic, and mixed. The construction of the MF is the same for these three approaches, whereas the construction of the NMF is different and it is shown in Sect. 2.2. The MF for these three approaches can be constructed by relaxing the uncertain parameter by 20% of the certain value. The linear MF has been assumed and decision-maker can select any other type of membership degree. These relaxation limits can be obtained by analyzing the collected data of the data acquisition system present in a plant, or by consulting several experienced operators/managers who have been running or observing the plant for long time. In this work, second approach is used to determine the fuzzy limits. The NMF for the optimistic approach can be fixed by relaxing the uncertain parameter 5% further, which means NMF can be constructed between the deterministic value and 25% deviated value either upper or lower side of the deterministic value. Similarly, in pessimistic approach, NMF can be constructed between 20 and 15% deviated values from the deterministic value either on the upper or the lower side. In the mixed approach, the NMF can be constructed between 30 and 10% deviated values from the deterministic value. In this way, membership and nonmembership degrees can be obtained and final membership degree can be obtained using Eq. 14.

To observe the effect of first set of parameters which are model-related parameters, uncertainty in six model parameters ($\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6$) are considered first. The remaining operational parameters are considered as certain and their values are fixed at deterministic values. The MF can be constructed by assuming lower and upper side uncertainties in six parameters. The NMF can be calculated using pessimistic approach in which uncertainty parameter is restricted. The final MF can be calculated using Eq. 14. Pareto optimal (PO) solutions for four objective optimization problems are obtained by considering the uncertainty either on the lower (Type a) or the upper (Type b) side of the deterministic value of the uncertain parameter in pessimistic approach. By observing Fig. 5a, it can be inferred that Type b uncertain parameters are giving better results based on the expected values of objective functions but SD values of Th are high, whereas SD values of Ms are less. However, from Fig. 5a, it is not that much clear to observe the Pareto front movement. Figure 5b represents PO solutions with respect to expected values of objective functions. By considering the

uncertainty on lower side (Type a) of uncertain parameters, movement of PO front is in left downward direction, which means that for a fixed value of Th, there is a decrease in Ms values (since both the objectives are maximized here). Similar analysis has been carried out with the uncertainty on upper side of uncertain parameter and it has been observed that there is a right upward movement in the Pareto front. Hence, based on expected values, it can be concluded that upper side uncertain values will result in the better PO solutions compared to deterministic and lower side uncertain values. However, Fig. 5c shows that SD values of Th are high for upper side (Type b) uncertain parameters. Since these SD values are of very small order of magnitude, they can be neglected. At the same time, the SD values for MS are high for lower side (Type a) uncertain values (Fig. 5d). Hence, it can be concluded that upper side uncertain parameter values (Type b) provide better solutions, which has high expected value and low SD values of objective function.

Next, the uncertainty is considered in operational parameters and all the model parameters are at their deterministic values and assumed as fixed. Similarly, effect of uncertain parameters has been determined for pessimistic approach by considering the uncertainty of Type a and Type b of the uncertain parameters one by one. With Type a uncertain parameters, Pareto front is moving in right upward direction (Fig. 6b). The newly evolved Pareto front is observed to be shorter in length due to the control variable constraint violation of coarse size feed fraction. On the other hand, with Type b uncertain parameters, a left and downward movement in the PO front has been observed (Fig. 6b) with no change in the span of the Pareto front. Trends in Fig. 6c show that the Pareto solutions obtained with Type a uncertain parameters are associated with higher SD values of throughput. Since the magnitude of the SD values of throughput is comparatively far less in order of magnitude, they can be neglected. Similarly, the lower side uncertainty Pareto solutions are associated with varied values of SD of midsize fraction, which is of the similar order of magnitude with the upper side uncertainty (Fig. 6d). Hence, it can be concluded from Fig. 6 that the uncertain values of the lower side give us relatively better Pareto solutions.

To see the behavior of optimization results in presence of all parameters, uncertainties in both the sets of parameters (model related and operational related) are considered together. Considering the uncertainties in all ten parameters simultaneously, PO solutions obtain Type a and Type b uncertain values. PO solutions between expected values of Th and Ms are shown for Type a and Type b uncertain parameters (Fig. 7b). Better PO front is obtained with values of uncertain parameters on the lower side, whereas uncertain parameter values on the upper side give us the deterioration in the Pareto front. Effect of SD values of the throughput can be ignored since the standard deviation values are very low. Looking at the SD values of the midsize fraction, both the upper and the lower side uncertain values provide equivalent nature of solutions (Fig. 7d). Hence, it can be concluded that Type a uncertain parameters are giving better Pareto solutions which are completely opposite to that of the uncertainty in model parameters alone and matches with the uncertainty in operational parameters. Hence, operational parameters are the dominating set of uncertain parameters where slight change in value can lead to the change in the PO solutions.

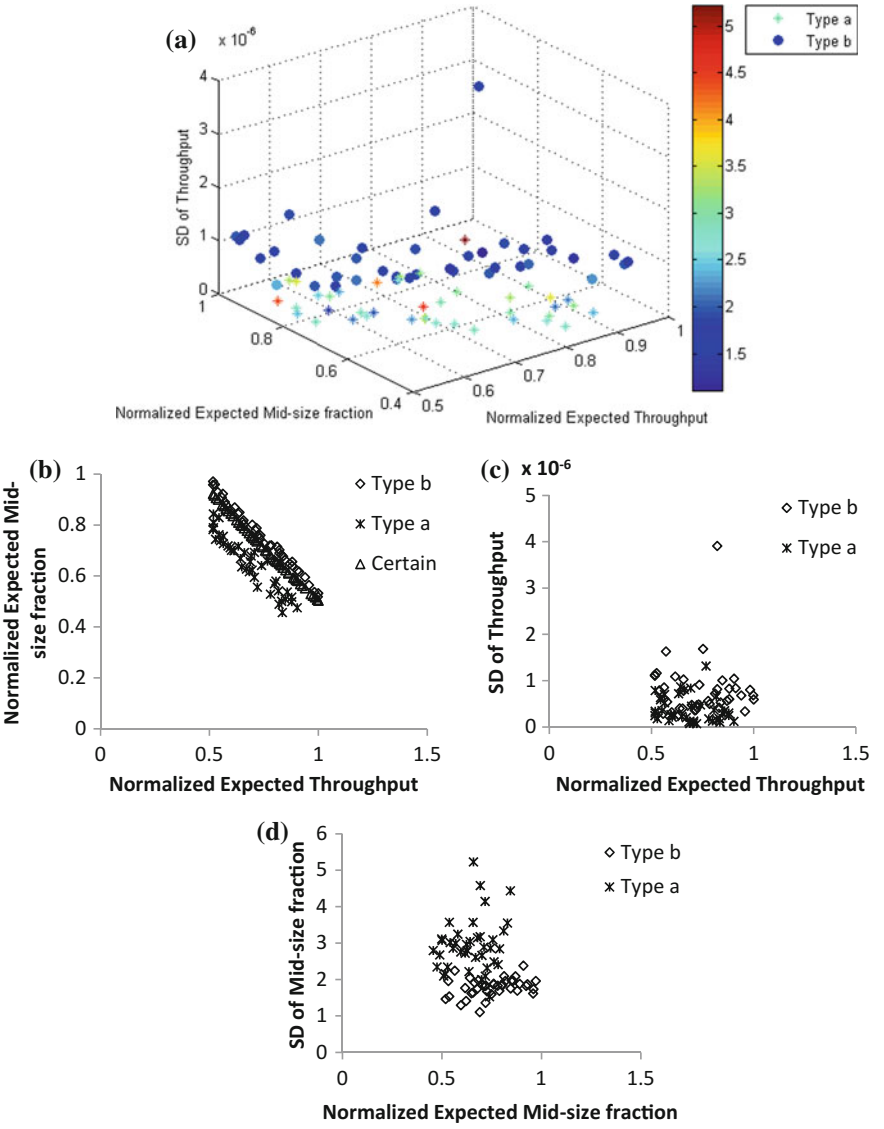


Fig. 5 Pareto optimal solutions with uncertainty in model parameters

Next, the effects of three different approaches such as optimistic, pessimistic, and mixed on the optimization results are studied considering the presence of ten uncertain parameters. From Fig. 8a, it can be observed that PO solutions by optimistic approach are conservative in nature, whereas PO solutions obtained by pessimistic approach are aggressive in nature. These three approaches are same in the definition of MF and the definition of NMF is different. In the first approach, the parameter

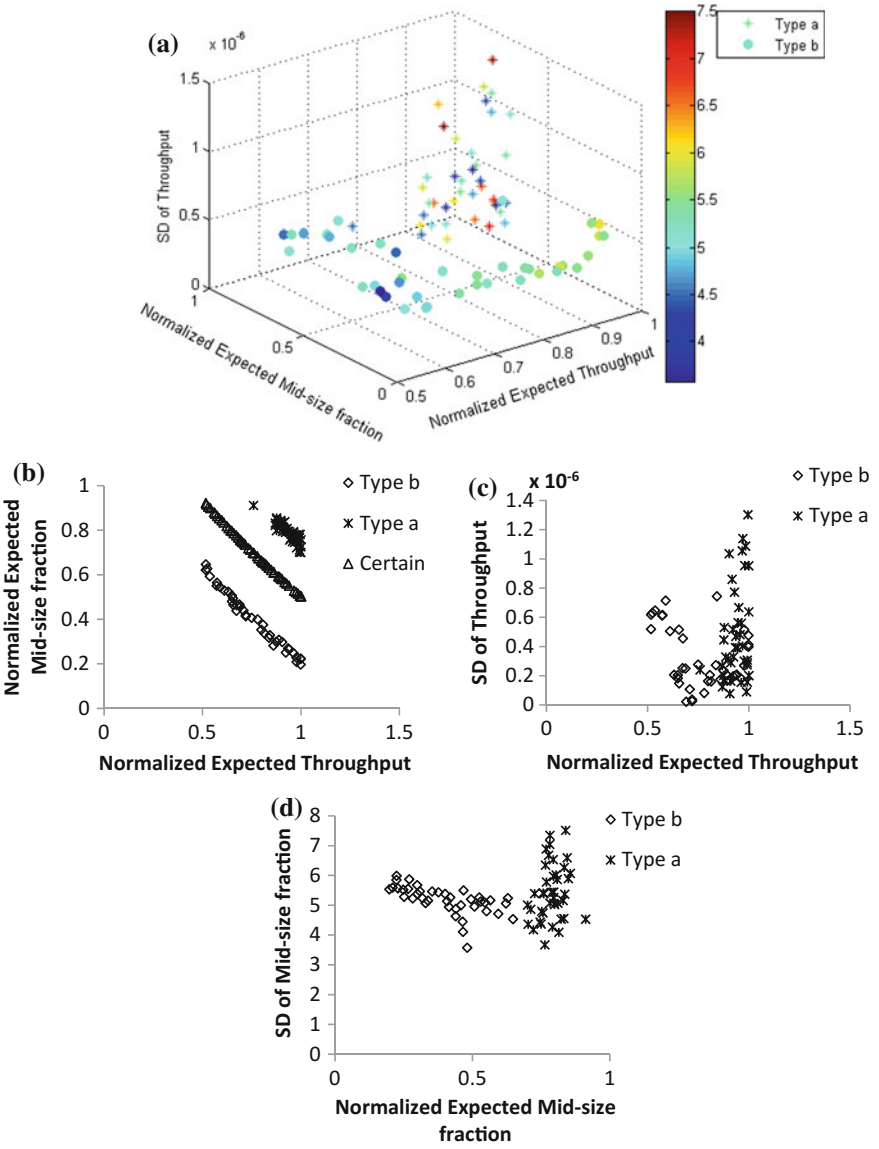


Fig. 6 Pareto optimal solutions with uncertainty in operational parameters

is relaxed further to construct the nonmembership degree which can bring more opportunity into the system. Thereby, we call this approach as optimistic approach though it gives conservative results which can be observed from Fig. 8a. Similarly, in the second approach, the uncertainty in the parameter is restricted to define non-membership degree. Hence, this approach is called as pessimistic approach and it

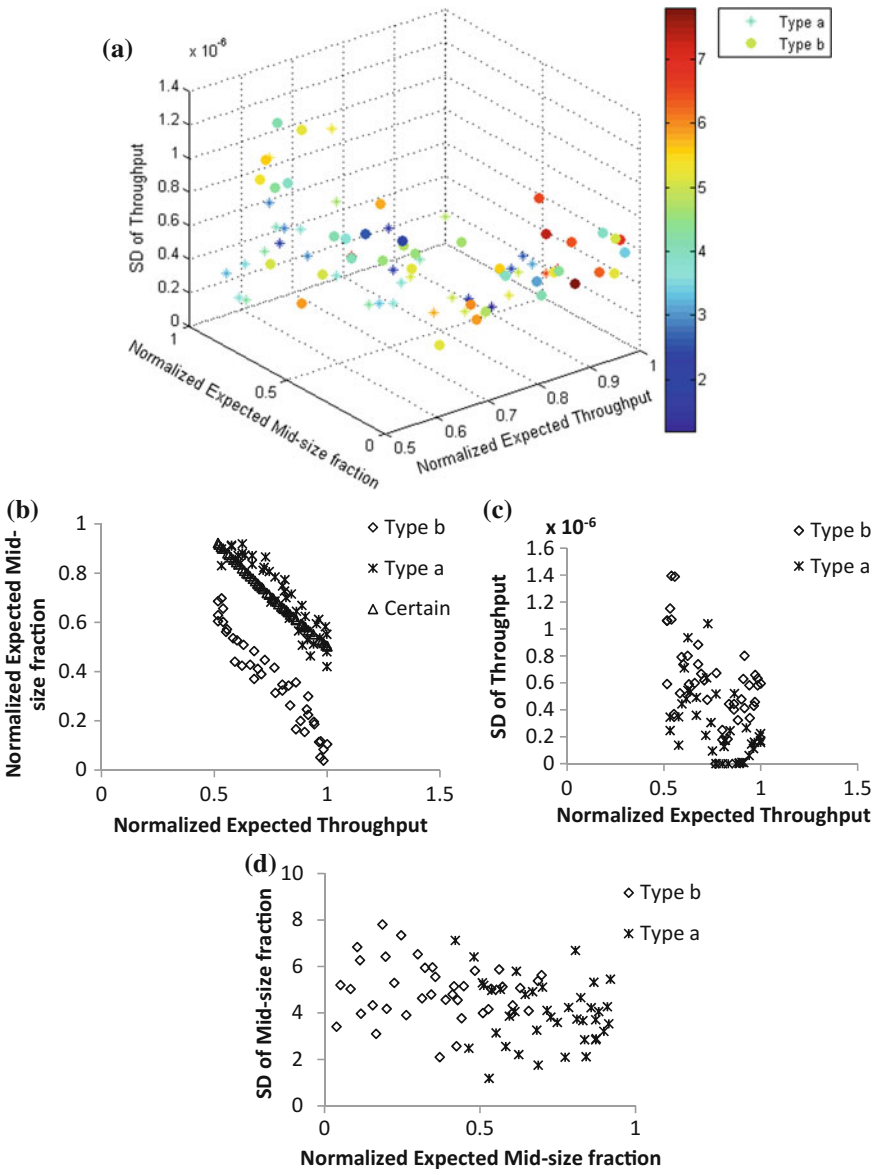


Fig. 7 Pareto optimal solutions by considering the uncertainty in model- and operational-related parameters

gives aggressive results (Fig. 8a). The third one, where the nonmembership degree is constructed between relaxed and restricted values of uncertainty in parameter, is called as the mixed approach and the optimization results by this approach are in

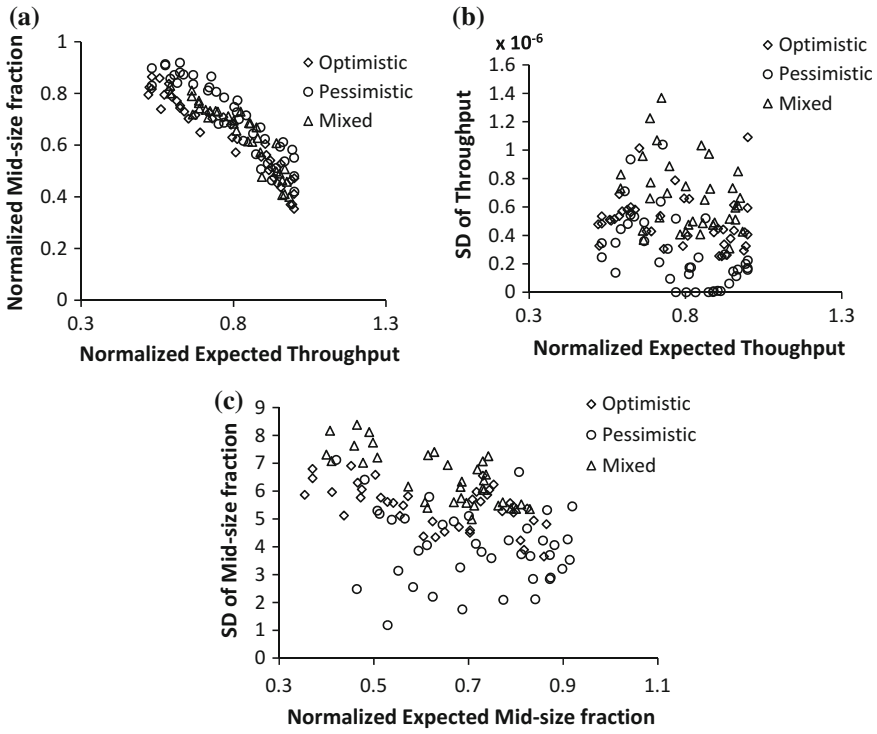


Fig. 8 Pareto optimal solutions for three different approaches

between conservative and aggressive nature results. From Figs. 8b and 8c, standard deviation values are low for pessimistic approach. Hence, it can be concluded that the PO solutions obtained by pessimistic approach give us the best optimal solutions as the objectives of the converted EDOP are maximization of expected value and minimization of SD values.

