Soft Computing
https://doi.org/10.1007/s00500-023-08603-0

OPTIMIZATION

MOOTLBO: a new multi-objective observer–teacher–learner-based optimization

Siyamak Doroudi[1] · Ahmad Sharafati[1,2] · Seyed Hossein Mohajeri[3]

Accepted: 22 May 2023

Abstract

Multi-objective problems are seen in most fields and represent challenges for researchers to solve them. Although a wide range of developed techniques has been provided to tackle these problems, the complexity of the proposed solutions requires the use of alternative algorithms. The innovation of this article is a multi-objective version of observer–teacher–learner-based optimization (OTLBO) called multi-objective observer–teacher–learner-based optimization (MOOTLBO), and an external archive has been employed to save the non-dominated Pareto-optimal solutions so far. This archive chooses the solutions using a leader selection strategy and a roulette wheel. Accordingly, a mutation operator was also added to the algorithm. To prove the effectiveness of the proposed algorithm, nineteen standard test functions were used and compared via MOPSO, MOGWO and MOMPA algorithms. The Taguchi–grey relational method was employed to adjust the parameters of the investigated algorithms. The ability of the proposed algorithm was investigated by seven metrics. The results show that MOOTLBO outperforms the other algorithms in fourteen of the nineteen test problems and produces viable results.

Keywords Multi-objective optimization · Multi-criterion optimization · Evolutionary algorithm · Observer–teacher–learner-based optimization

1 Introduction

Multi-objective optimization involves several objectives. Contrary to single-objective optimization, a set of solutions involves the optimum solutions for a multi-objective problem (Coello et al. 2007). Various optimization techniques have been developed to solve these types of problems. Many efforts have been made to apply analytical techniques to solve optimization problems in different fields, such as applied mathematics and computer science. Stagnation in local optima is a major drawback in most of these approaches, leading to the use of evolutionary and heuristic optimization techniques to solve optimization problems (Coello Coello 2006).

In addition to the above advantage, meta-heuristic algorithms are to some extent simple. This simplicity facilitates development of new meta-heuristics, combination of two or more meta-heuristics and optimizing existing stochastic meta-heuristics in addition to expediting the implementation of these algorithms. Furthermore, meta-heuristic algorithms are highly adaptable, eliminating the need for structural modifications to cater to different applications. These algorithms treat most problems as black-box assumptions, focusing solely on the inputs and outputs of a problem. As a result, designers need only present their problems accurately to the meta-heuristic algorithms to obtain optimal solutions for optimization problems (Mirjalili et al. 2014).

Unlike gradient-based optimization techniques, most meta-heuristic algorithms optimize the problems randomly and do not require computing the search space derivative to find the optimal solution, making them excellent for

✉ Ahmad Sharafati
asharafati@srbiau.ac.ir

[1] Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] New Era and Development in Civil Engineering Research Group, Scientific Research Center, Al-Ayen University, Nasiriyah, Thi-Qar 64001, Iraq

[3] Department of Civil and Environmental Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran

Published online: 05 July 2023

solving real problems with expensive or unknown derivability (Mirjalili et al. 2017). There are usually two approaches to dealing with multi-objective problems using random optimization techniques: The first approach involves reducing the multi-objective optimization problem to a single-objective problem by emphasizing a specific optimum Pareto solution. To find multiple solutions, the algorithm must be run multiple times as each simulation yields a different solution (Deb 2001). Nonetheless, this method fails to determine certain aspects of Pareto optimization (Das and Dennis 1998; Kim and De Weck 2005). The second approach is to develop algorithms to preserve multi-objective formulas and find the optimum Pareto set in a single run. Any Pareto front can be determined by this approach (S. Mirjalili et al. 2017).Several multi-objective meta-heuristic methods have been developed to tackle optimization problems. Among these, the non-dominated sorting genetic algorithm (NSGA-II) is one of the most widely used algorithms (Deb et al. 2002). This algorithm begins with a random population, and the individuals are classified based on the non-dominated sorting method. The non-domination level determines the fitness of each individual. The next population is generated using crossover and mutation operators and is merged with the previous population. The new population is then reordered using the non-dominated sorting approach, with higher non-domination levels given priority in selecting new individuals for the ultimate population of an iteration. This process of selecting non-dominated individuals is repeated until the population is as large as the initial population, and the stages are performed until the termination criterion is met.

