Journal of ICT, 20, No. 2 (April) 2021, pp: 171–211

JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
http://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Yasear, S. A., & Ku-Mahamud, K. R. (2021). Review of the multi-objective swarm intelligence optimization algorithms. *Journal of Information and Communication Technology, 20*(2), 171-211. https://doi.org/10.32890/jict2021.20.2.3

Review of Multi-Objective Swarm Intelligence Optimization Algorithms

Shaymah Akram Yasear & Ku Ruhana Ku-Mahamud
School of Computing
Universiti Utara Malaysia, Malaysia

shayma.akram.yasear@gmail.com
ruhana@uum.edu.my

Received: 30/8/2020 Revised: 11/10/2020 Accepted: 13/10/2020 Published: 23/2/2021

ABSTRACT

Multi-objective swarm intelligence (MOSI) metaheuristics were proposed to solve multi-objective optimization problems (MOPs) that consist of two or more conflict objectives, in which improving an objective leads to the degradation of the other. The MOSI algorithms were based on the integration of single objective algorithms and multi-objective optimization (MOO) approaches. The MOO approaches included scalarization, Pareto dominance, decomposition, and indicator-based. In this paper, the status of MOO research and state-of-the-art MOSI algorithms, namely multi-objective particle swarm, artificial bee colony, firefly algorithm, bat algorithm, gravitational search algorithm, grey wolf optimizer, bacterial foraging, and moth-flame optimization algorithms, were reviewed. These reviewed algorithms were mainly developed to solve continuous MOPs. The review was based on how the algorithms dealt with objective functions using MOO approaches, the benchmark MOPs used in the evaluation and performance metrics. Furthermore, it described the advantages and disadvantages of each MOO approach and provides some possible future research directions in this area. The results showed that several MOO approaches were used in most of the proposed MOSI algorithms. Integrating other different MOO approaches might help in developing more effective optimization algorithms, especially in solving complex MOPs. Furthermore, most of the MOSI algorithms were evaluated using MOPs with two objectives, which clarified open issues in this research area.

Keywords: Optimization, metaheuristic, nature-inspired, Pareto front, population-based.

Introduction

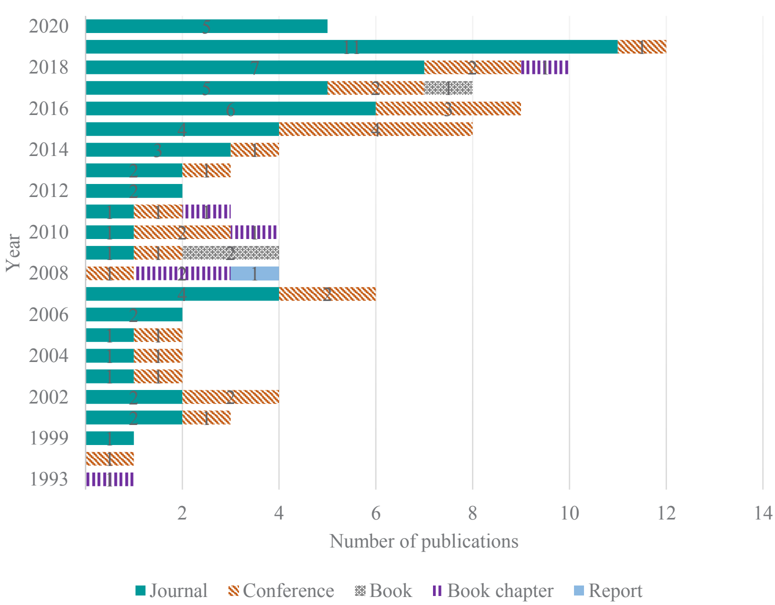
A real-world optimization problem usually consists of conflicting objectives that should be taken into consideration when making decisions. A problem associated with multiple objectives is commonly called a multi-objective optimization problem (MOP). The process of solving a MOP is known as multi-objective optimization (MOO). The solution of a MOP comprises a set of non-dominated solutions. The set of non-dominated solutions is called Pareto front. The MOP is a complex optimization problem and this complexity increases with the increasing number of objectives. Thus, the process of solving a MOP is non-trivial.

Metaheuristics are general optimization methods applicable to solve different optimization problems (Sörensen et al., 2018; Stojanović et al., 2017). In contrast to traditional methods, such as goal (Li, 2019), mixed-integer (Singh & Goh, 2019) linear programming, and weighted summation (Marler & Arora, 2010), metaheuristics apply a stochastic approach to find a feasible solution among randomly generated solutions. Metaheuristics are simple to implement practically and have proven their efficiency in different fields, such as operations research (Li et al., 2020), engineering (Dede et al., 2020; Sayed et al., 2018), and healthcare (Tsai et al., 2016). The strong point of metaheuristics is that they do not require detailed knowledge of the problem. One can represent metaheuristics by a black box carrying inputs (the variables) and outputs according to the objective functions (Talbi, 2009; Tamura & Gallagher, 2019).

The focus of this paper is on swarm intelligence (Beni & Wang, 1993) algorithms that have gained great attention as compared to evolutionary algorithms (Del Ser et al., 2019). Many swarm intelligence algorithms have been proposed and used to solve various optimization problems (Karaboga & Basturk, 2007; Kennedy & Eberhart, 1995; Mirjalili, 2015; Mirjalili et al., 2014; Rashedi et al., 2009; Yang, 2009, 2010). This due to their simple structure and high solution accuracy. However, these algorithms were mainly proposed to deal with single objective optimization problems (SOPs), where the goal is to minimize or maximize a single criterion (objective). To solve MOPs, several multi-objective swarm intelligence (MOSI) algorithms have been proposed (Coello et al., 2004; Hassanzadeh & Rouhani, 2010; Mirjalili et al., 2016; Niu et al., 2013; Savsani & Tawhid, 2017; Yang, 2012, 2013). In practice, a MOSI algorithm consists of combining a single objective swarm intelligence algorithm with a MOO approach to handle MOPs.

Figure 1

Year-Wise Distribution of Publications.



However, the number of published papers related to MOSI algorithms is relatively low as compared to multi-objective evolutionary algorithms. Most real-world problems are multi-objective in nature. Thus, the current trend is to either develop new algorithms and validate them with some of the metrics of MOPs or to develop interesting applications of existing algorithms.

In this paper, the MOSI algorithms are reviewed based on the MOO approaches. Despite the numerous MOSI optimization algorithms currently available, there is no review based on the MOO approaches that has been published so far to the best of the authors' knowledge. The MOSI algorithm papers that have been reviewed cover the benchmark MOPs and performance metrics used in the evaluation process. Figure 1 shows the year-wise distribution of the publications that have been reviewed.

A total of 100 publications related to swarm intelligence algorithms, MOO approaches, MOSI algorithms, benchmark MOPs, and performance metrics obtained from journals, conference proceedings, book chapters, and reports have been reviewed. Among these publications, 62 papers were published in journals, 28 papers appeared in conference proceedings, 6 papers were from book chapters, 3 books, and a technical report. The types and title of publications are shown in Table 1. The publications are listed in different databases, namely the Web of Science, Scopus, Association for Computing Machinery, Springer, Institute of Electrical and Electronics Engineers (IEEE) Xplore, ScienceDirect, and Google Scholar. Title, abstract, and index terms were used to conduct the search for publication.

Table 1

Types and Publication Titles

No.	Type	Title	No.	Type	Title
1.	Journal	IEEE Transactions on Evolutionary Computation.	40.	Journal	International Journal of Production Research.
2.	Journal	Evolutionary Computation.	41.	Journal	Phylogenetics and Evolution.
3.	Journal	IEEE Control Systems Magazine.	42.	Journal	Mathematical Problems in Engineering.
4.	Journal	Genetic Programming and Evolvable Machines, Springer.	43.	Journal	Journal of Experimental & Theoretical Artificial Intelligence.
5.	Journal	International Journal of Intelligent Systems.	44.	Journal	Springer Nature: Computer Science.
6.	Journal	Engineering Optimization.	45.	Journal	Journal of Cleaner Production.
7.	Journal	Information Sciences.	46.	Journal	Natural Computing.
8.	Journal	Structural and multidisciplinary Optimization, Springer.	47.	Journal	Transportation Research Part C: Emerging Technologies.
9.	Journal	SIAM Journal on Optimization.	48.	Proceeding	International Conference on Computer Communication and Informatics.
10.	Journal	Swarm and Evolutionary Computation.	49.	Proceeding	International Conference on Multimedia and Ubiquitous Engineering.
11.	Journal	International Journal of Bio-Inspired Computation.	50.	Proceeding	IEEE International Conference on Granular Computing.