However, based on the analysis carried out in Figs. 5, 6, 7, and 8, identification of the most sensitive parameter that is responsible for the maximum PO movement is very difficult. To find this, the uncertainty analysis is considered first for one parameter at a time (say, grindability index of rod mill) and deterministic values have been taken for the remaining nine uncertain parameters. In this way, the PO solutions are obtained for different parameters which are uncertain in nature, as shown in Fig. 9. It is observed that the Pareto front shift is maximum for the uncertainty in coarse size feed fraction. Based on this observation, it can be concluded that the coarse size feed fraction is the most sensitive parameter. From Fig. 9b, c, SD values for Ms are found high when the uncertainty in coarse size feed fraction and the GIBM are considered. Finally, it can be concluded that the coarse size feed fraction is the most sensitive parameter followed by the GIBM. This is a novel way of performing sensitivity analysis of uncertain parameters which comes as a by-

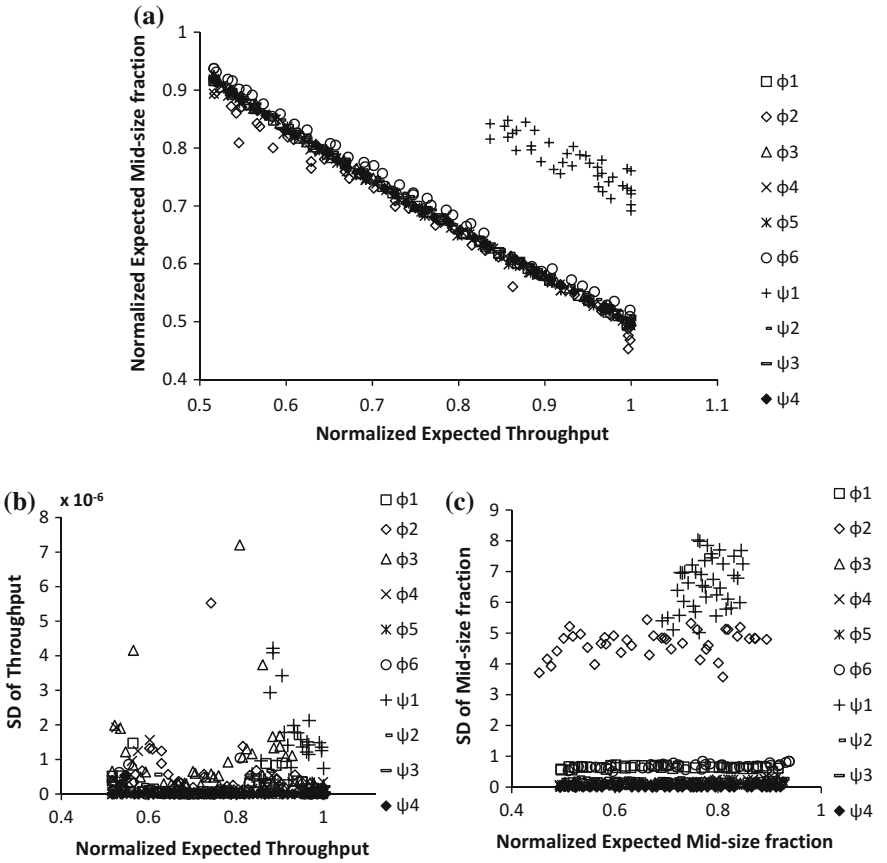


Fig. 9 Sensitivity analysis

product of this study and could be a formal way of performing the same instead of an ad hoc practice, where each of these parameters are varied randomly to see their effect on the final result. It is important to note here that it is not straightforward to find out the parameter sensitivity, otherwise, due to the nonlinear relationship among the uncertain parameters.

In Figs. 5, 6, 7, 8, and 9, the new MF can be determined using MF and NMF. The value, $\lambda = 0.5$ is used in the calculation of new membership degree which can resolve half of non-determinacy. To observe the effect of λ on the Pareto front, Pareto fronts are obtained for three different values of λ (i.e., $\lambda = 0$, 0.5, and 1) (Fig. 10). The value $\lambda = 0$ uses only MF to calculate new MF and it does not resolve non-determinacy. The PO solutions obtained with $\lambda = 0$ are conservative in nature because of the left

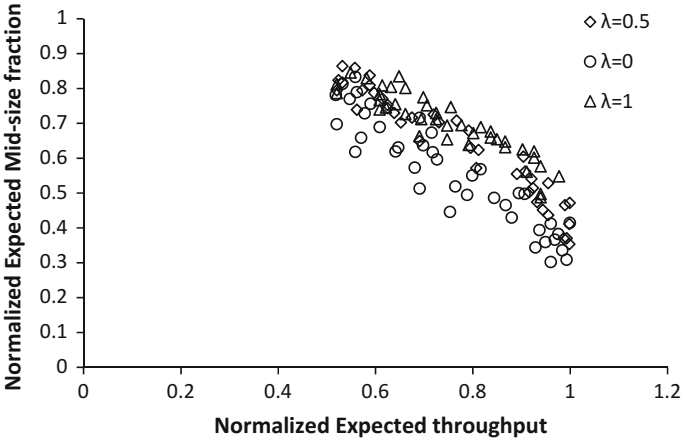


Fig. 10 Pareto optimal solutions with change in λ values

downward movement in the Pareto front. Similarly, PO solutions are obtained with $\lambda = 1$, which uses complement of NMF in place of new MF. PO front movement has been observed in the right upward direction and thus aggressive-natured Pareto front is obtained. The PO solutions obtained with $\lambda = 0.5$, which resolves half of non-determinacy, are in between conservative and aggressive-natured PO solutions.

Next, to compare the Pareto optimal solutions of FRO and IFRO, PO solutions for pessimistic approach are found out for uncertainty in second set of parameters (operational) alone. The Pareto front obtained with IFRO is restricted and shows better PO solutions because right upward movement in the PO solutions has been observed when compared with the Pareto front obtained with fuzzy robust optimization. Intuitionistic fuzzy robust optimization considers membership and nonmembership functions and resolves half of non-determinacy. As shown in Fig. 11, Pareto front obtained with intuitionistic fuzzy robust optimization got restricted when compared with the Pareto front obtained with fuzzy robust optimization which considers only membership degree to carry out the analysis. The PO solutions obtained with $\lambda = 0$ in intuitionistic fuzzy robust optimization programming matches with the PO solutions obtained with fuzzy robust optimization which uses only membership degree. In this sense, the intuitionistic fuzzy robust optimization is more generic concept which can show both the phenomena: (a) the effect of consideration of non-determinacy in the system and (b) how the concept of intuitionistic fuzzy robust optimization degenerates into the fuzzy robust optimization under special cases.

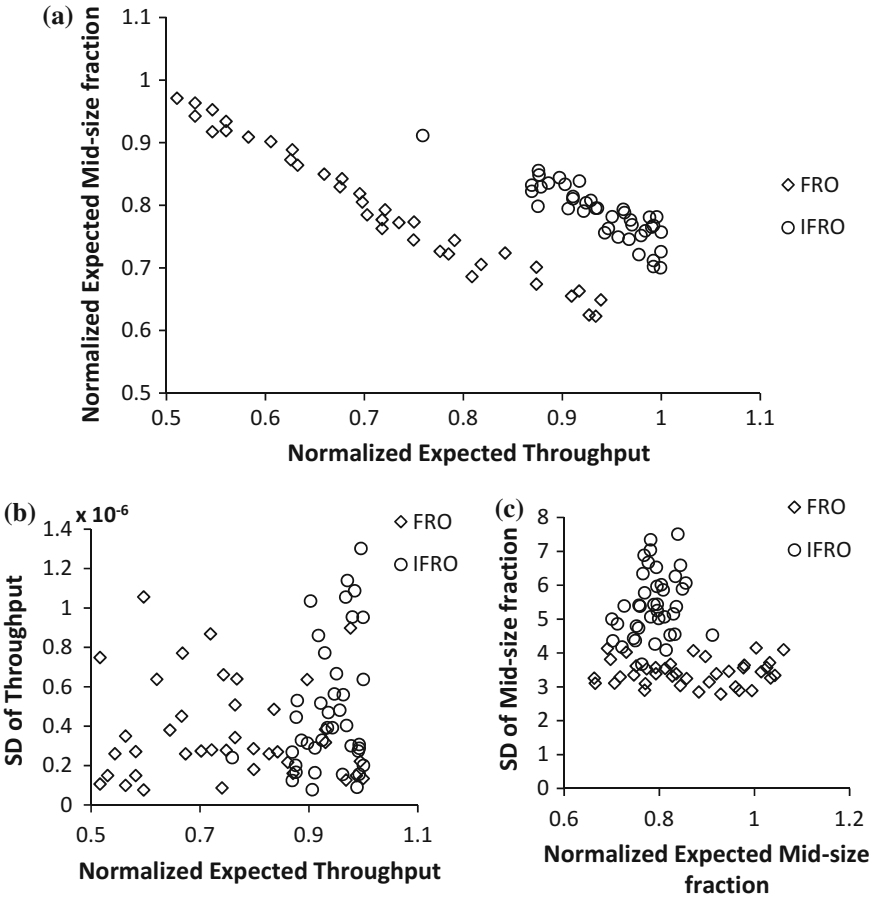


Fig. 11 Comparative study with fuzzy robust optimization and intuitionistic fuzzy robust optimization

Next, the PO solutions obtained by IFRO are compared with the well-developed robust optimization technique [2]. In both formulations, all ten uncertain parameters are relaxed 20% on lower side of their nominal values. The robust PO solutions between normalized Th and normalized Ms are obtained by the algorithm given in Sect. 3.3. The solutions obtained by RO methodology (Eq. 22) are conservative when compared to the PO solutions obtained by IFRO methodology (Fig. 12). As the two objective functions are of maximization type, PO solutions obtained by IFRO mixed

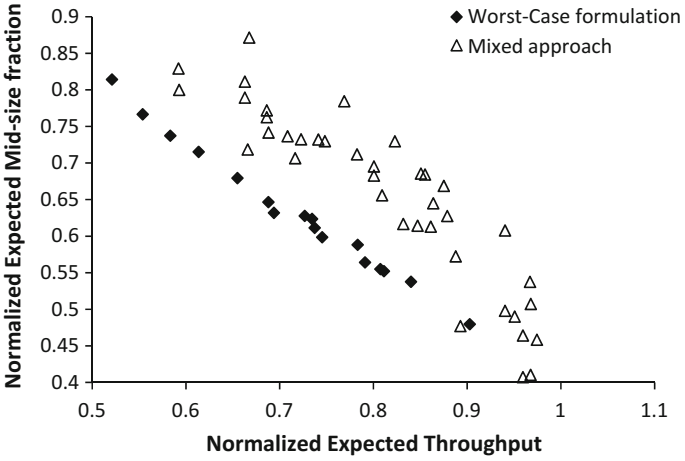


Fig. 12 Comparison of IFRO with worst-case formulation of robust optimization

approach give better results compared to the worst-case formulation [2]. This trend is evident not only for mixed approach but also for both optimistic and pessimistic approaches. Hence, the IFRO approach provides better results compared to that obtained by the worst-case formulation approaches.

In Fig. 10, PO solutions for different values of λ are plotted for optimistic approach considering all uncertain parameters. The importance of λ is the extent of non-determinacy that can be resolved while handling uncertainty. Under the special case of $\lambda = 0$ that considers only the MF and does not resolve non-determinacy term, PO solutions coincide with the solutions obtained by FRO. However, when compared with the PO solutions obtained by the worst-case formulation of Ben-Tal and Nemirovskii [2], the IFRO PO solutions for $\lambda = 0$ are not similar (see Fig. 13). The point to be noted here is that the robust optimization considers both expected value and SD terms of objective functions and the constraints, whereas the same robust optimization formulation degenerates into expected value model when standard deviation terms for the objective functions and constraints are not considered. Therefore, when the PO solutions obtained by intuitionistic fuzzy expected value model with $\lambda = 0$ are compared with the same obtained by the worst-case formulation of Ben-Tal and Nemirovskii [2], similar solutions are obtained. Though it might appear that the worst-case formulation is giving better PO solutions compared to the IFRO solutions with $\lambda = 0$, consideration of minimization of performance variability is also an equally important concept to consider which is additionally considered in IFRO. Hence, IFRO is found to be better when compared to intuitionistic fuzzy expected value model and the worst-case formulation of robust optimization because it decreases the variability in objective function and the constraints due to the presence of uncertain parameters in the formulation (Figs. 11 and 12).

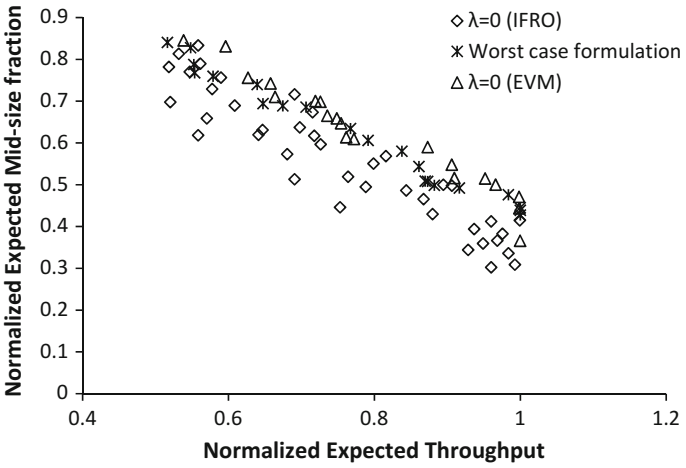


Fig. 13 Comparison of Pareto optimal solutions for $\lambda = 0$ of robust optimization and expected value model with worst-case formulation

5 Conclusions

In this work, IFRO technique has been adopted to carry out the uncertainty analysis of an IGP case study which comprises two conflicting objective functions. IFRO technique is used to convert the OUU problem into a DEF which is further solved by well-known multi-objective evolutionary algorithm, NSGA-II. The IFRO technique assumes uncertain parameters as IFNs which consider the membership and the nonmembership degrees together. IFRO technique has been applied to IGP and conclusions of the study are as follows:

- This technique shows that it is necessary to consider the parameters as uncertain when they are exposed to uncertain situations like experimental, regression errors.
- Though, there are different ways to represent the uncertain information, intuitionistic fuzzy set, generalization of fuzzy set, has been proved as a perfect implement to represent the uncertain situations when compared with FS.
- The entire set of uncertain parameters are divided into two classes, i.e., model-related uncertain parameters and operational uncertain parameters, and the uncertainty analysis has been carried out for both these classes of parameters, first considering them separately and next considering them together. It can be observed that the effect of operational parameters on Pareto front is more impactful as compared to model-related uncertain parameters.
- The IFRO technique has been applied considering three different uncertainty scenarios such as optimistic, pessimistic, and mixed approaches depicting different moods of a decision-maker. It has been found that the optimistic approach provides us with the most conservative Pareto front, whereas the pessimistic approach gives

us the most aggressive Pareto front and the solutions of the mixed approach lie in between the other two.

- Next, a novel IFRO-based sensitivity analysis study has been proposed where the uncertain parameters can be identified further based on their sensitivities on the results. In this study, the coarse size fraction and GIBM are found to be the most sensitive parameters where the former is followed by the latter in terms of their measure of impacts on the objective functions.
- It has been observed that the changing λ value, the extent of consideration of non-determinacy in the system, can also be another way of getting different natures of Pareto fronts, e.g., conservative, aggressive, etc.
- Comparative study between fuzzy robust optimization and intuitionistic fuzzy robust optimization reveals that IFRO is a more generic way of formulating robust optimization studies as it degenerates to FRO formulation as a special case. Further, IFRO approach provides with better results when compared with worst-case formulation of robust optimization in terms of minimization of variability of objective function and the constraints.

References

1. Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the-art and opportunities. *Computers & Chemical Engineering, 28,* 971–983.
2. Ben-Tal, A., & Nemirovskii, A. (2001). *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*. *MPS-SIAM Series on optimization*. Philadelphia: MPS-SIAM.
3. Liu, B. (2002). *Theory and practice of uncertain programming*. Heidelberg: Physica-Verlag.
4. Liu, M. L., & Sahinidis, N. V. (1996). Optimization in process planning under uncertainty. *Industrial and Engineering Chemistry Research, 35,* 4154–4165.
5. Charnes, A., & Cooper, W. W. (1959). Chance-constrained programming. *Management Science, 6,* 73–79.
6. Gupta, A., & Maranas, C. D. (2000). A two-stage modeling and solution framework for multisite midterm planning under demand uncertainty. *Industrial and Engineering Chemistry Research, 39,* 3799–3813.
7. Ayoub, N., Martins, R., Wang, K., Seki, H., & Naka, Y. (2007). Two levels decision system for efficient planning and implementation of bioenergy production. *Energy Conversion and Management, 48,* 709–723.
8. Mitra, K., Gudi, R. D., Patwardhan, S. C., & Sardar, G. (2008). Midterm supply chain planning under uncertainty: a multi-objective chance constrained programming framework. *Industrial and Engineering Chemistry Research, 47,* 5501–5511.
9. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8,* 338–353.
10. Palwak, Z. (1985). Rough sets and fuzzy sets. *Fuzzy Sets and Systems, 17,* 99–102.
11. Gau, W. L., & Buehrer, D. J. (1993). Vague sets. *IEEE Transactions on Systems, Man and Cybernetics, 23,* 610–614.
12. Sambuc, R. (1975). Fonctions φ-floues. Application l'aide au diagnostic en pathologie thyroidienne, Ph.D. Thesis, University of Marseille, France.
13. Zimmermann, H. J. (1991). *Fuzzy set theory and its application*. Boston: Kluwer Academic Publishers.
14. Dubois, D., & Prade, H. (1988). *Possibility theory: An approach to computerized processing of uncertainty*. New York: Plenum Press.

15. Virivinti, N., & Mitra, K. (2014). Fuzzy expected value analysis of an industrial grinding process. *Powder Technology, 286,* 9–18.
16. Tian, G., Chu, J., Liu, Y., Ke, H., Zhao, X., & Xu, G. (2011). Expected energy analysis for industrial process planning problem with fuzzy time parameters. *Computers & Chemical Engineering, 35,* 2905–2912.
17. Mitra, K. (2009). Multi-objective optimization of an industrial grinding operation under uncertainty. *Chemical Engineering Science, 64,* 5043–5056.
18. Virivinti, N., & Mitra, K. (2016). A comparative study of fuzzy techniques to handle uncertainty: An industrial grinding process. *Chemical Engineering and Technology, 39,* 1031–1039.
19. Marano, G. C., & Quaranta, G. (2009). Robust optimum criteria for tuned mass dampers in fuzzy environments. *Applied Soft Computing, 9,* 1232–1243.
20. Zhang, X., Huang, G. H., Chan, C. W., Liu, Z., & Lin, Q. (2010). A fuzzy-robust stochastic multiobjective programming approach for petroleum waste management planning. *Applied Mathematical Modelling, 34,* 2778–2788.
21. Diez, M., & Peri, D. (2010). Robust optimization for ship conceptual design. *Ocean Engineering, 37,* 966–977.
22. Capolei, A., Suwartadi, E., Foss, B., & Jorgensen J. B. (2015). A mean–variance objective for robust production optimization in uncertain geological scenarios. *Journal of Petroleum Science and Engineering, 125*, 23–37.
23. Atanassov, K. T. (1986). Intutionistic fuzzy sets. *Fuzzy Sets and Systems, 20,* 87–96.
24. Virivinti, N., & Mitra, K. (2015). Intuitionistic fuzzy chance constrained programming for handling parametric uncertainty: An industrial grinding case study. *Industrial and Engineering Chemistry Research, 54,* 6291–6304.
25. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley.
26. Yager, R. R. (2009). Some aspects of intuitionistic fuzzy sets. *Fuzzy Optimization and Decision Making, 8,* 67–90.
27. Bellman, R., & Zadeh, L. A. (1970). Decision-making in a fuzzy environment. *Management Science, 17,* 141–161.
28. Angelov, P. P. (1997). Optimization in an intuitionistic fuzzy environment. *Fuzzy Sets and Systems, 86,* 299–306.
29. Liu, B., & Liu, Y. K. (2002). Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems, 10,* 445–450.
30. Dubey, D., Chandra, S., & Mehra, A. (2012). Fuzzy linear programming under interval uncertainty based on IFS representation. *Fuzzy Sets and Systems, 188,* 68–87.
31. Petzold, L. R. (1983). A description of DASSL: A differential/algebraic system solver. In *Scientific computing* (pp. 65–68). North-Holland, Amsterdam.
32. Mitra, K., & Gopinath, R. (2004). Multiobjective optimization of an industrial grinding operation using elitist non-dominated sorting genetic algorithm. *Chemical Engineering Science, 59,* 385–396.

Optimisation in Friction Stir Welding: Modelling, Monitoring and Design

Qian Zhang and Xiaoxiao Liu

Abstract The friction stir welding process involves a highly complex microstructure evolution. This makes it very difficult to derive intricate relationships among operating conditions, in-process variables and characteristics of welds, and utilise the relationships into modelling, monitoring and optimal design of operating conditions. In this research, a heuristic optimisation paradigm Reduced Space Searching Algorithm, combined with soft-computing-based modelling and data analysis techniques, is employed to solve the problem. The research investigates an aluminium alloy AA5083 and includes three facets of research: first, developing a weld quality indicator that can provide a reliable indication of 'as-welded' defects for an online monitoring system; second, generating accurate and interpretable prediction models for both internal process attributes and post-weld properties; third, finding optimal operating conditions to enhance welding productivity, process reliability and cost efficiency.

