Multi-Objective Volleyball Premier League Algorithm

Reza Moghdani
Computational Intelligence and Intelligent Optimization Research Group (CIIORG), Persian Gulf University, Bushehr, Iran
Tel: (+98) 9177778337
reza.moghdani@gmail.com

Khodakaram Salimifard (corresponding author)
Computational Intelligence and Intelligent Optimization Research Group (CIIORG), Persian Gulf University, Bushehr, Iran
Tel: (+98) 9173783899
Office Tel: (+98) 77 3122 2218
salimifard@pgu.ac.ir

Emrah Demir
Cardiff Business School, Cardiff University, Cardiff, United Kingdom
Tel: (+44) 7774832127
demirE@cardiff.ac.uk

Abdelkader Benyettou
Département d'Informatique, USTOran, BP. 1505, El M'Naouer, Oran, Algeria
a_benyettou@yahoo.fr
Tel: (+21) 3552379949

Abstract

This paper proposes a novel optimization algorithm called the Multi-Objective Volleyball Premier League (MOVPL) algorithm for solving global optimization problems with multiple objective functions. The algorithm is inspired by the teams competing in a volleyball premier league. The strong point of this study lies in extending the multi-objective version of the Volleyball Premier League algorithm (VPL), which is recently used in such scientific researches, with incorporating the well-known approaches including archive set and leader selection strategy to obtain optimal solutions for a given problem with multiple contradicted objectives. To analyze the performance of the algorithm, ten multi-objective benchmark problems with complex objectives are solved and compared with two well-known multi-objective algorithms, namely Multi-Objective Particle Swarm Optimization (MOPSO) and Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D). Computational experiments highlight that the MOVPL outperforms the two state-of-the-art algorithms on multi-objective benchmark problems. In addition, the MOVPL algorithm has provided promising results on well-known engineering design optimization problems.

Keywords: Multi-Objective Evolutionary Algorithm; Global optimization; Pareto solution; Engineering design optimization problems.

1. Introduction

The Volleyball Premier League algorithm (VPL) [1] is an evolutionary algorithm in which a population of highly competitive teams is represented. The VPL algorithm attempts to solve global optimization problems by applying three volleyball metaphors, namely substitution, coaching, and learning. Like other metaheuristic algorithms, the VPL algorithm is initiated by creating random teams as initial solutions for a particular optimization problem. Each group in a solution contains specific possessions, including formation and substitutes. This algorithm applies the single-round robin (SRR) method to specify competitors during the iterations. To determine the winner of each game, the algorithm uses a power factor that is applied in a formulation to calculate the winning probability of each team. In the VPL algorithm, the coaching term is used with a knowledge sharing strategy to extract information from the game to train players and to substitute players during the match. Similar to any other Evolutionary Algorithms (EAs), VPL applies different neighborhood operators, such as repositioning and substitution strategies. These are used to alter the position of the current solution during the match based on their roles and match conditions for a better exploration and exploitation. In the VPL, each solution, which is called a team, is placed in the search space and calculated with regards to the predefined objective function.

The most challenging engineering design problems often consider multiple objectives with complicated, several linear and nonlinear constraints. In reality, conflicting objectives often regarded simultaneously, and optimal solutions may not be reachable even for small-sized instances [2]. There are many difficulties in resolving real-world problems that require specific tools to cope with them. In such cases, while there is more than one objective to be optimized, multi-objective algorithms (MOAs) come into play [3]. MOA has turned into a prevailing trend in recent years, and many powerful algorithms have been proposed to handle these problems [4]. Moreover, MO problems are mostly considered as NP-hard, which means there is no consensus on an exact algorithm, which can be used to solve that kind of problem. Meanwhile, it has been accepted among scholars that metaheuristic algorithms are compelling for such optimization problems, and there are several works that sincerely reviewed these methods [5-7].

There are two standard ways of handling multiple objectives, namely a priori and a posteriori [8, 9]. All information is needed before making any decision for a priori method, whereas a posteriori methods provide many Pareto optimal solutions to the decision-maker who will then select their preferred one [10]. These approaches include the global criterion method, goal programming, goal-attainment method, lexicographic method, min-max optimization, the weighting method, the weighting method with normalization, ϵ-constrained Method, hybrid method, and Pareto fronts [11]. Many studies have been conducted investigating and evaluating multi-objective evolutionary algorithms (MOEAs). Some of the most well-known stochastic optimization techniques include Multi-Objective Particle Swarm Optimization (MOPSO) [12], Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [13], Non-dominated Sorting Genetic Algorithm(NSGA) [14], Non-dominated Sorting Genetic Algorithm version two (NSGA-II) [14], Pareto Archived Evolution Strategy (PAES) [15], Pareto-frontier Differential Evolution (PDE) [16], Strength-Pareto Evolutionary Algorithm (SPEA) [17], Strength-Pareto Evolutionary Algorithm version two (SPEA-II) [18]. For more information regarding the MO techniques, interested readers are referred to the following studies [19-21]

This paper presents an optimization algorithm based on the artificial physical process for the MOO problem, namely Multi-Objective Volleyball Premier League (MOVPL) algorithm. To the best of our knowledge, only two metaheuristic algorithms presented by Kashan [22] Moosavian [23] have used sports metaphor for this purpose. The performance of MOEAs can be measured with their capability to estimate true Pareto optimal solutions of multi-objective problems. In this regard, a well-known theorem, named No Free Lunch (NFL) [24], declared that there is no unique method for solving optimization problems entirely. Considering this theorem, there is no guarantee that an optimizer algorithm has the same performance in the various class of problems. According to this fact, we develop the multi-objective version of a newcomer algorithm, which has interesting results in comparison with the state of art algorithms in the evolutionary computation context. Therefore, the contributions of this paper are multi-fold:

(i) a new multi-objective optimization algorithm has been proposed,

(ii) a new archive component has been incorporated into the algorithm to store non-dominated solutions,

(iii) top three teams, named rank1, rank2, and rank3, have been directed via a leader selection mechanism,

(iv) a new mechanism based on grid partitioning method has been developed to store the solutions in the archive,

(v) The MOVPL algorithm has been applied to both test and engineering design optimization problems and provides better solutions, compared to the ones in the literature.

This paper is organized as follows. Section 2 provides a brief overview of the recent developments in MOEAs. Section 3 defines the investigated problem along with its technical definitions used in Volleyball literature. This section also provides a mathematical formulation for the MOVPL algorithm. Section 4 introduces various test functions and performance metrics applied in this research. Computational results are presented in Section 5. Finally, the last section presents a brief summary and future research direction.

2. Literature review

This section provides brief background information related to the perceptions of the MOO problems and proposed methods in the context of MOEAs.

2.1 MO-based metaheuristic algorithms

MOEAs have been implemented in many particular problems, such as transit network design [25], vehicle routing problem [26], disassembly line balancing problem [27], and location-allocation problem [28]. Generally, the most forerunner method among MOEAs was derived from the most well-known evolutionary algorithm such as NSGA-II which was proposed in [29]. For example, Fonseca and Fleming [30] introduced a new type of MOEA named Multi-Objective Genetic Algorithm (MOGA), where the rank of any solution in the current population is computed based on the number of dominated individuals. Hence, all non-dominated individuals are assigned to the Pareto front, and population destiny is considered to penalize individuals, which are dominated by other solutions to determine their ranks. Having enhanced MOGA performance, its hybridization with neural networks was also proposed [31] in the literature.

In another study, Srinivas and Deb [14] represented one of the best-known MOEA approaches named the Non-dominated Sorting Genetic Algorithm (NSGA), which was initially proposed by Goldberg [29]. Unquestionably, the updated version of NSGA, NSGA-II, is still the most prevalent MOEA. The execution of NSGA-II contains choosing the structure of solution representation and the basic parameters of the algorithm [32]. Lately, Niched-Pareto Genetic Algorithm (NPGA) which is based on a tournament selection, was proposed [33], and accordingly, its new version, using Pareto ranking, NPGA 2, was introduced by Erickson et al. [34]. Later, Knowles and Corne [35] proposed MOEA, including a particular procedure that partitioned search space in a recursive manner. This approach called Pareto Archived Evolution Strategy (PAES) introduces a new diversity approach (i.e., histogram). In the work of Zitzler and Thiele [17], the Strength Pareto Evolutionary Algorithm (SPEA) is proposed. The SPEA has been suffered several shortcomings which were later revised by Zitzler et al., [36] and the second version of SPEA (SPEA-II) is developed.

Another MOEA variation with the GA algorithm (i.e., Multi-objective Messy Genetic Algorithm (MOMGA)) was presented by Veldhuizen and Lamont [37] as an effort to develop the messy GA [38]. A new version of MOMGA (known as MOMGA-II) [39] which was integrated with the fast-messy GA was later introduced by Zydallis et al., [40]. Recently, the MOMGA-III was introduced to enhance the exploitation of the search space [41]. In another study, the Pareto Envelope-based Selection Algorithm (PESA) was proposed by Corne et al., [42], which includes integrated small internal and sizeable external populations. In this algorithm, a hyper-grid division is used to keep diversity throughout the MOEA run. An updated version of PESA (PESA-II) [43] has been proposed with the difference in selection strategy.

We now briefly review other related metaheuristic algorithms, which are the most associated with MOVPL in terms of structures. Zabihi et al., [44] presented a new MOEA based on teaching–learning based optimization (TLBO), which is used a similar structure of NSGA-II to find optimal solutions. Wang et al., [45] proposed a multi-objective version of whale optimization algorithm, in which global grid ranking is used to enhance the performance presented MOEA. Pradhan and Panda [46] extended the new extension of cat swarm optimization (CSO) by applying an external archive to solve the MO problem. Sadollah et al., [47] presented a multi-objective water cycle algorithm (MOWCA), in which solutions are stored in an archive.

Got et al., [48] proposed a multi-objective algorithm by incorporating Pareto dominance and an external archive into a recently published algorithm, named Whale Optimization Algorithm [49], to deal with multiple objectives. Liu et al., [50] embedded the quantum approach into the PSO algorithm to the extended new multi-objective of this algorithm, named Multi-Objective Quantum-behaved Particle swarm optimization, to obtain promising results, in which cultural evolution mechanism was used to obtain high-quality results in Pareto optimal solutions. Zhang et al., [51] proposed a new method in developing a multi-objective approach, named Exploration/exploitation Maintenance multi-objective Evolutionary Algorithm″ (EMEA), in which various levels of the trade-off between exploration and exploitation has been balanced throughout the solving process. Cao et al., [52] presented a novel decomposition-based evolutionary dynamic multi-objective optimization using a different model (MOEA/D-DM), which is extended based on a centroid motion approach. Hultmann Ayala et al., [53] embedded Free search (FS) algorithm, inspired from animal behavioral, combined with differential evolution, named Multi Objective Fee Search based Differential Evolution (MOFSDE) to solve heat exchanger optimization problem. Zhang [54] presented a new multi-objective approach based on extending the immune optimization algorithm considering the interval number, named Micro Multi-objective Immune Optimization Algorithm (μMIOA). In this study, an uncertain programming model was used to deal with the uncertain environment of engineering problems, considering a non-dominated sorting approach. It has pointed out recently by [55] that the binary tree search procedure can be for solving multi-objective problems. In this study, a specific binary search, termed K-D tree, was combined to MOEA/D and used to tow operators including SelectRoot and SelectLeaf, to explore a higher level of neighborhoods in search space. Liang et al., [56] presented a multimodal multi objective Differential Evolution optimization algorithm (MMODE) in which diversity of obtained solutions were promoted thorough non-dominated sorting approach and crowding distance approach. Bora et al., [57] proposed new extension of NSGA-II by incorporating reinforcement techniques to solve multi-objective environmental/economic dispatch (EED) problem. Some recent studies have focused on specific real world application and presented extensive analysis to compare wide range of MOEAs. In this regard, one of the most engineering problem, wind turbine blade design, which is considered by [58] concerning varied range of MOEAs comprising NSGA-II, Quantum-inspired Multi-objective Evolutionary Algorithm (QMEA), MOEA/D, and Multi-objective Optimization Differential Evolution Algorithm (MODE).

The most well-known MOEAs are originated from GA, but many scholars turn their attention to hybrid MOEAs combining a specific local search technique within an MOEA. Table 1 shows the state of art MOEAs with various basic features where *EVOPS* denotes evolutionary operators including crossover ($\mathcal{C}$) and mutation ($\mathcal{M}$), *Fitness* is used to show the primary strategy of MOEAs to determine dominated solutions, and, the next column is used for real-value and binary representation, accordingly.

Table 1: The state of the are MOEAs and their basic features

MOEA	EVOPS	Fitness	Representation	Ref.
VEGA	$\mathcal{C}, \mathcal{M}$	Value of a single objective	$\{0,1\}$	[59]
M-PAES	$\mathcal{M}$	(1+1) single grid	$\mathbb{R}, \{0,1\}$	[60]
MOGA	$\mathcal{C}, \mathcal{M}$	Linear interpolation	$\mathbb{R}, \{0,1\}$	[30]
MOTLBO	$\mathcal{C}, \mathcal{M}$	Rank dominance	$\mathbb{R}, \{0,1\}$	[61]
MOCSO	$\mathcal{C}, \mathcal{M}$	Pareto ranking	$\mathbb{R}, \{0,1\}$	[46]
GENMOP	$\mathcal{C}, \mathcal{M}$	Pareto ranking	$\mathbb{R}$	[62]
MOGWO	$\mathcal{C}, \mathcal{M}$	Leader selection strategy	$\mathbb{R}$	[3]
MOPSO	$\mathcal{C}, \mathcal{M}$	Leader selection strategy	$\mathbb{R}$	[63]
PAES	$\mathcal{M}$	(1+1)single grid	$\mathbb{R}, \{0,1\}$	[35]
MOCS	$\mathcal{C}, \mathcal{M}$	Rank Dominance	$\mathbb{R}$	[2]
PESA	$\mathcal{C}, \mathcal{M}$	Pareto ranking	$\{0,1\}$	[64]
PESA-II	$\mathcal{C}, \mathcal{M}$	Region-based	$\{0,1\}$	[65]
MOGSA	$\mathcal{C}, \mathcal{M}$	Leader selection strategy	$\mathbb{R}$	[66]
NPGA	$\mathcal{C}, \mathcal{M}$	Tournament selection	$\mathbb{R}, \{0,1\}$	[67]
NPGA II	$\mathcal{C}, \mathcal{M}$	Rank dominance	$\mathbb{R}, \{0,1\}$	[32]
NSGA	$\mathcal{C}, \mathcal{M}$	Dummy fitness	$\mathbb{R}, \{0,1\}$	[14]
NSGA-II	$\mathcal{C}, \mathcal{M}$	Non-dominated sorting and crowding distance	$\mathbb{R}, \{0,1\}$	[68]
SPEA	$\mathcal{C}, \mathcal{M}$	Strength value based on dominance and clustering	$\{0,1\}$	[17]
SPEA2	$\mathcal{C}, \mathcal{M}$	Strength value based on dominance and clustering	$\mathbb{R}, \{0,1\}$	[69]
MOSGA	$\mathcal{C}, \mathcal{M}$	Linear interpolation	$\{0,1\}$	[70]
μGA	$\mathcal{C}, \mathcal{M}$	Pareto ranking	$\{0,1\}$	[71]
μGA2	$\mathcal{C}, \mathcal{M}$	Pareto ranking	$\mathbb{R}, \{0,1\}$	[72]
OMOEA	$\mathcal{C}$	Based on sub-niche evolution	$\mathbb{R}$	[73]
OMOEA-II	$\mathcal{C}$	Non-dominated sorting	$\mathbb{R}$	[74]
GPAWOA	$\mathcal{C}, \mathcal{M}$	Leader selection strategy	$\mathbb{R}$	[48]
MOQPSO	$\mathcal{C}, \mathcal{M}$	Leader selection strategy	$\mathbb{R}$	[50]
EMEA	$\mathcal{C}, \mathcal{M}$	Survivability-based Mechanism and Survival Length Indicator	$\mathbb{R}$	[51]
MOEA/D	$\mathcal{C}, \mathcal{M}$	centroid locations	$\mathbb{R}$	[52]
MOFSDE	$\mathcal{C}, \mathcal{M}$	Non-dominated sorting and crowding distance	$\mathbb{R}$	[53]
μMIOA	$\mathcal{C}, \mathcal{M}$	Non-dominated sorting approach	$\mathbb{R}$	[54]
KDT-MOEA	$\mathcal{C}, \mathcal{M}$	select root and select leaf operators	$\mathbb{R}$	[55]
MMODE	$\mathcal{C}, \mathcal{M}$	Non-dominated sorting and crowding distance	$\mathbb{R}$	[56]
NSGA-RL	$\mathcal{C}, \mathcal{M}$	Non-dominated sorting and crowding distance	$\mathbb{R}$	[57]

3. The Multi-Objective Volleyball Premier League Algorithm

This section provides the main features of the presented MOVPL algorithm along with its mathematical expressions. Please note that proposed MOEA is based on VPL including many steps summarized in this section. Interested readers maybe refer to [1] for more details.