(continued)

No.	Type	Title	No.	Type	Title
12.	Journal	Engineering with Computers.	51.	Proceeding	Stochastic Algorithms: Foundations and Applications, Springer.
13.	Journal	Neurocomputing.	52.	Proceeding	International Conference on Computational Intelligence, Communication Systems and Networks, IEEE.
14.	Journal	Advances in Engineering Software.	53.	Proceeding	Annual Conference on Genetic and Evolutionary Computation, Association for Computing Machinery.
15.	Journal	Algorithms.	54.	Proceeding	IEEE Innovative Smart Grid Technologies-Asia.
16.	Journal	Knowledge-Based Systems.	55.	Proceeding	Latin American Computing Conference.
17.	Journal	European Journal of Operational Research.	56.	Proceeding	International Conference on Parallel Problem Solving from Nature, Springer.
18.	Journal	International Journal of Electrical Power & Energy Systems.	57.	Proceeding	International Fuzzy Systems Association World Congress, Springer.
19.	Journal	International Journal of System Assurance Engineering and Management.	58.	Proceeding	International Conference on Neural Networks, IEEE.
20.	Journal	Expert Systems with Applications.	59.	Proceeding	International Conference on Evolutionary Multi-Criterion Optimization, Springer.
21.	Journal	Neural Computing and Applications.	60.	Proceeding	International Energy and Sustainability Conference.

(continued)

No.	Type	Title	No.	Type	Title
22.	Journal	Advanced Engineering Optimization Through Intelligent Techniques.	61.	Proceeding	IEEE Region 10 Conference.
23.	Journal	Computational Intelligence and Neuroscience.	62.	Proceeding	Congress on Evolutionary Computation, IEEE.
24.	Journal	Applied Intelligence.	63.	Proceeding	Genetic and Evolutionary Computation Conference, Springer.
25.	Journal	Engineering Review.	64.	Proceeding	Chinese Control Conference, IEEE.
26.	Journal	Engineering Applications of Artificial Intelligence.	65.	Proceeding	IEEE Congress on Evolutionary Computation.
27.	Journal	Applied Soft Computing.	66.	Proceeding	Power Systems Conference.
28.	Journal	Optimization Online.	67.	Proceeding	International Conference on Pattern Recognition Applications and Methods.
29.	Journal	IEEE Access.	68.	Book	Multiobjective Optimization.
30.	Journal	Mathematics.	69.	Book	Predator-Prey Interactions: Co-Evolution Between Bats and Their Prey.
31.	Journal	Journal of Risk and Reliability.	70.	Book	Metaheuristics: From design to implementation.
32.	Journal	IETE Journal of Research.	71.	Book	Multi-objective Evolutionary Optimisation for Product Design and Manufacturing.
33.	Journal	International Journal of Systems Science.	72.	Book chapter	Swarm Intelligence in Cellular Robotic Systems.

(continued)

No.	Type	Title	No.	Type	Title
34.	Journal	IAENG International Journal of Computer Science.	73	Book chapter	Nature Inspired Cooperative Strategies for Optimization, Springer.
35.	Journal	Wireless Networks.	74.	Book chapter	Classical and Recent Aspects of Power System Optimization.
36.	Journal	Computers & Electrical Engineering.	75.	Book chapter	Handbook of Heuristics.
37.	Journal	Complex & Intelligent Systems.	76.	Book chapter	Multiobjective Optimization: Interactive and Evolutionary Approaches.
38.	Journal	Engineering Optimization.	77.	Technical Report	University of Essex and Nanyang Technological University.
39.	Journal	Open Mathematics.			

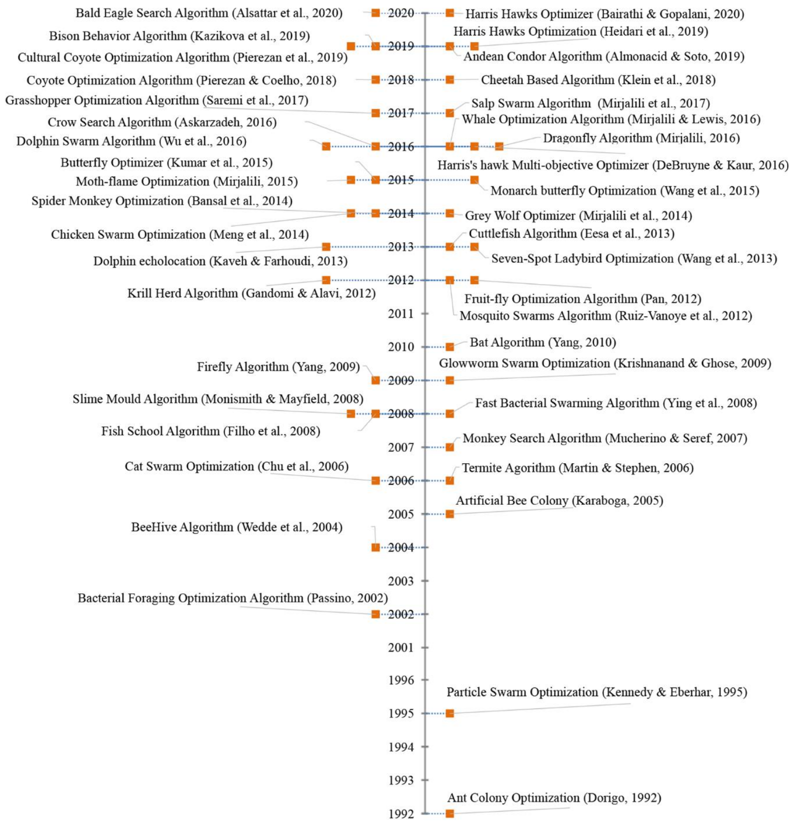
The sections in this paper are organized as follows. A brief definition of swarm intelligence and description of the most popular swarm intelligence optimization algorithms are presented in the next section. This is followed by describing the MOO approaches, in terms of the ways in dealing with objective functions and limitations. Next, the reviewed MOSI optimization algorithms based on the MOO approaches, benchmark MOPs, and performance metrics are presented. Lastly, the conclusion and future work of developing MOSI algorithms are highlighted.

SWARM INTELLIGENCE OPTIMIZATION ALGORITHMS

Swarm intelligence is an artificial intelligence technique that refers to the local interactions between agents or the environment by following some simple rules (Beni & Wang, 1993). Figure 2 shows the timeline of swarm intelligence algorithms that have been proposed from 1992 until 2020.

Figure 2

Chronology of Swarm Intelligence Metaheuristics.



Most swarm intelligence metaheuristics were developed according to the collective behavior of groups in biological systems, such as bird flocking, fish schooling, and animal herding. However, not all swarm intelligence metaheuristics are developed this way. Other algorithms have been developed using the inspiration of physical systems such as gravitational search algorithm (GSA) (Rashedi et al., 2009).

This paper briefly describes the popularly used swarm intelligence algorithms (Lones, 2020), namely particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), bacterial foraging optimization (BFO) (Passino, 2002), artificial bee colony (ABC) (Karaboga & Basturk, 2007), bat algorithm (BA) (Yang, 2010), grey wolf optimizer (GWO) (Mirjalili et al., 2014), firefly algorithm (FA) (Yang, 2009), GSA, and moth-flame optimization (MFO) (Mirjalili, 2015). Additionally, it reviews the MOSI algorithms based on MOO integrated with each algorithm.

The PSO algorithm mimics the swarm behavior of animals such as flocks of birds and schools of fish. The BFO algorithm proposed by Passino (2002) was developed according to the foraging behavior of *Escherichia coli* bacteria. The ABC algorithm was developed based on the foraging behavior of honeybees. The FA algorithm mimics the light-emitting behavior of fireflies. These insects use special organs to produce light inside their bodies. This light production is a form of chemical reaction called bioluminescence (Stanger-Hall et al., 2007). The attractiveness between fireflies is proportional to light intensity. For any two shining fireflies, the one of lesser intensity will move toward the greater one. If there is no brightness difference, the movement occurs at random. The BA mimics the echolocation behavior of microbats, which allows them to efficiently locate and hunt their prey even in complete darkness (Jacobs & Bastian, 2017). The GSA was developed according to the Newton's laws of gravity and motion. In the GSA, the collection of masses represents the searcher agents. The GWO was developed according to the leadership hierarchy and hunting mechanism of grey wolves. the algorithm is guided by the first three best solutions in the search space that are known as alpha, α, beta, β, and delta, δ. The remaining candidate solutions are omegas, ω. Searching for prey is an exploration or global search, while attacking the prey is exploitation or local search. The MFO algorithm (Mirjalili, 2015) was developed according to the navigation behavior of moths in nature. In the MFO algorithm, a moth spirally flies around lights and utilizes the transverse orientation technique to fly long distances in a straight path. This can be achieved by maintaining a constant angle relative to a distant point source of the moon. In MFO, the moths are modeled as candidate solutions for an optimization problem, while flames represent the best position found so far.

MULTI-OBJECTIVE OPTIMIZATION

The MOO problem can be defined as the search for a vector $X = (x_1, ..., x_n)$ that optimizes M objectives, $f_M(X)$ and satisfies constraints as shown in Equation 1 (Deb, 2011).

$$\begin{gathered}\text{Minimize } f_m(\vec{X}) \quad m = 1,2,\dots,M; \\ \text{subject to:} \\ g_k(\vec{X}) \leq 0 \quad k = 1,2,3,\dots,K \\ h_j(\vec{X}) = 0 \quad j = 1,2,3,\dots,J \\ x_i{}^{lb} \leq x\mathrm{i}_i \leq x_i{}^{ub} \; i = 1,2,\dots,D\end{gathered} \tag{1}$$

where $h(X)$ and $g(X)$ are the equality and inequality constraints, respectively. represent the ranges of the decision variables, X. D is the dimension of decision space.