The second popular multi-objective meta-heuristic is multi-objective particle swarm optimization (MOPSO) (Coello et al. 2004), which has been developed based on particle swarm optimization (PSO) concepts (Eberhart et al. 2001). The initial population is first created. The non-dominated individuals of the population are saved in an external archive. Additionally, a mutation operator is integrated into MOPSO to increase the randomness and variety of the solutions. The external archive comprises two main components: the archive controller and the grid. The archive controller is responsible for determining whether a new solution should be added to the archive. If the new solution cannot be dominated by any of the existing archive members, it is added to the archive. Finally, the adaptive grid technique is turned on if the archive is full. The network component must maintain as much diversity in the archiving solutions as possible. In this way, the objective space is divided into several partitions and a new solution inside the grid must be guided to the partition of the grid with the fewest number of particles (Reyes–Sierra and Coello 2006).

The multi-objective grey wolf optimizer (GWO) is another popular meta-heuristic developed in recent years (Mirjalili et al. 2016). An archive is integrated into the GWO algorithm to save the non-dominated solutions. The GWO additionally has a grid method to enhance non-dominated solutions in the archive. The techniques of these two components are similar to those of the MOPSO algorithm. However, to update and replace the accessible solutions in the archive, a leader selection technique based on three non-dominated solutions has been used. In contrast, MOPSO employs one non-dominated solution for leader selection. This causes a wider exploration of the search space by search agents.

The multi-objective marine predator algorithm (MOMPA) (Jangir et al. 2021) is one of the latest algorithms for multi-objective optimization. It is based on the marine predator algorithm and takes inspiration from the biological interaction between predator and prey. Similar to the NSGA-II, MOMPA is designed using elitist, non-dominated sorting and crowding distance mechanisms. The algorithm has been compared with the multi-objective water cycle algorithm (MOWCA) (Sadollah et al. 2015), multi-objective symbiotic organism search (MOSOS) (Tran et al. 2016) and the multi-objective moth flame optimizer (MOMFO) (Vikas and Nanda, 2016). The results illustrate that the proposed method is superior to other algorithms. In addition to the mentioned algorithms, other popular algorithms include.

Strength Pareto evolutionary algorithm (SPEA) (Zitzler 1999), Pareto archived evolution strategy (PAES) (Knowles and Corne 1999), multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang and Li 2007), multi-objective cat swarm optimization (MOSCO) (Pradhan and Panda 2012), the multi-objective bat algorithm (Yang 2012), multi-objective teaching–learning-based optimization algorithm (MOTLBO) (Zou et al. 2013), multi-objective dragonfly algorithm (MODA) (Mirjalili 2016), multi-objective ant lion optimization (MOALO) (Mirjalili et al. 2017), multi-objective grasshopper algorithm (Mirjalili et al. 2018), hybrid multi-objective cuckoo search (HMOCS) (Zhang et al. 2018), non-dominated sorting ions motion algorithm (NSIMO) (Buch and Trivedi 2020), multi-objective arithmetic optimization algorithm (MOAOA) (Premkumar et al. 2021) and multi-objective equilibrium optimizer (MOEO) (Premkumar et al. 2022).

Although the algorithms mentioned above have proved to be useful in multi-objective optimization, they still have some limitations that have prompted researchers to enhance the mechanisms of these methods. For example, the NSGA-II algorithm may suffer from a lack of diversity due to concentrated points in a small area, which can hinder global optimization (OuYang et al. 2008).

Algorithms that rely on an external repository, such as MOPSO, may face issues with updating procedures and maintaining solution diversity (Cabrera and Coello 2010). Accordingly, improving the leader selection methods can be a valuable solution to address these challenges.