The distinctive attribute of the algorithm is the solution structure, which is mostly different from other evolutionary algorithms. The structure of the MOVPL solution contains two divisions named active and passive parts. The first one, the active part, signifies the position of the team comprising six players, which are positioned in the court. The objective function of each group is evaluated concerning the first part of the solution. The second part, the passive part, embraces variable information, which is summoned with a unique inspiration rule, like substitution strategy. In the volleyball game, a substitute has placed the position of the player who is departing the court as the coach has ordered. Figure 1 shows the relationship between the special team structure and solution representation.

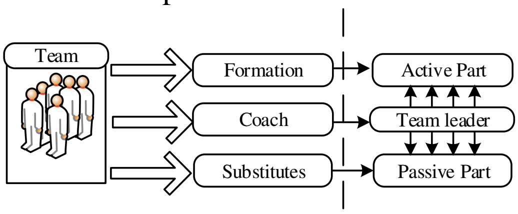


Figure 1: The solution representation structure of the proposed MOEA

As can be seen from Figure 1, the central part of the solution is affected by the team formation, whereas the second part of the solution is only affected by the substitutes.

In the standard volleyball league, several factors might have an influence on the results. To grasp an overall form of the implementation, we will now present underlying assumptions that can be used for the design of the algorithm. The main assumptions of the MOVPL are: (*i*) the result of the match is only known after the game; (*ii*) the term ‘‘team power” refers to the strength of a team in the league. It is used to highlight the point that it is more likely a better team can beat its rival; (*iii*) a team only respects to its upcoming game and does not ruminate any other games. Concerning the prior results, the coach defines the new line-up of the team, which is based on the analysis of the current situation of the team and the probability of winning in the upcoming match. And finally, (*iv*) once team i defeats team j, any strength aided team i to triumph is a weakness of team j to miss the match. That is to say, a weakness of a team is determined as the absence of its strength.

In the MOVPL algorithm, the term *league* is used as the *population* concept throughout this paper. What's more, the term *season number* is performed as the number of iteration, which is used in the main loop of the proposed algorithm, a team signifies a specific solution, and team i denotes the i^{th} member of the population. The term *week* indicates the league schedule. The following subsections will provide insights into the steps of the proposed algorithm.

To perform the MOVPL effectively, we use two components that were also used in the MOPSO algorithm proposed by Coello [75]. In the proposed algorithm, we use two parts to obtain Pareto solutions, which are an archive to keep non-dominated Pareto and leader selection strategy to choose the top three teams as the best teams of the learning process from the archive with the hope to move the algorithm toward the global optimum.

The archive of the proposed algorithm is quite similar to a simple storage unit that keeps non-dominated Pareto optimal solutions. The crucial element of the archive is an archive controller, which is used to determine whether the number of members is exceeded from the capacity of the archive.

The other component of the proposed algorithm, leader selection mechanism, is connected with the top three teams. These are also identified as rank 1, rank 2, and rank 3. As known in the multi-objective context, it is hard to compare solutions based on Pareto optimality to reach the best-obtained final Pareto front. Therefore, this component is used to cope with this problem. For this reason, teams are directed to the best solutions to find favorable search space by using the leader selection component, which selects a segment of search space with minimum crowding distance. In order to calculate time complexity, we use big O notation theory, where $\mathbb{N}$ denotes the number of individuals in the population and $\mathcal{M}$ expresses the number of objectives. Therefore, the complexity of MOVPL is $O(\mathcal{M}\mathbb{N}^2)$, which is identical to other acknowledged MOEAs, such as MOGWO[3], MOPSO[76], and NSGA-II [68].

Similar to other MOEAs, after initialization, all non-dominated teams are duplicated to the archive set which has two components (i.e., archive controller and the grid). At any iteration, each team's properties ($\mathcal{F}.\mathcal{S}$) and the best team information are updated. If any member of the archive dominates a solution, it will be added, and consequently dominated teams will be discarded from the archive. It is worthwhile mentioning here that regions dividing space of objective function may be changed based on the archive set, and the grid will be updated while a new solution was found at the outer side of current regions. The capacity of the archive is limited, and it will be checked throughout the solution process. The adaptive grid will be performed to achieve the Pareto fronts set. The steps of the MOVPL algorithm are iteratively repeated until the stop condition is reached. Algorithm 1 shows the pseudo-code of the proposed MOVPL algorithm.

Algorithm 1: Pseudocode of the MOVPL algorithm

Input: t (Generation)=0, parameters, cost function
Output: PF (Nondominated set)
Initialization
Record non-dominated teams in Archive
Generate the grid (hypercube)
While $t < T$
  Generate a league schedule
  For i=1: (N-1)×2
  Update Archive
  Best team =Select Best team (Archive)
  Apply Competition procedure between team A, and B
  Determine winner and loser teams
  Apply different strategies for winner and loser teams
  $X_i^g(t+1)_1$= Select Best team (Archive)
  Exclude $X_i^g(t+1)_1$ from the Archive
  $X_i^g(t+1)_2$= Select Best team (Archive)
  Exclude $X_i^g(t+1)_2$ from the Archive
  $X_i^g(t+1)_3$= Select Best team (Archive)
  Exclude $X_i^g(t+1)_3$ from the Archive
  For j=1: number of teams
    Update the position of the team(j)
  End
  Apply Promotion and relegation process
  Apply the season transfer process
  Update grid
  If the number of Archive members > $nArchive$
    Delete extra members
  End if
  $t = t + 1$
  End
End While

It is worthwhile mentioning that the convergence of an algorithm should be proven by an operator, which makes an individual solution to change its movement abruptly. Referring to [77], this manner arising from an operator has assured of convergence of the algorithm during the search. In the original VPL, some operators like b, defined in the learning phase and linearly decreased from β (a predefined constant) to 0, guarantees its convergence throughout the course of iterations. The convergence of MOVPL has proven due to the inheritance of all features of VPL, in which exploitation and exploration of search agents occur simultaneously.

3.1. Initialization stage

The MOVPL algorithm begins with the initialization of the teams representing initial solutions to the problem. As mentioned earlier, formation and substitutes are the main properties of each team. For the sake of simplicity throughout the paper, $\mathcal{F}$ and $\mathcal{S}$ are used as the notations of formation and substitute properties respectively, and $g = \{\mathcal{F}, \mathcal{S}\}$ represents either one in the appropriate formula. We consider N as the size of the initial population and terms $\mathcal{F}$ and $\mathcal{S}$ are assigned randomly between the lower lb_j and upper ub_jbound of each variable j using Eq.(1).

$$X_j^g = lb_j + r \times (ub_j - lb_j), \tag{1}$$

where $g = \{\mathcal{F}, \mathcal{S}\}$ and r represents a uniformly distributed random number between 0 and 1. The values of $F$and S are considered as the main properties of initial solutions, where the number of columns and rows specify the number of dimensions and the number of teams, respectively. The matrix G is representing that of $\mathcal{F}$ or that of $\mathcal{S}$, is defined according to the following Eq.(2)

$$G = \begin{bmatrix} X_{1.1}^g & X_{1.2}^g & \cdots & X_{1.j}^g \\ X_{2.1}^g & X_{2.2}^g & \dots & X_{2.j}^g \\ \vdots & \vdots & \ddots & \vdots \\ X_{i.1}^g & X_{i.2}^g & \dots & X_{i.j}^g \end{bmatrix}, \tag{2}$$

3.2. League schedule

Let N indicates the number of teams and each team will play $N - 1$ times, therefore $(N - 1)N/2$ games will be occurred during throughout the tournament. Suppose that we have eight teams, namely A, B, C, D, E, F, G, H in the league. We apply a specific method, single round robin (SRR), to generate the league's schedule randomly. The following example, including eight teams, is provided to show the implementation of the league schedule in our algorithm. In each round, every team will play to another team exactly once. In order to run SRR, a regular polygon that has $N - 1$ vertices is drawn so that each vertex (seven vertices) and the spot located in the center indicate a team. As seen in Figure 2, we draw horizontal lines and then join the vertex that has been left out to the center. Each red line signifies a match in which two teams located on the line are playing in the first round.

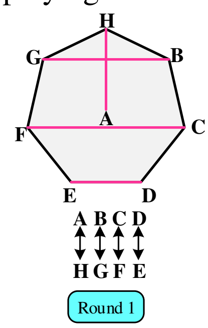


Figure 2: The first round of the SSR method

According to Figure 2, pairs $(A, H), (B, G), (C, F)$ and (C, F) play in the initial round. For the sake of implementing league schedule in the succeeding rounds, the figure is switched clockwise. This operation will carry on until the figure earnings to its original position.

3.3. Competition stage

We now recommend a mathematical equation expressing the probability of a winning team and its strength to win the match in the interest of determining the winning team in a competition. We believe that relationships among the power of groups stemmed from linear equations. Moreover, games among teams are assumed to be idealized, and there are no unforeseen elements that affect the results. Correspondingly, the following equation is given to gauge the strength of a team in a week.

$$\varphi(i) = \sum_{j=1}^{M} \frac{f(X_i^{\mathcal{F}})_j}{\sum_{i=1}^{N} f(X_i^{\mathcal{F}})_j}, \tag{3}$$

Where $\varphi(i)$ shows the power value of team i, and $f(X_i^{\mathcal{F}})_j$ is the value of solution $i$for jth objective function, which is computed using $\mathcal{F}$. Moreover, the denominator denotes the total summation of values in the current iteration for the jth objective function. Eq.(3) implies that the power index of team i is a function of its fitness value divided by all teams' fitness values. $\varphi(i)$ determines the weight of team i in the week where a better team has a higher φ. Suppose teams l and $k$are going to play in a match, with their $\mathcal{F}$ values, $X_l^{\mathcal{F}}$, and $X_k^{\mathcal{F}}$, respectively. The power indexes for both teams are defined as $\varphi(l)$ and $\varphi(k)$.

Let $p(l, k)$ denotes the probability of winning team l in competition with team k. Therefore, the following formula is given:

$$p(l, k) = \frac{\varphi(l)}{\varphi(l) + \varphi(k)}. \tag{4}$$

Since $p(l, k)$ expresses the probability of winning a match that can be obtained from uniformly distributed random number $r \in [0.1]$. If $r \leq p(l, k)$, team l can beat team l, otherwise, team k is the winner. Obviously, if $f(X_l^{\mathcal{F}})$ and $f(X_k^{\mathcal{F}})$are close to each other, $p(l, k)$ and $p(k, l)$ converge to 0.5. Afterward, the strategies for the winner and the loser teams are implemented to set new $\mathcal{F}$. In this manner, three strategies, namely, Knowledge sharing, repositioning, and leading role, is used for the wining team accordingly. Therefore, the procedure of competition is as outlined in Algorithm 2.

Algorithm 2: Competition between teams *i* and *j*

```
Function Competition (i, j)
Calculate φ(i) and φ(j) using Eq.,(3)
Calculate p(i, j) using Eq.(4)
Generate r ∈ [0, 1],
If r ≤ p(i, j)
     Team i is a winner, and team j is a loser,
Else
     Team j is a winner, and team i is a loser,
End if
Apply a winning strategy for the winning team,
Apply losing strategies for the loser team,
End
```

3.4. Knowledge sharing strategy

A coach has a significant impact on team performance and is responsible for coaching the team. During the match, the coaches continuously update both technical and tactical strategy of their teams, and they share his knowledge with players and substitutes, according to Eq.(5).

$$X_j^g(t+1) = X_j^g(t) + r\lambda^g(ub_j - lb_j), \tag{5}$$

where λ^g are coefficients of $\mathcal{F}$and $\mathcal{S}$, respectively. r is a generated random number that is uniformly distributed in the range [0-1]. Let δ_{ks} denotes the rate of knowledge sharing in each team, the number of knowledge sharing in each competition is shown by $N_{ks} = [M\delta_{ks}]$ where N_{ks} denote the number of knowledge sharing positions for each team used this strategy and M symbolizes the total number of positions in the team. The pseudo-code for this strategy is presented in Algorithm 3.

Algorithm 3: Knowledge-sharing strategy

For k=1 :N_{ks}
Select a position randomly
For j=1 to J
Update position j of $\mathcal{F}$ and that of $\mathcal{S}$ using Eq.(5),
End For
End For

3.5. Repositioning strategy

We assume that the coach tries to assign the best player for each position to obtain the best presentation in the match. Therefore, players can be allocated to various positions throughout a game, based on the coaching strategy. We named this procedure *repositioning strategy* where the coach alters the positions of active players to obtain the best performance. This strategy can be used for the substitution part as well.