The MOO approaches, based on the way of dealing with objective functions, can be divided into four main categories, namely scalarization, Pareto dominance, decomposition, and indicator-based (Emmerich & Deutz, 2018). Scalarization is a traditional approach to solve MOPs. This approach transforms a MOP problem into a SOP. A common scalarization method is the weighted sum. This method consists of adding all the objectives by assigning a weight for each objective (Emmerich & Deutz, 2018). The Pareto dominance approach uses Pareto dominance relation to select non-dominated solutions. According to the Pareto dominance relation, a solution p is said to dominate q, if a solution p is better than q in at least one objective, and p is better than or equal to q in all $f_M(X)$ (Emmerich & Deutz, 2018). The Pareto dominance is the most popular approach in the field of MOO. A decomposition-based approach transforms the MOP into a set of SOPs that are solved by using a single objective optimization algorithm. A scalarization method is used to calculate the fitness value of each sub-problem. Each sub-problem is associated with a weight vector (Tan et al., 2019). The indicator-based approach was first proposed as a general framework by Zitzler and Künzli (2004). This approach uses performance indicators, such as the hypervolume (Zitzler & Thiele, 1999), to score solutions. The goal is to maximize (in the case of hypervolume) the value of the indicator associated with the approximation (Emmerich & Deutz, 2018).

The scalarization-based approaches are strongly dependent on the aggregation function. In the weighted sum method, the weights may not reflect the relative importance of the objectives. Thus, the problem with new weights need to be resolved (Brück et al., 2018; Jakob &

Blume, 2014). The Pareto dominance approach has become the main approach in solving MOPs. However, the Pareto dominance-based algorithms may face the loss of a selection pressure (Li et al., 2018; Ochoa et al., 2000), which leads to poor convergence toward the Pareto front (Coello et al., 2019; Liu et al., 2019). Obtaining a uniformly distributed solution set for many decomposition-based algorithms still remains a challenge (Coello et al., 2019). The decomposition-based approaches are strongly affected by the method used to generate weights and scalarization function. Improper weight vector leads to poor convergence toward the true Pareto front (Weiszer et al., 2018). Furthermore, the number of weights grows exponentially with the number of objectives (Emmerich & Deutz, 2018). The indicator-based approach has been recently used by several studies as an alternative to deal with MOP. However, the advantages of this approach are still not as clear as compared to other MOO approaches (Coello et al., 2019).

REVIEW OF THE MOSI OPTIMIZATION ALGORITHMS

The section provides a review for the multi-objective PSO (MOPSO), multi-objective ABC (MOABC), multi-objective FA (MOFA), multi-objective BA (MOBA), multi-objective GSA (MOGSA), multi-objective GWO (MOGWO), multi-objective BFO (MOBFO), and multi-objective MFO (MOMFO) algorithms. These algorithms can be considered as extensions to the single objective optimization algorithms, which are integrated with MOO to solve MOPs.

Scalarization-Based Approach

Several MOSI algorithms have been proposed based on the scalarization approach (Mellal & Zio, 2019; Yang, 2012; 2013). Yang (2012) proposed MOBA, which extends the BA algorithm, to solve a MOP. This algorithm was developed according to the weighted sum method. The proposed algorithm was evaluated by using different MOPs with two objectives. However, the performance of MOBA was not compared to the performance of other MOSI algorithms. In Yang (2013), the same author of MOBA proposed MOFA, which was also developed based on the weighted sum approach and used Lévy flights to maintain population diversity. MOFA was used to solve a set of MOPs and engineering problems. According to Yang (2013), the MOFA outperformed other MOO algorithms. Following the same approach, Mellal and Zio (2019) proposed a MOPSO algorithm based on the weighted sum method, where Lévy flight was used to maintain the population diversity. The authors showed that the results of the proposed algorithm were superior to the standard PSO. In solving MOPs with non-convex Pareto front, some solutions may not be accessible using the weighted sum method (Brück et al., 2018; Jakob & Blume, 2014). Therefore, there is no guarantee that the Pareto curve will be well distributed.

Pareto Dominance-Based Approach

Many MOSI algorithms have been proposed according to the Pareto dominance approach (Akbari et al., 2012; Bhowmik & Chakraborty, 2015; Chen et al., 2019; Coello et al., 2004; Hassanzadeh & Rouhani, 2010; Huang et al., 2006; Janga Reddy & Nagesh Kumar, 2007; Kumawat et al., 2017; Li, 2003; Man-Im et al., 2015; Mirjalili et al., 2016; Niu et al., 2013; Prakash et al., 2016; Savsani & Tawhid, 2017; Sierra & Coello, 2005; Sun & Gao, 2019; Yang & Ji, 2016). These algorithms employed different strategies to maintain the population diversity. Some of these algorithms used the crowding distance to maintain the diversity of population (Bhowmik & Chakraborty, 2015; Chen et al., 2019; Huang et al., 2006; Janga Reddy & Nagesh Kumar, 2007; Li, 2003; Man-Im et al., 2015; Niu et al., 2013; Prakash et al., 2016; Sierra & Coello, 2005; Sun & Gao, 2019; Yang & Ji, 2016). However, in some cases, the crowding distance approach cannot be used to select appropriate solutions, which may affect the diversity of solutions (Savsani & Tawhid, 2017; Vachhani et al., 2016).

The grid mechanism proposed by Knowles and Corne (2000) has been used in algorithms proposed by Coello et al. (2004), Mirjalili et al. (2016), Akbari et al. (2012), Hassanzadeh and Rouhani (2010), and Kumawat et al. (2017) to maintain the diversity of non-dominated solutions stored in an external archive. However, the grid mechanism depends heavily on the number of cells and has a high computational complexity.

Although Pareto dominance-based algorithms (Bhowmik & Chakraborty, 2015; Hassanzadeh & Rouhani, 2010; Huang et al., 2006; Kumawat et al., 2017; Li, 2003; Man-Im et al., 2015; Niu et al., 2013; Sierra & Coello, 2005; Yang & Ji, 2016) showed good performance in terms of convergence and diversity in solving different MOPs, they have not been tested in solving MOPs with more than two objectives. Therefore, further testing needs to be conducted to determine the performance in solving more complex MOPs. In Akbari et al. (2012) and Mirjalili et al. (2016), the algorithms were evaluated by solving MOPs with two and three objectives. Based on the results, the algorithms showed superior performance as compared to other state-of-the-art algorithms such as multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang & Li, 2007) and MOPSO (Coello et al., 2004).

The proposed Pareto dominance-based algorithms have been mainly developed to solve particular MOPs (Mahmoodabadi & Shahangian, 2019; Mohamed et al., 2016). The MOGWO proposed by Mohamed et al. (2016) was used to solve the optimal power flow of MOPs and it showed superior performance as compared to other MOO algorithms. However, according to the no-free-lunch theorem, there is no optimization algorithm that works well on all optimization problems. An optimization algorithm may achieve very good results on a set of optimization problems; nevertheless, it is not suitable for others. Therefore, further testing needs to be conducted to evaluate the performance of this algorithm in solving different MOPs. Mahmoodabadi and Shahangian (2019) proposed a MOABC algorithm to solve MOPs where the diversity of solutions in the archive was maintained using a pruning technique. The proposed algorithm was used to design an adaptive controller for the ball-beam system. Furthermore, the MOABC was used to solve a set of MOPs with two objectives. However, the results were not compared with other MOO algorithms, which was required to validate the performance of the algorithm.

Decomposition-Based Approach

Some of the decomposition-based MOSI algorithms followed the same concept used in Zhang and Li (2007) and replaced the genetic algorithm with a swarm intelligence algorithm (Peng & Zhang, 2008; Sapre & Mini, 2020). However, in these algorithms, the old solutions were replaced by new solutions with respect to the aggregation function values. This replacement did not take into consideration the diversity of new solutions in the objective space, which might lead to poor population diversity (Dai et al., 2015). To overcome this limitation, Dai et al. (2015) proposed a MOPSO algorithm based on the decomposition approach where the Pareto optimal solution was generated for each sub-region in the objective space. In the proposed algorithm, different strategies were used to preserve the diversity of population. The crossover operations with selection strategy and neighborhood correction were used to perform the search process. Furthermore, the selection operation of the best solutions was performed based on the crowding distance, which was used as a fitness value for each solution. According to the results, the proposed algorithms could significantly outperform other MOO algorithms such as non-dominated sorting genetic algorithm (NSGA-II) (Deb, Pratap et al., 2002) and MOEA/D in solving a set of MOPs. However, the usage of crowding distance might lead to a loss of population diversity in some situations.