Keywords Friction stir welding · Aluminium · Monitoring
Data-driven modelling · Optimal design · Operating conditions
Multi-objective optimisation · Reduced space searching · Fuzzy systems

1 Introduction

As a relatively new solid-state joining method, Friction Stir Welding (FSW) [1] has been proven a very practical joining technique that produces excellent mechanical and microstructural properties for post-weld materials. It has found various and increasing industrial applications in aerospace, railway and shipbuilding industries.

Q. Zhang (✉)
Department of Electronics and Electrical Engineering, Liverpool John Moores University, Liverpool L33AF, UK
e-mail: Q.Zhang@ljmu.ac.uk

X. Liu
General Engineering Research Institute, Liverpool John Moores University, Liverpool L33AF, UK

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_11

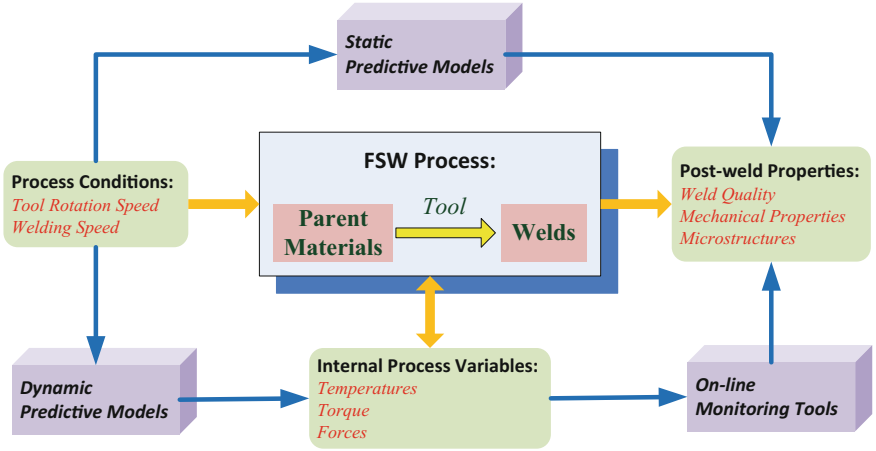


Fig. 1 The FSW diagram with important variables, predictive models and online monitoring tools

Generally, FSW uses a non-consumable rotating tool in the welding process. The tool has a pin on the top of a shoulder. During an FSW process, the tool pin needs to be inserted into the rigidly clamped edges of two adjoining workpieces. Its shoulder needs to contact the workpieces' top surface rigidly. The welding tool then rotates and travels along a desired joint line. The rotation generates heat from friction. With the rotation of the tool, softened material moves from the front to the trailing edge of the tool and then the material is forged into the joint [2]. FSW includes intricate and severe thermo-mechanical and recrystallisation processes, which develop a highly complex texture of welds.

In FSW, operating conditions, in-process attributes and properties of welded materials are three important groups of attributes, as shown in Fig. 1. The operating conditions, for example, tool rotational speed and tool travel speed, are used to control the welding process. The in-process variables, for example, tool temperature profile and forces on tool, are important features that can be observed and serve essential information for understanding the status of the undergoing process. The post-weld properties, for example, microstructural properties and mechanical properties of welds, represent the quality and information of the final product.

For an optimal design of FSW operating conditions, or monitoring and control the FSW process, it is very important to investigate the relationships between operating conditions, in-process variables and post-weld properties. It is also crucial to further design practical predictive models and optimisation paradigms. Due to the high complexity of the friction stir process, it is quite difficult to elicit accurate physical correlations and models. In such a case, some novel intelligent systems, heuristic optimisation algorithms, and data analytics and modelling strategies may be utilised into the FSW research.

In the last two decades, multi-objective optimisation techniques using heuristic approaches have started to be utilised in manufacturing and materials processes

[3–6, 7]. In the field of FSW, Tansel et al. [8] implemented a Genetic Algorithm (GA) to locate the optimal process conditions from the ANN models. In the work of Roshan et al. [9], a neuro-fuzzy model was developed to predict the mechanical properties for alloy AA7075. The simulated annealing optimisation technique was then employed to find the best mechanical properties. In Parida and Pal's work [10], a fuzzy-assisted Taguchi method was designed to improve different operating conditions of FSW, where the multi-objective optimisation case was converted to a single-objective problem. In the researches [11, 12], a multi-objective optimisation tool NSGA-II was employed in thermal models to tackle an optimal design problem with two objectives, i.e. maximising welding speed and minimising the residual stress of the welded material simultaneously [11], and maximising production efficiency and tool life simultaneously [12]. In the work of Shojaeefard et al. [13], the researchers employed a Multi-objective Particle Swarm Optimisation to optimise the operating conditions to achieve desired post-weld properties. All of the above researches only addressed single-objective [8–10] or two-objective [11–13] design problems. In the current chapter, more conflicting objectives are considered, which include welding quality, mechanical properties, economic cost and internal process variables.

The chapter will provide examples showing how heuristic optimisation algorithms help in data analysis, modelling and optimal design of FSW. A nature-inspired searching and optimisation technique, Reduced Space Searching Algorithm (RSSA) [14, 15] is employed across a set of studies relating to the FSW of the AA5083 aluminium alloy. First, RSSA helps analyse relationships between in-process variables and weld quality, and then design reliable indices for online monitoring [16]. Second, the optimisation algorithm is used to elicit accurate and robust prediction models for internal process features and post-weld properties where operating conditions being inputs [17]. Last, RSSA is used to exploit the constructed models to design defect-free, structurally sound and low-cost welds via a 'reverse-engineering' fashion [18].

The remaining parts of this chapter are arranged as follows. In Sect. 2, an introduction about the experimental details is provided. Section 3 introduces the key information about RSSA. In Sect. 4, several data analyses methods are used to study the correlations between the in-process and post-weld variables, and RSSA is then implemented to produce a more precise and practical weld quality indicator. Section 5 presents some work of constructing models for FSW-related properties, where RSSA is employed to improve the structure and parameters of fuzzy models. In Sect. 6, RSSA is utilised to get best operating conditions to realise some quality and economic objectives, ranging from two to five objectives. In the end, conclusions are provided in Sect. 7.

2 Material and Experiments

2.1 Material and Experimental Settings

The aluminium alloy AA5083-O was used in this research. It is a non-heat-treatable material with very good strength and good corrosion resistance [19]. In the FSW trials, the samples of 5.8 mm thick were FSW-joined to be butt welds. A high-performance welding tool MX-Triflute^{TM} with a 25 mm wide scroll shoulder was adopted [20]. This welding tool can improve material flow of FSW and enables a large increase in welding speed, which makes the welding more practical and efficient [21].

There are two attributes used for controlling the FSW process, i.e. the clockwise tool Rotational Speed (RS) (rpm) and the Welding Speed (WS) following the joint line. The latter is also represented by the Forward Movement (FM) per revolution (mm/rev). In this research, the experiments were undertaken according to a 5×5 testing matrix, whose parameters include five tool rotational speeds, 280, 355, 430, 505 and 580 rpm, and 5 forward feed rates, 0.6, 0.8, 1.0, 1.2 and 1.4 mm/rev.

2.2 Assessment of In-process Variables

The Artemis platform (Advanced Rotating Tool Environment Monitoring and Information System) is an advanced online sensory system that was employed to in-process collect the data representing the internal status of welding. It was developed by TWI and works as an extensively instrumented tool holder. The data that may be collected include temperatures of various parts, forces and torque of the tool, for instance, bending force, axial compression, traverse force, etc. Figure 2 shows an example of some data collected by Artemis.

Moreover, Artemis is able to continuously test the lateral bending forces of the welding tool in 48 channels of a rotation. Displaying the multichannel bending forces on a polar plot provides instant visualisation for the force profile in a whole 360 degree rotation [21]. This is also called ‘tool force footprint’. Figure 3 gives an example of a tool force footprint.

2.3 Assessment of Post-weld Properties

Tensile tests were carried out by utilising a Digital Image Correlation (DIC) device, where a 2-Megapixel camera and a LaVision system worked together to measure and acquire displacement. The specimens were tested for tensile properties, including Reduction of Area (ROA), elongation, Yield Strength (YS) and Ultimate Tensile Strength (UTS), all at room temperature.

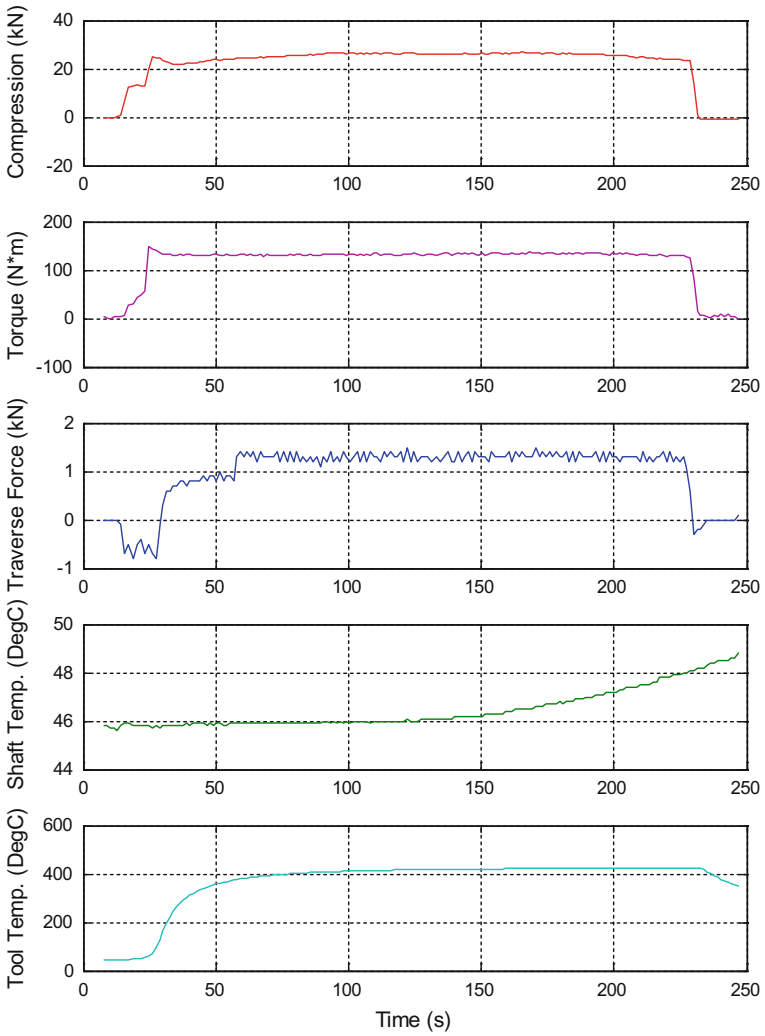


Fig. 2 An example of in-process variables: RS 355 rpm and FM 0.8 mm/rev

For each welding profile, five specimens with two geometries were tested. All of them were machined in the transverse orientation. In this way, the strength measured reflects the weakest zone of welds. However, the ductility measured reflects an average situation across all zones. There are two failure types that may happen in the welds, whose examples are shown in Fig. 4. The first type of failure is a shear fracture and the second type of failure includes defects and voids. The former happens in the heat-affected zone while the latter happens in the nugget zone.

For FSW, the normal defects are volumetric defects caused by the inadequate stir and insufficient material flow [2]. To assess the Welding Quality (WQ), four

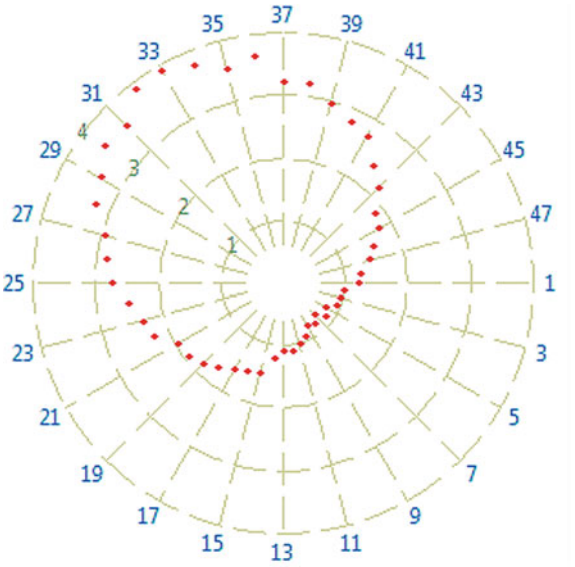


Fig. 3 A force footprint example: Angle 37 is the tool travel direction

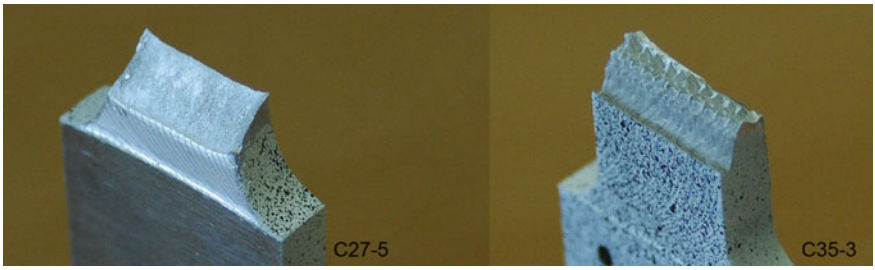


Fig. 4 Comparison of two types of failures

independent tests were conducted, (1) inspection of surface, (2) inspection of cross section, (3) bend test from root side and (4) bend test from surface side. In every subtest, an index with a value range between 0 and 3 is designed to indicate the quality degree. They are integrated to form an overall WQ index to show the general status of welding quality, where its value ranges from 0 (best quality) to 12 (worst quality).

The FSW process involves significant microstructural evolution, which greatly changes grain size and grain boundary character [22]. The Average Grain Size (AGS) at welds' nugget zone is measured in this research. Figure 5 provides examples of micrographs, including a pre-weld sample and a post-weld sample. The value ranges of the relevant variables are illustrated in Table 1.

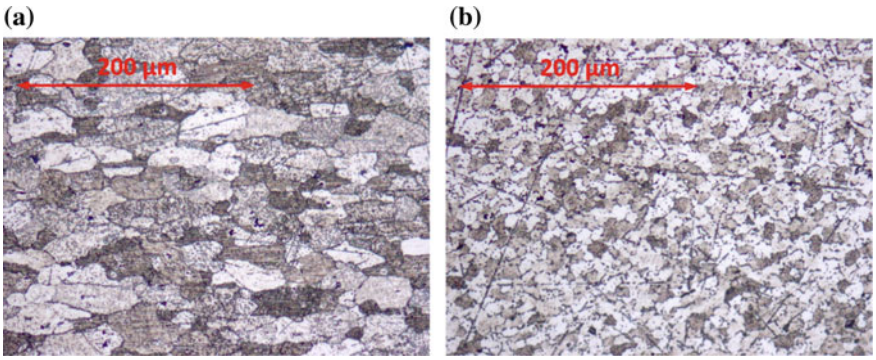


Fig. 5 Micrograph examples: **a** a parent sample and **b** a weld with RS 580 rpm and FM 0.6 mm/rev

Table 1 Value ranges of relevant variables

Variables	Value ranges
Traverse speed	168–812 mm/min
Forward feed rate	0.6–1.4 mm/rev
Tool rotational speed	280– 580 rpm
Reduction of area	13.0–33.3%
Elongation	9.8–21.7%
Ultimate tensile strength	229–320 MPa
Yield strength	162–184 MPa
Weld quality index	0–8
Average grain size	7.0–14.5 µm
Torque	−3.4–174 N * m
Compression	−3.0–34.4 kN
Traverse force	−1.2–4.9 kN
Tool temperature	46.0–438 °C
Shaft temperature	44.1–54.4 °C

3 Reduced Space Searching Algorithm

People have designed many efficient optimisation techniques based on the inspiration from natural and social behaviours. Following a similar manner, an optimisation paradigm Reduced Space Searching Algorithm (RSSA) was developed [14, 15]. It was inspired from a human experience in seeking for the optimal solution in our daily life. The most distinct difference between Reduced Space Searching and other optimisation algorithms comes from the operational emphases. The majority of other optimisation methods focus on producing candidate solutions by implementing different mechanisms, such as mutation operators, recombination operators, PSO equations or derivative-related equations, however, RSSA focuses on changing and

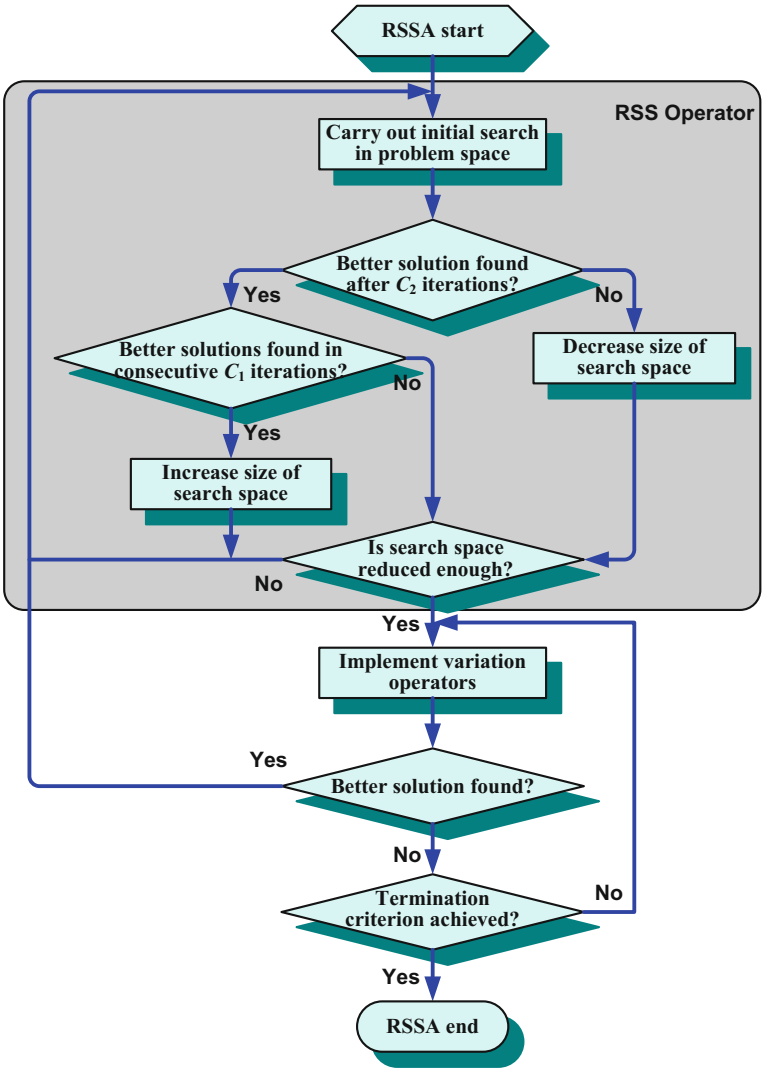


Fig. 6 The flow diagram of reduced space searching algorithm

shifting the search space to find the best sub-area. Figure 6 gives its flow chart and more details can be found in the literatures [14, 15].

RSSA was initially designed to solve single-objective optimisation problems. In order to extend it to cope with the cases with multiple objectives, a varying weighted aggregation strategy [23, 24] was utilised. For retaining non-dominated (Pareto-optimal) solutions, an archive mechanism was also designed. The multi-objective RSSA is summarised as a number of steps as follows:

Table 2 Parameters of RSSA and MO-RSSA used in the FSW experiments

Parameters	Values
Increasing parameter C_2	1
Decreasing parameter C_1	9
Exponent threshold m	20
Changing ratio k	0.5
Maximal function evaluation $E_{\max}$	100,000
Frequency parameter H	1000

1. Initialising weights for different optimisation objectives.
2. Employing the RSS operator to optimise an updated problem with a weighted-sum objective.
3. Adding the current best solution to the archive.
4. For the archive, executing a non-dominated selection.
5. For the archive, executing a diversity selection.
6. Varying the objectives' weights using Random Weighted Aggregation (RWA) [14, 24].
7. Repeating Steps 2–6 if the termination criterion is not achieved.