This study aims to develop a new multi-objective model with a slightly modified leader selection mechanism compared to MOGWO, MOPSO and MOMPA methods to update and explore the solution space efficiently. The no free lunch (NFL) theorem (Wolpert and Macready 1997) posits that the performance of an optimizer in a certain set of problems cannot guarantee the same performance in dealing with other problems, allowing researchers to either propose new algorithms or improve the available ones. This present research is based on this foundation in which a new multi-objective optimization algorithm—multi-objective observer–teacher–learner-based optimization (MOOTLBO)—is suggested based on the optimizer observer–teacher–learner-based optimization (OTLBO) (Mohsen Shahrouzi et al. 2017). The new multi-objective version was developed by adding an external repository, mutation, grid and leader operators. The steps are as follows:

i. A mutation operator is added to the OTLBO algorithm to prevent local search and achieve high convergence speed.
ii. A grid mechanism is merged into OTLBO to obtain a more proper distribution of the Pareto front by keeping the diversity in selecting the non-dominated solutions available in the external repository.
iii. To update and change the accessible solutions in the repository, a leader selection technique based on the teacher operator is suggested. The leaders are updated for each position.

2 Methodology

The general framework of multi-objective algorithms is based on an almost identical population. Non-dominant solutions are saved in the repository at each optimization stage so that the algorithm can try to make them more favourable in the next iteration. The application of various methods for enhancing non-dominated solutions is the factor that differentiates an algorithm from others. In the following section, a new multi-objective version of OTLBO is put forward as a substitution for the present algorithms for resolving multi-objective optimization issues.

2.1 Multi-objective observer–teacher–learner-based optimization (MOOTLBO)

The development of the OTLBO algorithm took inspiration from the TLBO (teaching–learning-based optimization) optimizer. This algorithm has been used to optimize different engineering science problems in recent years (Doroudi et al. 2021; Lavasani and Doroudi 2020). TLBO is a meta-heuristic introduced by Rao et al. (Rao et al. 2011). It is a population-based method that simulates the objective space search process (knowledge growth process) in two separate stages: The first stage focuses on enhancing the average grade level of the students (populations) through the teachers, while the second stage involves learning through interactions among the students. An additional method has been introduced in the OTLBO algorithm for building and leading new solutions. In this process, the information regarding students is randomly obtained from various students to create a new solution named observer. This strategy thus produces an additional way of analysing the search space. Hence, the capability of random search by the algorithm is expected to improve (Shahrouzi 2011).

Elitism has shown to prevent the loss of the best solutions obtained till now via iterations of the search method. In the first generation, the most proper population solution is saved as the elite solution. Then, the best solutions of the new generation are compared and updated in each iteration through the teacher phase mechanism, the observer and the learner. The OTLBO algorithm is as follows:

2.1.1 OTLBO algorithm

The OTLBO algorithm is divided into three sections: the teacher phase, the learner phase and the observer phase. The section below presents a detailed, step-by-step approach for implementing the OTLBO algorithm:

Step 1: Class start: generating a population matrix with N_p classmates, between X_j^L—the lower limit—and X_j^U—the upper limit—for each subject j using Eq. (1):

$$X_j^i = \text{round}\left(X_j^L + \text{rand} \times \left(X_j^U - X_j^L\right)\right) \quad i \in \{1.2\ldots.N_p\} \cdot j \in \{1.2\ldots.N_d\} \tag{1}$$

N_d is the number of subjects or components of any classmate vector X_i, and rand is a function that generates random numbers in the [0–1] range.

Step 2 Elitist Update: In the first iteration, the most proper classmate is saved as the elite solution. In each next iteration, the previously saved elite is compared with the best current individuals, and the best is selected as the solution for the new elites.

Step 3 The teacher phase or observer phase is performed with identical probability:

a. Teacher phase:

The teacher phase refers to learning from the teacher. The teacher vector, X^{Teacher}, can be the best solution for the present population or the elite solution until the present generation. Hence, the search agents are guided towards the global best solution until the present repetition, forcing the student to move towards the best-found location in the same direction as the average class. In this procedure, the teaching factor, which varies randomly between 1 and 2, is employed. Therefore, the effort is to improve the average class score for any *i*th student using the teacher operator according to the following equation:

$$X^{\text{new}} = X_i + \text{rand} \times \left(X^{\text{Teacher}} - T_f X^{\text{Mean}}\right) \tag{2}$$