Others studies proposed a decomposition-based MOSI algorithm by utilizing a penalty boundary intersection (PBI) method, which is used as a scalarization function (Bai & Liu, 2016; Zapotecas Martínez & Coello Coello, 2011). According to Bai and Liu (2016), the proposed algorithm showed superior performance as compared to other state-of-the-art algorithms such as Pareto archive evolution strategy (Knowles & Corne, 2000), MOEA/D, NSGA-II, and optimal multi-objective optimization based on PSO (Niu & Shen, 2007). The performance of the algorithm proposed by Zapotecas Martínez and Coello Coello (2011) was evaluated by solving different MOPs with two and three objectives. Based on the results, the proposed algorithm outperformed smart multi-objective particle swarm optimizer using decomposition (Al Moubayed et al., 2010) and MOEA/D algorithms in solving most MOPs. Although the PBI method produced more uniform solutions than other scalarization functions, such as Tchebycheff, its performance depended on penalty parameter (Mohammadi et al., 2015).

Indicator-Based MOMH

The indicator-based approach is relatively new as compared to the Pareto dominance and decomposition-based approaches. Therefore, it has received little attention in the area of MOSI algorithms. García et al. (2014) proposed a MOPSO algorithm based on the hypervolume (Zitzler & Thiele, 1999) indicator. The proposed algorithm used the hypervolume contribution value to select the leaders from an external archive and as a mechanism for updating the external archive during the optimization process. Although the proposed hypervolume-based algorithm showed competitive performance as compared to other Pareto dominance-based and hypervolume-based algorithms, the main disadvantage of this approach was the computational complexity of the hypervolume, which increased by raising the number of objectives (Riquelme et al., 2015).

Other studies followed the same concept by using the R2 indicator instead of the hypervolume (Díaz-Manríquez et al., 2016; Wei et al., 2018). Díaz-Manríquez et al. (2016) proposed an R2-based MOPSO algorithm where the leaders of the swarm were selected based on the R2 indicator contribution value.

Furthermore, the usage of a Pareto dominance approach has been eliminated from the evolution process and applied only on the external archive. This leads to a reduction in the computational cost of the algorithm. Results were compared to other well-known algorithms such as MOEA/D and NSGA-II, which showed a competitive performance in solving MOPs with two and three objectives. Wei et al. (2018) proposed a MOPSO algorithm based on R2 indicator. The R2 indicator contribution value was used to select individuals from the external archive instead of the crowding distance. The swarm diversity in the archive was maintained through polynomial mutation (Deb, Pratap et al., 2002). Wei et al. (2018) highlighted that the performance, in terms of convergence and diversity achieved by R2-based MOPSO, was competitive as compared to those obtained by four state-of-the-art MOO algorithms. However, the performance of the proposed algorithm depended on the value of parameters, namely maximum age of particle, probability of crossover, and probability of mutation. In general, the R2-based algorithm requires a weight vector associated with the specific objective function. The number of weights increases with the number of objectives (Zitzler et al., 2008). Furthermore, the convergence toward the Pareto front depends strongly on the weight vector.

Other MOSI algorithms combined two or more MOO approaches to handle the multiple objectives (Al Moubayed et al., 2014; Li et al., 2015; Lin et al., 2015; Liu et al., 2019; Luo et al., 2017; Wei et al., 2017). The combination of Pareto dominance and decomposition-based approaches was first proposed by Al Moubayed et al. (2014). These approaches were integrated with MOPSO and the PBI method was used as a scalarization function in the decomposition approach. The proposed algorithm used the Pareto dominance relation to select and store non-dominated solutions in an external archive. The crowding distance was calculated for both objective and decision spaces to maintain the diversity of population. The particle leaders were selected based on the crowding distance values. The performance of the proposed algorithm was evaluated by solving a set of MOPs. Results showed that the proposed algorithm outperformed other MOO algorithms such as MOEA/D and OMOPSO.

Lin et al. (2015) followed the same concept and proposed a MOPSO algorithm by combining the Pareto dominance and decomposition approaches. In the proposed algorithm, two search strategies were utilized to preserve the search process. The leaders of particles were selected based on the best values of each sub-problem and all SOPs. The non-dominated solutions in the archive were updated based on the Pareto dominance approach and crowding distance. Results showed that the performance of the proposed algorithm outperformed other MOO algorithms in solving most MOPs.

Wei et al. (2017) proposed a MOPSO algorithm based on the decomposition and Pareto dominance approaches. The comprehensive learning strategy and mutation operator were applied in the algorithm to control the exploration and exploitation and avoid falling into local optima. To maintain the diversity of the external archive, the crowding distance was used. The performance of the proposed algorithm was evaluated by using a set of MOPs. The results were compared with other MOO algorithms, which showed that the proposed algorithm was competitive in solving most MOPs. In the proposed algorithm, the Tchebycheff method was used as a scalarization function. However, the main drawback of this method was the computational complexity as it minimized each objective when using the reference point (Ramirez et al., 2018). In general, the algorithms that have been developed based on the Pareto dominance and decomposition approaches and employed crowding-distance and PBI method inherit their drawbacks as described earlier.

Luo et al. (2017) and Luo et al. (2019) combined the indicator-based approach with Pareto dominance approach, while Li et al. (2015) combined the indicator approach with the decomposition approach. Luo et al. (2017) and Luo et al. (2019) proposed MOABC and MOPSO by integrating the epsilon indicator and Pareto dominance approach with the ABC and PSO algorithms. The epsilon indicator was used to evaluate the solutions and the Pareto dominance approach was applied to compare the solutions. An external archive was utilized to store the obtained non-dominated solutions. Based on the results, the proposed algorithms outperformed other state-of-the-art algorithms in solving MOPs with two and three objectives. However, the performance of an algorithm highly depends on the value of the epsilon indicator, which is determined by the decision maker. Improper value leads to poor approximation to the true Pareto front (Hernández-Díaz et al., 2007). Li et al. (2015) proposed a MOPSO algorithm based on the decomposition and R2-indicator approaches. The personal best position is updated by using the decomposition approach with different scalarization functions. The external archive based on the R2-indicator contribution value is used to select the global best solution. The performance of the proposed algorithm was evaluated by using MOPs with two and three objectives. However, according to Li et al. (2015), this algorithm was not suitable to solve high-dimensional MOPs with more than three objectives. Inspired by R2-MOPSO that was earlier proposed in Li et al. (2015), Liu et al. (2019) proposed a MOPSO algorithm to deal with high-dimensional MOPs. In the proposed algorithm, a bi-level archiving strategy based on the R2-indicator and decomposition approach was introduced to guide the search process. In the proposed algorithm, the personal best position was selected according to Pareto dominance relation, while the global-best position was selected based on the R2 contribution value. The performance of the algorithm was evaluated by solving high-dimensional MOPs and the results showed that it was superior than several MOO algorithms. Table 2 summarizes the MOO approaches applied in some of the well-known swarm intelligence metaheuristics.

Table 2

Summary of MOSI Optimization Algorithms with Respect to the MOO Approach

No.	Algorithm Reference	MOO Approach: Scalarization	Pareto Dominance	Decomposition	Indicator-based	Archive
1	MOBA Yang (2012)	✓	-	-	-	-
2	MOFA Yang (2013)	✓	-	-	-	-
3	MOPSO Mellal and Zio (2019)	✓	-	-	-	-
4	MOPSO Coello and Lechuga (2002)	-	✓	-	-	✓
5	MOPSO Coello et al. (2004)	-	✓	-	-	✓
6	MOPSO Janga Reddy and Nagesh Kumar (2007)	-	✓	-	-	✓
7	MOGSA Hassanzadeh and Rouhani (2010)	-	✓	-	-	✓
8	MOABC Akbari et al. (2012)	-	✓	-	-	✓
9	MOBFO Niu et al. (2013)	-	✓	-	-	-
10	MOPSO Man-Im et al. (2015)	-	✓	-	-	-
11	MOGSA Bhowmik and Chakraborty (2015)	-	✓	-	-	✓
12	MOBFO Yang and Ji (2016)	-	✓	-	-	✓

(continued)

No.	Algorithm Reference	MOO Approach: Scalarization	MOO Approach: Pareto Dominance	MOO Approach: Decomposition	MOO Approach: Indicator-based	Archive
14	MOGWO Mirjalili et al. (2016)	-	✓	-	-	✓
15	MOGWO Mohamed et al. (2016)	-	✓	-	-	✓
16	MOABC Kishor et al. (2016)	-	✓	-	-	✓
17	MOMFO Savsani and Tawhid (2017)	-	✓	-	-	-
18	MOGSA Zellagui et al. (2017)	-	✓	-	-	✓
19	MOGWO Jangir and Jangir (2018)	-	✓	-	-	✓
20	MOPSO Sun and Gao (2019)	-	✓	-	-	✓
21	MOBA Chen et al. (2019)	-	✓	-	-	✓
22	MOABC Mahmoodabadi and Shahangian (2019)	-	✓	-	-	✓
23	MOPSO Peng and Zhang (2008)	-	-	✓	-	✓
24	MOPSO Zapotecas Martínez and Coello Coello (2011)	-	-	✓	-	-
25	MOPSO Dai et al. (2015)	-	-	✓	-	-
26	MOABC Bai and Liu (2016)	-	-	✓	-	-

(continued)