The algorithms RSSA and MO-RSSA have been validated in solving challenging benchmark problems. They showed to outperform many other optimisation algorithms [14]. In the experiments of the following sections, the parameters setting in Table 2 applies.

4 Optimal Design of Weld Quality Indicator

In this research, we first identified the correlations between Weld Quality (WQ) and in-process variables by implementing statistical correlation analysis, Fourier frequency analysis and wavelet analysis. After this, we elicited a new and more precise indicator for online monitoring of the weld quality through employing an optimisation algorithm.

4.1 Statistical Analysis

In this work, two dependency measures were used. Pearson's correlation coefficient (r) is reflective of the linear correlation. Correlation ratio (η) is able to detect the more general type of dependency. Torque, axial compression force, traverse force (TF) and temperatures of tool were considered. The most significant statistical features in the

Table 3 The in-process variables detected using correlation measures

Measures	1st best correlated variable	2nd best correlated variable	3rd best correlated variable
Correlation coefficient r	Min of TF ($r = 0.706$)	Mean of TF ($r = 0.693$)	Max of TF ($r = 0.684$)
Correlation ratio η	Max of TF ($\eta = 0.906$)	Mean of TF ($\eta = 0.886$)	Min of TF ($\eta = 0.881$)

Table 4 The tool bending forces detected using correlation measures

Measures	1st best correlated variable	2nd best correlated variable	3rd best correlated variable
Correlation coefficient r	Mean of BF on Angle 4 ($r = 0.774$)	Mean of BF on Angle 3 ($r = 0.773$)	Min of BF on Angle 4 ($r = 0.770$)
Correlation ratio η	Mean of BF on Angle 2 ($\eta = 0.904$)	Max of BF on Angle 2 ($\eta = 0.904$)	Min of BF on Angle 2 ($\eta = 0.900$)

steady state, i.e. minimum, maximum, mean and standard deviation, were correlated with weld quality.

Table 3 shows the variables which were detected to possess the biggest dependence. From this experiment, it was found that the variables relating to shaft temperature and traverse force have the best correlation with the WQ. This observation is reasonable, since the tool traverse force reflects the status of material flow, where abnormal or insufficient material flow causes the formation of volumetric defects. On the other hand, temperature reflects to the situation of heat input, which directly relates to material fluidity and flow. Both inadequate and excessive heat input will increase the possibility of the cavities formation in the weld zone [25].

In the second experiment, detailed bending forces (BFs) of the tool were investigated, where the minimum, maximum, mean and standard deviation of them on 48 channels were correlated with the WQ. Table 4 demonstrates the attributes best correlated with WQ. From this experiment, we observed that WQ has a better correlation with the BFs located at two regions, Angles 21–31 and Angles 2–5. From Fig. 3 we can see that Angles 21–31 locate at the advancing side's edge and Angles 2–5 locate at the retreating side's back. Both of the regions are the place where the formation of voids often happens.

4.2 Frequency Analysis

It was noted that oscillations in welding tool's force profile are relevant to the dynamics of the FSW's material flow [26]. They may thus indicate in-depth information for FSW. We utilised a Fast Fourier Transform (FFT) method to evaluate the in-process data and try to reveal the hidden frequency-domain information that relates to the WQ circumstance.

Table 5 The in-process variables detected using frequency analysis

Measures	1st best correlated variable	2nd best correlated variable	3rd best correlated variable
Correlation coefficient r	BF on Angle 35 ($r =$ 0.895)	BF on Angle 34 ($r =$ 0.893)	BF on Angle 36 ($r =$ 0.890)
Correlation ratio η	BF on Angle 34 ($\eta =$ 0.955)	BF on Angle 33 ($\eta =$ 0.934)	BF on Angle 32 ($\eta =$ 0.924)

From some preliminary experiments, it was observed that the high-frequency components of the frequency spectra would relate to WQ. Based on this observation, all the in-process variables were analysed. The largest 100 amplitudes, ranging from tool rotational frequency to sampling frequency, were aggregated and assessed. Table 5 gives the in-process variables identified. It is found that the defect formation is likely correlated with the high-frequency oscillation of BFs at the front region of the advancing side (Angles 32–36).

4.3 Wavelet Analysis

The wavelet transform is a method deemed to represent time and frequency information of temporal signals at the same time. It is a technique good at analysing non-stationary signals, non-periodic signals and the signals with sharp and discontinuities peaks. The wavelet transform can be described using the following equation:

$$s(t) = A\varphi(t) + \sum_{i=1}^{L} D_i \psi_i(t) \tag{1}$$

where $s(t)$ is a signal to be decomposed, $\varphi(t)$ is a scaling function ('approximation' function used to represent low-frequency information) and $\psi_i(t)$ are wavelet functions ('detail' functions used to represent high-frequency information). L is the wavelet transform level ($L = 3$ in this study). A is the approximation coefficient and D_i are the detail coefficients.

Generally, in a wavelet transform, the types of scaling and wavelet functions are given. One only needs to generate the coefficients A and D_i. In this chapter, square-shaped Haar wavelet was used, which has been proven to be good at monitoring machining signals [27].

Figure 7 gives examples of Haar wavelet transform for the BF of three specimens. The high-WQ specimen has relatively small D_1–D_3 (detail coefficients), while the low-WQ specimen has larger D_1–D_3. Based on this observation, all the in-process variables were wavelet transformed and assessed. We summed the detail coefficients of various scale levels and analysed the sums against the overall WQ. Table 6 shows the three best variables identified, whose dependency values are quite high.

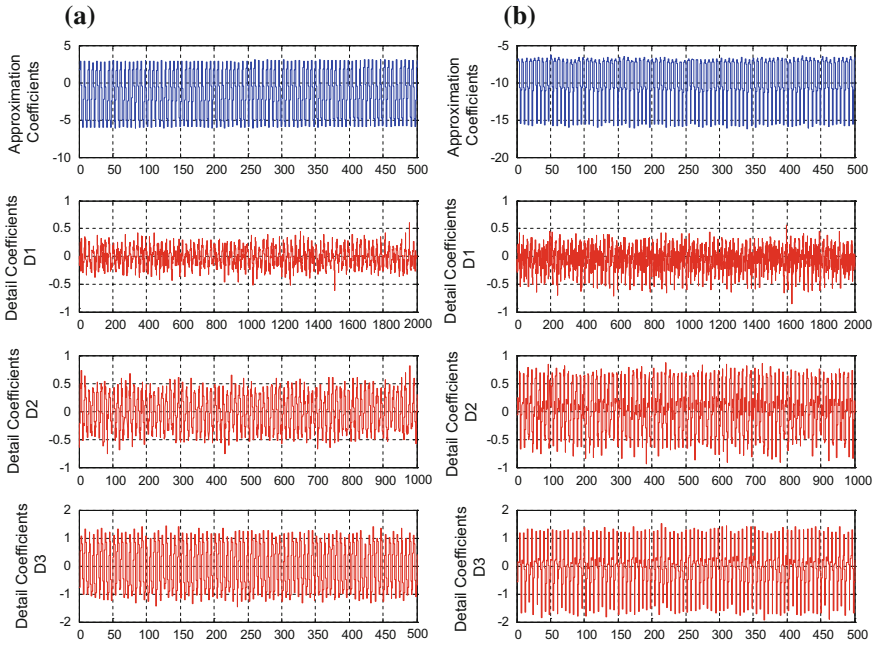


Fig. 7 The wavelet transform of the BF on Angle 46: **a** a high-WQ specimen and **b** a low-WQ specimen

Table 6 The in-process variables detected using wavelet analysis

Measures	1st best correlated variable	2nd best correlated variable	3rd best correlated variable
Correlation coefficient r	Sum of D_2 of the BF on Angle 46 ($r = 0.864$)	Sum of D_3 of the BF on Angle 35 ($r = 0.847$)	Sum of D_2 of the BF on Angle 45 ($r = 0.845$)
Correlation ratio η	Sum of D_2 of the BF on Angle 46 ($\eta = 0.955$)	Sum of D_2 of the BF on Angle 29 ($\eta = 0.926$)	Sum of D_3 of the BF on Angle 36 ($\eta = 0.919$)

4.4 Optimal Design of Weld Quality Indicator

In Sects. 4.1–4.3, multiple attributes were found helpful in assessing the welding quality. However, in a real-life online monitoring system, users may demand a more precise monitoring indicator. A practical but simple method to obtain such an indicator is integrating several in-process variables as the following linear combination:

$$Ind(x) = \sum_i w_i V_i(x) + c, \tag{2}$$

where $Ind(x)$ represents a desired WQ indicator with the value ranging between 0 (best quality) and 1 (worst quality); $V_i(x)$ are the internal variables previously found

Table 7 The correlation coefficient values of Ind_1 and other in-process-variable indicators

Indicators	r between an indicator and overall WQ
Max of TF	0.684
Mean of ST	0.252
Mean of the BF on Angle 2	0.758
Sum of highest frequency spikes of the BF on Angle 34	0.893
Sum of D_2 in wavelet analysis of the BF on Angle 46	0.864
Ind_1	0.951

in Sects. 4.1–4.3; weights w_i and constant c are coefficients to be identified using an optimisation algorithm.

The following optimisation problem was constructed to find appropriate w_i and c:

Objective: Maximising the correlation coefficient value r_{Ind} (between $Ind(x)$ and the overall WQ index, using experimental samples x_k) towards 1.
Constraint: $-1 \leq Ind(x_k) \leq 1$

In this research, RSSA was employed and its parameters settings are as described in Sect. 3.

It was assumed that five in-process factors were selected as V_i (x). They include (1) mean of shaft temperature (ST) $ST_{mean}(x)$, (2) maximum of TF $TF_{max}(x)$, (3) mean of BF on Angle 2 $BF2_{mean}(x)$, (4) sum of detail coefficients D_2 in wavelet analysis of the BF on Angle 46 $SD2_{BF46}(x)$ and (5) sum of highest 100 amplitudes in frequency spectrum of the BF on Angle 34 $SA_{BF34}(x)$. After the optimisation process, a new indicator was obtained as follows:

$$\begin{aligned} Ind_1(x) = {} & 0.027TF_{\max}(x) + 0.016ST_{mean}(x) - 0.088BF2_{mean}(x) \\ & + 0.509SA_{BF34}(x) + 0.134SD2_{BF46}(x) - 1.913 \end{aligned} \tag{3}$$

Table 7 summarises the r value of Ind_1 and every single in-process variable involved in Ind_1. Figure 8 exhibits their correlations with the weld quality intuitively by using scatter plots. The integrated indicator is clearly better than any single in-process indicator. Therefore, it is more suitable to be applied to real-life quality monitoring.

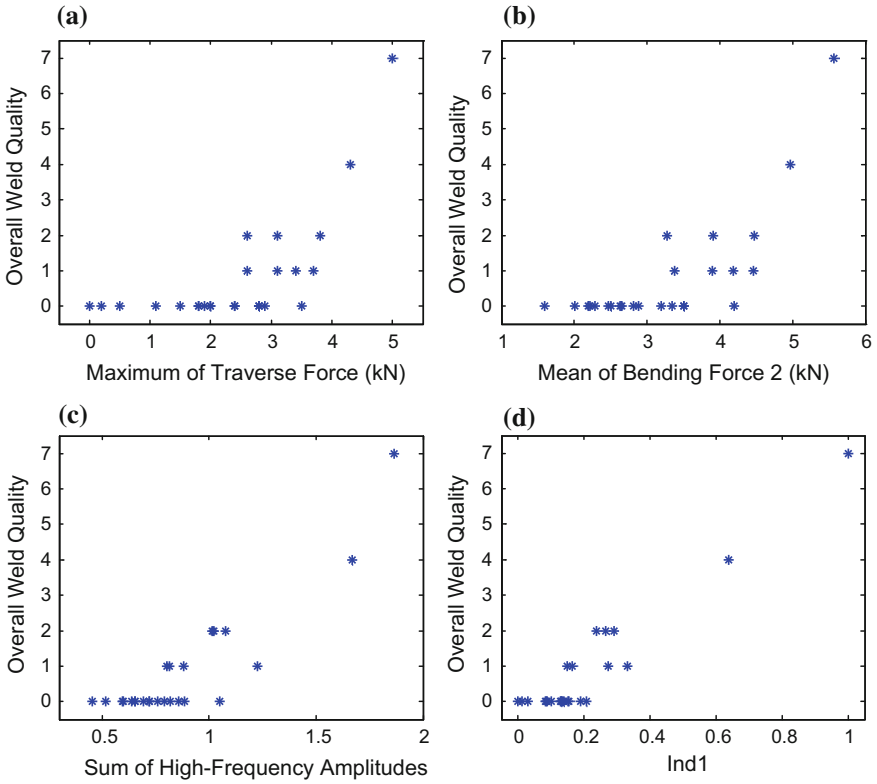


Fig. 8 The scatter plots of the overall WQ index against **a** the maximum of TF, **b** the mean of the BF on Angle 2, **c** the sum of the highest 100 frequency-domain amplitudes of the BF on Angle 34 and **d** Ind_1

5 Modelling Using Fuzzy Rule-Based Systems and RSSA

Due to the high complexity of FSW, it is very difficult to obtain physical models of the process. In this case, a data-driven modelling approach based on fuzzy systems was developed to elicit prediction models for internal process features, mechanical properties and microstructural features (See Fig. 1). Fuzzy Rule-Based Systems (FRBSs) [28] are able to learn complex relationships from data while needing very little prior knowledge about the process and materials. They are more interpretable than some other black box modelling methods, as FRBSs implement descriptive IF-THEN rules.

5.1 Hierarchical Multi-objective Fuzzy Modelling

An FRBS normally includes four parts: fuzzifiers, defuzzifiers, a fuzzy inference engine and a fuzzy rule-base. Two general approaches are used for fuzzy modelling, which are (1) knowledge acquisition from experts and (2) knowledge discovery from data (data-driven modelling). The knowledge acquisition method has a big limitation due to the lack of reliable expert knowledge. Nowadays, data-driven modelling becomes the mainstream method because of the availability of 'big data'.

For data-driven modelling using FRBSs, the learning techniques can be gradient descent, linear least squares, neural-fuzzy training or evolutionary optimisation. Evolutionary fuzzy systems show to be very effective to improve both the structure and the parameters of FRBSs [29, 30]. In addition, multi-objective evolutionary optimisation is very valuable in investigating the trade-off between transparency and accuracy of FRBSs [29, 30].

Figure 9 shows the designed fuzzy modelling framework, in which a hierarchical structure in optimisation is suggested. In this method, two learning phases are executed iteratively and sequentially to improve two aspects of FRBSs: MO-RSSA is mainly utilised to improve the structure of models and the single-objective RSSA is utilised to improve the parameters of models.

First, an initial fuzzy model is constructed by using clusters information. Such an initial model is then structurally improved, i.e. moderation of fuzzy sets and rules, considering the transparency issues. It is achieved via a four-step operation, i.e. (1) merging similar fuzzy rules, (2) removing redundant fuzzy rules, (3) merging similar fuzzy sets and (4) removing redundant fuzzy sets. The four operations are controlled using four threshold coefficients and the generation of these threshold coefficients is determined by MO-RSSA. Given a fuzzy system with an improved structure, RSSA is further employed to improve the modelling accuracy. More details can be found in the literatures [17, 30].

5.2 Modelling of In-process Variables

Development of models for in-process variables is crucial for designing an efficient and safe FSW process. For instance, the models of temperatures and forces can help avoid tool wear or overheating problems in the design of welding conditions. In the research, the in-process variables modelled include bending force, compression force, traverse force, torque, shaft temperature and tool temperature.

The following shows the case of modelling the Tool Peak Temperature (TPT), and Fig. 10 demonstrates the trade-offs between different non-dominated models with respect to various criteria, including RMSE, fuzzy rules number, fuzzy rules length and fuzzy sets number. In the experiments in Sects. 5.2–5.4, 20 data samples were employed in training and 5 data samples were employed in testing.

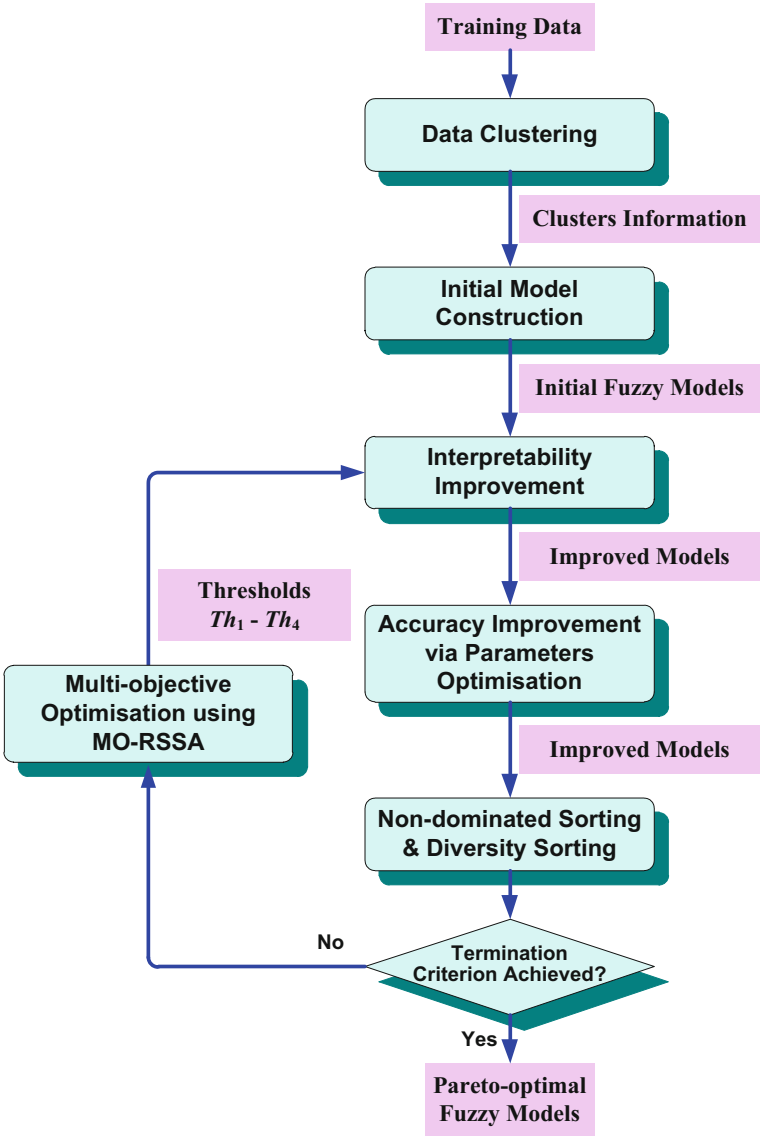


Fig. 9 The flow diagram of the proposed hierarchical multi-objective fuzzy modelling

Table 8 shows the parameters of the obtained FRBSs, which are with 8, 5 and 3 rules, respectively, and are selected from the Pareto-optimal models. It is observed that, for such improved systems, more fuzzy rules and more parameters will result in better accuracy. On the other hand, the systems with fewer fuzzy rules and fewer parameters are easier to understand and simpler in structure.