Let δ_{rs} indicates the rate that a team uses the repositioning strategy during the game. $N_{rs} = [M\delta_{rs}]$ defines the number of repositioning strategies in any match. After choosing two positions i and j randomly, two variable A and $B$with two properties $(\mathcal{F}, \mathcal{S})$ are redefined, and then we consign properties of positions i and $j$to properties $(\mathcal{F}, \mathcal{S})$ of A and B, respectively.The following formulas are obtained.

$$A^g = X_i^g \text{ , } B^g = X_j^g. \quad (6)$$

Then, properties of variables A and B are consigned to properties of$j$ and i, respectively. Therefore, we get the following formulas:

$$X_i^g = B^g \text{ , } X_j^g = A^g. \quad (7)$$

The pseudo-code of the repositioning strategy is expressed in Algorithm 4.

Algorithm 4: The repositioning process

For$k$=1 to N_{rs}
Select randomly two members (i, j) of team k
Define two variables A and B
Use Eq. (6)
Reverse two positions i and j using Eq. (7)
End

3.6. Substitution strategy

Throughout a match, a player may be swapped with another player sitting on a substitution bench. Here, different $\mathcal{F}$ could be used to change the player position in this strategy.

The main goal of performing the substitution process is to achieve a better search process in the algorithm. In the classic version of the volleyball game, no strict regulations were limiting the number of players coming into the game to substitute for other players. Whereas, nowadays, teams are restricted in the number of substitutions which they can make in a typical match. We assume that the original version of volleyball roles in the algorithm, and the number of substitutions is expected to be free. Let r represents a random number uniformly distributed between zero and one. The following equation calculates the number of substitutions $N_s = [rM]$ for a team in the competition.

In this stage, the losing team is selected randomly, and then, sets F and S are delineated containing selected players and substitutions, and accordingly, all members of sets F and $S$are swapped randomly. The pseudo-code of the substitutions process is presented in Algorithm 5.

Algorithm 5: The substitution process

Compute N_s : a number of the substitution process
Define Sets h,F, and S
For k=1 to N_s
$X_{h(k)}^{\mathcal{F}} = S(k)$
$X_{h(k)}^{\mathcal{S}} = F(k)$
End

3.7. Winner strategy

To apply this strategy to winner teams, the position of a solution is given by its position $X^g(t)$, the best team $X^g(t)^*$, the inertia weight ψ^g, and the set $g = \{\mathcal{F}, \mathcal{S}\}$. The next formula is presented to compute the winning strategy.

$$X^g(t+1) = X^g(t) + r\psi^g(X^g(t)^* - X^g(t)), \tag{8}$$

where ψ^g denotes the inertia weight of $\mathcal{F}$ and $\mathcal{S}$, and r represents a random number between zero and one.

3.8. Learning phase

In this stage, the best team is considered as the Φ_1, accordingly, the teams with rank 2 and rank 3 are termed as $\Phi 2$ and $\Phi 3$, respectively. In the MOVPL, any new solution is directed by $\Phi 1, \Phi 2,$ and $\Phi 3$ which are considered as leaders for the other teams of the league. At the beginning of this stage, we set constant value β. The following equations are given to model the learning phase of the algorithm:

$$\theta = dbr_1 - b, \tag{9}$$

$$\vartheta = dr_2, \tag{10}$$

Where θ and ϑ denote coefficient values, d is equal to β,r_1 and r_2 are generated random numbers in the range [0-1], and b is linearly lessened from β to 0. Therefore, b is calculated using the following formula:

$$b = \beta - (t(\beta/T)), \tag{11}$$

where t symbolizes the present iteration, and T represents the total number of iterations in the proposed algorithm. It is worth mentioning that the value of b has a great impact on the balance between exploration and exploitation that both have an influential impact on the performance of metaheuristics. Exploration ensures the algorithm reaches different promising regions of the search space, whereas exploitation ensures the searching for optimal solutions within the given region. At the initiating of the optimization process, the value of b forces the proposed algorithm to make higher exploration, whereas the exploitation process has more strength at the end of the main loop of MOEAs. At the next step, the values of θ and ϑ are used in the following formula:

$$X_j^g(t+1)_\Phi = (X_j^g(t))_\Phi - \theta(|\vartheta\, (X_j^g(t))_\Phi - X_j^g(t)|). \tag{12}$$

In the above equation, $g = \{\mathcal{F}, \mathcal{S}\}$ and$\Phi = \{1,2,3\}$, where indices in Φ, 1 to 3, represent the $\Phi 1, \Phi 2,$ and $\Phi 3$ of the current iteration, respectively. $X_j^g(t)$ is the value of position$j$, and $X_j^g(t+1)_\Phi$ indicates the value of the position$j$of property$g$ related to the best solutions Φ. To show more clarification on this formula, it would be mentioned here that we have six sets, generated by the sets g and Φ. The index Φ may take a value 1 to 3, representing rank1, rank2, and rank3 teams of the current iteration. Therefore, the following three equations can be grasped for the top three teams for the formation property.

$$X_j^{\mathcal{F}}(t+1)_1 = (X_j^{\mathcal{F}}(t))_1 - \theta(|\vartheta\, (X_j^{\mathcal{F}}(t))_1 - X_j^{\mathcal{F}}(t)|). \tag{13}$$

$$X_j^{\mathcal{F}}(t+1)_2 = (X_j^{\mathcal{F}}(t))_2 - \theta(|\vartheta\, (X_j^{\mathcal{F}}(t))_2 - X_j^{\mathcal{F}}(t)|) \tag{14}$$

$$X_j^{\mathcal{F}}(t+1)_3 = (X_j^{\mathcal{F}}(t))_3 - \theta(|\vartheta\, (X_j^{\mathcal{F}}(t))_3 - X_j^{\mathcal{F}}(t)|) \quad (15)$$

In the same vein, the following three equations can be used for measuring the three best teams for the substitute property.

$$X_j^{\mathcal{S}}(t+1)_1 = (X_j^{\mathcal{S}}(t))_1 - \theta(|\vartheta\, (X_j^{\mathcal{S}}(t))_1 - X_j^{\mathcal{S}}(t)|). \quad (16)$$

$$X_j^{\mathcal{S}}(t+1)_2 = (X_j^{\mathcal{S}}(t))_2 - \theta(|\vartheta\, (X_j^{\mathcal{S}}(t))_2 - X_j^{\mathcal{S}}(t)|) \quad (17)$$

$$X_j^{\mathcal{S}}(t+1)_4 = (X_j^{\mathcal{S}}(t))_3 - \theta(|\vartheta\, (X_j^{\mathcal{S}}(t))_3 - X_j^{\mathcal{S}}(t)|) \quad (18)$$

Coaches usually teach players according to the performance of the best team. We have no idea about the optimum values $\mathcal{F}$and $\mathcal{S}$ of the best possible team, and the top three teams are considered to be a good measure.

We consider the three best teams ($\Phi = 1,2,$ and 3) and induce a current team to update its properties toward the best team's properties. In this regard, the following formula is presented.

$$X_j^{g}(t+1) = \sum_{\Phi=1}^{3} \frac{X_j^{g}(t+1)_\Phi}{3}. \quad (19)$$

3.9. Season transfers

To mimic the season transfer, we have set H where teams are chosen randomly from the set N. All positions of each member of set $H$is selected randomly from the currently available teams if r, which is a random number from [0, 1], is greater than 0.5.it is assumed that only some, and not all, teams will participate season transfers process. Let δ_{st} indicates the percentage of teams contributing in season transfer and $N_{pr} = [N\delta_{st}]$, the number of teams participating in season transfer. The pseudo-code for season transfers is shown in Algorithm 6.

Algorithm 6: The season transfers process

```
For k=1 to N_st
H = {select randomly i from N | i ∉ H}
End For
For k=1 to N_st
For j=1 to M
r=rand()
    If r>0.5
        w=select randomly from current available teams
        H_j^F(k) = w_j^F
        H_j^S(k) = w_j^S
    End If
End For
CostFunction (k) = f(X^F(k))
End For
```

3.10. Promotion and relegation process

In many sports competitions, there is a hierarchy of leagues in which the premier league is the top league. After finishing a season, best teams are moved up to an upper-division of the league, and the worst teams are moved down to a lower division for the next season. This process is called relegation and promotion in sports literature. Let δ_{pr} shows the rate of promoted and regulated teams at the end of a season. We define $N_{pr} = \lceil N\delta_{pr} \rceil$ to determine the number of teams be moved.

In this process, teams are selected from the lowest-ranked teams to go down to a lower division. To determine which team is promoted to the premier league, and since there is just one league in our algorithm, a unique process has been implemented to reach this goal. Hence, the position of the promoted team is selected randomly from positions of available teams in the premier league. According to the above elucidations, the pseudo-code of the promotion and relegation process is shown in Algorithm 7.

Algorithm 7: Promotion and relegation process

```
Remove N_pr worst teams of the league.
Define N_pr empty teams NT with two properties: F and S
For k=1 to N_pr
For j=1 to J
    s=select randomly from currently available teams (N - N_pr)
    NT_j^F(k) = s_j^F
    NT_j^S(k) = s_j^S
End For
    CostFunction (k) = f(NT^F(k))
End For
Add NT teams to the league
```

4. Multi-objective test functions

There are numerous MOO test functions attainable in the literature [78, 79]. To show the validity of the proposed algorithm, we have selected a subset of these functions with different features. Moreover, different test functions with more sophisticated search space are considered in this paper. That said, we have chosen the test functions proposed in CEC 2009 including seven bi-objective and three tri-objective test functions as listed in . These test problems are considered as the most challenging test problems in the literature that provide different multi-objective search spaces with different Pareto optimal fronts: convex, non-convex, discontinuous, and multi-modal.

Table 2 and Table 3. These test problems are considered as the most challenging test problems in the literature that provide different multi-objective search spaces with different Pareto optimal fronts: convex, non-convex, discontinuous, and multi-modal.

Table 2: Bi-objective test functions

Function	Mathematical expression
UF1	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left[x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2, f_2 = 1-\sqrt{x} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left[x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $J_1 = \{j\lvert j\ is\ odd\ and\ 2\le j\le n\}, J_2 = \{j\lvert j\ is\ even\ and\ 2\le j\le n\}$
UF2	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2, f_2 = 1-\sqrt{x} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$ $J_1 = \{j\lvert j\ is\ odd\ and\ 2\le j\le n\}, J_2 = \{j\lvert j\ is\ even\ and\ 2\le j\le n\},$ $y_j = \begin{cases} x_j - \left[0.3x_1^2 cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\ if\ j\in J_1 \\ x_j - \left[0.3x_1^2 cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\ if\ j\in J_2 \end{cases}$
UF3	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $f_2 = \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\left(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $J_1\ and\ J_2\ are\ the\ same\ as\ those\ of\ UF1, y_j = x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}, j = 2,3,\dots,n$
UF4	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j), f_2 = 1-\sqrt{x} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h(y_j)$ $J_1\ and\ J_2\ are\ the\ same\ as\ those\ of\ UF1, y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2,3,\dots,n, h(t) = \frac{\lvert t\rvert}{1+e^{2\lvert t\rvert}}$
UF5	$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right)\lvert sin(2N\pi x_1)\rvert + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j),\ f_2 = 1-\sqrt{x} + \left(\frac{1}{2N} + \varepsilon\right)\lvert sin(2N\pi x_1)\rvert \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h(y_j)$

	J_1 *and* J_2 *are identcal to those those of* $UF1, \varepsilon > 0$, $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2,3,\ldots,n,$ $h(t) = 2t^2 - cos(4\pi t) + 1$
UF6	$f_1 = x_1 + max\left\{0,2\left(\left(\frac{1}{2N}+\varepsilon\right)\lvert sin(2N\pi x_1)\rvert\right)\right\} + \frac{2}{\lvert J_1\rvert}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $f_2 = 1 - x_1 + max\left\{0,2\left(\left(\frac{1}{2N}+\varepsilon\right)\lvert sin(2N\pi x_1)\rvert\right)\right\} + \frac{2}{\lvert J_1\rvert}\left(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ J_1 *and* J_2 *are identcal to those those of* $UF1, \varepsilon > 0$, $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2,3,\ldots,n,$
UF7	$f_1 = \sqrt[5]{x_1} + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2, f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$ J_1 *and* J_2 *are identcal to those those of* $UF1, \varepsilon > 0$, $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2,3,\ldots,n$

Table 3: Tri-objective test functions

Function	Mathematical formulation
UF8	$f_1 = cos(0.5x_1\pi)cos(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left[x_j - 2x_2 sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $f_2 = cos(0.5x_1\pi)sin(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_2}\left[x_j - 2x_2 sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $f_3 = sin(0.5x_1\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_3}\left[x_j - 2x_2 sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $J_1 = \{j \mid 3 \le j \le n. and\ j-1\ is\ a\ multiplication\ of\ 3\}.$ $J_2 = \{j \mid 3 \le j \le n. and\ j-2\ is\ a\ multiplication\ of\ 3\}.$ $J_3 = \{j \mid 3 \le j \le n. and\ j\ is\ a\ multiplication\ of\ 3\}.$
UF9	$f_1 = 0.5[\,max\{0.(1+\varepsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left[x_j - 2x_2 sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $f_1 = 0.5[\,max\{0.(1+\varepsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left[x_j - 2x_2 sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $f_1 = 1 - x_2 + \frac{2}{\lvert J_3\rvert}\sum_{j\in J_2}\left[x_j - 2x_2 sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$ $J_1 = \{j \mid 3 \le j \le n. and\ j-1\ is\ a\ multiplication\ of\ 3\}.$ $J_2 = \{j \mid 3 \le j \le n. and\ j-2\ is\ a\ multiplication\ of\ 3\}.$ $J_3 = \{j \mid 3 \le j \le n. and\ j\ is\ a\ multiplication\ of\ 3\}.$
UF10	$f_1 = cos(0.5x_1\pi)cos(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}[4y_j^2 - \cos(8\pi y_{j1}) + 1]$ $f_2 = cos(0.5x_1\pi)sin(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_2}[4y_j^2 - \cos(8\pi y_{j1}) + 1]$ $f_3 = sin(0.5x_1\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_3}[4y_j^2 - \cos(8\pi y_{j1}) + 1]$ $J_1 = \{j \mid 3 \le j \le n. and\ j-1\ is\ a\ multiplication\ of\ 3\}.$ $J_2 = \{j \mid 3 \le j \le n. and\ j-2\ is\ a\ multiplication\ of\ 3\}.$ $J_3 = \{j \mid 3 \le j \le n. and\ j\ is\ a\ multiplication\ of\ 3\}.$

The MOVPL algorithm is statistically analyzed, and compared with two eminent MOEAs in the literature, which have recorded encouraging performance including MOPSO and MOEA/D. This approach is inspired by a recent study of [3] on MOEAs contexts to analyze all proposed MOEAs. To homogenize our experiment, the number of function evaluations is set to 300K for all algorithms. Like in other studies [3, 80], we run each test function 30 times.

One of the most important aspects of implementing an algorithm is parameter tuning which plays a crucial role in the performance of proposed MOEA. Since the proposed algorithm has not been published yet, there is no study that determines the optimal values for all parameters of MOVPL, and that is, the Taguchi method is used for this consideration. The first step of the Taguchi method is to find the parameters and their levels, which are shown in Table 4.