No.	Algorithm Reference	MOO Approach: Scalarization	Pareto Dominance	Decomposition	Indicator-based	Archive
28	MOPSO García et al. (2014)	-	-	-	✓	✓
29	MOPSO Díaz-Manríquez et al. (2016)	-	-	-	✓	✓
30	MOPSO Wei et al. (2018)	-	-	-	✓	✓
31	MOPSO Sierra and Coello (2005)	-	✓	✓	-	✓
32	MOPSO Al Moubayed et al. (2014)	-	✓	✓	-	✓
33	MOPSO Lin et al. (2015)	-	✓	✓	-	✓
34	MOPSO Wei et al. (2017)	-	✓	✓	-	✓
35	MOABC Luo et al. (2017)	-	✓	-	✓	✓
36	MOPSO Luo et al. (2019)	-	✓	-	✓	✓
37	MOPSO Li et al. (2015)	✓	-	✓	✓	-
38	MOPSO Liu et al. (2019)	-	✓	✓	✓	✓
Total number of usage for each approach		4/38	26/38	11/38	7/38	28/38

It can be concluded that most of the previous MOSI algorithms (19 out of 38) have been developed according to the Pareto dominance approach. This is due to its ability to find a potentially effective set of non-dominated solutions. The non-dominated sorting approach and crowding distance (Sierra & Coello, 2005) have been used with the

Pareto dominance approach in numerous algorithms in maintaining the population diversity and selecting the non-dominated solutions. On the other hand, five out of 40 MOSI algorithms have been developed according to the decomposition approach. This small number of studies is due to the difficulties in determining the weight vector and limitations of the aggregation functions.

From this review, very few studies (3 out of 38) have been developed based on the scalarization and indicator approaches. This is because the weighted sum method that has been used in the scalarization-based algorithms cannot provide efficient performance in solving complex and non-convex problems (Brück et al., 2018; Jakob & Blume, 2014). Furthermore, the indicator approach is relatively new as compared to scalarization, Pareto dominance, and decomposition approaches. Except for PSO and ABC algorithms, none of the present MOSI algorithms is developed according to the indicator-based approach. Most of the indicator-based MOPSO algorithms are developed based on the R2 indicator. This is due to the high computational complexity of hypervolume and the other indicators, such as generational distance and inverted generational distance (Coello & Cortés, 2005); their performance depends on the reference set (Ishibuchi et al., 2017).

Studies are moving toward the usage of Pareto dominance and combined approaches. Furthermore, most of the reviewed MOSI algorithms (28 out of 38) use an external archive to save the obtained non-dominated solutions. During the optimization process, the solutions in the archive are updated at each iteration. This is achieved by generating new solutions and comparing them, one by one, with solutions in the archive. The new solution that dominates solutions in the archive will join the archive and the dominated solutions will be eliminated. The external archive technique has a limitation of high cost in terms of computation especially for large archives. Furthermore, the population of archives are often filled with many similar solutions (Coello et al., 2009).

Most of the proposed MOSI algorithms have been developed without incorporating the decision-maker's (user) preferences into the algorithms. However, in a real situation, the decision-maker is interested in one solution, and not the whole Pareto front set. Thus, such incorporation helps in improving optimization efficiency, in terms of effectively finding the most satisfactory solutions and reducing computational cost.

Benchmark MOPs with different features have been widely used in the literature to evaluate the performance of MOO algorithms. These benchmark MOPs include test functions and real-world problems. Test functions are normally used in the literature to validate the performance of a MOO algorithm or to compare two or more algorithms. In comparison to real-world problems, test functions have advantages whereby if their true Pareto front is known, their difficulty degree can be controlled, and in most problems, the number of objectives and decision variables can also be controlled (Tanabe & Ishibuchi, 2020). Several test problems have been used in the literature over the years (Deb, Thiele et al., 2002; Huband et al., 2006; Zhang et al., 2008; Zitzler et al., 2000). Real-world problems have been used by many researchers to evaluate the performance of optimization algorithms. Most real-world problems in the continuous domain are the engineering problems (Stewart et al., 2008; Tanabe & Ishibuchi, 2020).

In the area of MOO, several metrics have been proposed to evaluate the performance of MOO algorithms. In general, these performance metrics are used to measure two criteria, namely the convergence and diversity of non-dominated solutions (Mohammadi et al., 2013). These metrics include but are not restricted to generational distance, epsilon (Zitzler et al., 2003), inverted generational distance, hypervolume, spread (Custódio et al., 2011), maximum spread, and spacing (Mirjalili et al., 2016). Table 3 shows the benchmark MOPs used in the reviewed publications, the number of objectives of the problems, and the performance metrics that were used to evaluate the performance of the proposed MOSI algorithms.

Table 3

Benchmark Mops, Number of Objectives, and Performance Metrics Used in Mops

No.	Algorithm Reference	Test Problem	Real-World MOP	Number of Objectives	Performance Metrics Used
1	MOPSO Coello and Lechuga (2002)	✓	-	2	Maximum spread
2	MOPSO Coello et al. (2004)	✓	-	2	Spacing, generational distance, error ratio
3	MOPSO Sierra and Coello (2005)	✓	-	2, 3	Success counting inverted generational distance, two Set coverage, hypervolume
4	MOPSO Janga Reddy and Nagesh Kumar (2007)	✓	✓	2	Set coverage metric, generational distance, spread
5	MOPSO Peng and Zhang (2008)	✓	-	2	Inverted generational distance
6	MOGSA Hassanzadeh and Rouhani (2010)	✓	-	2	Spacing, generational distance
7	MOPSO Zapotecas Martínez and Coello Coello (2011)	✓	-	2, 3	Hypervolume, spacing, two set coverage
8	MOBA Yang (2012)	✓	✓	2	Distance
9	MOABC Akbari et al. (2012)	✓		2, 3	Inverted generational distance
10	MOBFO Niu et al. (2013)	✓	-	2	Diversity, generational distance

(continued)

No.	Algorithm Reference	Test Problem	Real-World MOP	Number of Objectives	Performance Metrics Used
11	MOFA Yang (2013)	✓	✓	2	Distance
12	MOPSO García et al. (2014)	✓	-	2, 3	Spread, inverted generational distance, hypervolume
13	MOPSO Al Moubayed et al. (2014)	✓	-	2, 3	inverted generational distance, hypervolume, epsilon
14	MOPSO (Li et al., 2015)	✓	-	2, 3	generational distance, inverted generational distance
15	MOPSO Lin et al. (2015)	✓	-	2, 3	Inverted generational distance
16	MOPSO Dai et al. (2015)	✓	-	2, 3	Generational distance, inverted generational distance, hypervolume
17	MOPSO Man-Im et al. (2015)	-	✓	2	-
18	MOGSA Bhowmik and Chakraborty (2015)	-	✓	2	-
19	MOBFO Yang and Ji (2016)	✓	-	2	Spacing, generational distance
20	MOBA Prakash et al. (2016)	✓	✓	2, 3	Generational distance, hypervolume, spacing
21	MOGWO Mirjalili et al. (2016)	✓	-	2, 3	Maximum spread, spacing, inverted generational distance

(continued)

No.	Algorithm Reference	Test Problem	Real-World MOP	Number of Objectives	Performance Metrics Used
22	MOGWO Mohamed et al. (2016)	-	✓	2	-
23	MOABC Kishor et al. (2016)	✓	✓	2	Inverted generational distance
24	MOPSO Díaz-Manríquez et al. (2016)	✓	-	2, 3	Hypervolume
25	MOABC Bai and Liu (2016)	✓	-	2, 3	Inverted generational distance, hypervolume, spread, epsilon
26	MOMFO Savsani and Tawhid (2017)	✓	✓	2	Generational distance, spacing, spread
27	MOGSA Zellagui et al. (2017)	-	✓	2	-
28	MOPSO Wei et al. (2017)	✓	-	2, 3	Inverted generational distance
29	MOABC Luo et al. (2017)	✓	-	2, 3, 5, 8	Hypervolume, inverted generational distance
30	MOGWO Jangir and Jangir (2018)	✓	✓	2	Generational distance, diversity
31	MOPSO Wei et al. (2018)	✓	-	2, 3	Inverted generational distance

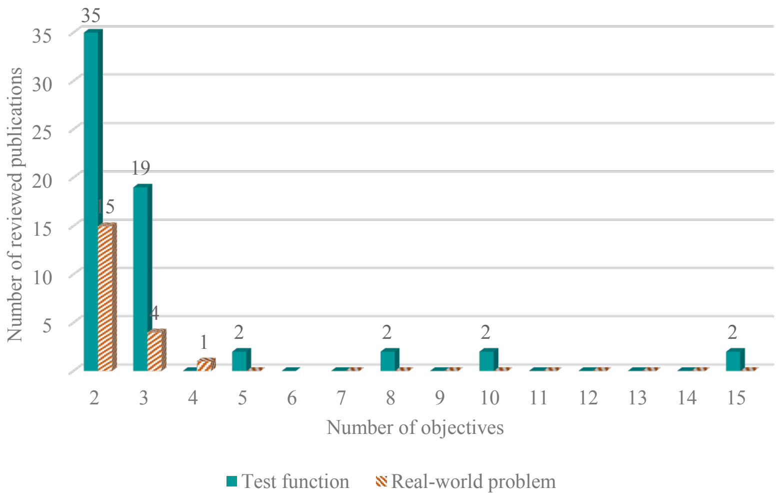
(continued)