X^{Teacher} is the current elite solution, and T_f denotes a teaching factor that randomly switches to either 1 or 2. Also, any component X^{Mean} in the above relation is determined by averaging over all classmates in their *j*th subject.

b. The observer phase

In the observer phase, a search agent creates a new solution by utilizing the memory of the present population. This process selects the *I*th component (the *I*th decision variable) of the *K*th student and considers it for the observer vector. The procedure is repeated for the number of N_d decision variables to create the observer solution for the *K*th student. In the absence of the teacher in the class, such an observer takes its position to guide other students. Since this strategy introduces an extra way to analyse the search space, it is expected to improve the random searching capability of the algorithm. An observer solution is generated for any *i*th student, X^{EXP} is calculated according to the following equation:

$$\begin{aligned} &X_l^{\text{EXP}} = X_l^k \\ &k = \text{randi}(N_P) \cdot l = 1.2\ldots.N_d \end{aligned} \tag{3}$$

Randi (*Np*) generates a random integer between 1 and N_p. N_p is the number of members of the population. The observer vector is generated via receiving each component (subject) related to the k student,N_d is the number of components (decision variables) and X^{new} is selected as the X^{EXP}.

Step 4 Evaluation of X^{new}: if X^{new} is more proper than the current X^i, it will substitute it.

$$X_i = X^{\text{new}} \quad \text{if} \quad \text{Fit}(X^{\text{new}}) > \text{Fit}(X^{\text{new}}) \tag{4}$$

Step 5 Learner Phase.

Apart from the teacher and observer phases, the learner phase is also available for students to improve their knowledge. During the learner phase, students learn through interactions with each other. They engage in group discussions, presentations and formal communications with other students randomly. By doing so, learners can acquire new knowledge from those who possess higher knowledge. Hence, students interact with each other during this phase to elevate their scientific level through the following steps:

- A pair of students with different numbers are selected randomly: $i \neq j$
- X^{new} is updated through the following equations:

$$\begin{aligned} &X^{\text{new}} = (X^i + \text{rand} \times (X^j - X^i)) \quad \text{if} \quad \text{Fit}(X^j) > \text{Fit}(X^i) \\ &X^{\text{new}} = (X^i + \text{rand} \times (X^i - X^j)) \quad \text{otherwise} \end{aligned} \tag{5}$$

- X^{new} and current X^i are evaluated and compared. If X^{new} is fitter than X^i, it will substitute it as Eq. (4).

Step 6 Steps four to six 4–6 are repeated for all classmates.

Step 7 If the number of iterations has not reached the maximum iteration, one is added to it and returns to Step 3. Figure 1 illustrates the flowchart of the OTLBO algorithm.

2.2 Description of integrated components

A common method for demonstrating multi-objective optimization problems and basic definitions of multi-objective optimization are provided below:

$$\begin{aligned} &\frac{\text{Min}}{\text{Max}} F(\vec{x}) = \{f_1(\vec{x}) \cdot f_2(\vec{x}) \ldots f_o(\vec{x})\} \\ &\text{subject to } g(\vec{x}) \geq 0.i = 1.2\ldots.m \; h(\vec{x}) = 0.i = 1.2\ldots.p \\ &L_i \leq x_i \leq U_i.i = 1.2\ldots.N \end{aligned} \tag{6}$$

where O is the number of objective functions, m is the number of inequality constraints, p is the number of equality constraints, N is the number of variables,h_i is the equality constraints of the *i*th variable, g_i is the inequality constraints of the *i*th variable, and L_i and U_i are the lower and upper bounds of the *i*th variable, respectively.

The most useful definitions in this field are as follows:

Definition 1 (*Pareto Dominance*): Suppose there are two solution vectors, namely x and y; x dominates y if and only if:

$$\begin{aligned} &\forall i \in \{1.2\ldots.k\} : f_i(\vec{x}) \leq f_i(\vec{y}) \wedge \exists i \in \{1.2\ldots.k\} \\ &\quad : f_i(\vec{x}) \leq f_i(\vec{y}) \end{aligned} \tag{7}$$

Definition 2 (*Pareto Optimality*): A solution vector x is located on the Pareto-optimal front if and only if.

$$\nexists\, \vec{y} \in X | F(\vec{y}) \prec F(\vec{x}) \tag{8}$$

Definition 3 (*Pareto-Optimal Set*): The set that includes all Pareto-optimal solutions is called the Pareto set (P_S).