Table 4: MOVPL parameters and levels.

Parameter	Level 1	Level 2	Level 3
$nPop$	100	150	200
T	100	200	400
λ^f	1	1.4	1.8
λ^s	1	1.4	1.8
ψ^f	1	1.4	1.8
ψ^s	1	1.4	1.8
β	4	7	10
δ_{pr}	0.1	0.5	0.9
δ_{st}	0.1	0.5	0.9
δ_{tr}	0.1	0.5	0.9
$MaxIt$	100	200	400
$nArchive$	50	75	100

It is worth mention here that Taguchi method is not implemented for other algorithms, and the optimal values of parameters for corresponding MOEAs can be grasped from [3], thus, the following parameters are selected for MOPSO:

- $\phi_1 = \phi_2 = 2.05$
- $\phi = \phi_1 + \phi_2$
- $w = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}$
- $c_1 = w \times \phi_1$
- $c_2 = w \times \phi_2$
- $\alpha = 0.1$
- $\beta = 4$
- $nGrid$=10

where w denotes inertial weight, c_1 and c_2 are personal coefficient and social coefficient, respectively, $nGrid$ α be grid inflation parameter, β is leader selection pressure parameter, and finally, the number of grids per each dimension. It is worth mentioning here that $nGrid$ and β have the most influential effects on the performance of MOPSO to reach diversity throughout the optimization process. As mentioned in the literature (e.g., [3]), the best values for these components ($\beta = 4$ and $nGrid = 10$) have guaranteed the proposed algorithm to maintain diversity in the most multi-objective optimization problems. The initial parameters of MOEA/D are selected as follows:

- $N = 100$
- $T = 10$
- $n_r = 1$
- $\gamma = 0.9$
- $m_r = 0.5$
- $\eta = 30$

where N denotes the number of Subproblems, T is number of neighbors, n_r is the maximal copies of a new child in update, γ is the probability of selecting parents from the neighborhood, m_r is mutation rates, and finally, η is the distribution index.

To investigate all factors simultaneously, orthogonal arrays have been used for the Taguchi method. In this way, the L27 design is applied for MOVPL, which is shown in Table 5. This approach has been implemented in many studies, such as [81, 82].

Table 5: Used L27 design in the Taguchi method

Experiment number	Parameters levels											
	$nPop$	T	λ^f	λ^s	ψ^f	ψ^s	β	δ_{pr}	δ_{st}	δ_{tr}	$MaxIt$	$nArchive$
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2
11	2	1	2	3	2	3	1	2	3	1	2	3
12	2	1	2	3	3	1	2	3	1	2	3	1
13	2	2	3	1	1	2	3	2	3	1	3	1
14	2	2	3	1	2	3	1	3	1	2	1	2
15	2	2	3	1	3	1	2	1	2	3	2	3
16	2	3	1	2	1	2	3	3	1	2	2	3
17	2	3	1	2	2	3	1	1	2	3	3	1
18	2	3	1	2	3	1	2	2	3	1	1	2
19	3	1	3	2	1	3	2	1	3	2	1	3
20	3	1	3	2	2	1	3	2	1	3	2	1
21	3	1	3	2	3	2	1	3	2	1	3	2
22	3	2	1	3	1	3	2	2	1	3	3	2
23	3	2	1	3	2	1	3	3	2	1	1	3
24	3	2	1	3	3	2	1	1	3	2	2	1
25	3	3	2	1	1	3	2	3	2	1	2	1
26	3	3	2	1	2	1	3	1	3	2	3	2
27	3	3	2	1	3	2	1	2	1	3	1	3

After determining Taguchi design, we adjusted the MOVPL parameters over different problems (10 test functions and three classical engineering problems including welded beam design (WBD), disc brake design (DBD), and speed reducer design (SRD) problems), which are expressed in the Section 6). The parameters are shown in Table 6. As seen in this table, MOVPL is tuned for each problem, separately.

Table 6: Best obtained parameter values used based on various problems

Parameter	Test functions										Engineering		
	UF1	UF2	UF3	UF4	UF5	UF6	UF7	UF8	UF9	UF10	WBD	DBD	SRD
$nPop$	1	1	1	3	1	1	1	2	1	1	1	1	3
T	2	2	2	1	1	2	1	1	1	2	2	2	1
λ^f	3	2	2	1	2	2	2	3	1	1	2	2	1
λ^s	3	1	2	2	2	2	2	2	2	2	1	3	2
ψ^f	3	2	2	2	2	2	1	1	1	1	2	1	3
ψ^s	2	3	1	2	2	1	2	1	1	1	3	1	3
β	1	1	1	1	2	1	2	2	2	3	1	1	2
δ_{pr}	2	2	3	2	2	1	1	3	2	2	3	2	2
δ_{st}	1	1	2	2	2	2	2	1	1	3	1	1	3
δ_{tr}	2	2	2	2	2	3	1	1	1	1	1	1	1

$MaxIt$	2	2	3	1	3	2	3	3	3	3	2	3	3
$nArchive$	3	1	2	3	1	1	3	1	2	1	3	2	1

In this section, four standard metrics of MOEA algorithms are introduced.

Spacing (SP) metric: This metric [83] is functioned to measure the coverage of our algorithm as:

$$SP = \sqrt{\frac{1}{|PF|}\sum_{i=1}^{n}(\bar{d} - d_i)^2}. \quad (20)$$

The main statistic parameter used in this metric is the standard deviation which considers distances among solutions of the Pareto front, where $d_i = \min_{\vec{y}\in PF} \sum_{k=1}^{K}(|o_k(\vec{x}) - o_k(\vec{y})|)$ is a minimal divergence of an individual $\vec{x}$ from all individuals and $\bar{d}$ is the average of d_i.

Maximum Spread (MS) metric: This metric considers the extension of Pareto solutions, as shown in Eq. (21) [84]:

$$MS = \sqrt{\sum_{i=1}^{l}(min\, f_i - max\, f_i)^2}, \quad (21)$$

where $min\, f_i$ and $max\, f_i$ denote the best and the worst values of the fitness function within all non-dominated individuals [85].

Inverted Generational Distance (IGD) metric: the average value of the distance between obtained individuals on Pareto front and obtained Pareto optimal set is calculated via this metric as: [46]:

$$IGD = \frac{\left(\sum_{i=1}^{N} d_i^2\right)^{1/2}}{N}, \quad (22)$$

where N means the total number of obtained non-dominated individuals in Pareto front, and d_i states the minimum Euclidean distance between the ith solutions to the Pareto optimal set.

Diversity (Δ) metric: This metric calculates the variety of obtained individuals as following [46]:

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{N}|d_i - \bar{d}|}{\sum_{m=1}^{M} d_m^e + N * \bar{d}}, \quad (23)$$

where N symbolizes the total number of non-dominated individuals in Pareto front, M is defined as the total number of objectives, d_i states Euclidean distance between neighboring individuals, and $\bar{d}$ is the mean of d_i. Moreover, the parameter d_m^e denotes the Euclidean distance between the individuals of Pareto optimal set and the obtained non-dominated solution set with respect to the mth objective function.

5. Computational experiments

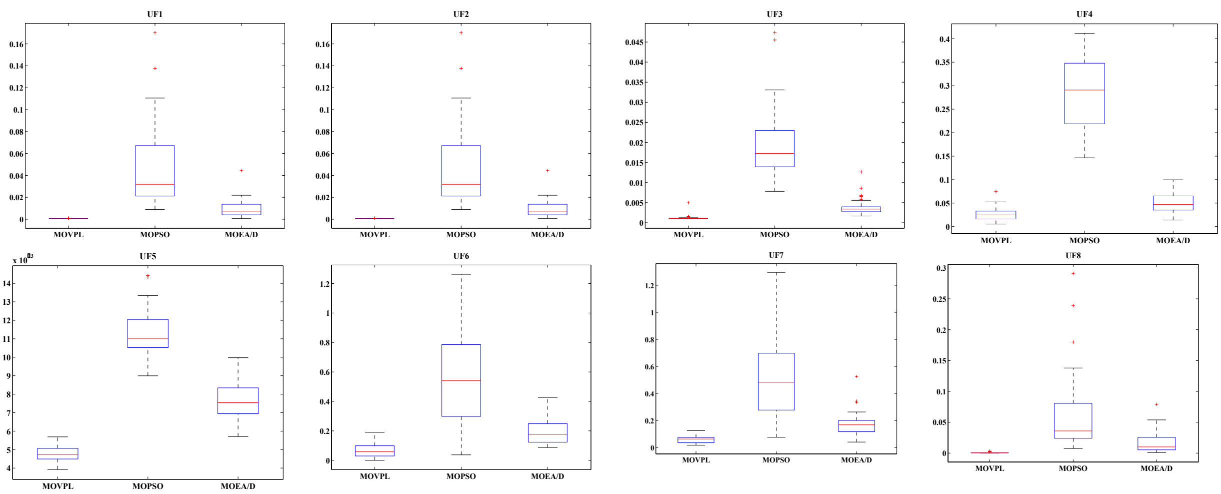
In the course of the experiment, statistical reports played an important role. Therefore, the experiments were performed statistically with presented in tables and plots. Table 7 demonstrates the statistical results of the algorithms for the first metric, IGD, in which MOVPL shows exceptionally better than the others do in UF1. As seen in Table 7, the statistical results indicated that MOVPL obtains the best ranks for nine out of all test problems indicating promising performances of the proposed algorithm in IGD metric on the multi-objective test functions. It can be concluded that high convergence of MOVPL is coined from the learning procedures and updating the main properties of a team with respect to other teams

Table 7: Statistical results for the IGD metric on all test functions

Test function	Statistical features	MOVPL	MOPSO	MOEA/D
UF1	Mean	**0.0005**	0.0474	0.0096
	Std.	**0.0002**	0.0389	0.0086

	Rank	1	3	2
UF2	Mean	**0.0012**	0.0202	0.0040
	Std.	**0.0007**	0.0097	0.0022
	Rank	1	3	2
UF3	Mean	**0.0268**	0.2874	0.0501
	Std.	**0.0144**	0.0766	0.0202
	Rank	1	3	2
UF4	Mean	**0.0048**	0.0113	0.0076
	Std.	**0.0004**	0.0013	0.0011
	Rank	1	3	2
UF5	Mean	**0.0719**	0.5708	0.1983
	Std.	**0.0472**	0.3383	0.0908
	Rank	1	3	2
UF6	Mean	**0.0581**	0.4942	0.1754
	Std.	**0.0275**	0.2652	0.0948
	Rank	1	3	2
UF7	Mean	**0.0008**	0.0646	0.0170
	Std.	**0.0008**	0.0663	0.0178
	Rank	1	3	2
UF8	Mean	0.0209	0.622	**0.0169**
	Std.	0.0353	0.2957	**0.0100**
	Rank	2	3	1
UF9	Mean	0.0401	0.6529	**0.0211**
	Std.	0.0336	0.1597	**0.0096**
	Rank	2	3	1
UF10	Mean	**0.0994**	2.6392	0.1597
	Std.	**0.0348**	1.1245	0.0764
	Rank	1	3	2
Average ranking		1.2	3	1.8
Total rank		1	3	2

This statistical result, also, can be seen in the boxplot. According to Figure 3, the corresponding boxplot for MOVPL is tighter and lower than the others. With this in mind, this algorithm shows better accuracy convergence than its rivals. It can be expressed that the MOVPL algorithm is immensely talented to afford exceptional convergence and accuracy on UF1.



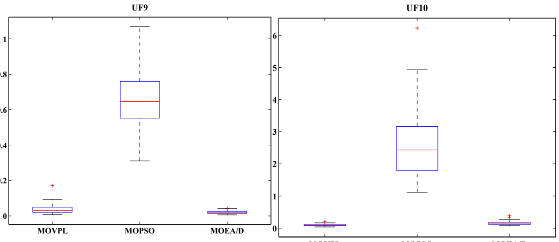


Figure 3: Boxplot of the statistical results for IGD on all test functions

It can be seen a comprehensive overview of the results of Pareto optimal solution for all MOEAs, which is shown in Figure 4. According to Table 8 and Table 9, it can be concluded that the proposed algorithm is able to achieve better convergence and coverage than the others are. While there are breaks on the Pareto optimal front of MOVPL, the convergence of the whole front is more extensive than the others on this test function. Hence, for the first test function, UF1, true Pareto optimal front is not widely distributed along with both objectives. It can be concluded that the results of MOVPL are better to achieve a true Pareto front, and it can cover a large part of the true Pareto front. For the UF2, the results of MOVPL are quite close to the results of MOEA/D, which shows the superior result in the average. MOEA/D obtains the best results, and the obtained results of Table 8 and Table 9 show that MOVPL is more stable than the others are. The significance of the results in Figure 3 indicates that MOVPL has promising performance in converging related to the true Pareto optimal front.

To analyze the coverage of the proposed MOEAs, Figure 4 is provided to show obtained Pareto optimal solutions in which MOPSO is less distributed then MOVPL and MOEA/D which indicate the poor performance of that algorithm. Concerning obtained optimal results from MOEAs, MOVPL proves comparatively better coverage as outcomes of SP and MS indicated in Table 8 and Table 9. In UF3, MOVPL has the best average in all statistical values for IGD. By focusing on Figure 3, the superior performance of this algorithm can be grasped according to its narrower and lower boxplot. However, according to statistical results, similar to those on UF1 and UF2, MOPSO shows the worst performance for the IGD metric. The optimal Pareto solutions of all MOEAs can be seen in Figure 4, indicating the superiority of the MOVPL. This figure also reveals that the poor performance results of all proposed MOEAs obtained front to show that they would never be able to afford good coverage on UF3 test function. Despite this fact, the convergence and accuracy of MOVPL are better than the others are.

The next test function problem, UF4, is discussed in this section to show the validity of the algorithms. Considering the statistical and in-depth analysis of this test function, MOVPL shows superior performance than MOPSO and MOEA/D. The tighter boxplot of MOVPL for the aforementioned test function (Figure 3) placed to lower the minimum values of MOPSO, and MOEA/D verifies that statistics results of MOVPL are relatively significant. The results of the experiment on this test function find that MOPSO obtains worst performance. Generally, the poor performance of MOEAs on this test function could be derived from the complex shape of the optimal Pareto front.

As shown in Figure 4, the UF5 has been considered as one of the most discontinuous test functions. According to statistical results shown in the tables, MOVPL has defeated both MOPSO and MOPSO concerning all metrics. Even though MOVPL has better performance than others do, but statistically, not all MOEAs are able to provide promising performance on UF5. Given these points, it might be derived from complications arising from search space with a large number of discontinuous regions that impede algorithms from achieving a remarkable result on this test problem. Similar to test functions aforementioned before, MOVPL states impressive performance on approximating Pareto front on UF6. According to statistical result, MOVPL has superior performance on this test function. MOPSO shows the poor results concerning all metrics. With considering results shown in Table 7-14, MOVPL shows better convergence ability in statistical indices. Because of sophisticated search space, none of these algorithms can estimate Pareto optimal solutions to the true Pareto front, while MOVPL is much closer to the Pareto front than others.