No.	Algorithm Reference	Test Problem	Real-World MOP	Number of Objectives	Performance Metrics Used
32	MOPSO Sun and Gao (2019)	✓	-	2	Generational distance, diversity, maximum spread
33	MOBA Chen et al. (2019)	-	✓	2, 3	-
34	MOPSO Mellal and Zio (2019)	-	✓	4	-
35	MOABC Mahmoodabadi and Shahangian (2019)	✓	✓	2	-
36	MOPSO Liu et al. (2019)	✓	-	3, 5, 8, 10, 15	Inverted generational distance, hypervolume
37	MOPSO Luo et al. (2019)	✓	-	3, 5, 8, 10, 15	Inverted generational distance, hypervolume
38	MOMFO Sapre and Mini (2020)	✓	✓	2, 3	Inverted generational distance, spacing, maximum spread
Total number of usage for benchmark MOPs		32/38	15/38		

Most of the studies (23 out of 38) used only test functions in evaluating the performance of MOSI optimization algorithms, while six out of 38 studies used only real-world problems. On the other hand, nine out of 38 MOSI algorithms were evaluated by using both test functions and real-world problems. Most of the MOSI algorithms (18 out of 38) were evaluated by solving low-dimensional MOPs (with two objectives). Figure 3 shows the number of objectives for the benchmark MOPs that were used in evaluating the performance of the reviewed MOSI algorithms.

Figure 3

The Number of Objectives for Benchmark MOPs used in the Reviewed Publications.

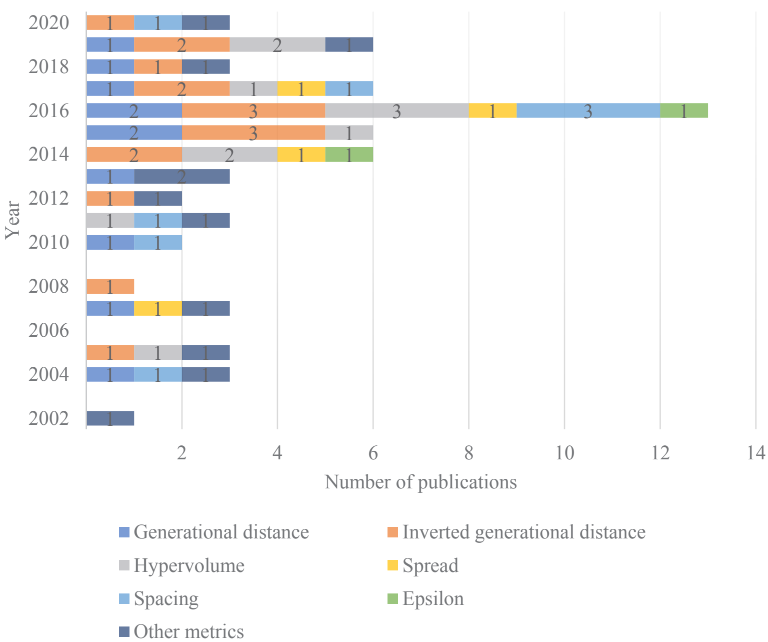


Most of the studies used test functions and real-world problems with two objectives to evaluate the performance of MOSI algorithms. However, in a real situation, an optimization problem may consist of more than two objectives. In this case, these algorithms need to be extended to deal with this type of problem. Several studies (17 out of 39) used MOPs with two and three objectives to evaluate the performance of algorithms. Several MOSI algorithms (4 out of 39) were evaluated by using high-dimensional MOPs (with more than three objectives).

Several performance metrics were applied to evaluate the MOSI algorithms. The most widely used were generational distance, inverted generational distance, hypervolume, spread, spacing, and epsilon metrics. The generational distance metric was used to measure the convergence toward the true Pareto front. However, this metric could not effectively measure the diversity of solutions. The inverted generational distance was utilized to measure both convergence and diversity (Riquelme et al., 2015). On the other hand, the spread and spacing metrics were used to measure the diversity of solutions. The hypervolume and epsilon metrics were employed to measure both convergence and diversity. Figure 4 shows the usage frequency of each performance metric.

Figure 4

Usage of the Performance Metrics.



The most used performance metric in measuring the performance of MOSI algorithms was inverted generational distance (17 out of 38 studies), followed by generational distance and hypervolume metrics (11 out of 38 studies). Compared to the inverted generational distance and generational distance metrics, the epsilon metric does not require a reference set, and it has been widely used in the area of MOO. Furthermore, the epsilon metric has a low computational complexity as compared to hypervolume, especially when dealing with high-dimension MOPs (Riquelme et al., 2015; Zitzler et al., 2003) However, in evaluating the performance of MOSI algorithms, it was used in several studies (2 out of 38). Thus, the usage of epsilon metric needs to be taken into consideration when evaluating the performance of MOSI algorithms. Both spread and spacing metrics were used in several studies (4 and 8 out of 38, respectively). The small number was because of the limitations of these metrics to measure the diversity of solutions and not the convergence. Furthermore, the spread metric was only useful when the Pareto front was composed of several solutions (Audet et al., 2018).

CONCLUSION

The MOSI metaheuristics have become popular MOO methods. Several approaches have been proposed to handle MOPs, namely scalarization-, Pareto-, decomposition-, and indicator-based. This paper provided a review for MOSI algorithms according to the MOO approaches. Most of the researchers focused on the Pareto dominance or decomposition approach in developing MOSI algorithms and they used an external archive to collect the obtained non-dominated solutions. The non-dominated sorting and crowding distance are widely used by many algorithms in selecting non-dominated solutions and maintaining the population diversity. For future work, it is possible to propose other MOSI algorithms by integrating an algorithm with other indicators such as inverted generational distance and generational distance, proposing different approaches to handle MOO, and using another method to preserve the population diversity. More MOSI algorithms need to be proposed to solve high-dimensional MOPs. In real-world applications, the user only needs one Pareto optimal solution and not the whole set as normally assumed by MOSI researchers. Thus, incorporating the preferences of a user into MOSI algorithms is very important to narrow the search and reduce the computational cost.

ACKNOWLEDGMENT

This work was supported by the Higher Education Ministry of Malaysia under the Fundamental Research Grant Scheme (FRGS/1/2017/ICT02/UUM/02/1) (S/O code 13794).

REFERENCES

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and*

Evolutionary Computation, *2*, 39–52. https://doi.org/10.1016/j.swevo.2011.08.001

Al Moubayed, N., Petrovski, A., & McCall, J. (2010). A novel smart multi-objective particle swarm optimisation using decomposition. In R. Schaefer, C. Cotta, J. Kołodziej, & G. Rudolph, *Parallel Problem Solving from Nature*, *PPSN XI*, (pp. 1–10). Springer. https://doi.org/10.1007/978-3-642-15871-1_1

Al Moubayed, N., Petrovski, A., & McCall, J. (2014). D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evolutionary Computation*, *22*(1), 47–77. https://doi.org/10.1162/evco_a_00104

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., & Salomon, L. (2018). Performance indicators in multiobjective optimization. *Optimization Online*. https://doi.org/10.1109/clei.2015.7360024

Bai, J., & Liu, H. (2016). Multi-objective artificial bee algorithm based on decomposition by PBI method. *Applied Intelligence*, *45*(4), 976–991. https://doi.org/10.1007/s10489-016-0787-x

Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In P. Dario, G. Sandini & P. Aebischer (Eds.), *Robots and Biological Systems: Towards a New Bionics?* (Vol. 102, pp. 703–712). Springer. https://doi.org/10.1007/978-3-642-58069-7_38

Bhowmik, A. R., & Chakraborty, A. K. (2015). Solution of optimal power flow using non dominated sorting multi objective opposition based gravitational search algorithm. *International Journal of Electrical Power & Energy Systems*, *64*, 1237–1250. https://doi.org/10.1016/j.ijepes.2014.09.015

Brück, A., Faßbender, S., & Waffenschmidt, E. (2018). Single- and multi-objective parameter optimization in a tool for designing PV-diesel-battery systems. In *2018 7th International Energy and Sustainability Conference (IESC)* (pp. 1–5). IEEE. https://doi.org/10.1109/iesc.2018.8439998

Chen, G., Qian, J., Zhang, Z., & Sun, Z. (2019). Multi-objective improved bat algorithm for optimizing fuel cost, emission and active power loss in power system. *IAENG International Journal of Computer Science*, *46*(1), 118–133. https://doi.org/10.1504/ijbic.2011.042259

Coello, C. A. C., Brambila, S. G., Gamboa, J. F., Tapia, M. G. C., & Gómez, R. H. (2019). Evolutionary multiobjective optimization: Open research areas and some challenges lying ahead. *Complex & Intelligent Systems*, 1–16. https://doi.org/10.1007/s40747-019-0113-4

Coello, C. A. C., & Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, *6*(2), 163–190. https://doi.org/10.1007/s10710-005-6164-x

Coello, C. A. C., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002)*, *2*, (pp. 1051–1056). https://doi.org/10.1109/cec.2002.1004388

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *8*(3), 256–279. https://doi.org/10.1109/tevc.2004.826067

Coello, C. C., Dehuri, S., & Ghosh, S. (2009). *Swarm intelligence for multi-objective problems in data mining* (Vol. 242). Springer. https://doi.org/10.1007/978-3-642-03625-5

Custódio, A. L., Madeira, J. A., Vaz, A. I. F., & Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, *21*(3), 1109–1140.