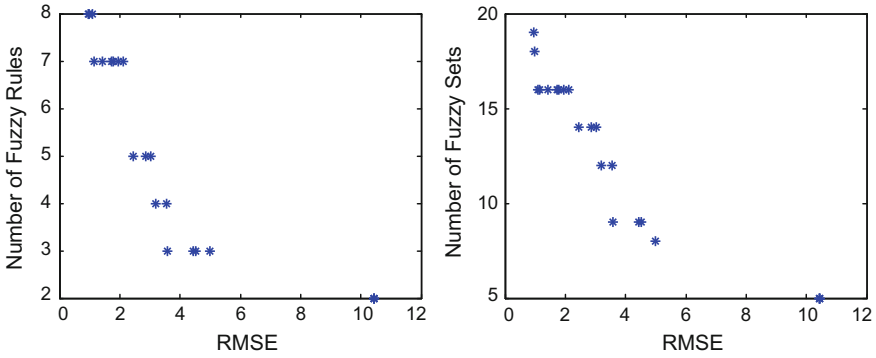


Fig. 10 Performance of the Pareto-optimal FRBS models for TPT

Table 8 Main parameters of three TPT models

FRBS model	Fuzzy sets number	Rule length	Training RMSE	Testing RMSE
TPT of 8 rules	Input variables: [6, 6] Output variable: 6	[3, 3, 3, 2, 2, 3, 3, 3]	0.978	2.354
TPT of 5 rules	Input variables: [5, 5] Output variable: 4	[3, 3, 3, 3, 3]	2.452	1.781
TPT of 3 rules	Input variables: [3, 3] Output variable: 3	[3, 3, 3]	3.576	2.800

To verify the physical interpretation, Fig. 11 gives 3D response surfaces of two selected FRBSs. From this figure, it is observed that the system containing more fuzzy rules captures more details from the data samples. It is also seen that, with the increase in the forward feed rate, the temperature of the tool will go down. Such behaviour is consistent with the finding of the literatures [2, 31, 32] and follows the expected behaviour from the field experts.

5.3 Modelling of Mechanical Properties

The proposed modelling method was then used in building the predictive models for some mechanical properties. As an example, Fig. 12 demonstrates the trade-offs between the multiple criteria for Yield Strength (YS) models and Ultimate Tensile

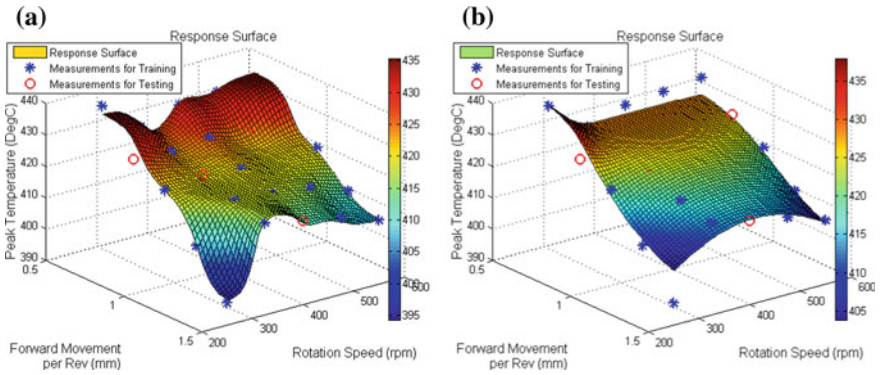


Fig. 11 3D response surfaces of TPT models: **a** the 8-rule FRBS and **b** the 3-rule FRBS

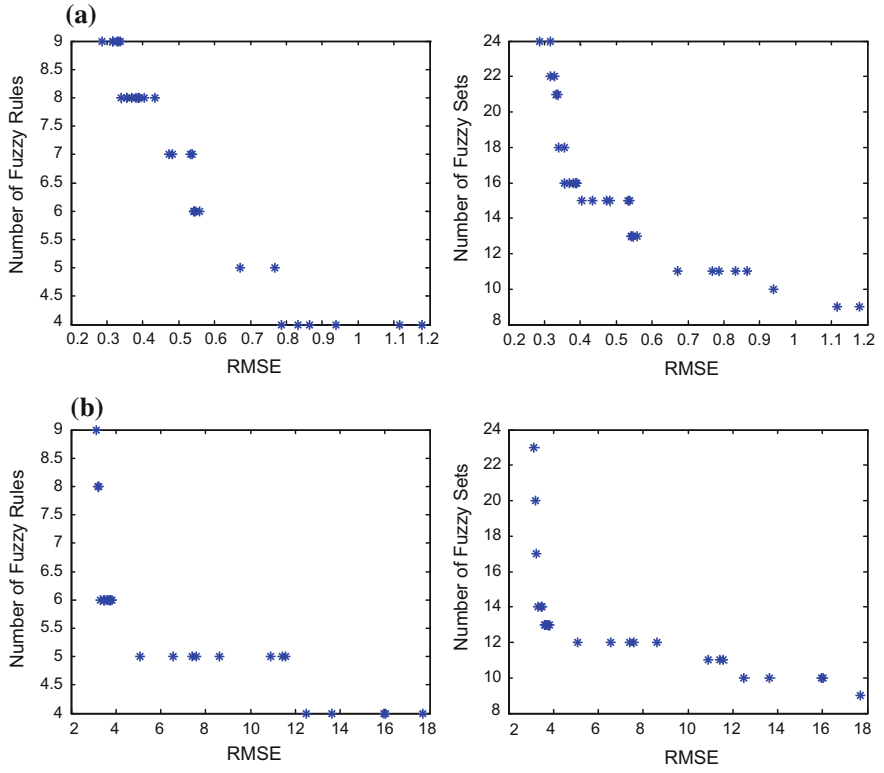


Fig. 12 Performance of the Pareto-optimal FRBS models: **a** YS and **b** UTS

Strength (UTS) models. Figure 13 shows the 3D input–output surfaces of these generated models.

From these surfaces, we can observe that, with the increase in welding feed rate, YS has a trend to increase; while with the increase in tool rotational speed, UTS has a trend to go down. Such observations show consistency with the findings in the literatures [33]. It is also worth noting that the FRBSs demonstrate excellent generalisation capability evidenced by very smooth response surfaces.

5.4 Modelling of Grain Size

Figure 14 demonstrates the trade-offs in multiple criteria among the non-dominated models. Table 9 gives the detailed parameters of three optimised FRBSs that are with 4, 7 and 9 rules, respectively. It is easily observed that more parameters and more fuzzy rules will bring a better accuracy; however, they are more complex in structure and more difficult to interpret.

Figure 15 illustrates the rule base of the 4-rule FRBS. The rule base can also be represented using the approximate linguistic rules [6, 30] with linguistic hedges [34] as follows:

R_1: IF *RS* is *medium small* AND *FM* is *medium large*, THEN *AGS* is *medium small*.
R_2: IF *RS* is *more or less medium* AND *FM* is *medium small*, THEN *AGS* is *more or less medium*.
R_3: IF *RS* is *medium large* AND *FM* is *medium*, THEN *AGS* is *large*.
R_4: IF *RS* is *large* AND *FM* is *large*, THEN *AGS* is *medium small*.

By interpreting such linguistic fuzzy rules, people are enabled to understand more about the process and its behaviours.

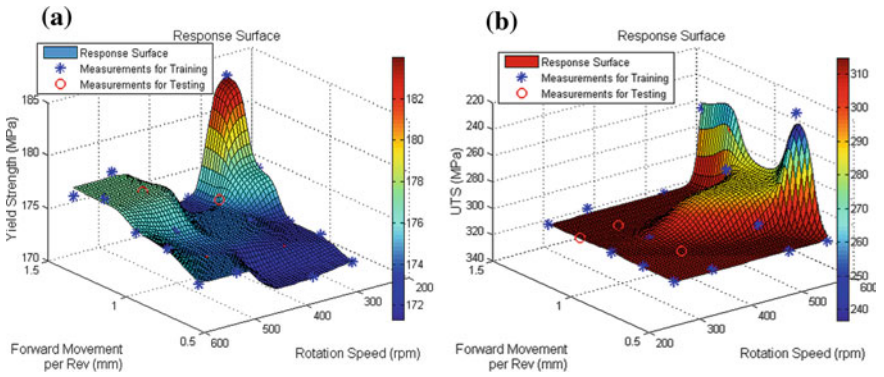


Fig. 13 3D response surfaces of the mechanical properties models: **a** the 9-rule FRBS of YS **b** the 5-rule FRBS of UTS

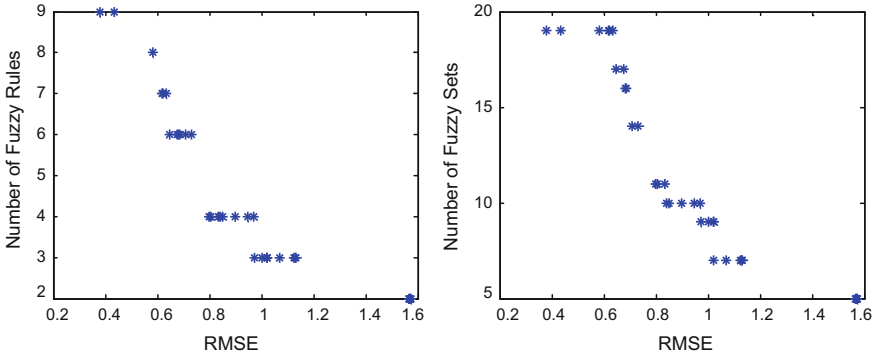


Fig. 14 Performance of the Pareto-optimal FRBS models for AGS

Table 9 Main parameters of three AGS models

FRBS model	Fuzzy sets number	Rule length	Training RMSE	Testing RMSE
AGS of 9 rules	Input variables: [7, 7] Output variable: 5	[3, 3, 3, 3, 3, 3, 3, 3, 3]	0.435	0.581
AGS of 7 rules	Input variables: [6, 6] Output variable: 7	[3, 3, 3, 3, 3, 3, 3]	0.633	0.625
AGS of 4 rules	Input variables: [4, 4] Output variable: 3	[3, 3, 3, 3]	0.799	0.747

6 Multi-objective Optimal Design of Welding Conditions

Up to five conflicting optimisation objectives are considered in this research. The objectives consist of in-process attributes, weld quality, mechanical properties and cost of welding.

6.1 Cost of Welding

The cost for friction stir welding one workpiece consists of four parts of costs as follows:

$$C_U = C_M + C_L + C_E + C_T \tag{4}$$

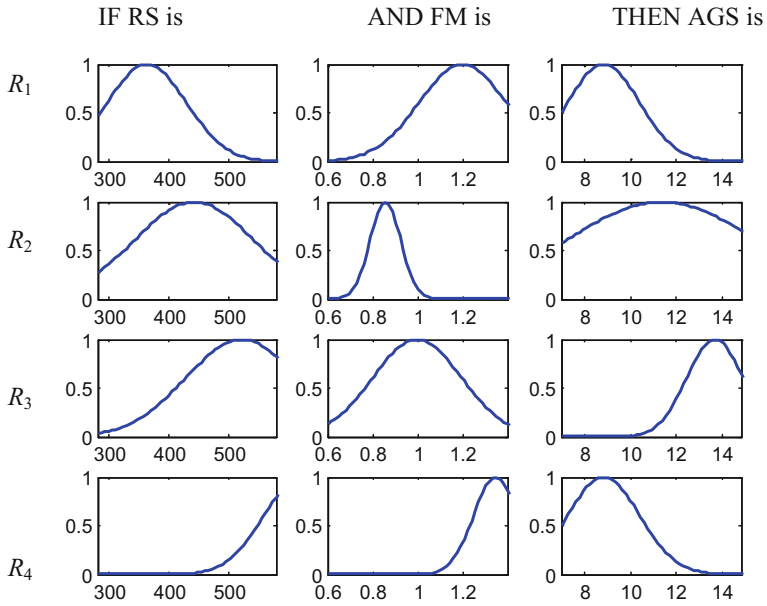


Fig. 15 The rule-base of the 4-rule FRBS of AGS

where C_U (£) is the overall cost for each piece (overall unit cost). C_T (£), C_E (£) and C_L (£) represent the unit tool wear, energy and labour costs, respectively. C_M (£) is the unit material cost.

The tool wear cost is written as the following equation:

$$C_T = K_T \frac{t_w}{T} = K_T \frac{L}{v_w T} \tag{5}$$

where K_T (£) is the tool value; t_w (min) is the unit welding time; T (min) represents the tool life; v_w (mm/min) is the welding speed; L (mm) is the workpiece length. Applying the Taylor equation of tool life [35], we can get

$$\pi D v_r T^n = K \tag{6}$$

where v_r (rpm) is the tool rotation speed; D (mm) represents the pin diameter of tool; n and K are the constants for a given FSW tool.

The unit energy (electricity) cost is expressed using the following equation:

$$C_E = K_E P_w t_w = K_E P_w \frac{L}{v_w} \tag{7}$$

Table 10 Parameters relating to welding cost

Parameters	Values
C_M	10.9 £
K_L	0.5 £/min
L	1000 mm
K_E	0.095 £/kWh
P_w	10 kW
K_T	2000 £
D	10 mm
n	0.2
K	100

where P_w (kW) is the power of the FSW machine; K_E (£/kWh) represents the electricity cost every kWh. Similarly, the unit labour cost is described using the following equation:

$$C_L = K_L t_w = K_L \frac{L}{v_w} \tag{8}$$

where K_L (£/min) represents the unit labour cost.

Above all, the overall unit cost can be further written as follows:

$$C_U = C_M + C_L + C_E + C_T = C_M + (K_L + K_E P_w)\frac{L}{v_w} + K_T \frac{L(\pi D v_r)^{1/n}}{v_w K^{1/n}} \tag{9}$$

C_M (£) is a fixed value across this work as we were using the same material in all experiments. The relevant parameters are provided in Table 10 and some are approximate values.

Figure 16 demonstrates the framework of the proposed multi-objective optimal design of operating conditions for the FSW process. For each designing experiment in Sects. 6.2–6.4, 10 optimisation runs were conducted. Only some solution sets with medium performance are discussed as examples. We found that the solutions obtained from different optimisation runs show very similar and consistent behaviours.

6.2 Two-Objective Optimal Design

We aim to maximise the weld quality and yield strength at the same time in the first experiment. The objectives are described as follows:

Objective 1: minimising WQ ($\boldsymbol{x}$)
Objective 2: maximising YS ($\boldsymbol{x}$)

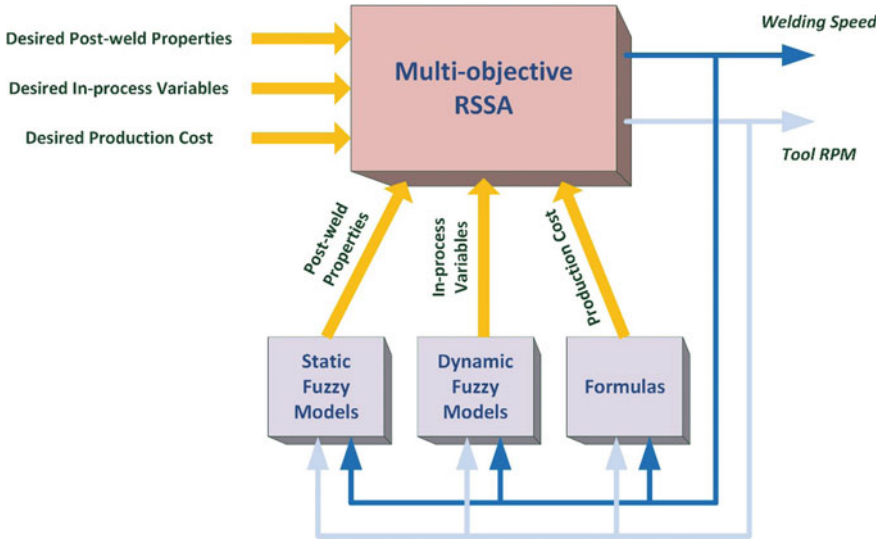


Fig. 16 The framework of FSW multi-objective optimal design

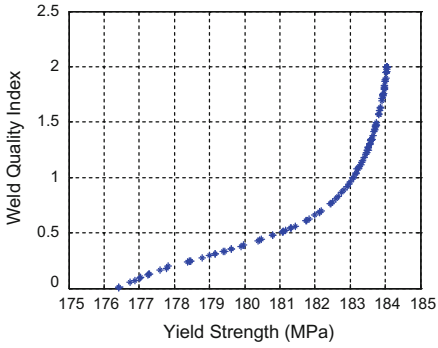


Fig. 17 Pareto-optimal solutions of the two-objective design problem

where $WQ(\boldsymbol{x})$ is the variable of the weld quality index and $YS(\boldsymbol{x})$ is the yield strength variable; $\boldsymbol{x}$ is the decision vector consisting of operating conditions, i.e. Forward Feed Rate (FFR) and Tool Rotation Speed (RS).

Figure 17 gives a set of Pareto solutions in a two-objective plot. To provide more information about the solutions, ten of them are chosen and shown in Table 11. They have relatively high forward feed rates and low tool rotation speeds. This observation follows the general principles of recrystallization [36], as a high forward feed speed and a low rotation speed produce a low heat input and thus lead to the generation of fine grains. This often causes high strength. However, a large welding speed may also cause the formation of void defects and thus worsen the WQ. In a real-life application, users could implement a tool rotational speed around 280 rpm and the feed rate close to 1.3 mm/rev, which will guarantee both good WQ and strong YS.

Table 11 Ten solutions of the two-objective design problem

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
FFR (mm/rev)	1.17	1.23	1.26	1.28	1.30	1.31	1.33	1.35	1.37	1.39
WS (mm/min)	329	345	353	359	364	368	373	378	385	389
RS (rpm)	280	280	280	280	280	280	280	280	280	280
WQ Index	0	0.115	0.216	0.334	0.448	0.566	0.785	1.085	1.555	1.860
YS (MPa)	176.4	177.2	178.1	179.4	180.6	181.5	182.5	183.2	183.8	184.0

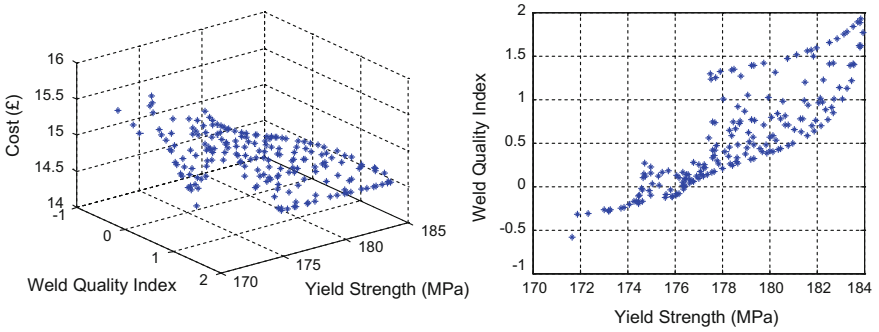


Fig. 18 Pareto-optimal solutions of the three-objective design problem

6.3 Three-Objective Optimal Design

In the second design problem, we considered the following objectives:

Objective 1: minimising $WQ\ (\boldsymbol{x})$
Objective 2: maximising $YS\ (\boldsymbol{x})$
Objective 3: minimising $Cost\ (\boldsymbol{x})$

where $Cost\ (\boldsymbol{x})$ is the cost variable C_U introduced in Sect. 6.1.

The Pareto-optimal solutions in both 2D and 3D objective spaces are shown in Fig. 18. Table 12 provides 10 solutions to them. From the figures, one could clearly see the trade-off of the solutions against different objectives. For instance, the solutions with good WQ (indicated by low-WQ index value) generally have low YS, while the solutions with high YS generally have bad WQ (high-WQ index value).

If a practitioner prefers to achieve the perfect WQ, he needs to implement the solutions with a relatively low FFR and a relatively fast RS. If a practitioner concerns more about cost or yield strength, he may use the solutions with a relatively higher FFR. By using a moderate solution (RS around 295 rpm and FFR around 1.34 mm/rev), people could achieve a low cost (≈£14.5) and a strong YS (>180 MPa), while maintaining a good WQ.