Scholars have regarded that some linear test functions, like UF7, is not challenging to estimate true Pareto solutions [2]. The proposed algorithm, MOVPL, has better performance in determining true Pareto front in comparison with others. Even though it could be seen in Table 7-Considering obtained results from Table 9, the excellent agreement was achieved for evaluating the performance of all proposed algorithms indicating that MOVPL algorithm approach produced good quality results in MS metrics.

Table 10 that the proposed algorithm shows better results, but, this eminence is not very outstanding as a result of the linear-shaped Pareto front of UF7. As seen in Figure 3, the proposed algorithm has a narrower lower average than both MOEA/D and MOPSO, which is proven its superiority. In analyzing this test function, MOPSO is not uniformly scattered along the true Pareto optimal front, and achieved solutions are placed on spots where the convergence of algorithm rarely occurred.

Table 8: Statistical results for spacing (SP) metric on all test functions

Test problem	Statistical features	MOVPL	MOPSO	MOEA/D
UF1	Mean	**0.0028**	0.1341	0.0387
	Std.	**0.0013**	0.1494	0.0695
	Rank	1	3	2
UF2	Mean	**0.0092**	0.0778	0.0264
	Std.	**0.0074**	0.0549	0.0211
	Rank	1	3	2
UF3	Mean	**0.0549**	0.2528	0.0674
	Std.	**0.0618**	0.1442	0.1045
	Rank	1	3	2
UF4	Mean	**0.0123**	0.0248	0.0289
	Std.	**0.0024**	0.0062	0.0078
	Rank	1	2	3
UF5	Mean	**0.1636**	0.5702	0.2999
	Std.	**0.1274**	0.3941	0.3698
	Rank	1	3	2
UF6	Mean	**0.1590**	0.5651	0.1757
	Std.	**0.0955**	0.5626	0.4210
	Rank	1	3	2
UF7	Mean	**0.0057**	0.1836	0.0772
	Std.	**0.0018**	0.2148	0.1428
	Rank	1	3	2
UF8	Mean	0.1592	2.7551	**0.1011**
	Std.	**0.0287**	1.4626	0.0844
	Rank	2	3	1
UF9	Mean	0.2895	2.7612	**0.1202**
	Std.	0.2507	0.8406	**0.0602**
	Rank	2	3	1
UF10	Mean	0.6476	9.3447	**0.4684**
	Std.	0.2427	4.2827	**0.4621**
	Rank	2	3	1
Average ranking		1.3	2.9	1.8
Total rank		1	3	2

The results obtained from Table 8, the SP metrics of all proposed algorithms show that MOVPL has gained the best rank for 7 out of 10 test problems, and in terms of UF8, UF9, and UF10 where MOVPL has gained rank 2, the difference between top 2 ranks can be overlooked.

The other three test functions, UF8, UF9, and UF10, are defined as the most challenging test problems, which have more than two objectives. In two of these test functions, the results show that MOVPL has a better value in all metrics, which is seen in boxplot (Figure 3). The obtained Pareto front of UF8, as the first three-objectives test function problems, can be seen in Figure 4. It would be clear that obtained Pareto optimal solutions of all algorithms are not surely close to the true Pareto optimal front. Another challenging test function problem, the UF9, provides a distinguished Pareto front, which is more challenging to be solved. Figure 4 shows obtained Pareto front of UF9 for all applied MOEAs. According to Figure 4, MOEA/D shows better convergence despite the fact that its poor results correspond to all metrics. Even though MOEA/D records the best performance, but this difference is not highly significant with MOVPL, as shown in Figure 3.

The last test function, UF10, has similar performance to UF8 test function in estimating Pareto front. As shown in Figure 4, all proposed MOEAs are not able to coverage true Pareto front, but MOVPL shows superior statistical results on different aspects for IGD. As quantitative analytical results of all metrics shown in Table 7-Considering obtained results from Table 9, the excellent agreement was achieved for evaluating the performance of all proposed algorithms indicating that MOVPL algorithm approach produced good quality results in MS metrics.

Table 10, MOVPL dominates other proposed algorithms and affords spectacular results on this test problem. We now provide the performance analysis of all proposed MOEAs for different metrics. As mentioned before, Table 7 offers statistical results for IGD, which is considered one of the primary metrics for analyzing the performance of MOEAs. Hence, the other performance metrics, maximum spread (MS), diversity metric (Δ), and for spacing (SP), which provide quantitative assessments for the performance of the proposed MOEAs, are shown in Table 8 -- Table 10, respectively.

Table 9: Statistical results for maximum spread (MS) metric on all test functions

Test function	Statistical features	MOVPL	MOPSO	MOEA/D
UF1	Mean	**2.2518**	1.3532	1.1661
	Std.	**0.7952**	0.241	0.6436
	Rank	1	2	3
UF2	Mean	**2.3855**	1.7687	1.594
	Std.	**0.35**	0.1932	0.2908
	Rank	1	2	3
UF3	Mean	**2.5778**	0.8948	0.559
	Std.	**1.2757**	0.6872	0.6731
	Rank	1	2	3
UF4	Mean	**2.0713**	1.995	2.0504
	Std.	**0.0542**	0.044	0.0942
	Rank	1	3	2
UF5	Mean	**4.1626**	1.7331	1.9152
	Std.	**2.9018**	1.1498	1.6763
	Rank	1	3	2
UF6	Mean	**3.5934**	1.9239	1.1978
	Std.	**2.4578**	0.9794	1.2694
	Rank	1	2	3
UF7	Mean	**2.2424**	0.8101	1.106
	Std.	**1.1106**	0.4298	0.9429
	Rank	1	3	2
UF8	Mean	**4.2128**	3.671	2.8741
	Std.	0.373	**2.0775**	0.9458
	Rank	1	2	3
UF9	Mean	**6.5246**	4.1932	2.7404
	Std.	**5.6954**	1.8129	0.8121
	Rank	1	2	3
UF10	Mean	**9.7357**	8.0222	3.6231

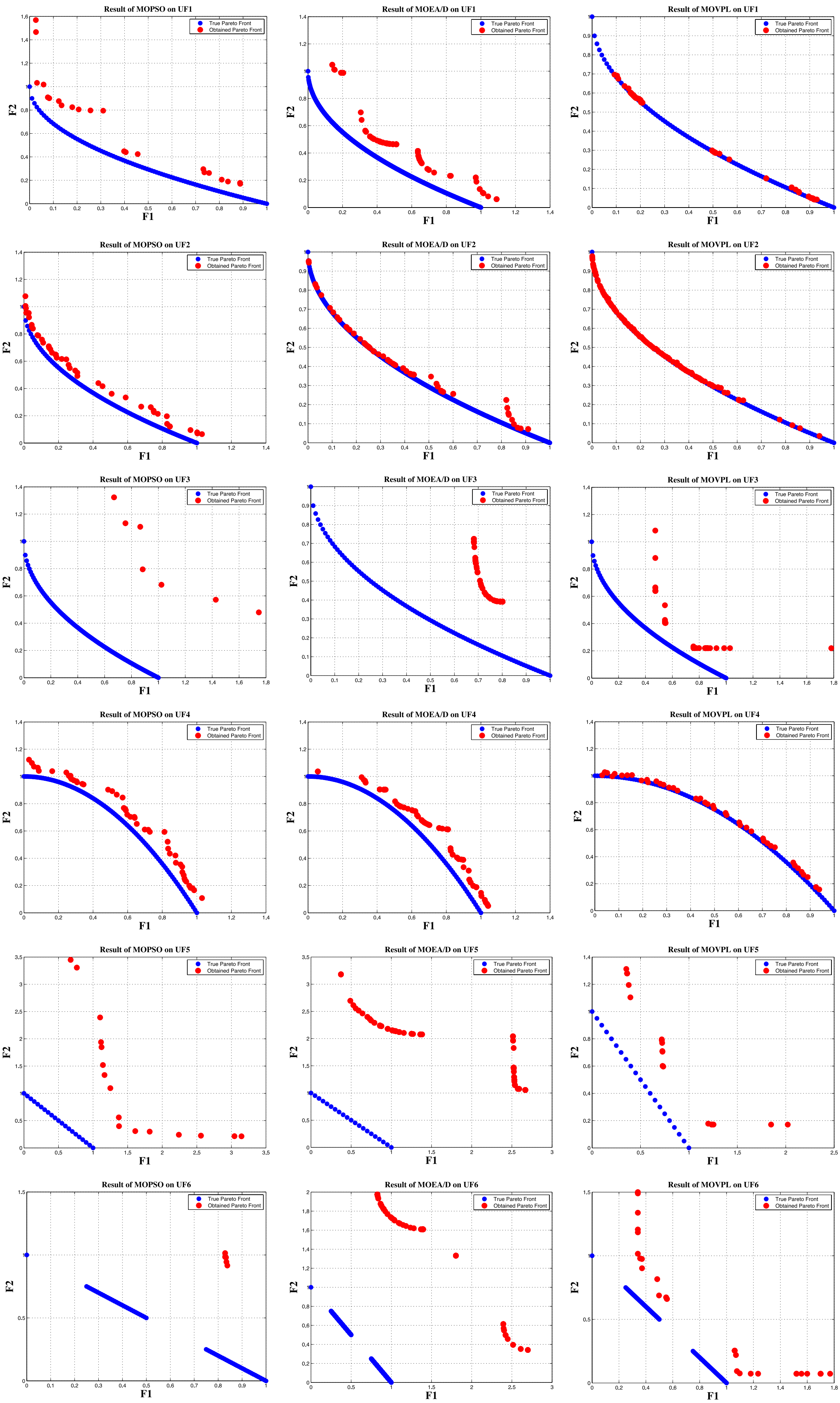
	Std.	**2.9007**	2.7391	2.8233
	Rank	1	2	3
Average ranking		1	2.3	2.7
Total rank		1	2	3

Considering obtained results from Table 9, the excellent agreement was achieved for evaluating the performance of all proposed algorithms indicating that MOVPL algorithm approach produced good quality results in MS metrics.

Table 10: Statistical results for diversity metric (Δ) on all test functions

Test function	Statistical features	MOVPL	MOPSO	MOEA/D
UF1	Mean	**0.1214**	0.2766	0.1489
	Std.	**0.0535**	0.0540	0.0681
	Rank	1	3	2
UF2	Mean	**0.1632**	0.2320	0.1734
	Std.	**0.0048**	0.0267	0.0209
	Rank	1	3	2
UF3	Mean	**0.0975**	0.3652	0.1038
	Std.	**0.0741**	0.0743	0.0951
	Rank	1	3	2
UF4	Mean	**0.1842**	0.1937	0.1957
	Std.	0.0289	**0.0147**	0.0175
	Rank	1	3	2
UF5	Mean	**0.1329**	0.4016	0.1438
	Std.	0.1176	**0.0998**	0.1207
	Rank	1	3	2
UF6	Mean	**0.0893**	0.4113	0.1057
	Std.	**0.0854**	0.1177	0.1104
	Rank	1	3	2
UF7	Mean	**0.0937**	0.2829	0.1466
	Std.	**0.0451**	0.0814	0.0850
	Rank	1	3	2
UF8	Mean	**0.1205**	0.5077	0.1936
	Std.	0.0886	0.0939	**0.0330**
	Rank	1	3	2
UF9	Mean	**0.1962**	0.5188	0.2195
	Std.	0.0839	0.0655	**0.0331**
	Rank	1	3	2
UF10	Mean	**0.2795**	0.5356	0.3755
	Std.	**0.0554**	0.0844	0.1233
	Rank	1	3	2
Average ranking		1	3	2
Total rank		1	3	2

According to results grasped from Table 10, the performance of a diversity metric has completely similar to MS metrics indicated that MOVPL has obtained the best rank in comparison with its rivals. The increasing diversity of obtained solutions may be explained by using specific operators such as learning strategy, which is originally driven from the basic version of VPL.


Result of MOPSO on UF1
Result of MOEA/D on UF1
Result of MOVPL on UF1
Result of MOPSO on UF2
Result of MOEA/D on UF2
Result of MOVPL on UF2
Result of MOPSO on UF3
Result of MOEA/D on UF3
Result of MOVPL on UF3
Result of MOPSO on UF4
Result of MOEA/D on UF4
Result of MOVPL on UF4
Result of MOPSO on UF5
Result of MOEA/D on UF5
Result of MOVPL on UF5
Result of MOPSO on UF6
Result of MOEA/D on UF6
Result of MOVPL on UF6
True Pareto Front
Obtained Pareto Front
F1
F2

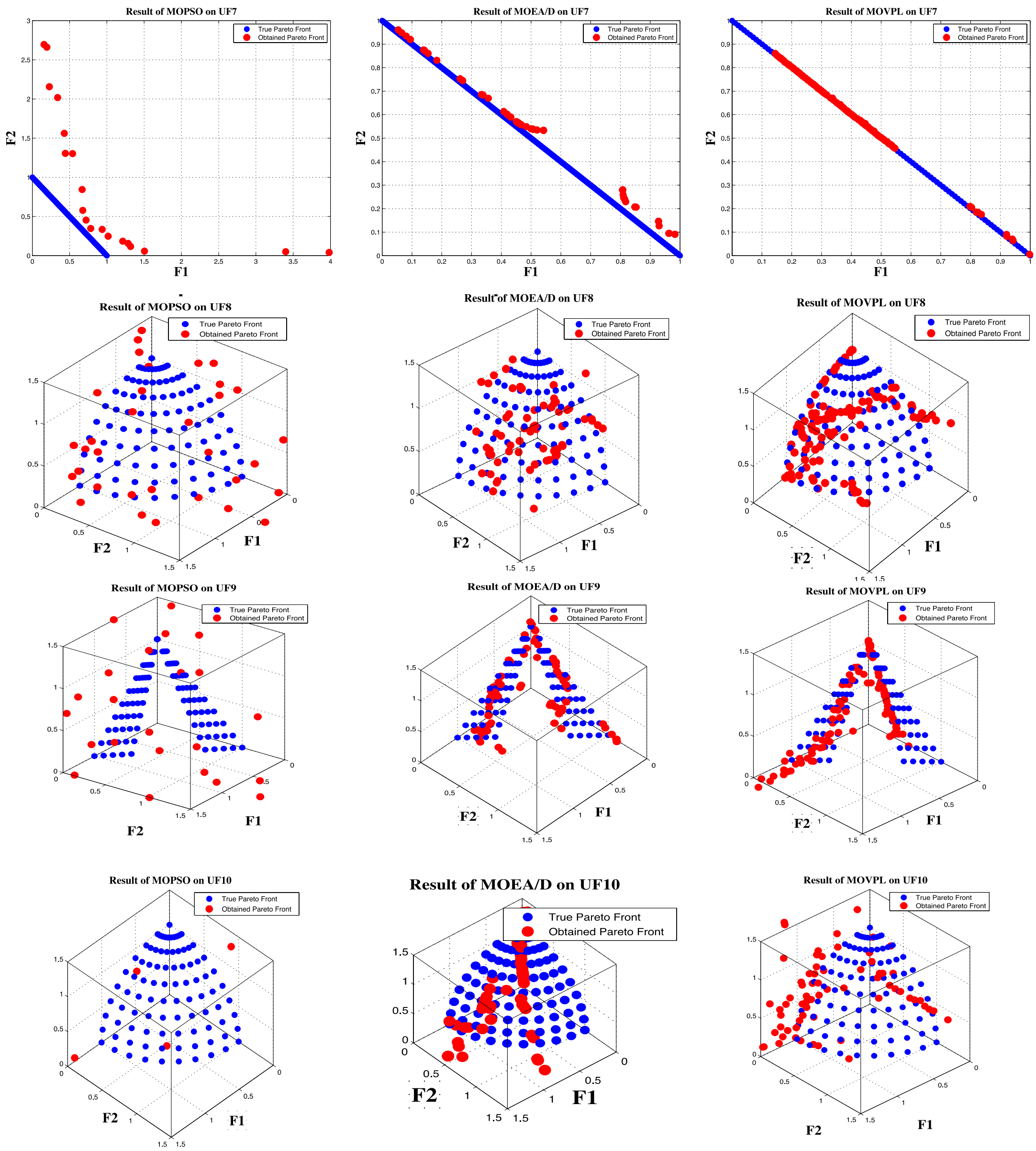


Figure 4: Pareto optimal solutions of all MOEAs for all test functions

It can be concluded that kind of pattern takes place as a result of the hyper grid and archive which were embedded to MOEAs. The archive applied to the MOVPL algorithm compels algorithm to store and retrieve the best teams.