Dai, C., Wang, Y., & Ye, M. (2015). A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences*, *325*, 541–557. https://doi.org/10.1016/j.ins.2015.07.018

Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: An introduction. In L. Wang, A. H. C. Ng, & K. Deb (Eds.), *Multi-objective evolutionary optimisation for product design and manufacturing* (pp. 3–34). Springer London. https://doi.org/10.1007/978-0-85729-652-8_1

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, *1* (pp. 825–830). IEEE. https://doi.org/10.1109/cec.2002.1007032

Dede, T., Grzywiński, M., & Venkata Rao, R. (2020). Jaya: A new meta-heuristic algorithm for the optimization of braced dome structures. In *Advanced Engineering Optimization Through Intelligent Techniques*, Singapore (pp. 13–20). Springer. https://doi.org/10.1007/978-981-13-8196-6_2

Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello, C. A. C., & Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, *48*, 220–250. https://doi.org/10.1016/j.swevo.2019.04.008

Díaz-Manríquez, A., Toscano, G., Barron-Zambrano, J. H., & Tello-Leal, E. (2016). R2-based multi/many-objective particle swarm optimization. *Computational Intelligence and Neuroscience*, *2016*. https://doi.org/10.1155/2016/1898527

Emmerich, M. T., & Deutz, A. H. (2018). A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. *Natural Computing*, *17*(3), 585–609. https://doi.org/10.1007/s11047-018-9685-y

García, I. C., Coello, C. A. C., & Arias-Montaño, A. (2014). Mopsohv: A new hypervolume-based multi-objective particle swarm optimizer. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 266–273). IEEE. https://doi.org/10.1109/cec.2014.6900540

Hassanzadeh, H. R., & Rouhani, M. (2010, 28–30 July 2010). A multi-objective gravitational search algorithm. In *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks* (pp. 7–12). https://doi.org/10.1109/cicsyn.2010.32

Hernández-Díaz, A. G., Santana-Quintero, L. V., Coello Coello, C. A., & Molina, J. (2007). Pareto-adaptive ε-dominance. *Evolutionary Computation*, *15*(4), 493–517. https://doi.org/10.1162/evco.2007.15.4.493

Huang, V. L., Suganthan, P. N., & Liang, J. J. (2006). Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. *International Journal of Intelligent Systems*, *21*(2), 209–226. https://doi.org/10.1002/int.20128

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, *10*(5), 477–506. https://doi.org/10.1109/tevc.2005.861417

Ishibuchi, H., Setoguchi, Y., Masuda, H., & Nojima, Y. (2017). Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, *21*(2), 169–190. https://doi.org/10.1109/tevc.2016.2587749

Jacobs, D. S., & Bastian, A. (2017). *Predator-prey interactions: Co-evolution between bats and their prey*. Springer. https://doi.org/10.1007/978-3-319-32492-0

Jakob, W., & Blume, C. (2014). Pareto optimization or cascaded weighted sum: A comparison of concepts. *Algorithms*, *7*(1), 166–185. https://doi.org/10.3390/a7010166

Janga Reddy, M., & Nagesh Kumar, D. (2007). An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. *Engineering Optimization*, *39*(1), 49–68. https://doi.org/10.1080/03052150600930493

Jangir, P., & Jangir, N. (2018). A new non-dominated sorting grey wolf optimizer (NS-GWO) algorithm: Development and application to solve engineering designs and economic constrained emission dispatch problem with integration of wind power. *Engineering Applications of Artificial Intelligence*, *72*, 449–467. https://doi.org/10.1016/j.engappai.2018.04.018

Karaboga, D., & Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In *International Fuzzy Systems Association World Congress* (pp. 789–798). Springer. https://doi.org/10.1007/978-3-540-72950-1_77

Kennedy, J., & Eberhart, R. (1995, 27 Nov.–1 Dec. 1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN'1995)*, *4* (Vol. 1994, pp. 1942–1948). https://doi.org/10.1109/icnn.1995.488968

Kishor, A., Singh, P. K., & Prakash, J. (2016). NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering. *Neurocomputing*, *216*, 514–533. https://doi.org/10.1016/j.neucom.2016.08.003

Knowles, J. D., & Corne, D. W. (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, *8*(2), 149–172. https://doi.org/10.1162/106365600568167

Kumawat, I. R., Nanda, S. J., & Maddila, R. K. (2017). Multi-objective whale optimization. In *TENCON 2017 - 2017 IEEE Region 10 Conference* (pp. 2747–2752). IEEE. https://doi.org/10.1109/tencon.2017.8228329

Li, C. (2019). A fuzzy multi-objective linear programming with interval-typed triangular fuzzy numbers. *Open Mathematics*, *17*(1), 607–626. https://doi.org/10.1515/math-2019-0048

Li, F., Liu, J., Tan, S., & Yu, X. (2015). R2-MOPSO: A multi-objective particle swarm optimizer based on R2-indicator and decomposition. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3148–3155). IEEE. https://doi.org/10.1109/cec.2015.7257282

Li, J.-q., Han, Y.-q., Duan, P.-y., Han, Y.-y., Niu, B., Li, C.-d., Zheng, Z.-x., & Liu, Y.-p. (2020). Meta-heuristic algorithm for solving vehicle routing problems with time windows and synchronized visit constraints in prefabricated systems. *Journal of Cleaner Production*, *250*, 119464. https://doi.org/10.1016/j.jclepro.2019.119464

Li, K., Wang, R., Zhang, T., & Ishibuchi, H. (2018). Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access*, *6*, 26194–26214. https://doi.org/10.1109/access.2018.2832181

Li, X. (2003). A non-dominated sorting particle swarm optimizer for multiobjective optimization. In E. Cantú-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, & J. Miller, *Genetic and Evolutionary Computation — GECCO 2003 Genetic and Evolutionary Computation Conference*, Chicago, IL, USA (pp. 37–48). Springer. https://doi.org/10.1007/3-540-45105-6_4

Lin, Q., Li, J., Du, Z., Chen, J., & Ming, Z. (2015). A novel multi-objective particle swarm optimization with multiple search strategies. *European Journal of Operational Research*, *247*(3), 732–744. https://doi.org/10.1016/j.ejor.2015.06.071

Liu, J., Li, F., Kong, X., & Huang, P. (2019). Handling many-objective optimisation problems with R2 indicator and decomposition-based particle swarm optimiser. *International Journal of Systems Science*, *50*(2), 320–336. https://doi.org/10.1080/00207721.2018.1552765

Lones, M. A. (2020). Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms. *SN Computer Science*, *1*(1), 49. https://doi.org/10.1007/s42979-019-0050-8

Luo, J., Huang, X., Li, X., & Gao, K. (2019). A novel particle swarm optimizer for many-objective optimization. In *2019 IEEE*

Congress on Evolutionary Computation (CEC) (pp. 958–965). IEEE. https://doi.org/10.1109/cec.2019.8790343

Luo, J., Liu, Q., Yang, Y., Li, X., Chen, M.-r., & Cao, W. (2017). An artificial bee colony algorithm for multi-objective optimisation. *Applied Soft Computing*, *50*, 235–251. https://doi.org/10.1016/j.asoc.2016.11.014

Mahmoodabadi, M. J., & Shahangian, M. M. (2019). A new multi-objective artificial bee colony algorithm for optimal adaptive robust controller design. *IETE Journal of Research*, 1–14. https://doi.org/10.1080/03772063.2019.1644211

Man-Im, A., Ongsakul, W., Singh, J., & Boonchuay, C. (2015). Multi-objective optimal power flow using stochastic weight trade-off chaotic NSPSO. In *2015 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA)* (pp. 1–8). IEEE. https://doi.org/10.1109/isgt-asia.2015.7387120

Marler, R. T., & Arora, J. S. (2010). The weighted sum method for multi-objective optimization: New insights. *Structural and Multidisciplinary Optimization*, *41*(6), 853–862. https://doi.org/10.1007/s00158-009-0460-7

Mellal, M. A., & Zio, E. (2019). An adaptive particle swarm optimization method for multi-objective system reliability optimization. *Journal of Risk and Reliability*, *233*(6), 990–1001. https://doi.org/10.1177/1748006X19852814

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, *89*, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. D. S. (2016). Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, *47*, 106–119. https://doi.org/10.1016/j.eswa.2015.10.039

Mohamed, A.-A. A., El-Gaafary, A. A., Mohamed, Y. S., & Hemeida, A. M. (2016). Multi-objective modified grey wolf optimizer for optimal power flow. In *2016 Eighteenth International Middle East Power Systems Conference (MEPCON)* (pp. 982–990). IEEE. https://doi.org/10.1109/mepcon.2016.7837016