6.4 Five-Objective Optimal Design

We then considered the third design problem with the following five objectives:

Objective 1: minimising $WQ\ (\boldsymbol{x})$
Objective 2: maximising $YS\ (\boldsymbol{x})$
Objective 3: minimising $Cost\ (\boldsymbol{x})$
Objective 4: minimising $\sum_{i=1}^{p} \left(TT(\boldsymbol{x}, t_i) - TT_{target}\right)^2 / p$
Objective 5: minimising $\sum_{i=1}^{p} TF(\boldsymbol{x}, t_i)/p$

Table 12 Ten solutions of the three-objective design problem

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
FFR (mm/rev)	0.71	1.20	1.22	1.23	1.34	1.40	1.34	1.35	1.40	1.40
WS (mm/min)	362	427	388	345	400	428	396	377	414	396
RS (rpm)	509	356	318	280	298	305	295	280	296	283
WQ Index	0	0.149	0.100	0.117	0.793	1.347	0.813	1.002	1.567	1.928
YS (MPa)	173.0	174.7	176.2	177.2	180.0	178.5	180.7	183.1	181.7	183.9
Cost (£)	15.49	14.38	14.69	15.13	14.56	14.33	14.59	14.78	14.43	14.59

where $TF\ (\boldsymbol{x}, t_i)$ is the traverse force variable and $TT\ (\boldsymbol{x}, t_i)$ is the tool temperature variable; TT_{target} represents the target tool temperature ($TT_{target} = 380$ °C for this design problem); p represents the sample size; t_i represent the time points in welding.

Figure 19 demonstrates the Pareto solutions in 2D and 3D graphs and Table 13 provides 10 example solutions. From Fig. 19, people could observe intricate relationships between various factors. For instance, by increasing the welding speed, the mean tool temperature (TT) is generally going down because of less heat input. Meanwhile, the production cost can be either decreasing (when energy and labour costs playing the major role) or increasing (when tool wear cost playing the major role) depending on other factors.

It shows that MO-RSSA is able to generate well-distributed solutions near the desired objectives. They offer users with diverse design solutions. For a real-life application, the 7th solution in Table 13 shows a good compromise in different aspects. If we compare the solution to one with no optimal design (with the mean welding cost £16.4), a big save of 16.6% contributes to the overall welding cost. For an FSW machine operating on the full load, the obtained optimal process conditions would save tens of thousands pounds every year.

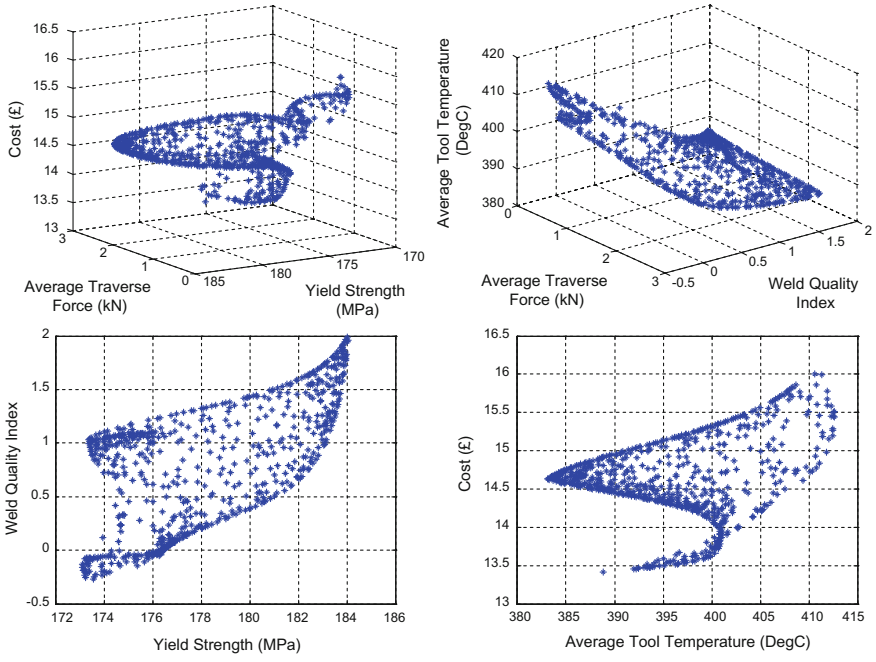


Fig. 19 Pareto-optimal solutions of the five-objective design problem

Table 13 Ten solutions of the five-objective design problem

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
FFR (mm/rev)	1.38	1.40	1.28	1.24	1.38	1.38	1.29	1.18	1.11	1.07
WS (mm/min)	389	414	360	354	437	482	556	350	316	304
RS (rpm)	283	297	281	286	317	350	430	296	284	286
WQ Index	1.564	1.472	0.311	0.140	0.974	0.939	0.902	−0.023	−0.073	−0.104
YS (MPa)	183.7	181.6	179.0	177.0	175.3	173.5	176.0	176.4	174.8	173.4
Cost (£)	14.65	14.44	14.96	15.02	14.27	13.98	13.67	15.08	15.53	15.71
Average TF (kN)	2.260	1.652	2.002	1.429	1.424	1.695	2.489	0.133	0.739	0.116
Average TT (°C)	385.0	391.3	391.1	397.1	399.0	401.0	396.6	405.9	406.7	409.2

7 Conclusions

In this chapter, a heuristic optimisation algorithm RSSA has been employed to improve the understanding and exploitation of FSW in the aspects of modelling, monitoring and process conditions design. First, RSSA helped the design of an accurate weld quality indicator used to supply an online indication of defects formation. Second, a data-driven fuzzy modelling approach was successfully proposed to build interpretable models for in-process and post-weld properties. In the generation of the models, RSSA and MO-RSSA have been employed to improve both the structure and parameters of FRBSs. Last, the developed models were exploited to design the optimal process parameters via a reverse-engineering fashion. MO-RSSA has been applied to solve a range of optimal design problems with two to five objectives, where both quality and cost issues have been successfully addressed.

References

1. Thomas, W. M., Nicholas, E. D., Needham, J. C., Murch, M. G., Templesmith, P., & Dawes, C. J. (1991). G.B. Patent Application No. 9125978.8, December 1991.
2. Nandan, R., DebRoy, T., & Bhadeshia, H. K. D. H. (2008). Recent advances in friction stir welding—process, weldment structure and properties. *Progress in Materials Science, 53,* 980–1023.
3. Chakraborti, N. (2014). Critical assessment 3: The unique contributions of multi-objective evolutionary and genetic algorithms in materials research. *Materials Science and Technology, 30*(11), 1259–1262.
4. Paszkowicz, W. (2013). Genetic algorithms, a nature-inspired tool: A survey of applications in materials science and related fields: Part II. *Materials and Manufacturing Processes, 28*(7), 708–725.
5. Datta, S., Zhang, Q., Sultana, N., & Mahfouf, M. (2013). Optimal design of titanium alloys for prosthetic applications using a multi-objective based genetic algorithm. *Materials and Manufacturing Processes, 28*(7), 741–745.
6. Zhang, Q., Mahfouf, M., Yates, J. R., Pinna, C., Panoutsos, G., Boumaiza, S., et al. (2011). Modeling and optimal design of machining-induced residual stresses in aluminium alloys using a fast hierarchical multiobjective optimization algorithm. *Materials and Manufacturing Processes, 26*(3), 508–520.
7. Zhang, Q., & Mahfouf, M. (2009). A modified PSO with a dynamically varying population and its application to the multi-objective optimal design of alloy steels. In *Proceedings of the 2009 IEEE Cong. on 'Evolutionary Computation'*, Trondheim, Norway (pp. 3241–3248). IEEE.
8. Tansel, I. N., Demetgul, M., Okuyucu, H., & Yapici, A. (2010). Optimizations of friction stir welding of aluminum alloy by using genetically optimized neural network. *International Journal of Advanced Manufacturing Technology, 48,* 95–101.
9. Roshan, S. B., Jooibari, M. B., Teimouri, R., Asgharzadeh-Ahmadi, G., Falahati-Naghibi, M., & Sohrabpoor, H. (2013). Optimization of friction stir welding process of AA7075 aluminum alloy to achieve desirable mechanical properties using ANFIS models and simulated annealing algorithm. *International Journal of Advanced Manufacturing Technology, 69,* 1803–1818.
10. Parida, B., & Pal, S. (2015). Fuzzy assisted grey Taguchi approach for optimisation of multiple weld quality properties in friction stir welding process. *Science and Technology of Welding and Joining, 20*(1), 35–41.
11. Tutum, C. C., & Hattel, J. H. (2010). Optimization of process parameters in friction stir welding based on residual stress analysis: A feasibility study. *Science and Technology of Welding and Joining, 15*(5), 369–377.

12. Tutum, C. C., Deb, K., & Hattel, J. H. (2013). Multi-criteria optimization in friction stir welding using a thermal model with prescribed material flow. *Materials and Manufacturing Processes, 28*(7), 816–822.
13. Shojaeefard, M. H., Behnagh, R. A., Akbari, M., Givi, M. K. B., & Farhani, F. (2013). Modelling and Pareto optimization of mechanical properties of friction stir welded AA7075/AA5083 butt joints using neural network and particle swarm algorithm. *Materials and Design, 44,* 190–198.
14. Zhang, Q., & Mahfouf, M. (2010). A nature-inspired multi-objective optimisation strategy based on a new reduced space searching algorithm for the design of alloy steels. *Engineering Applications of Artificial Intelligence, 23*(5), 660–675.
15. Zhang, Q., & Mahfouf, M. (2007). A new reduced space searching algorithm (RSSA) and its application in optimal design of alloy steels. In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore (pp. 1815–1822).
16. Zhang, Q., Mahfouf, M., Panoutsos, G., Beamish, K., & Norris, I. (2012a). Knowledge discovery for friction stir welding via data driven approaches: Part 1—correlation analyses of in-process variables and weld quality. *Science and Technology of Welding and Joining*, *17*(8), 672–680.
17. Zhang, Q., Mahfouf, M., Panoutsos, G., Beamish, K., & Norris, I. (2012b). Knowledge discovery for friction stir welding via data driven approaches part 2—multiobjective modelling using fuzzy rule based systems. *Science and Technology of Welding and Joining*, *17*(8), 681–693.
18. Zhang, Q., Mahfouf, M., Panoutsos, G., Beamish, K., & Liu, X. (2015). Multiobjective optimal design of friction stir welding considering quality and cost issues. *Science and Technology of Welding and Joining, 20*(7), 607–615.
19. El-Danaf, E. A., El-Rayes, M. M., & Soliman, M. S. (2010). Friction stir processing: An effective technique to refine grain structure and enhance ductility. *Materials and Design, 31,* 1231–1236.
20. Thomas, W. M., & Gittos, M. F. (1999). Development of friction stir tools for the welding of thick (25 mm) aluminium alloys. TWI Members Report 694/1999, TWI, UK.
21. Beamish, K. A., & Russell, M. J. (2010). Relationship between the features on an FSW tool and weld microstructure. In *Proceedings of the 8th International Symposium on 'Friction Stir Welding'*, Timmendorfer Strand, Germany.
22. Mishra, R. S., & Ma, Z. Y. (2005). Friction stir welding and processing. *Materials Science and Engineering: R: Reports, 50*(1–2), 1–78.
23. Jin, Y., Olhofer, M., & Sendhoff, B. (2001). Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? In *Proceedings of the Genetic and Evolutionary Computation Conf*erence (pp. 1042–1049). ACM: San Francisco, USA.
24. Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & Industrial Engineering, 30*(4), 957–968.
25. Gaafer, A. M., Mahmoud, T. S., & Mansour, E. H. (2010). Microstructural and mechanical characteristics of AA7020-O Al plates joined by friction stir welding. *Materials Science and Engineering A, 527,* 7424–7429.
26. Boldsaikhana, E., Corwinb, E. M., Logarb, A. M., & Arbegast, W. J. (2011). The use of neural network and discrete Fourier transform for real-time evaluation of friction stir welding. *Applied Soft Computing, 11,* 4839–4846.
27. Lee, B., & Tarng, Y. S. (1999). Application of the discrete wavelet transform to the monitoring of tool failure in end milling using the spindle motor current. *International Journal of Advanced Manufacturing Technology, 15*(4), 238–243.
28. Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics, 3,* 28–44.
29. Zhang, Q., & Mahfouf, M. (2008). Mamdani-type fuzzy modelling via hierarchical clustering and multi-objective particle swarm optimisation (FM-HCPSO). *International Journal of Computational Intelligence Research, 4*(4), 314–328.
30. Zhang, Q., & Mahfouf, M. (2011). A hierarchical Mamdani-type fuzzy modelling approach with new training data selection and multi-objective optimisation mechanisms: A special application for the prediction of mechanical properties of alloy steels. *Applied Soft Computing, 11*(2), 2419–2443.

31. Hashimoto, T., Jyogan, S., Nakata, K., Kim, Y. G., & Ushio, M. (1999). FSW joints of high strength aluminum alloy. In *Proceedings of the First International Symposium on Friction Stir Welding*, Thousand Oaks, CA.
32. Ma, Z. Y., Mishra, R. S., & Mahoney, M. W. (2002). Superplastic deformation behaviour of friction stir processed 7075Al alloy. *Acta Materialia, 50*(17), 4419–4430.
33. Biallas, G., Braun, R., Donne, C. D., Staniek, G., & Kaysser, W. A. (1999). Mechanical properties and corrosion behavior of friction stir welds. In *Proceedings of the First International Symposium on Friction Stir Welding*, Thousand Oaks, CA.
34. Zadeh, L. A. (1972). A fuzzy-set-theoretic interpretation of linguistic hedges. *Journal of Cybernetics, 2,* 4–34.
35. Chu, T. C., Ranson, W. F., & Sutton, M. A. (1985). Applications of digital-image-correlation techniques to experimental mechanics. *Experimental Mechanics, 25*(3), 232–244.
36. Humphreys, F. J., & Hotherly, M. (1995). Recrystallization and related annealing phenomena. New York, USA: Pergamon Press.

Comparative Study of Multi/Many-Objective Evolutionary Algorithms on Hot Rolling Application

Prateek Mittal, Affan Malik, Itishree Mohanty and Kishalay Mitra

Abstract Handling multiple number of objectives in industrial optimization problems is a regular affair. The journey of development of evolutionary algorithms for handling such problems occurred in two phases. In the first phase, multi-objective optimization algorithms are developed that worked quite satisfactorily while finding Pareto set of solutions for two to three objectives. However, their success rates for finding the Pareto optimal solutions for higher number of objectives were limited which triggered the development of different sets of evolutionary algorithms under the name of many-objective optimization algorithms. In this work, we intend to compare the performance of these two classes of algorithms for an industrial hot rolling operation from a real-life steel plant. Several process, chemistry and geometry related parameters are modelled to yield different mechanical properties such as % elongation, ultimate tensile strength and yield strength of final hot rolled steel product through data-based techniques such as artificial neural networks (ANN) . Using this ANN model, the mechanical properties are maximized to obtain the Pareto trade-off solutions using both non-dominated sorting genetic algorithms II (NSGA-II) and many-objective evolutionary algorithm decomposition and dominance (MOEA/DD) and their solutions are compared using a suitable metric for identifying the extent of convergence and diversity. This kind of Pareto set provides a designer with ample of alternatives before choosing a solution for final implementation.

Keywords Many-objective optimization · NSGA-II · MOEA-DD · Hot rolling
Chemistry parameter · Geometry parameter · Elongation
Ultimate tensile strength · Yield strength · ANN · IGD

P. Mittal · A. Malik · K. Mitra (✉)
Department of Chemical Engineering, Indian Institute of Technology Hyderabad, Kandi, Sangareddy 502285, Telangana, India
e-mail: kishalay@iith.ac.in

I. Mohanty
Research & Development, Tata Steel Limited, Jamshedpur 831001, Jharkhand, India

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8_12

1 Introduction

Most of the real-world applications demand simultaneous satisfaction of multiple numbers of conflicting objectives to influence the performance of a system. This leads to design, development and implementation of efficient multi-objective optimization problems (MOP). Generally, a MOP is defined as [1]

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), \ldots, f_m(x)) \\ &\text{subject to } g_p(x) \leq 0, p = 1, \ldots, P \\ &\qquad\qquad h_q(x) = 0, \quad q = 1, \ldots, Q \\ &\qquad\qquad x \in \Omega \end{aligned} \tag{1}$$

where P and Q are the numbers of inequality and equality constraints, respectively; x is the decision vector in Ω, (refers to feasible search space), R^m refers to the m-dimensional real-valued objective space and $F: \Omega \rightarrow R^m$ consists of m real-valued objective functions. Since objectives in MOP are of conflicting nature, there is no single optimal solution possible to find that optimizes all the objectives at the same time. Instead, a set of trade-off solutions is sought, known as the Pareto optimal (PO) set, which can help the designer to understand the conflicting nature of the objectives under consideration and choose a particular solution out of the alternatives present [2]. Multi-objective evolutionary algorithms (MOEA) have shown their effectiveness in determining the set of converged and well-diversified solutions for two/three objective optimization problems since 1990s. However, due to continuous endeavour for having the competitive advantage over the peers, the enterprise system demands many numbers of objectives to be satisfied simultaneously, leading to the formation of many-objective optimization problems (MaOP) involving four or more number of objectives. Recent studies [3, 4] have shown that the performance of the algorithms specifically designed for solving MOEAs is deteriorated while tackling more number of objectives, i.e. MaOPs. In this work, a comparative study on two state-of-the-art evolutionary algorithms, one from the domain of MOP (Non-dominated Sorting Genetic Algorithm II) while the other from the domain of MaOP (many-objective evolutionary algorithm/Dominance and decomposition), has been performed while solving an industrial multi-objective optimization problem from the steel industrial sector. The enhanced performance of the latter over the former is reported. Also, as a result, multiple numbers of solutions are obtained on a single Pareto front, where each solution corresponds to a set of optimized parameters and the alternative objective trade-off values. Such a set of solutions provides the decision-maker with ample of alternatives to think through before implementing one solution on the shop floor.