For more supporting our study statistically, The Wilcoxon Signed-Rank Test was used to verify statistical significance [86]. Therefore, the null hypothesis (H_0) and alternative hypothesis (H_1) are defined to show the performance of the two MOEAs:

$$\begin{cases} H_0: \mu_1 = \mu_2 \\ H_1: \mu_1 \neq \mu_2 \end{cases} \tag{24}$$

Significance was defined as a probability value of 0.5. It is worth mention here that p-value would be respected as the primary criteria to accept the null hypothesis (H_0). The p-values of each unrelated hypothesis have been measured and expressed in Table 11. In this table, R denotes statistical results of the Wilcoxon signed-rank test, if it is equal to 1, there is a significant difference between MOVPL and the other algorithms; conversely, if it is equal to

0, there is no significant difference between MOVPL and the others. From the value of H rows, if it is termed as ''+'', indicated that the proposed algorithm outperforms other rivals statistically, conversely, ''-'' shows that MOVPL is inferior to the other MOEAs, and finally, ''='' indicated that there is no significant difference between proposed algorithm and its rivals. Additionally, in the last row of this table, it is seen the term $w/t/l$ which is considered the number of the win, tie, and loss of proposed MOEA compared to its rivals.

Table 11: Statistical results of Wilcoxon signed-rank test for all MOEAs

MOVPL	Statistic features	MOVPL vs #							
		IGD		SP		MS		Δ	
		MOPSO	MOEA/D	MOPSO	MOEA/D	MOPSO	MOEA/D	MOPSO	MOEA/D
UF1	p-value	4.20E-05	1.40E-04	4.42E-04	2.46E-04	1.60E-04	3.91E-05	4.20E-04	1.40E-04
	R	1	1	1	1	1	1	1	1
	H	+	+	+	+	+	+	+	+
UF2	p-value	3.93E-06	6.35E-04	6.58E-04	4.49E-04	1.95E-05	5.29E-06	3.93E-05	6.35E-04
	R	1	1	1	1	1	1	1	0
	H	+	+	+	+	+	+	+	=
UF3	p-value	2.05E-04	1.92E-04	6.02E-05	5.52E-04	3.15E-04	4.84E-04	2.05E-04	1.92E-01
	R	1	1	1	1	1	1	1	1
	H	+	+	+	+	+	+	+	+
UF4	p-value	5.11E-05	3.13E-04	1.90E-04	1.50E-05	1.03E-04	2.99E-05	5.11E-04	3.13E+01
	R	1	1	1	1	1	1	1	0
	H	+	+	+	+	+	+	+	=
UF5	p-value	1.51E-06	4.00E-04	7.77E-05	2.14E-04	2.90E-04	2.42E-04	1.51E-04	4.00E-01
	R	1	1	1	1	1	1	1	0
	H	+	+	+	+	+	+	+	=
UF6	p-value	8.45E-05	3.45E-04	3.30E-04	2.25E-04	1.79E-04	5.61E-04	8.45E-05	3.45E-04
	R	1	1	1	1	1	1	1	1
	H	+	+	+	+	+	+	+	+
UF7	p-value	3.41E-5	3.08E-04	3.51E-04	1.82E-04	4.12E-05	3.09E-04	3.41E-04	3.08E-04
	R	1	1	1	1	1	1	1	1
	H	+	+	+	+	+	+	+	+
UF8	p-value	3.66E-04	8.45E-05	1.50E-04	2.57E-04	6.19E-04	2.35E-04	3.66E-05	8.45E-05
	R	1	1	1	1	1	1	1	1
	H	+	+	+	-	+	+	+	+
UF9	p-value	1.86E-05	5.55E-03	6.15E-05	5.65E-05	4.86E-04	3.26E-04	1.86E-05	5.55E-04
	R	1	1	1	1	1	1	1	1
	H	+	-	+	-	+	+	+	+
UF10	p-value	8.34E-04	1.00E+00	4.26E-04	3.26E-04	7.77E-04	1.78E-04	8.34E-04	4.97E-04
	R	1	0	1	1	1	1	1	1
	H	+	=	+	-	+	+	+	+
$w/t/l$		10/0/0	8/1/1	10/0/0	7/0/3	10/0/0	10/0/0	10/0/0	7/3/0

In this regard, an interesting point found from Table 11 is the fact that MOVPL outperforms all test function in terms of all metrics compared with MOPSO, meanwhile proposed method has shown different behavior in comparison with MOEA/D, that is, MOVPL superiors to MOEA/D in all test problems in terms of MS, but the proposed method cannot surpass MOEA/D on UF9 and UF10 for IGD, UF8, UF9, and UF 10 for SP and UF2, UF4, and UF5 for Δ. Consequently, it can be concluded that the difference in obtained results between the proposed method and its rivals was statistically significant, and generally, MOVPL excel both methods on all metrics.

6. Application of MOVPL in classical engineering problems

Multi-objective optimization has many implementations in engineering and industry. Three engineering design problems are analyzed to show the validity of our proposed algorithm. To show the capability of the proposed algorithm, we will provide a comprehensive experimental analysis for different classical engineering benchmark test functions.

6.1. The welded beam design problem

Multi-objective design of a welded beam is a well-known traditional engineering benchmark applied by many scholars. The objective of this problem is the minimization of overall fabrication. There are various constraints such as shear stress (τ), buckling load on the bar (P_c), end deflection of the beam (δ), and bending stress in the beam (σ) considering in this problem. The general sketch of the problem is exposed in Figure 5.

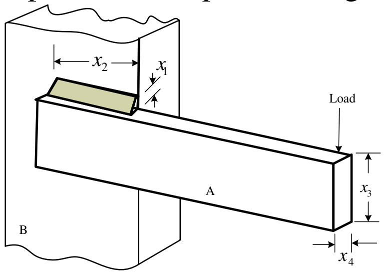


Figure 5: the welded beam design and its features

In this problem, there are four variables as the width $h\ (x_1)$, length $l\ (x_2)$ of the welded area, the depth $t\ (x_3)$, and the thickness $b(x_4)$ of the main beam. This problem can be stated as:

$$Consider\ \vec{x} = [x_1\ x_2\ x_3\ x_4\ ] = [h\ l\ t\ b] \quad (25)$$

$$Min\ f(\vec{x}) = 1.10471 x_2 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) \quad (26)$$

$$Min\ f(\vec{x}) = \delta(\vec{x}) \quad (27)$$

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0 \quad (28)$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0 \quad (29)$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0 \quad (30)$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0 \quad (31)$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0 \quad (32)$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0 \quad (33)$$

$$g_7(\vec{x}) = 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0 \quad (34)$$

$$0.10 \leq x_1 \leq 2.00, \quad (35)$$

$$0.10 \leq x_2 \leq 10.00, \quad (36)$$

$$0.10 \leq x_3 \leq 10.00, \quad (37)$$

$$0.10 \leq x_4 \leq 2.00, \quad (38)$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{4PL^3}{Ex_4x_3^2}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$P = 6000$ lb, $L14$ In, $\delta_{max} = 0.25$ In.,
$E = 30 \times 10^6$ psi, $G = 30 \times 10^6$ psi
$\tau_{max} = 13600$ psi, $\sigma_{max} = 30000$ psi

Eqs. (26) and (27) consider the objective functions; all constraints of the problem are discussed in the Eqs. (28)-(34), and variables are defined Eqs. (35)-(38). The IGD, SP, MS, and Δ metrics are calculated over thirty runs for MOVPL and the two other algorithms. The obtained results in the form of min, worst, mean, and standard deviation values are exposed in Table 12, including the values of MOVPL, MOPSO, and MOEA/D algorithms. In Table 15, the best results in terms of all metrics are attained by MOVPL. In this benchmark function problem, it is perceived that MOVPL has lower IGD, as well as, lower SP, higher MS, and lower Δ which are indicated that proposed algorithm has superior performance than its rivals to find true Pareto front.

Table 12: Statistics results of the welded beam design problem

Metrics	MOEAs	Min	Max	Mean	Std.	Rank
IGD	MOVPL	2.85E+04	3.23E+05	2.92E+04	2.85E+04	1
	MOPSO	3.95E+04	4.65E+05	6.12E+04	2.85E+04	3
	MOEA/D	2.85E+04	3.23E+05	2.92E+04	2.85E+04	2
SP	MOVPL	4.23E+03	7.65E+03	5.37E+03	1.23E+02	1
	MOPSO	4.77E+03	8.23E+03	5.94E+03	1.42E+02	3
	MOEA/D	4.44E+03	8.12E+03	6.01E+03	1.73E+02	2
MS	MOVPL	1.84E+04	5.23E+04	3.54E+04	0.07E+04	1
	MOPSO	1.23E+04	4.43E+04	2.44E+04	0.03E+04	3
	MOEA/D	1.65E+04	4.75E+04	2.67E+04	0.04E+04	2
Δ	MOVPL	0.6451	0.9221	0.8332	0.0321	1
	MOPSO	0.7787	0.9801	0.8741	0.0511	3
	MOEA/D	0.6812	0.9723	0.8865	0.0921	2

The general comparison of corresponding Pareto fronts obtaining from the best individuals of algorithms is illustrated in Figure 6, where it is clearly grasped that the results are the mostly better than the results obtained from its rivals.

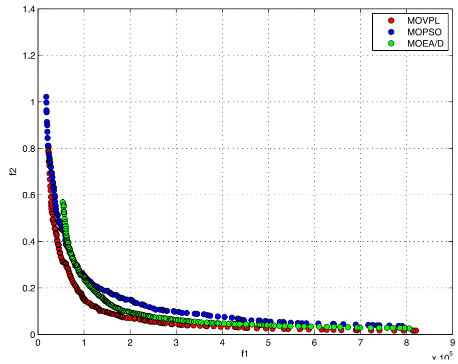


Figure 6: Pareto optimal front for the welded beam design problem

6.2. Disc brake design problem

The second benchmark for multi-objective optimization, considering in many papers [2, 87, 88], is the design of multiple disc brakes. There are many variables regarded in this problem as follows: r (the inner radius), R (outer radius of the discs), R (outer radius R the discs), R (the engaging force), and s (the number of the friction surface). The objectives of this problem are minimizing the overall mass and the braking time. Some constraints such as the torque, pressure, temperature, and length of the brake are considered in this problem. This problem can be expressed as follows:

$$Min\ f_1(\vec{x}) = 4.9 \times (R^2 - r^2)(s - 1) \quad (39)$$

$$Min\ f_2(\vec{x}) = \frac{9.82 \times 10^6 (R^2 - r^2)}{Fs(R^3 - r^3)} \quad (40)$$

$$g_1(\vec{x}) = 20 - (R - r) \leq 0 \quad (41)$$

$$g_2(\vec{x}) = 2.5(s + 1) - 30 \leq 0 \quad (42)$$

$$g_3(\vec{x}) = \frac{F}{3.14(R^2 - r^2)} - 0.4 \leq 0 \quad (43)$$

$$g_4(\vec{x}) = \frac{2.22 \times 10^{-3} F(R^3 - r^3)}{(R^2 - r^2)^2} - 1 \leq 0 \quad (44)$$

$$g_5(\vec{x}) = 900 - \frac{0.0266 Fs(R^3 - r^3)}{(R^2 - r^2)} \leq 0 \quad (45)$$

$$55 \leq r \leq 80, \quad (46)$$

$$75 \leq R \leq 110, \quad (47)$$

$$1000 \leq F \leq 3000, \quad (48)$$

$$2 \leq s \leq 20, \quad (49)$$

Eqs. (39) and (40) are represented as objective functions, Eqs.(41)-(45) are related to the constraints, and variables are presented in Eqs. (46)-(49). Having shown the performance of the proposed algorithm from the viewpoint of statistical analyses, results of the design of a disc brake problem are provided in Table 13 with respect to all metrics.

Table 13: Descriptive statistics results of the design of a disc brake problem

Metrics	MOEAs	Min	Max	Mean	Std.	Rank
IGD	MOVPL	3.27E+03	3.27E+03	3.27E+03	3.27E+03	1
	MOPSO	5.87E+03	9.65E+04	8.45E+04	2.12E+02	3
	MOEA/D	3.27E+03	1.77E+04	7.56E+03	1.13E+01	2
SP	MOVPL	1.13E+03	1.13E+03	1.13E+03	1.13E+03	1
	MOPSO	2.71E+03	10.18E+03	6.42E+03	4.67E+03	3
	MOEA/D	1.13E+03	8.88E+03	4.61E+03	3.15E+03	2
MS	MOVPL	1.23E+04	6.11E+04	4.29E+04	0.03E+04	3
	MOPSO	4.71E+04	9.33E+04	6.48E+04	0.04E+04	1
	MOEA/D	2.45E+04	7.61E+04	5.54E+04	0.06E+04	2
Δ	MOVPL	0.5788	0.9011	0.7933	0.0341	3
	MOPSO	0.4548	0.9123	0.8734	0.0412	2
	MOEA/D	0.4511	0.8712	0.7565	0.0311	1

The results, which are obtained from all algorithms for solving design of a disc brake problem, are generally different from those calculated in the first engineering test problem, and MOVPL has received the best rank in IGD and SP metrics, conversely, in terms of MS and Δ , the proposed algorithm is inferior to others. The comparison of the best solution of the Pareto fronts for all algorithms is depicted in Figure 9. We can see that the results of MOVPL are much superior to the results achieved by its competitors.