Mohammadi, A., Omidvar, M. N., & Li, X. (2013). A new performance metric for user-preference based multi-objective evolutionary algorithms. In *2013 IEEE Congress on Evolutionary*

Computation (pp. 2825–2832). IEEE. https://doi.org/10.1109/cec.2013.6557912

Mohammadi, A., Omidvar, M. N., Li, X., & Deb, K. (2015). Sensitivity analysis of penalty-based boundary intersection on aggregation-based EMO algorithms. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2891–2898). https://doi.org/10.1109/cec.2015.7257248

Niu, B., Wang, H., Wang, J., & Tan, L. (2013). Multi-objective bacterial foraging optimization. *Neurocomputing*, *116*, 336–345. https://doi.org/10.1016/j.neucom.2012.01.044

Niu, Y., & Shen, L. (2007). The optimal multi-objective optimization using PSO in blind color image fusion. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)* (pp. 970–975). https://doi.org/10.1109/mue.2007.204

Ochoa, G., Harvey, I., & Buxton, H. (2000). Optimal mutation rates and selection pressure in genetic algorithms. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation* (pp. 315–322). Morgan Kaufmann Publishers Inc.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, *22*(3), 52–67. https://doi.org/10.1109/mcs.2002.1004010

Peng, W., & Zhang, Q. (2008). A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *IEEE International Conference on Granular Computing (GRC)* (pp. 534–537). IEEE. https://doi.org/10.1109/grc.2008.4664724

Prakash, S., Trivedi, V., & Ramteke, M. (2016). An elitist non-dominated sorting bat algorithm NSBAT-II for multi-objective optimization of phthalic anhydride reactor. *International Journal of System Assurance Engineering and Management*, *7*(3), 299–315. https://doi.org/10.1007/s13198-016-0467-6

Ramirez, J. M., Medina, M. A., & Coello, C. A. C. (2018). A multiobjective teaching-learning algorithm for power losses reduction in power systems. In *Classical and Recent Aspects of Power System Optimization* (pp. 505–542). Elsevier. https://doi.org/10.1016/B978-0-12-812441-3.00018-5

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

Riquelme, N., Von Lücken, C., & Baran, B. (2015). Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)*, Arequipa, Peru (pp. 1–11). IEEE. https://doi.org/10.1109/clei.2015.7360024

Sapre, S., & Mini, S. (2020). Moth flame optimization algorithm based on decomposition for placement of relay nodes in WSNs. *Wireless Networks*, *26*(2), 1473–1492. https://doi.org/10.1007/s11276-019-02213-1

Savsani, V., & Tawhid, M. A. (2017). Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Engineering Applications of Artificial Intelligence*, *63*, 20–32. https://doi.org/10.1016/j.engappai.2017.04.018

Sayed, G. I., Darwish, A., & Hassanien, A. E. (2018). A new chaotic multi-verse optimization algorithm for solving engineering optimization problems. *Journal of Experimental & Theoretical Artificial Intelligence*, *30*(2), 293–317. https://doi.org/10.1080/0952813x.2018.1430858

Sierra, M. R., & Coello, C. A. C. (2005). Improving PSO-based multi-objective optimization using crowding, mutation and $\in$-dominance. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 505–519). Springer. https://doi.org/10.1007/978-3-540-31880-4_35

Singh, S. K., & Goh, M. (2019). Multi-objective mixed integer programming and an application in a pharmaceutical supply chain. *International Journal of Production Research*, *57*(4), 1214–1237. https://doi.org/10.1080/00207543.2018.1504172

Sörensen, K., Sevaux, M., & Glover, F. (2018). A history of metaheuristics. In R. Martí, P. M. Pardalos, & M. G. C. Resende (Eds.), *Handbook of Heuristics* (pp. 791–808). Springer International Publishing. https://doi.org/10.1007/978-3-319-07124-4_4

Stanger-Hall, K. F., Lloyd, J. E., & Hillis, D. M. (2007). Phylogeny of North American fireflies (Coleoptera: Lampyridae): Implications for the evolution of light signals. *Molecular Phylogenetics and Evolution*, *45*(1), 33–49. https://doi.org/10.1016/j.ympev.2007.05.013

Stewart, T., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., Jin, Y., Nakayama, H., Poles, S., & Di Stefano, D. (2008). Real-world applications of multiobjective optimization. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (pp. 285–327). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-88908-3_11

Stojanović, I., Brajević, I., Stanimirović, P. S., Kazakovtsev, L. A., & Zdravev, Z. (2017). Application of heuristic and metaheuristic algorithms in solving constrained weber problem with feasible region bounded by arcs. *Mathematical Problems in Engineering*, *2017*. https://doi.org/10.1155/2017/8306732

Sun, Y., & Gao, Y. (2019). A multi-objective particle swarm optimization algorithm based on gaussian mutation and an improved learning strategy. *Mathematics*, *7*(2), 148. https://doi.org/10.3390/math7020148

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation* (Vol. 74). John Wiley & Sons.

Tamura, K., & Gallagher, M. (2019). Quantitative measure of nonconvexity for black-box continuous functions. *Information Sciences*, *476*, 64–82. https://doi.org/10.1016/j.ins.2018.10.009

Tan, Y., Lu, X., Liu, Y., Wang, Q., & Zhang, H. (2019). Decomposition-based multiobjective optimization with invasive weed colonies. *Mathematical Problems in Engineering*, *2019*. https://doi.org/10.1155/2019/6943921

Tanabe, R., & Ishibuchi, H. (2020). An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 106078. https://doi.org/10.1016/j.asoc.2020.106078

Tsai, C.-W., Chiang, M.-C., Ksentini, A., & Chen, M. (2016). Metaheuristic algorithms for healthcare: Open issues and challenges. *Computers & Electrical Engineering*, *53*, 421–434. https://doi.org/10.1016/j.compeleceng.2016.03.005

Vachhani, V. L., Dabhi, V. K., & Prajapati, H. B. (2016). Improving NSGA-II for solving multi objective function optimization problems. In *2016 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1–6). IEEE. https://doi.org/10.1109/iccci.2016.7479921

Wei, L.-X., Li, X., Fan, R., Sun, H., & Hu, Z.-Y. (2018). A hybrid multiobjective particle swarm optimization algorithm based on R2 indicator. *IEEE Access*, *6*, 14710–14721. https://doi.org/10.1109/access.2018.2812701

Wei, L., Fan, R., & Li, X. (2017). A novel multi-objective decomposition particle swarm optimization based on comprehensive learning strategy. In *2017 36th Chinese Control Conference (CCC)* (pp. 2761–2766). https://doi.org/10.23919/chicc.2017.8027783

Weiszer, M., Chen, J., Stewart, P., & Zhang, X. (2018). Preference-based evolutionary algorithm for airport surface operations. *Transportation Research Part C: Emerging Technologies*, *91*, 296–316. https://doi.org/10.1016/j.trc.2018.04.008

Yang, C., & Ji, J. (2016). Multiobjective bacterial foraging optimization using archive strategy. In *5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016)* (pp. 185–192). https://doi.org/10.5220/0005668601850192

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International Symposium on Stochastic Algorithms: Foundations and Applications, SAGA 2009*, *5792* (pp. 169–178). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04944-6_14

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Nature inspired cooperative strategies for optimization* (pp. 65–74). Springer. https://doi.org/10.1007/978-3-642-12538-6_6

Yang, X. S. (2012). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, *3*(5), 267–274. https://doi.org/10.1504/ijbic.2011.042259

Yang, X. S. (2013). Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, *29*(2), 175–184. https://doi.org/10.1007/s00366-012-0254-1

Zapotecas Martínez, S., & Coello Coello, C. A. (2011). A multi-objective particle swarm optimizer based on decomposition. In *13th Annual Conference on Genetic and Evolutionary Computation* (pp. 69–76). ACM. https://doi.org/10.1145/2001576.2001587

Zellagui, M., Hassan, H. A., & Abdelaziz, A. Y. (2017). Non-dominated sorting gravitational search algorithm for multi-objective optimization of power transformer design. *Engineering Review*, *37*(1), 27–37.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731. https://doi.org/10.1109/tevc.2007.892759

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P., Liu, W., & Tiwari, S. (2008). *Multiobjective optimization test instances for the CEC 2009 special session and competition*. https://www3.ntu.edu.sg/home/epnsugan/index_files/CEC09-MOEA/CEC09-MOEA.htm

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, *8*(2), 173–195. https://doi.org/10.1162/106365600568202

Zitzler, E., Knowles, J., & Thiele, L. (2008). Quality assessment of pareto set approximations. In *Multiobjective Optimization* (pp. 373–404). Springer. https://doi.org/10.1007/978-3-540-88908-3_14

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, & H.-P. Schwefel, *Parallel Problem Solving from Nature - PPSN VIII International Conference On Parallel Problem Solving From Nature* (pp. 832–842). Berlin, Heidelberg. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30217-9_84

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271. https://doi.org/10.1109/4235.797969

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132. https://doi.org/10.1109/tevc.2003.810758