2 Literature Review

Multi-objective optimization problems (MOPs) are commonly seen in real-world applications, especially in the areas of engineering, biology and economics [5–8]. The characteristic feature of such optimization problems is the presence of conflicts among the multiple objectives. Due to the presence of trade-off among objectives, no single solution can be found as optimal under this conflicting scenario that can satisfy all the objectives at the same time leading to the generation of PO set. Generally, two classes of approaches are prevalent to obtain the solutions in the PO set, i.e. the first class belongs to population-based nature inspired evolutionary algorithms (EAs) and the second class evolves around the gradient-based classical techniques. However, the aim in both the cases is to find solutions close to PO set which can be measured in terms of convergence, closer the solutions to the PO set is a sign of better convergence and the solutions should be as diverse as possible as that leads to more number of alternatives to a designer. Over the last few decades, evolutionary algorithms have shown their supremacy in achieving both these targets (i.e. convergence and diversity) as compared to their classical counterparts [9]. Since evolutionary algorithms work with population of points, they are designed in such a fashion exploiting the concept of non-dominance, the entire PO set was generated using single simulation run whereas their classical counterparts use a multiple number of simulations to generate the entire PO set. Even with respect to the aspects of convergence and diversity [4], EAs are generally designed to achieve a balance between the two whereas in the classical route, there is no separate treatment for these aspects. In order to meet the above requirements of convergence and diversity, a large number of multi-objective evolutionary algorithms (MOEAs) , such as non-dominated sorting genetic algorithm II (NSGA-II) [10], strength pareto evolutionary Algorithm 2 (SPEA 2) [11], Pareto archived evolutionary algorithm (PAES) [12], etc. are developed. These algorithms utilize the Pareto-based dominance relationship, i.e. the solutions in a population are ranked into several fronts on the basis of dominance and the solutions with better rank of non-dominance will be preferred [10] to solve the multi-objective optimization problems. This concept worked well while solving for two or three objectives, but it reported inferior performance while solving MaOPs having more than three objectives. Since many real-world applications are of many-objective in nature [13, 14], efforts have been applied to develop efficient algorithms to tackle the problems involving more numbers of objectives. Generally, MOEAs are observed to solve these problems unsatisfactorily. There are certain issues that can create a differentiator between these two sets of algorithms (MOP and MaOP). With the increase in the number of objective functions, the chance of every candidate solution becoming non-dominated with respect to other solutions becomes very high at a very early stage and this might lead to a case of stalemate because the algorithm faces difficulty to differentiate between these solutions [9]. As most of the parameters responsible for maintaining diversity prefer dominance resistant solutions, they are not able to influence and fasten the selection process towards the final Pareto front hence, the conflict between convergence and diversity becomes more aggravated [15]. More-

over, with the increase in dimensions, it might be difficult to generate offspring solutions that are very far away from parents and thereby create less diversity. So far, several MOEAs have been modified to handle the MaOPs [3]. These algorithms are classified into three types: (1) the first one is conceptualized based on modified Pareto dominance operator to influence better ranking of solutions, e.g. shift density-based evolutionary algorithm (SDE) [16], grid-based evolutionary algorithm (GrEA) [17], etc. (2) the second one is based on better kind of performance indicator indicating the convergence and diversity of a non-dominating front, e.g. approximated hypervolume based evolutionary algorithm (HyPE) [18], etc., and (3) the third one is based on decomposition of objective space, such as non-dominated sorting genetic algorithm III, many-objective optimization based on decomposition (MOEA/D) [19] and many-objective optimization based on decomposition and dominance (MOEA/DD) [1]. Moreover, several other promising approaches such as bi-goal evolutionary algorithm, knee point driven evolutionary algorithm (KnEA), etc. are also developed to solve many-objective optimization problems [20]. Among all, the decomposition-based approaches are reported as the most effective approach for solving MaOPs so far [3]. In decomposition-based approaches, many-objective optimization problem is decomposed into a number of scalar optimization sub-problems represented by weight vectors and solved in a collaborative manner. Due to the diverse nature of weight vectors, the well-spread solutions are obtained in this method. Using the decomposition-based method, MOEA/D [19] algorithm has been developed where each solution is associated with a sub-problem, and each sub-problem is optimized using the information from its neighbourhood. Recently, in order to obtain faster divergence and well diverse solutions, NSGA-III [21] has been proposed, where the decomposition-based idea is employed to have diverse solutions and convergence is controlled by Pareto dominance. Moreover, in order to obtain well-spread solutions, MOEA/DD [1] algorithm is developed by utilizing the merits of both Pareto dominance and decomposition-based approaches. In this method, the efforts are applied to associate each solution with each weight vector (this shows diversity) and using Pareto dominance criteria, the solutions are updated simultaneously.

In this work, the effectiveness of both multi-objective and many-objective evolutionary algorithm has been studied. In case of MOEA, the state-of-the-art NSGA-II is chosen whereas in case of MaOEA, another state-of-the-art and recently developed many-objective evolutionary algorithm/Dominance and decomposition (MOEA/DD) has been selected. The working principle of each of these algorithms is provided in the next section. Both algorithms are applied on an industrial case study of hot rolled steel plant. Here, the aim is to determine the chemistry and processing parameters of the hot rolling process in order to achieve the desired mechanical properties, i.e. maximization of yield strength (YS), ultimate tensile strength (UTS) and % elongation (El). Moreover, the performance of both algorithms in terms of convergence and diversity has been compared using inverted generational distance (IGD) metric. The chapter is organized as follows. Section 3 talks about the evolutionary algorithms (NSGA-II and MOEA/DD) and Sect. 4 describes the problem formulation. The results and discussion are presented in Sect. 5.

3 Methods

In this section, the working principle of two multi-objective algorithms, i.e. NSGA-II (as MOEA) and MOEA/DD (as MaOEA) is described in details.

3.1 Non-dominated Sorting Genetic Algorithm (NSGA-II)

A well-established MOEA, read coded NSGA-II, is used here as one of the algorithms. The flow of the algorithm goes like this. First, the parent population N is initialized by creating real variables randomly within their respective bounds (*step: initialization*). After computing objective function and constraint values for these solutions (*step: fitness evaluation*), the population is undergone a selection process where better candidates are selected and N new offsprings are generated from them using the genetic operators such as crossover and mutation (*step: selection, crossover and mutation*). Next, the N parent and N offspring solutions are combined to sort them into different fronts using the principles of non-domination. In this way, the entire 2 N population is classified into different levels of non-dominated solutions (*step: Ellitism*), where these levels of non-domination and a distance metric called crowding distance are used as the metrics of the selection process in the next generation. During the selection process, the solutions with better non-dominating fronts are preferred to accelerate the convergence and within the same front, solutions with higher crowding distance, signifying less crowded solutions, are preferred to maintain diversity in the population. From the combined set of 2 N population, the number of candidates is judiciously reduced to N during the selection process in every generation. During this process, solutions from the better non-dominated fronts are filled up first till a front appears which cannot be filled up in entirety, where the solutions are filled up sorted in the preference of higher to lower crowding distance. The combined action of crossover and mutation to generate new solutions and the action of non-dominated sorting and selection to select the best possible set of non-dominated solutions guide the solutions to merge with the set of Pareto optimal solutions. Figure 1 shows the complete flowchart of the algorithm.

For the sake of brevity, the description of classical NSGA-II is restricted here and readers can refer [10] for more details.

3.2 Multi-objective Evolutionary Algorithm/Dominance and Decomposition (MOEA/DD)

Next, a very recently developed many-objective evolutionary algorithm which utilizes the concepts of both dominance and decomposition has been described. In this algorithm, a set of weight vectors is uniformly placed in the given objective space

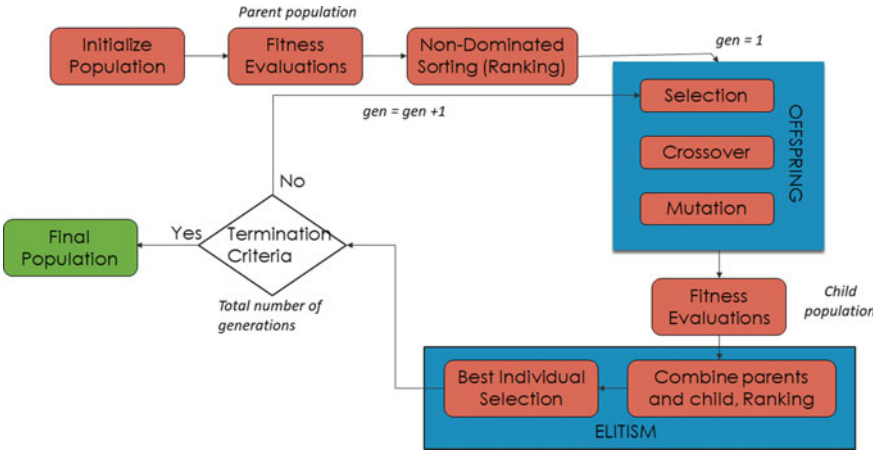


Fig. 1 Flowchart of elitist NSGA-II algorithm

while decomposing the mentioned space into several subregions. If different solutions can be associated with such different weight vectors during the progress of the algorithm, the idea of maintaining the diversity in the final PO front can be achieved. The responsibility of generation of new solutions is still based on the genetic operators and selection of solutions is still based on the concepts of non-dominated sorting [10]. The algorithm works like this. First, the *initialization* procedure generates N initial solutions and N weight vectors. These initial solutions are created randomly in the given decision variable space honouring the bounds on the decision variables and weight vectors are created in a uniform manner in the objective space. Afterwards, for each generation and each weight vector, in the *mating and reproduction step,* some parents are selected for offspring generation. Then, accordingly the population is updated with better solutions by removing the worst solution one by one (*step update population*) [1]. The schematic flowchart of the MOEA/DD algorithm is shown in Fig. 2 and the detailed description of each step implemented in this algorithm is described below:

Step1 (Initialization): In this step, N number of candidate solutions (or population (P)) are initialized with random sampling of decision variable set (x_{gcp}) while satisfying their respective bounds. Using Das and Dennis approach [22], the uniform set of weight vectors (also called reference vectors) $\mathbf{V_{Wt}} = \{V^1_{Wt}, \ldots, V^N_{Wt}\}$ is created to decompose and thereby represent the whole objective space. In this approach, weight vectors are uniformly spaced for any number of objectives with sub-spacing parameter $\Delta = 1/D$, where $D > 0$ is the number of divisions along each objective coordinate. Figure 3a, illustrates the simple example of weight generation method with $D = 3$ for three objectives and their representations in the three-dimensional objective space are shown in Fig. 3b. Surrounding each weight vector V^i_{Wt}, $i = 1, \ldots, N$, N_{Wt} neighbouring weight vectors are computed such that $N_{Wt} \ll N$. Next, each member of randomly generated population P (i.e. population point) is associated

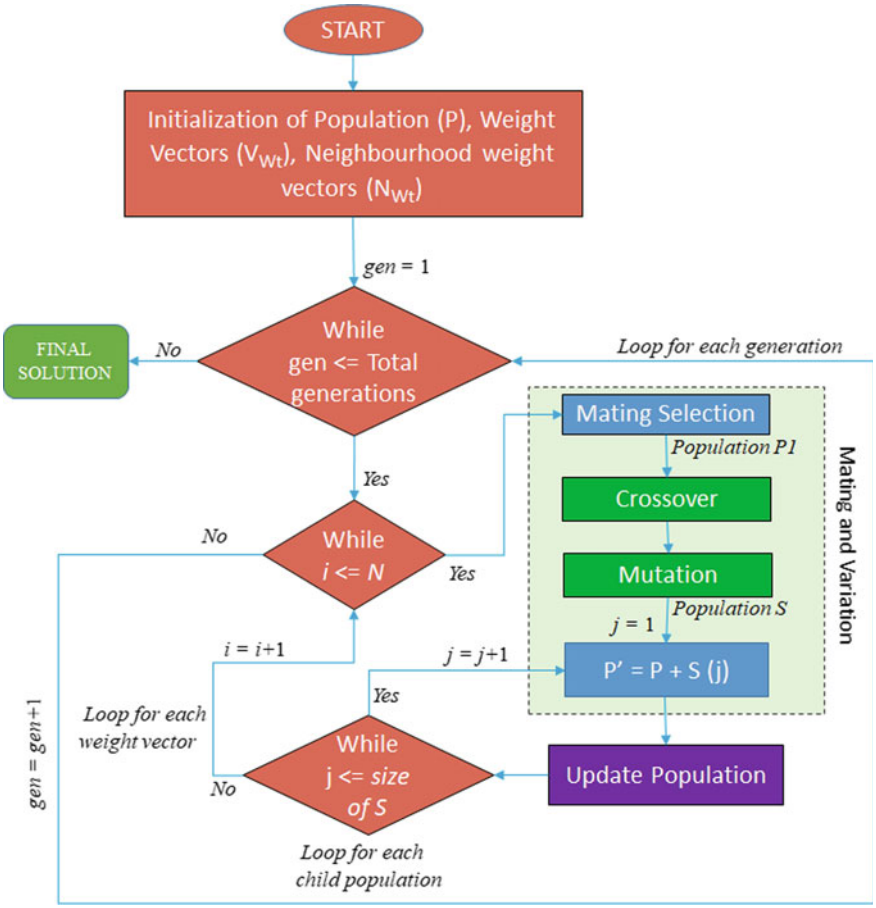


Fig. 2 Schematic flow chart of MOEA/DD algorithm

with one of the generated reference vectors (Fig. 3c). Due to the random nature of solution generation during the initialization process, there are chances that more than one initialized solution can be associated with a weight vector (for, e.g. dashed circle in Fig. 3c) and some weight vectors may not have any solution in their premises. Next, different non-dominated levels (F_i) where, i=1, …, nd, ($nd \leq N$) are created for the entire population P, using the non-dominated sorting technique [10].

Step I1 (Mating selection and Reproduction): After initializing the population and associating the solutions to one of the reference vectors, the offspring population (S) is generated using genetic operators (crossover and mutation) and the parent population is updated. Basically, this step is executed in two parts. In the first part, i.e. mating selection, some parent populations (P1) are selected for offspring generation. Here, the approach is to proceed by each of the reference vectors one at a time. First, a weight vector is selected and for the selected weight vector 'i', P1 solutions

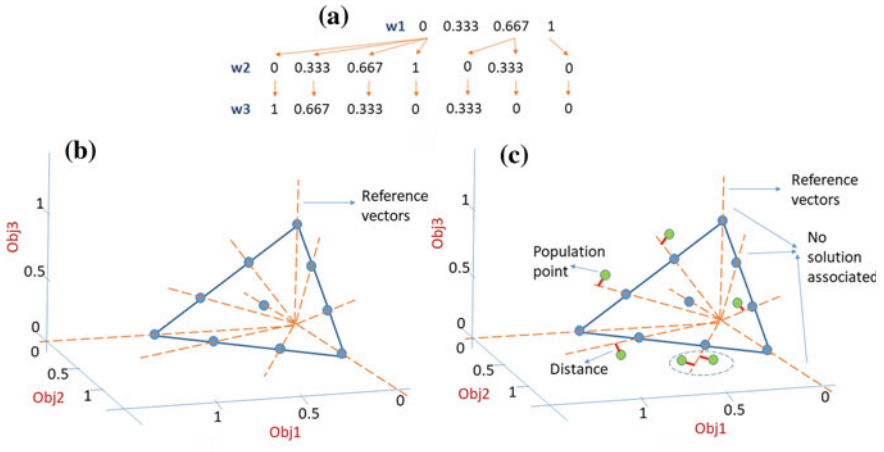


Fig. 3 **a** Das and Dennis weight generation approach for 3 objectives and D = 3, Δ = 0.333; **b** representation of reference vectors in three-dimensional objective space; **c** association of population points with reference vectors (dashed circle shows the association of 2 points with one reference vector). (*Obj1, Obj2 and Obj3 are the normalized objective function values*)

are selected either from its neighbourhood weight vectors. In case the neighbouring weight vectors cannot provide with P1 parents to participate in the mating, the balance candidates are selected randomly from the entire parent population. After selecting the P1 solutions, the variation operators such as crossover and mutation are utilized to create the offspring (or child) population (S). Here, simulated binary crossover (SBX) crossover [23] and polynomial mutation [24] techniques are applied to generate the child population.

Step III (Update Population): The generated offspring population S is further utilized to update the parent population by finding the worst solution from the parent population and replacing it by a better solution from S one at a time. This is implemented in this way. Each offspring solution '*j*' is added to the Parent population of size N to create a new population (P′) of size $N+1$ as shown in Fig. 2. Next, P′ is passed through the update population procedure where the worst solution is removed from it, and this process is repeated for each offspring solution (one by one) to update the entire parent population P. Figure 4 shows the flowchart of the update population step. First, the selected offspring solution S(j) is assigned with the most suitable reference vector based on the minimum acute angle between the reference vector and the solution (*step: associate child*). Next, an updated non-dominated levels (F_k^1), where, $k=1, \ldots, nd1$, ($nd1 \leq N+1$) are generated for P′ using effective non-dominated sorting method [25] (*step: Effective NDS before*). In this step, instead of creating the non-dominated fronts for all solutions of P′ (as in Step I), the solution S(j) is compared with the solutions inside each generated front of the parent population 'P' (called frontier solutions) to update the fronts. Two options appear here: (1) either S(j) is added inside a front which is non-dominating in nature with it, or (2) a new

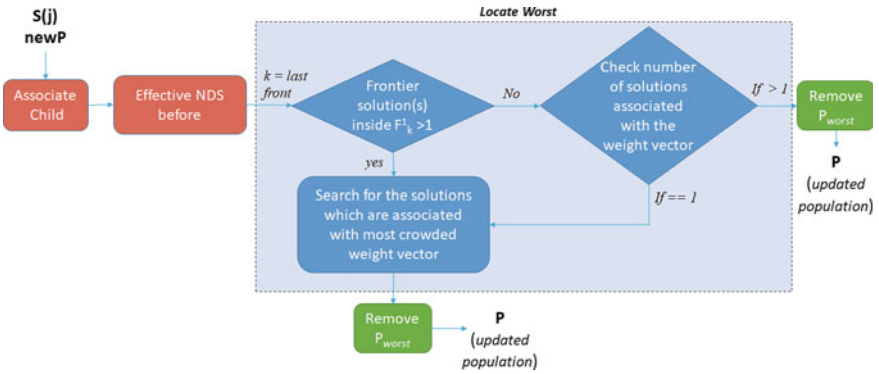


Fig. 4 Schematic flow chart of Step III: update population step

front is created with the non-dominated solutions of a particular front if part of solutions in that front are dominating nature and rest of the part is non-dominating in nature with respect to S(j). After representing the solutions in terms of an updated non-dominated fronts (F^1), the worst solution (P_{worst}) is removed (*step: locate worst*) from P′ to maintain the final population size as N. Generally, the last non-dominated front consists of many P_{worst} solution(s) as compared to the previous fronts; so, the *locate worst* step starts the removal process from the last front (F^1_{nd1}) towards the first front (F^1_1). In this step, it is assured that every reference vector should have at least one solution associated with it to maintain the diversity of the final solution. This means, if the selected worst solution (to be removed) is the only solution associated with a particular weight vector, it is not removed to maintain diversity; instead, another solution from the same front which is associated with the most crowded weight vector is selected for removal. In this way, each offspring solution is utilized to update the parent population by removing the worst solution from it. For the sake of brevity, the description of whole MOEA/DD is restricted here and readers can refer [1] for more details. By repeating the Step II and Step III for every weight vector and for every generation, the non-dominated set of solutions is obtained. This completes one generation in MOEA/DD. These steps are continued for every generation till the number of generations reaches the maximum number specified by a designer and final set of non-dominated solutions is reported as the final PO set.

4 Problem Formulation

Rolling of any material is done by passing the metal stock through one or more pairs of rolls to reduce the thickness and to make the thickness uniform. Rolling is classified according to the temperature of the metal rolled. If the temperature of the metal is above its recrystallization temperature, then the process is known as hot

rolling. The main objective of this rolling is to change the dimension of the slabs (generally 210 mm) to meet the dimension of the final product (2–12 mm). In order to achieve this after casting, the slabs are reheated to the desired temperature required for homogenization of chemistry and dissolution of precipitates formed during casting and soaked within a furnace to permit hot rolling to strip. After coming out from furnaces, heated slabs are rolled in roughing mill to break the cast structure of the slab through repeated recrystallization, and finish with a homogeneous austenite grain size. During the roughing, complete recrystallization is expected for all steel grades and the dimension generally reduces from 210 to 30 mm. Following roughing mill, the coil enters a coil box where the temperature of the coil will typically be around 1050 °C. The primary reason for the coil being in coil box is to 'hold' the coil as it is fed into the finishing mill which allows homogenization in temperature throughout the coil.

The finishing mill consists of 6–7 stands, each of which can be independently controlled in terms of roll gap, roll speed, work roll shifting and bending. During rolling, the coil will be in all the stands at the one time with peripheral roll and coil speed synchronized to maintain strip tension with the aid of inter-stand balanced rolls. In addition, inter-stand cooling can also be applied to aid rolling and also to cool the coil. The reduction pattern employed is designed to reduce the developed rolling loads in the mill, as the strain rate will substantially increase due to the increased roll speed. As the temperature of the coil decreases, the load in the mill increases. The finish rolling temperature (FRT) of the coil is around is 900 °C.