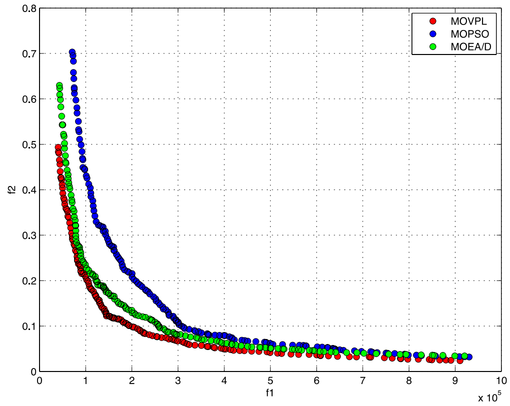


Figure 7: Pareto optimal front for the design of a disc brake problem

6.3. Speed reducer design problem

The latest problem studied in this paper is the speed reducer design problem, which is important in the context of mechanical engineering [89-91]. In this problem, the main objectives of this problem are the weight (f_1) and stress (f_1) of a speed reducer that should be minimized. The main variables of this problem include gear face width (x_1), teeth module (x_2), number of teeth of pinion (x_3), distance between bearings 1 (x_4), distance between bearings 2 (x_5), diameter of shaft 1 (x_6), and diameter of shaft 2 (x_7) as well as eleven constraints. The general scheme of this problem is shown in Figure 8.

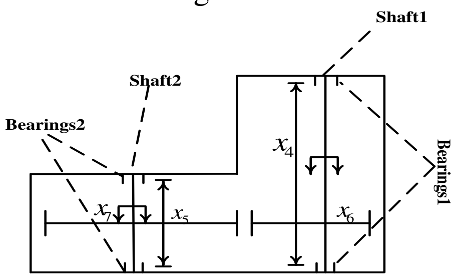


Figure 8: the speed reducer design and its features [92]

The corresponding objectives and constraints of this problem can be written as follows:

$$Min\ f_1(x) = 0.7854x_1x_2^2\left(\frac{10x_3^2}{3} + 14.933x_3 - 43.0934\right) - 1.508x_1(x_6^2 + x_7^3) + 7.477(x_6^2 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (50)$$

$$Min\ f_2(x) = \sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2} + 1.69 \times 10^7/0.1x_6^3 \quad (51)$$

$$e_1(x) = 1/27 - 1/(x_1x_2^2x_3) \geq 0 \quad (52)$$

$$e_2(x) = 1/397.5 - 1/(x_1x_2^2x_3^2) \geq 0 \quad (53)$$

$$e_3(x) = 1/1.93 - x_4^3/(x_2x_3x_6^4) \geq 0 \quad (54)$$

$$e_4(x) = 1/1.93 - x_5^3/(x_2x_3x_7^4) \geq 0 \quad (55)$$

$$e_5(x) = 40 - x_2x - 3 \geq 0 \quad (56)$$

$$e_6(x) = 12 - x_1/x_2 \geq 0 \quad (57)$$

$$e_7(x) = x_1/x_2 - 5 \geq 0 \quad (58)$$

$$e_8(x) = x_4 - 1.5x_2 - 1.9 \geq 0 \quad (59)$$

$$e_9(x) = x_5 - 1.1x_7 - 1.9 \geq 0 \quad (60)$$

$$e_{10}(x) = 1300 - f_2(x) \geq 0 \quad (61)$$

$$e_{11}(x) = 1100 - \sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2} + 1.275 \times 10^8/0.1x_7^3 \geq 0 \quad (62)$$

$$2.6 \leq x_1 \leq 3.6 \quad (63)$$

$$0.7 \leq x_2 \leq 0.8 \quad (64)$$

$17 \leq x_3 \leq 28$ (65)
$7.3 \leq x_4$ (66)
$x_5 \leq 8.3$ (67)
$2.9 \leq x_6 \leq 3.9$ (68)
$5.0 \leq x_7 \leq 5.5$ (69)

Accordingly, the objective function is expressed in Eq. (50) and Eq. (51), all the constraints of the problem are reflected in constraints Eqs. (52)-(62), and variables are defined in Eqs. (63)-(69). The statistics results from the design of the speed reducer design problem are shown in Table 14. As shown in this table, the performance of MOVPL is superior in the case of IGD, SP, and MS metrics. On the other hand, MOEA/D has better a presentation in case of Δ metrics.

Table 14: Statistics results of speed reducer design problem

Metrics	MOEAs	Min	Max	Mean	Std.	Rank
IGD	MOVPL	2.77E+03	8.85E+03	7.22E+03	2.14E+01	1
	MOPSO	4.11E+03	8.23E+04	8.14E+04	4.23E+02	2
	MOEA/D	5.51E+03	2.03E+04	8.52E+03	2.41E+03	3
SP	MOVPL	2.42E+03	7.72E+03	4.93E+03	2.11E+03	1
	MOPSO	3.90E+03	8.68E+03	5.52E+03	2.97E+03	2
	MOEA/D	4.36E+03	9.14E+03	6.67E+03	3.04E+03	3
MS	MOVPL	5.12E+04	8.61E+04	6.23E+04	0.07E+04	1
	MOPSO	4.31E+04	8.73E+04	5.79E+04	0.04E+04	2
	MOEA/D	3.23E+04	7.82E+04	5.38E+04	0.03E+04	3
Δ	MOVPL	0.5431	0.9223	0.7213	0.0332	3
	MOPSO	0.6543	0.9156	0.8312	0.0211	2
	MOEA/D	0.3123	0.8765	0.6712	0.0159	1

The Pareto fronts obtained for a standard multi-objective function, the design of speed reducer design problem, with the proposed MOVPL and comparative algorithms (MOPSO and MOEA/D) are presented in Figure 9, in which indicates the superior performance of the proposed algorithm in comparison with its rivals.

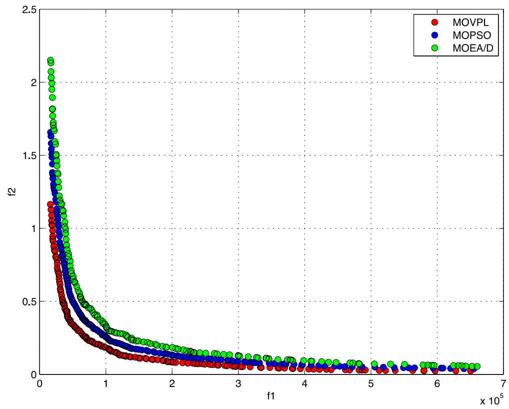


Figure 9: Pareto optimal front for the Speed reducer design problem

6.4. Statistical analysis for engineering problems

More detail analysis is required to determine the validity of the proposed method. In this section, we provide an extensive statistical test based on Wilcoxon signed-rank to show the validity of the proposed algorithm in the engineering test problem. Therefore, the results produced by the Wilcoxon signed-rank test with a significance level α = 0.05 are exhibited in Table 15.

Table 15: Statistical results of Wilcoxon signed-rank test in engineering test problems

Engineering test problems	Statistical features	MOVPL vs #							
		IGD		SP		MS		Δ	
		MOPSO	MOEA/D	MOPSO	MOEA/D	MOPSO	MOEA/D	MOPSO	MOEA/D
WBD	p-value	4.20E-05	1.40E-04	4.42E-04	2.46E-04	1.60E-04	3.91E-05	4.20E-04	1.40E-04
	R	1	1	1	1	1	1	1	1
	H	+	+	+	+	+	+	+	+
DBD	p-value	3.93E-06	6.35E-04	6.58E-04	4.49E-04	2.24E+01	6.19E-05	7.05E+01	5.23E-05
	R	1	1	1	1	0	1	0	1
	H	+	+	+	+	=	-	=	-
SRD	p-value	2.05E-04	1.92E-04	6.02E-05	5.52E-04	1.60E-04	1.60E-04	2.05E-04	1.92E-01
	R	1	1	1	1	1	1	1	1
	H	+	+	+	+	+	+	-	-

Interestingly, the obtained results in Table 15 support extensively those obtained from Table 12-Table 14. Statistically, the proposed method has a better performance in terms of IGD and SP for all engineering test problems. Even though, MOVPL obtains a better performance on welded beam design problem in all metrics, but its rivals mostly have outperformed our proposed method in disk break design and space reducer design problems in terms of MS and Δ metrics. Although, these facts have been considered as a negative point for the performance of the proposed method, but overall statistical results show the superiority of our algorithm.

7. Conclusions and directions for future research

This paper has introduced a novel multi-objective evolutionary algorithm, which was inspired by the formal and informal interaction among teams, coaches, and players in a volleyball. New procedures, strategies, and components have also been proposed in this research. We have described the main steps of MOVPL, which consider solution structure, various operators, and different mathematical expressions. The solutions of the presented algorithm are defined as teams which effort to attain the best position in the league by applying some predefined procedures associated with the volleyball game. The coach, which has the most prominent role in the match, organizes all players in consonance with the condition of their teams and the rivals. One of the main steps of this algorithm, the competition process, has simulated the rivalry action occurred between two teams to reach a better position in the league. In this step, teams explore the rival teams' condition to adopt the best strategy in the match. Also, they concentrate on learning from the best teams available in the league to move toward them. This concept is performed in the learning phase in which solution update its properties concerning three best teams. Furthermore, two new operators, which are called season transfer, and promotion and relegation process, have been implanted in the algorithm to improve search space in the proposed algorithm. The last two components, an archive, and leader selection were embedded into our algorithm to keep and retrieve the best non-dominated obtained individuals during the optimization process and to select the best teams, respectively.

To show the performance of the MOVPL algorithm and compared with two well-known MOEAs, including MOEA/D and MOPSO, ten standard test functions were used and analyzed to various metrics. We have examined all test functions statistically in terms of four metrics, and the results show that proposed method has won 72 out of 80 (10 test functions, 2 compared algorithms, and 4 metrics) computations in this class of test function, Meanwhile, the proposed algorithm has won in four test functions (UF1, UF3, UF6, and UF7) in terms of all metrics. According to these metrics, results showed that the results of the MOVPL algorithm are superior to its competitors. Finally, three eminent engineering problems were solved; the results demonstrated in this part dominate the state of the art methods. The results of these engineering test problems showed that proposed method has won 18 out of 24 computations indicating, accordingly, the performance of the proposed algorithm is significantly better than other algorithms.

Future research on MOVPL might extend the theory and application of MOEA in the scientific works, and hybridization with other well-known nature-inspired algorithms would be reasonably prolific. Moreover, various methods of parameter tuning can be explored, and to cope with uncertainty, the robust optimization method on MOVPL may be investigated in the future.

Acknowledgment

We would like to thank Professor Ponnuthurai N. Suganthan and Dr. Seyyed Mostafa Kalami Haris for giving MO test functions and providing MOPSO and MOEA/D codes. This research is also supported by CIIORG (Computational Intelligence and Intelligent Optimization Research Group) of the Persian Gulf University.

References

[1] R. Moghdani and K. Salimifard, "Volleyball Premier League Algorithm," *Applied Soft Computing,* vol. 64, pp. 161-185, 2018/03/01/ 2018, doi: https://doi.org/10.1016/j.asoc.2017.11.043.

[2] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research,* vol. 40, no. 6, pp. 1616-1624, 2013/06/01/ 2013, doi: https://doi.org/10.1016/j.cor.2011.09.026.

[3] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. d. S. Coelho, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Systems with Applications,* vol. 47, pp. 106-119, 4/1/ 2016, doi: http://dx.doi.org/10.1016/j.eswa.2015.10.039.

[4] S. Tiwari, G. Fadel, P. Koch, and K. Deb, "Performance assessment of the hybrid Archive-based Micro Genetic Algorithm (AMGA) on the CEC09 test problems," in *2009 IEEE Congress on Evolutionary Computation*, 18-21 May 2009 2009, pp. 1935-1942, doi: 10.1109/CEC.2009.4983177.

[5] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation,* vol. 1, no. 1, pp. 32-49, 2011/03/01/ 2011, doi: https://doi.org/10.1016/j.swevo.2011.03.001.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software,* vol. 69, pp. 46-61, 3// 2014, doi: http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

[7] M. A. Tawhid and V. Savsani, "Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems," *Neural Computing and Applications,* 2017/06/12 2017, doi: 10.1007/s00521-017-3049-x.

[8] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software,* vol. 32, no. 6, pp. 499-507, 6// 2001, doi: http://dx.doi.org/10.1016/S0965-9978(00)00110-1.

[9] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization,* vol. 26, no. 6, pp. 369-395, 2004// 2004, doi: 10.1007/s00158-003-0368-6.

[10] E. Demir, T. Bektaş, and G. Laporte, "The bi-objective Pollution-Routing Problem," *European Journal of Operational Research,* vol. 232, no. 3, pp. 464-478, 2014/02/01/ 2014, doi: https://doi.org/10.1016/j.ejor.2013.08.002.

[11] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuisen, *Evolutionary algorithms for solving multi-objective problems* (Genetic and evolutionary computation series). New York: Springer, 2007.

[12] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, 24-26 April 2003 2003, pp. 26-33, doi: 10.1109/SIS.2003.1202243.

[13] R. Carvalho, R. R. Saldanha, B. N. Gomes, A. C. Lisboa, and A. X. Martins, "A Multi-Objective Evolutionary Algorithm Based on Decomposition for Optimal Design of Yagi-Uda Antennas," *IEEE Transactions on Magnetics,* vol. 48, no. 2, pp. 803-806, 2012, doi: 10.1109/TMAG.2011.2174348.

[14] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation,* vol. 2, no. 3, pp. 221-248, 1994.

[15] J. Knowles and D. Corne, "The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999 1999, vol. 1, pp. 1-105 Vol. 1, doi: 10.1109/CEC.1999.781913.

[16] H. A. Abbass, R. Sarker, and C. Newton, "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001 2001, vol. 2, pp. 971-978 doi: 10.1109/CEC.2001.934295.

[17] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation,* vol. 3, no. 4, pp. 257-271, 1999, doi: 10.1109/4235.797969.

[18] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the Performance of the Strength Pareto Evolutionary Algorithm 2," in *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings*, X. Yao *et al.* Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 742-751.

[19] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-Objective Optimization: An Engineering Design Perspective," in *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005, Guanajuato, Mexico, March 9-11, 2005. Proceedings*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 14-32.

[20] D. F. Jones, S. K. Mirrazavi, and M. Tamiz, "Multi-objective meta-heuristics: An overview of the current state-of-the-art," *European Journal of Operational Research,* vol. 137, no. 1, pp. 1-9, 2/16/ 2002, doi: http://dx.doi.org/10.1016/S0377-2217(01)00123-0.

[21] H.-Z. Huang, Y.-K. Gu, and X. Du, "An interactive fuzzy multi-objective optimization method for engineering design," *Engineering Applications of Artificial Intelligence,* vol. 19, no. 5, pp. 451-460, 8// 2006, doi: http://dx.doi.org/10.1016/j.engappai.2005.12.001.