Apart from reduction in thickness during finish rolling, combined effect of temper and strain induce many metallurgical changes in the material. The most prominent of them is recrystallization. Proper choices of rolling parameters and temperatures are decided keeping in mind the chemical composition of the material which helps in maximizing the effect of recrystallization and obtaining the best combination of mechanical property.

After coming out from finishing mill, the gauge and width of the strip thickness are accurately measured and the strip is cooled on run out table by following a particular spray pattern from the top and bottom surface of the strip to achieve the target coiling temperature and the desired cooling rate. The cooling of the strip is probably one of the most important stages of rolling in hot strip mill, which require a precise control. Once the coiler captures the head end of the strip, tension is applied to the strip and the mill will accelerate to maintain the finishing temperature. Such variations in the strip velocity will result in different rates of cooling to be applied and inconsistent mechanical properties if the cooling rates are not controlled along the entire strip length.

The precise control of reheating temperature, FRT, CT and the cooling rate is the key behind getting the right combination of microstructure and precipitation which helps in achieving the target mechanical properties. The detailed understanding of evolution of the austenite microstructure during hot rolling plays an important role in converting carbon steel into sophisticated high strength low alloy (HSLA) steel. When steel is processed thermos-mechanically with micro-alloy such as Nb, it helps in pinning austenite grain boundaries and thus prevents recrystallization.

The tendency of grains to coarsen between rolling passes is prevented by allowing recrystallization of austenite at the first few passes and then by preventing recrystallization altogether which results in elongated austenite grains which are strained due to rolling. The ferrite forms in these elongated austenite grains are finer and help in achieving higher strength with higher toughness. Micro-alloy addition helps in achieving the fine ferrite grain size in final microstructure and thus this process helps in achieving fine austenite grains and transforms into fine ferrite subsequently.

The synergistic effects of chemistry and different processing parameters of hot rolled steel mechanism (HSM) such as chemical composition, the speed of rollers, thickness of sheet, etc. are optimized to achieve the desired mechanical properties. Mathematically, the multi-objective optimization problem of maximization of % elongation (EI), Ultimate tensile Strength (UTS) and Yield strength (YS) can be represented as (2)–(3)

$$\begin{aligned} &\text{Objective} : \underset{(x_{gcp})}{\text{Min}}(-\text{EI}(x_{gcp}), -\text{UTS}(x_{gcp}), -\text{YS}(x_{gcp})) \\ &\text{Subject to,} \end{aligned} \tag{2}$$

$$\begin{aligned} &24 \leq \text{El} \leq 100 \\ &530 \leq \text{UTS} \leq 590 \\ &440 \leq \text{YS} \leq 500 \\ &\text{lb} \leq x_{gcp} \leq \text{ub} \end{aligned} \tag{3}$$

Here, El (x_{gcp}), UTS (x_{gcp}) and YS(x_{gcp}) are the calculated values of percentage elongation, ultimate tensile strength and yield strength, respectively, which are, in turn, functions of decision variable set x_{gcp}. The calculated values are going to be obtained from an ANN model that connects the decision variable set x_{gcp} and the objective functions, i.e. El (x_{gcp}), UTS (x_{gcp}) and YS [26]. There are 21 such decision variables representing various geometrical, chemical and processing parameters in the decision variable set bounded by their respective lower and upper bounds (denoted as lb and ub, respectively). The bounds of this decision variable set, x_{gcp}, for different case studies are shown in Table 1. On the other hand, objective functions are also bounded by their lower and upper allowable values as shown in Eq. (2). In this formulation, the decision variable set is optimized by solving a multi-objective optimization problem and a Pareto optimal front is generated where each point on the Pareto curve corresponds to decision variable set along with the trade-off values of three objectives.

5 Results and Discussions

Both NSGA-II and MOEA/DD algorithms are applied on two hot rolling mill case studies based on the realistic plant data. Details of this case study are given in Table 1. Values used in Table 1 for different parameters are primarily driven by the require-

Table 1 The lower bounds (lb) and upper bounds (ub) of different geometrical, chemical and process parameters for two case studies

S. no.	Decision variables	Case 1		Case 2	
		lb	ub	lb	ub
1	Thickness/mm	5	8	8	12
2	Carbon (C)/wt%	0.03	0.07	0.03	0.07
3	Manganese (Mn)/wt%	1	1.45	1	1.45
4	Aluminium (Al)/wt%	0.01	0.05	0.01	0.05
5	Phosphorus (P)/wt%	0.006	0.02	0.006	0.02
6	Sulphur (S)/wt%	1.00E–03	0.02	1.00E–03	0.02
7	Nitrogen (N)/ppm	20	50	20	50
8	Titanium (Ti)/wt%	0.015	0.04	0.015	0.04
9	Chromium (Cr)/wt%	0.008	0.15	0.008	0.15
10	Copper (Cu)/wt%	0.002	0.35	0.002	0.35
11	Niobium (Nb)/wt%	0.005	0.065	0.005	0.065
12	Vanadium (V)/wt%	0.005	0.05	0.005	0.05
13	Silicon (Si)/wt%	0.005	0.03	0.005	0.03
14	Boron (B)/wt%	0	0.0039	0	0.0039
15	F6 Speed/ms^{-1}	3.9	5.8	2.8	3.9
16	FRT/^{o}C	860	920	840	900
17	CT/°C	600	650	600	650
18	Roughing mill exit temp. (RMET)/°C	1000	1.12E + 03	1000	1.12E + 03
19	Slab drop out temp. (SDOT)/°C	1.18E + 03	1230	1.18E + 03	1230
20	Soak time/min	4.3333	8.00E + 01	4.3333	8.00E + 01
21	Slab retention time/min	1	300	1	300

Table 2 The specifications of evolutionary multi/many-objective (MOEA/DD & NSGA-II) approaches

Multi-objective evolutionary algorithm specifications		
Algorithm	MOEA/DD	Elitist NSGA-II
Population size (N)	91	100
Weight vectors (V_{Wt})	91	–
Number of generations	1000	1000
Weight vector partition (D)	0.0833	–
Neighbourhood weight vector (N_{Wt})	20	–
Crossover probability	0.95	0.80
Mutation probability	0.05	0.05

ments of the target steel product. Both algorithms are utilized to find solutions that maximize the mechanical properties such as YS, UTS and El by choosing the parameters in the given ranges as presented in Table 1. The indigenously developed real coded NSGA-II and MOEA/DD are used here. All calculations are performed on Intel® Xeon® CPU E5-2690 0 @ 2.90 GHz (2 processors) 128 GB RAM machine.

Case I: Hot rolled case study 1

Based on the values of geometrical, chemical and process parameters for this case as given in Table 1, multi-objective optimization is performed while maximizing the YS, UTS and El properties (Eqs. (2) and (3)). The parameter settings used for NSGA-II and MOEA/DD are described in Table 2. As a result, a non-dominated front (*PO front*) is constructed using NSGA-II and MOEA/DD, when initialized with random population. As opposed to single solution, multiple number of equally competent solutions are obtained. Figure 5 shows the final PO front obtained as a result of applying both the approaches. The performance of both approaches is further compared for this case study. Since a multiple number of equally competent solutions are obtained on each Pareto front, and these fronts are intersecting with one another in the objective space, it is hard to determine which one is better among them. Under this circumstance, generally different metrics such as an inverted generational distance (IGD) [27], etc. can be used to compare the performance of these Pareto fronts in terms of convergence and diversity. Convergence is the measure of movement towards the ideal point [10] and more diverse the front, better choices a decision-maker can have. As the lower IGD value is an indication of a better Pareto front in terms of convergence and diversity, the superior performance of MOEA/DD can be observed over the NSGA-II results (Table 3).

As explained, every point on the front contains the information about the geometrical, chemical and process parameters along with the conflicting objective values. Four different points (P1, P2, P3 and P4) are selected from the final front (Fig. 5b), where points P1, P2 and P3 are the extreme points of the final PO front corresponding to the scenarios where each of these objective functions is optimized when consid-

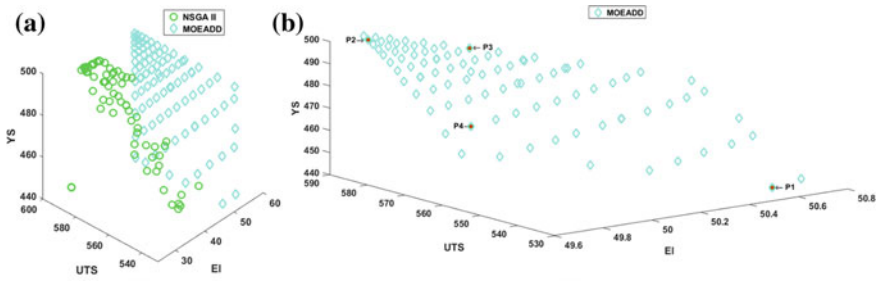


Fig. 5 Final non-dominated fronts generated using both NSGA-II (MOEA) and MOEA/DD (MaOEA) approaches for case I

Table 3 IGD metric based performance calculation of all approaches NNC, NSGA-II, MOEA/DD and hybrid for two case studies

S. no.	Multi/many-objective approaches	Case 1	Case 2
1	NSGA-II [10]	0.4958	0.3985
2	MOEADD [1]	0.0090	0.0400

ered individually, i.e. P1 corresponds to the only maximization of objective EI, P2 is for only maximization of UTS not considering any other objectives and so on (Table 4). However, point P4 is chosen from a region that has some contribution from all the three objectives showing the trade-off among all of them. Table 3 shows the difference in tri-objective function values and processing parameters for each of these points (P1, P2, P3 and P4). This shows that these points are different in nature in terms of decision variables as well as the objective values and they are alternative solutions to a decision-maker. This also shows why the conflicting multi-objective optimization is not a trivial case of considering each of the objectives and performing their individual optimization because in that way, only the terminal points (P1, P2 and P3) can be identified, not so many other trade-off solutions lying in between (e.g. P4). Of course, out of these many solutions from the Pareto set, only one solution has to be chosen, it is always better for a decision-maker to observe the trade-off present among the solutions first before converging to the final solution based on some higher level information (e.g. cost) that can differentiate these solutions.

Case II: Hot rolled case study 2

Similar to the case I, the values of the bounds for chemistry and processing parameters in this case study are given in Table 1. The only difference is in the bounds (lb and ub) of some decision variables (e.g. thickness, F6 speed and FRT) by keeping the other variables same as the previous case study. The rationale of doing this is lying in the fact that changing certain parameter bounds is going to generate PO solutions in certain different ranges. Both NSGA-II and MOEA/DD are applied to perform the multi-objective optimization with an aim of maximizing YS, UTS and El

Table 4 Different geometrical, chemical and process parameters and objective function values of points P1, P2, P3 and P4 for Case 1 and Case 2, respectively

S. no.		Case I				Case II			
		Point P1	Point P2	Point P3	Point P4	Point P1	Point P2	Point P3	Point P4
Decision Variables									
1	Thickness	5.0003	5.0002	5.0001	5.0004	8.0001	8.0003	8.0002	8.0001
2	C	**0.0334**	**0.0377**	**0.0375**	**0.0351**	**0.0324**	**0.0328**	**0.0336**	**0.0327**
3	Mn	1.4485	1.4498	1.4497	1.4494	**1.3767**	**1.3617**	**1.3936**	**1.3948**
4	Al	0.0100	0.0100	0.0100	0.0100	0.0103	0.0106	0.0104	0.0104
5	P	0.0060	0.0060	0.0060	0.0060	0.0060	0.0060	0.0060	0.0060
6	S	0.0010	0.0010	0.0010	0.0010	**0.0107**	**0.0121**	**0.0126**	**0.0117**
7	N	49.89	49.70	49.59	49.88	49.95	49.98	49.97	49.98
8	Ti	**0.0350**	**0.0399**	**0.0398**	**0.0367**	0.0400	0.0400	0.0400	0.0400
9	Cr	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500
10	Cu	**0.0777**	**0.0502**	**0.0506**	**0.0608**	0.0020	0.0021	0.0020	0.0020
11	Nb	0.0112	0.0103	0.0075	0.0153	0.0102	0.0202	0.0082	0.0112
12	V	0.0050	0.0050	0.0050	0.0050	0.0050	0.0050	0.0050	0.0050
13	Si	0.0050	0.0050	0.0050	0.0050	**0.0083**	**0.0124**	**0.0111**	**0.0103**
14	B	0.0010	0.0007	0.0008	0.0007	0.0000	0.0000	0.0000	0.0000
15	F6 speed	5.80	5.80	5.80	5.80	3.90	3.90	3.90	3.90
16	FRT	860.00	860.00	860.00	860.01	840.00	840.00	840.00	840.00
17	CT	600.06	600.02	600.02	600.05	600.00	600.00	600.00	600.00
18	RMET	1119.64	1119.86	1119.80	1119.66	**1104.30**	**1118.55**	**1119.62**	**1118.99**
19	SDOT	1229.99	1229.99	1229.99	1229.98	1229.99	1229.99	1229.99	1229.99
20	Soak Time	**53.13**	**50.29**	**47.83**	**56.16**	**48.70**	**58.61**	**50.39**	**53.62**
21	Retention Time	**114.56**	**116.43**	**105.03**	**129.09**	**1.11**	**1.23**	**1.16**	**1.15**

(continued)

Table 4 (continued)

S. no.		Case I				Case II			
		Point P1	Point P2	Point P3	Point P4	Point P1	Point P2	Point P3	Point P4
Objective functions									
1	% Elongation (El)	**50.62**	49.76	49.89	50.10	**49.73**	49.39	49.51	49.58
2	Ultimate tensile strength (UTS)	538.15	**590.00**	571.96	585.01	535.49	**590.00**	545.79	554.36
3	Yield strength (YS)	440.00	497.99	**500.00**	455.12	453.70	465.58	**500.00**	480.27

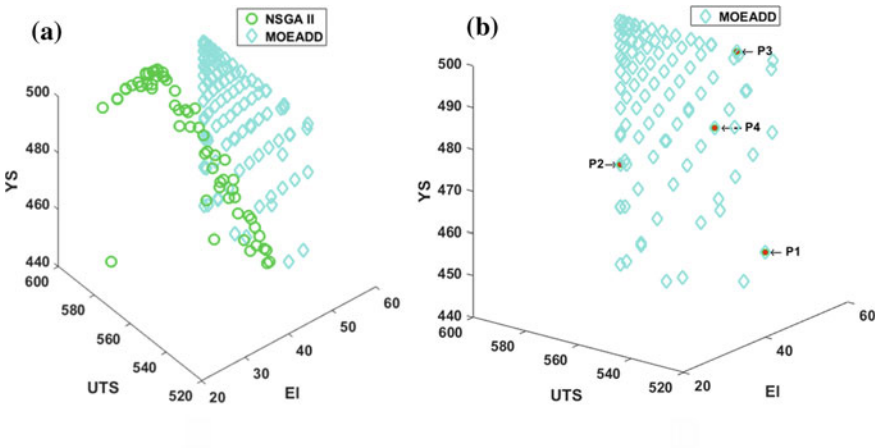


Fig. 6 Final non-dominated fronts generated using both NSGA-II (MOEA) and MOEA/DD (MaOEA) approaches for case II

(Eqs. 1 and 2) using parameter settings as given in Table 2. Figure 6a shows the final non-dominated front generated after applying both the algorithms. The performance metric IGD [27] is used to compare the performance of each PO front as shown in Table 3 which shows MOEA/DD performance is better. Similar to case 1, four different points P1, P2 and P3 and P4 are selected on the MOEA/DD front (Fig. 6b) in a way such that P1, P2 and P3 are the extreme points and point P4 is to represent some region in the objective space where the effect of all the objectives are present. Table 4 shows the difference in the objective function values and the corresponding decision variables values for each point, which shows that each and every solution on the generated PO front is of different nature and important for a decision-maker to choose based on the requirement.

6 Conclusion

A comparative study of multi-objective (NSGA-II) and many-objective (MOEA/DD) optimization algorithms has been performed in this study. Two case studies of a hot rolling mill operation of steel industry have been considered where the mechanical properties such as % elongation, ultimate tensile Strength and Yield strength are maximized to find high-quality hot rolled steel products. Optimized values of twenty-one parameters depicting geometry, chemistry and operating process condition, etc. of the process are obtained through this study which are connected with the mechanical properties using data-based modelling approach such as artificial neural networks. MOEA/DD algorithm is found to deliver better set of PO solutions compared to well-established NSGA-II validated through the IGD metrics, a measure of convergence

and diversity. The final PO set provides a decision-maker with several alternative solutions which can act a decision set before implementing a final solution to the process.

References

1. Li, K., Deb, K., Zhang, Q., & Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation, 19*(5), 694–716.
2. Mitra, K. (2008). Genetic algorithms in polymeric material production, design, processing and other applications: A review. *International Materials Reviews, 53*(5), 275–297.
3. Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., & Zhang, T. (2016). Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*.
4. Cai, X., Yang, Z., Fan, Z., & Zhang, Q. (2017). Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization. *IEEE Transactions on Cybernetics*.
5. Herrero, J. G., Berlanga, A., & Lopez, J. M. M. (2009). Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters. *IEEE Transactions on Evolutionary Computation, 13*(1), 151–168.
6. Yeung, S. H., Man, K. F., Luk, K. M., & Chan, C. H. (2008). A trapeizform U-slot folded patch feed antenna design optimized with jumping genes evolutionary algorithm. *IEEE Transactions on Antennas and Propagation, 56*(2), 571–577.
7. Handl, J., Kell, D. B., & Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*(2).
8. Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2013). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation, 17*(3), 321–344.
9. Zhang, X., Tian, Y., & Jin, Y. (2015). A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 19*(6), 761–776.
10. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (Vol. 16). John Wiley & Sons.
11. Ziztler, E., Laumanns, M., & Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design, Optimization, and Control*, 95–100.
12. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.
13. Fu, G., Kapelan, Z., Kasprzyk, J. R., & Reed, P. (2012). Optimal design of water distribution systems using many-objective visual analytics. *Journal of Water Resources Planning and Management, 139*(6), 624–633.
14. Lygoe, R. J., Cary, M., & Fleming, P. J. (2013, March). A real-world application of a many-objective optimisation complexity reduction process. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 641–655). Springer, Berlin, Heidelberg.
15. Purshouse, R. C., & Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation, 11*(6), 770–784.
16. Li, M., Yang, S., & Liu, X. (2014). Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation, 18*(3), 348–365.
17. Yang, S., Li, M., Liu, X., & Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 17*(5), 721–736.

18. Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation, 19*(1), 45–76.
19. Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation, 11*(6), 712–731.
20. He, Z., & Yen, G. G. (2016). Many-objective evolutionary algorithm: Objective space reduction and diversity improvement. *IEEE Transactions on Evolutionary Computation, 20*(1), 145–160.
21. Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation, 18*(4), 577–601.
22. Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization, 8*(3), 631–657.
23. Agrawal, R. B., Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems, 9*(2), 115–148.
24. Deb, K., & Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics, 26,* 30–45.
25. Li, K., Deb, K., Zhang, Q., & Kwong, S. (2014). Efficient non-domination level update approach for steady-state evolutionary multiobjective optimization. *Department of Electrical and Computer Engineering, Michigan State University, East Lansing, USA, Tech. Rep. COIN Report*, (2014014).
26. Mohanty, I., Sarkar, S., Jha, B., Das, S., & Kumar, R. (2014). Online mechanical property prediction system for hot rolled IF steel. *Ironmaking and Steelmaking, 41*(8), 618–627.
27. Mohammadi, A., Omidvar, M. N., & Li, X. (2013, June). A new performance metric for user-preference based multi-objective evolutionary algorithms. In *2013 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2825–2832). IEEE.

Index

S. Datta and J. P. Davim (eds.), *Optimization in Industry*, Management and Industrial Engineering, https://doi.org/10.1007/978-3-030-01641-8