[22] A. Husseinzadeh Kashan, "League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships," *Applied Soft Computing,* vol. 16, pp. 171-200, 2014/03/01/ 2014, doi: https://doi.org/10.1016/j.asoc.2013.12.005.

[23] N. Moosavian and B. Kasaee Roodsari, "Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks," *Swarm and Evolutionary Computation,* vol. 17, pp. 14-24, 2014/08/01/ 2014, doi: https://doi.org/10.1016/j.swevo.2014.02.002.

[24] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 67-82, 1997, doi: 10.1109/4235.585893.

[25] S. B. Jha, J. K. Jha, and M. K. Tiwari, "A multi-objective meta-heuristic approach for transit network design and frequency setting problem in a bus transit system," *Computers & Industrial Engineering,* vol. 130, pp. 166-186, 2019/04/01/ 2019, doi: https://doi.org/10.1016/j.cie.2019.02.025.

[26] R. Baños, J. Ortega, C. Gil, A. L. Márquez, and F. de Toro, "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows," *Computers & Industrial Engineering,* vol. 65, no. 2, pp. 286-296, 2013/06/01/ 2013, doi: https://doi.org/10.1016/j.cie.2013.01.007.

[27] K. Wang, X. Li, and L. Gao, "A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem," *Computers & Industrial Engineering,* vol. 130, pp. 634-649, 2019/04/01/ 2019, doi: https://doi.org/10.1016/j.cie.2019.03.017.

[28] V. Hajipour, S. H. A. Rahmati, S. H. R. Pasandideh, and S. T. A. Niaki, "A multi-objective harmony search algorithm to optimize multi-server location–allocation problem in congested systems," *Computers & Industrial Engineering,* vol. 72, pp. 187-197, 2014/06/01/ 2014, doi: https://doi.org/10.1016/j.cie.2014.03.018.

[29] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Machine Learning,* journal article vol. 3, no. 2, pp. 95-99, 1988, doi: 10.1023/a:1022602019183.

[30] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization," in *Icga*, 1993, vol. 93, no. July, pp. 416-423.

[31] N. M. Duarte, A. E. Ruano, C. M. Fonseca, and P. J. Fleming, "Accelerating multi-objective control system design using a neuro-genetic approach," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, 2000 2000, vol. 1, pp. 392-397 vol.1, doi: 10.1109/CEC.2000.870322.

[32] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *International Conference on Parallel Problem Solving From Nature*, 2000: Springer, pp. 849-858.

[33] J. rey Horn, N. Nafpliotis, and D. E. Goldberg, "Multiobjective optimization using the niched pareto genetic algorithm," *IlliGAL report,* no. 93005, pp. 61801-2296, 1993.

[34] M. Erickson, A. Mayer, and J. Horn, "The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems," in *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7–9, 2001 Proceedings*, E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 681-695.

[35] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary computation,* vol. 8, no. 2, pp. 149-172, 2000.

[36] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Eurogen*, 2001, vol. 3242, no. 103, pp. 95-100.

[37] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective optimization with messy genetic algorithms," presented at the Proceedings of the 2000 ACM symposium on Applied computing - Volume 1, Como, Italy, 2000.

[38] K. Deb, "Binary and floating-point function optimization using messy genetic algorithms (IlliGAL Report No. 91004 and doctoral dissertation, Unversity of Alabama, Tuscaloosa)," *Urbana: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory,* 1991.

[39] D. E. Goldberg, K. Deb, H. Kargupta, and G. R. Harik, "RapidAccurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms," in *ICGA*, 1993, vol. 5, pp. 56-64.

[40] J. B. Zydallis, D. A. Van Veldhuizen, and G. B. Lamont, "A Statistical Comparison of Multiobjective Evolutionary Algorithms Including the MOMGA-II," in *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7–9, 2001 Proceedings*, E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 226-240.

[41] R. O. Day, "Explicit building block multiobjective evolutionary computation: methods and applications," DTIC Document, 2005.

[42] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization," in *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings*, M. Schoenauer *et al.* Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 839-848.

[43] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: region-based selection in evolutionary multiobjective optimization," presented at the Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, San Francisco, California, 2001.

[44] S. Zabihi, M. Rashidi Kahag, H. Maghsoudlou, and B. Afshar-Nadjafi, "Multi-objective teaching-learning-based meta-heuristic algorithms to solve multi-skilled project scheduling problem," *Computers & Industrial Engineering,* vol. 136, pp. 195-211, 2019/10/01/ 2019, doi: https://doi.org/10.1016/j.cie.2019.07.032.

[45] W. L. Wang, W. K. Li, Z. Wang, and L. Li, "Opposition-based multi-objective whale optimization algorithm with global grid ranking," *Neurocomputing,* vol. 341, pp. 41-59, 2019/05/14/ 2019, doi: https://doi.org/10.1016/j.neucom.2019.02.054.

[46] P. M. Pradhan and G. Panda, "Solving multiobjective problems using cat swarm optimization," *Expert Systems with Applications,* vol. 39, no. 3, pp. 2956-2964, 2012/02/15/ 2012, doi: https://doi.org/10.1016/j.eswa.2011.08.157.

[47] A. Sadollah, H. Eskandar, and J. H. Kim, "Water cycle algorithm for solving constrained multi-objective optimization problems," *Applied Soft Computing,* vol. 27, pp. 279-298, 2015/02/01/ 2015, doi: https://doi.org/10.1016/j.asoc.2014.10.042.

[48] A. Got, A. Moussaoui, and D. Zouache, "A guided population archive whale optimization algorithm for solving multiobjective optimization problems," *Expert Systems with Applications,* vol. 141, p. 112972, 2020/03/01/ 2020, doi: https://doi.org/10.1016/j.eswa.2019.112972.

[49] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016/05/01/ 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[50] T. Liu, L. Jiao, W. Ma, J. Ma, and R. Shang, "A new quantum-behaved particle swarm optimization based on cultural evolution mechanism for multiobjective problems," *Knowledge-Based Systems,* vol. 101, pp. 90-99, 2016/06/01/ 2016, doi: https://doi.org/10.1016/j.knosys.2016.03.009.

[51] H. Zhang, J. Sun, T. Liu, K. Zhang, and Q. Zhang, "Balancing exploration and exploitation in multiobjective evolutionary optimization," *Information Sciences,* vol. 497, pp. 129-148, 2019/09/01/ 2019, doi: https://doi.org/10.1016/j.ins.2019.05.046.

[52] L. Cao, L. Xu, E. D. Goodman, and H. Li, "Decomposition-based evolutionary dynamic multiobjective optimization using a difference model," *Applied Soft Computing,* vol. 76, pp. 473-490, 2019/03/01/ 2019, doi: https://doi.org/10.1016/j.asoc.2018.12.031.

[53] H. V. Hultmann Ayala, P. Keller, M. d. F. Morais, V. C. Mariani, L. d. S. Coelho, and R. V. Rao, "Design of heat exchangers using a novel multiobjective free search differential evolution paradigm," *Applied Thermal Engineering,* vol. 94, pp. 170-177, 2016/02/05/ 2016, doi: https://doi.org/10.1016/j.applthermaleng.2015.10.066.

[54] Z. Zhang, "Fast multiobjective immune optimization approach solving multiobjective interval number programming," *Swarm and Evolutionary Computation,* vol. 51, p. 100578, 2019/12/01/ 2019, doi: https://doi.org/10.1016/j.swevo.2019.100578.

[55] A. S. M. Lacerda and L. S. Batista, "KDT-MOEA: A multiobjective optimization framework based on K-D trees," *Information Sciences,* vol. 503, pp. 200-218, 2019/11/01/ 2019, doi: https://doi.org/10.1016/j.ins.2019.07.011.

[56] J. Liang *et al.*, "Multimodal multiobjective optimization with differential evolution," *Swarm and Evolutionary Computation,* vol. 44, pp. 1028-1059, 2019/02/01/ 2019, doi: https://doi.org/10.1016/j.swevo.2018.10.016.

[57] T. C. Bora, V. C. Mariani, and L. d. S. Coelho, "Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm," *Applied Thermal Engineering,* vol. 146, pp. 688-700, 2019/01/05/ 2019, doi: https://doi.org/10.1016/j.applthermaleng.2018.10.020.

[58] J. X. Vianna Neto, E. J. Guerra Junior, S. R. Moreno, H. V. Hultmann Ayala, V. C. Mariani, and L. d. S. Coelho, "Wind turbine blade geometry design based on multi-objective optimization using metaheuristics," *Energy,* vol. 162, pp. 645-658, 2018/11/01/ 2018, doi: https://doi.org/10.1016/j.energy.2018.07.186.

[59] J. D. Schaffer and J. J. Grefenstette, "Multi-Objective Learning via Genetic Algorithms," in *Ijcai*, 1985, vol. 85, pp. 593-595.

[60] J. D. Knowles and D. W. Corne, "M-PAES: A memetic algorithm for multiobjective optimization," in *evolutionary computation, 2000. Proceedings of the 2000 congress on*, 2000, vol. 1: IEEE, pp. 325-332.

[61] W. Lin *et al.*, "Multi-objective teaching–learning-based optimization algorithm for reducing carbon emissions and operation time in turning operations," *Engineering Optimization,* vol. 47, no. 7, pp. 994-1007, 2015/07/03 2015, doi: 10.1080/0305215X.2014.928818.

[62] M. P. Kleeman, G. B. Lamont, A. Cooney, and T. R. Nelson, "A multi-tiered memetic multiobjective evolutionary algorithm for the design of quantum cascade lasers," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2007: Springer, pp. 186-200.

[63] M. J. Reddy and D. Nagesh Kumar, "Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation," *Hydrological Processes: An International Journal,* vol. 21, no. 21, pp. 2897-2909, 2007.

[64] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," in *International conference on parallel problem solving from nature*, 2000: Springer, pp. 839-848.

[65] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001: Morgan Kaufmann Publishers Inc., pp. 283-290.

[66] H. R. Hassanzadeh and M. Rouhani, "A Multi-objective Gravitational Search Algorithm," in *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*, 28-30 July 2010 2010, pp. 7-12, doi: 10.1109/CICSyN.2010.32.

[67] J. rey Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence*, 1994, vol. 1: Citeseer, pp. 82-87.

[68] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation,* vol. 6, no. 2, pp. 182-197, 2002.

[69] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-report,* vol. 103, 2001.

[70] J. Andersson and P. Krus, "Multiobjective optimization of mixed variable design problems," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2001: Springer, pp. 624-638.

[71] C. A. C. C. Coello and G. T. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2001: Springer, pp. 126-140.

[72] G. T. Pulido and C. A. C. Coello, "The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2003: Springer, pp. 252-266.

[73] L. Ding, Y. Chen, and L. Kang, "A new multiobjective evolutionary algorithm: OMOEA," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, 2003, vol. 2: IEEE, pp. 898-905.

[74] S. Zeng, S. Yao, L. Kang, and Y. Liu, "An efficient multi-objective evolutionary algorithm: OMOEA-II," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2005: Springer, pp. 108-119.

[75] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation,* vol. 8, no. 3, pp. 256-279, 2004, doi: 10.1109/TEVC.2004.826067.

[76] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2002, vol. 2: IEEE, pp. 1051-1056.

[77] F. van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Sciences,* vol. 176, no. 8, pp. 937-971, 4/22/ 2006, doi: http://doi.org/10.1016/j.ins.2005.02.003.

[78] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, 12-17 May 2002 2002, vol. 1, pp. 825-830, doi: 10.1109/CEC.2002.1007032.

[79] K. Deb, "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation,* vol. 7, no. 3, pp. 205-230, 1999/09/01 1999, doi: 10.1162/evco.1999.7.3.205.

[80] R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, "A multi-objective artificial bee colony algorithm," *Swarm and Evolutionary Computation,* vol. 2, pp. 39-52, 2// 2012, doi: http://doi.org/10.1016/j.swevo.2011.08.001.

[81] S. M. Mousavi, V. Hajipour, S. T. A. Niaki, and N. Alikar, "Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: two calibrated meta-heuristic algorithms," *Applied Mathematical Modelling,* vol. 37, no. 4, pp. 2241-2256, 2013.

[82] S. M. Mousavi, J. Sadeghi, S. T. A. Niaki, and M. Tavana, "A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NRGA, and MOPSO," *Applied Soft Computing,* vol. 43, pp. 57-72, 2016.

[83] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2002.

[84] V. Hajipour, P. Fattahi, M. Tavana, and D. Di Caprio, "Multi-objective multi-layer congested facility location-allocation problem optimization with Pareto-based meta-heuristics," *Applied Mathematical Modelling,* vol. 40, no. 7, pp. 4948-4969, 2016/04/01/ 2016, doi: http://dx.doi.org/10.1016/j.apm.2015.12.013.

[85] H. Maghsoudlou, M. R. Kahag, S. T. A. Niaki, and H. Pourvaziri, "Bi-objective optimization of a three-echelon multi-server supply-chain problem in congested systems: Modeling and solution," *Computers & Industrial Engineering,* vol. 99, pp. 41-62, 2016/09/01/ 2016, doi: http://dx.doi.org/10.1016/j.cie.2016.07.008.

[86] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for global optimization," *Applied Soft Computing,* vol. 30, pp. 614-627, 5// 2015, doi: https://doi.org/10.1016/j.asoc.2015.02.014.

[87] K. Deb, "Evolutionary algorithms for multi-criterion optimization in engineering design," *Evolutionary algorithms in engineering and computer science,* vol. 2, pp. 135-161, 1999.

[88] T. Ray and K. M. Liew, "A Swarm Metaphor for Multiobjective Design Optimization," *Engineering Optimization,* vol. 34, no. 2, pp. 141-153, 2002/01/01 2002, doi: 10.1080/03052150210915.

[89] C. A. Coello Coello and G. T. Pulido, "Multiobjective structural optimization using a microgenetic algorithm," *Structural and Multidisciplinary Optimization,* journal article vol. 30, no. 5, pp. 388-403, November 01 2005, doi: 10.1007/s00158-005-0527-z.

[90] A. Farhang-Mehr and S. Azarm, "Entropy-based multi-objective genetic algorithm for design optimization," *Structural and Multidisciplinary Optimization,* journal article vol. 24, no. 5, pp. 351-361, November 01 2002, doi: 10.1007/s00158-002-0247-6.

[91] W. Gong, Z. Cai, and L. Zhu, "An efficient multiobjective differential evolution algorithm for engineering design," *Structural and Multidisciplinary Optimization,* journal article vol. 38, no. 2, pp. 137-157, April 01 2009, doi: 10.1007/s00158-008-0269-9.

[92] G. Dhiman and V. Kumar, "Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems," *Knowledge-Based Systems,* vol. 150, pp. 175-197, 2018/06/15/ 2018, doi: https://doi.org/10.1016/j.knosys.2018.03.011.