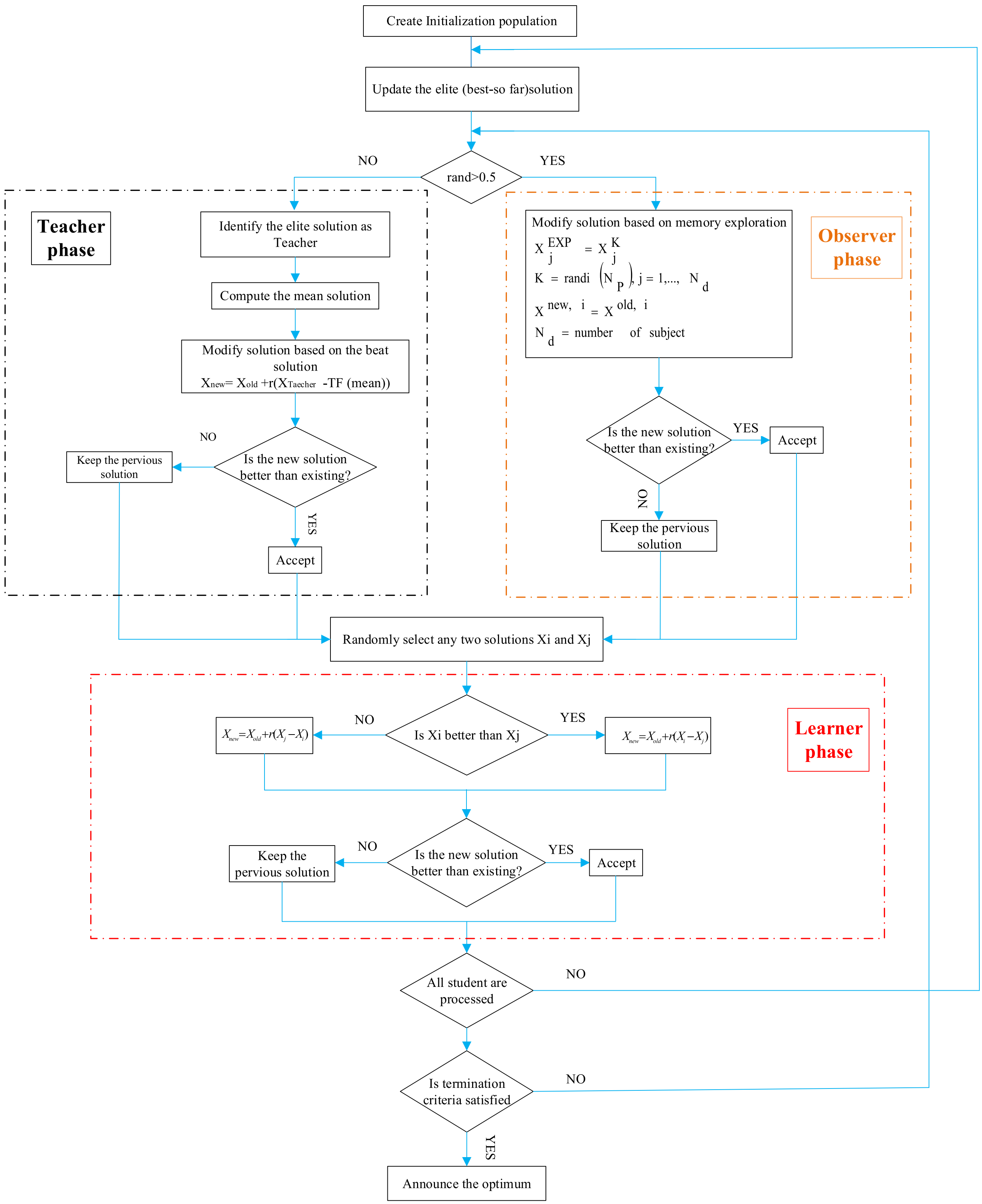

Fig. 1 Flowchart of OTLBO

$$P_s := \{x.y \in X | \exists F(\vec{y}) \succ F(\vec{x})\} \tag{9}$$

Definition 4 (*Pareto-Optimal Front*): A set containing the value of objective functions for the Pareto solutions set:

$$P_f := \{F(\vec{x}) | \vec{x} \in P_s\} \tag{10}$$

Here we merged three new components to perform multi-objective optimization by OTLBO. The first component is the mutation operator, which operates on students (population members) as well as the amplitude of each problem decision variable that must be solved. The second one is the archive, which is responsible for storing the current non-dominated Pareto-optimal solutions. The third component is the leader selection approach, which helps in selecting a leader solution from the archive. Each component will be described in detail below.

2.2.1 Mutation operator

OTLBO has a high convergence speed, which might be disadvantageous for multi-objective optimization. An OTLBO-based algorithm might converge to a false front. This means that the optimization model for a local optimum has converged to a global optimization.

To overcome this problem, a mutation operator has been created. The pseudocode of the mutation operator is presented in Fig. 2.

As shown in Fig. 2, every time the mutation operator is applied, a number of decision variables of a particle are randomly selected, and based on their number, the decision

```
% particle = particle to be mutated
%mu = mutation rate
%It = current iteration
% MaxIt = maximum of iteration
% position= decision making any student
%lb = lower bound of any desiction variable
%ub = upper bound of any desition variable
function = mutation operation (particle.MaxIt.mu.lb.ub)
Pm = (1 − (it − 1)/(MaxIt − 1)^(1/mu)

% random selection of several decision variables
% Which position is the array of selected decision variables.
Which position = random [0, number of decision variables]
dx = Pm × (ub − lb)
% N is number of Which position array elements
for i=1:N
lower bound = which position(i) − dx
upper bound = which position(i) + dx
If lower bound < lb then lower bound=lb
if upper bound > ub then upper bound = ub
Which position(i) = real random(lower bound , upper bound)
end
```

Fig. 2 Pseudocode of mutation operator

variables of a particle (student) change, and a new solution is obtained.

At the beginning of the search, this operator attempts to explore every particle (student). After that, according to the number of iterations, the number of particles influenced by the mutation operator decreases.

2.2.2 External repository

The external repository is composed of two main sections: the archive controller and the grid. The first section is the archive controller, which controls the archive when a solution is entered or when the archive is filled. If the archive capacity is filled, the grid mechanism will be activated to readjust the partitioning of the objective space. The grid is a space formed by the hypercube. Each hypercube can be interpreted as a geographical region with fewer individuals. In order to preserve diversity, the solution with the least role in diversity must be eliminated. Hence, the elimination mechanism is designed so that there is a higher possibility of elimination for those solutions that are present in the most crowded archive segmentation (hypercube).

2.2.3 Leader selection

The third component is the leader selection mechanism. Leader selection in MOPSO updates the location of particles based on a non-dominated solution obtained till now. However, MOGWO utilizes three non-dominated solutions found thus far. The comparison between these two models has shown better performance for MOGWO than MPOSO. According to Mirjalili et al. (2016), MOGWO outperforms MPOSO in a comparison between the two models (Mirjalili et al. 2016). In the multi-objective OTLBO model, non-dominated solutions are used to update the location of students multiple times. Specifically, at each iteration, three non-dominated solutions are selected as leaders based on Eq. (11) to calculate at least three new solutions. The final solutions are then obtained using Eq. (12). Figure 3 shows the leader selection mechanism and new solutions obtained in the MOOTLBO algorithm.

$$X_m^{It} = X_m^{It} + \text{rand} \times \left(L_m - T_f X_j^{It}\right) \tag{11}$$

$$X_{It}^{\text{new}} = \sum_{m=1}^{3} \left(\frac{X_m^{It}}{3}\right) \tag{12}$$

$$m = 1.2.3.j = 1.2\ldots.N_P$$

It is the current iteration, L_m denotes the *m*th selected leader, X_j^{It} is the *j*th student in the *It*th iteration, X_m^{It} shows the *m*th obtained solution in the *It* iteration, and X_{It}^{new} reveals the obtained solution in the *It* iteration.

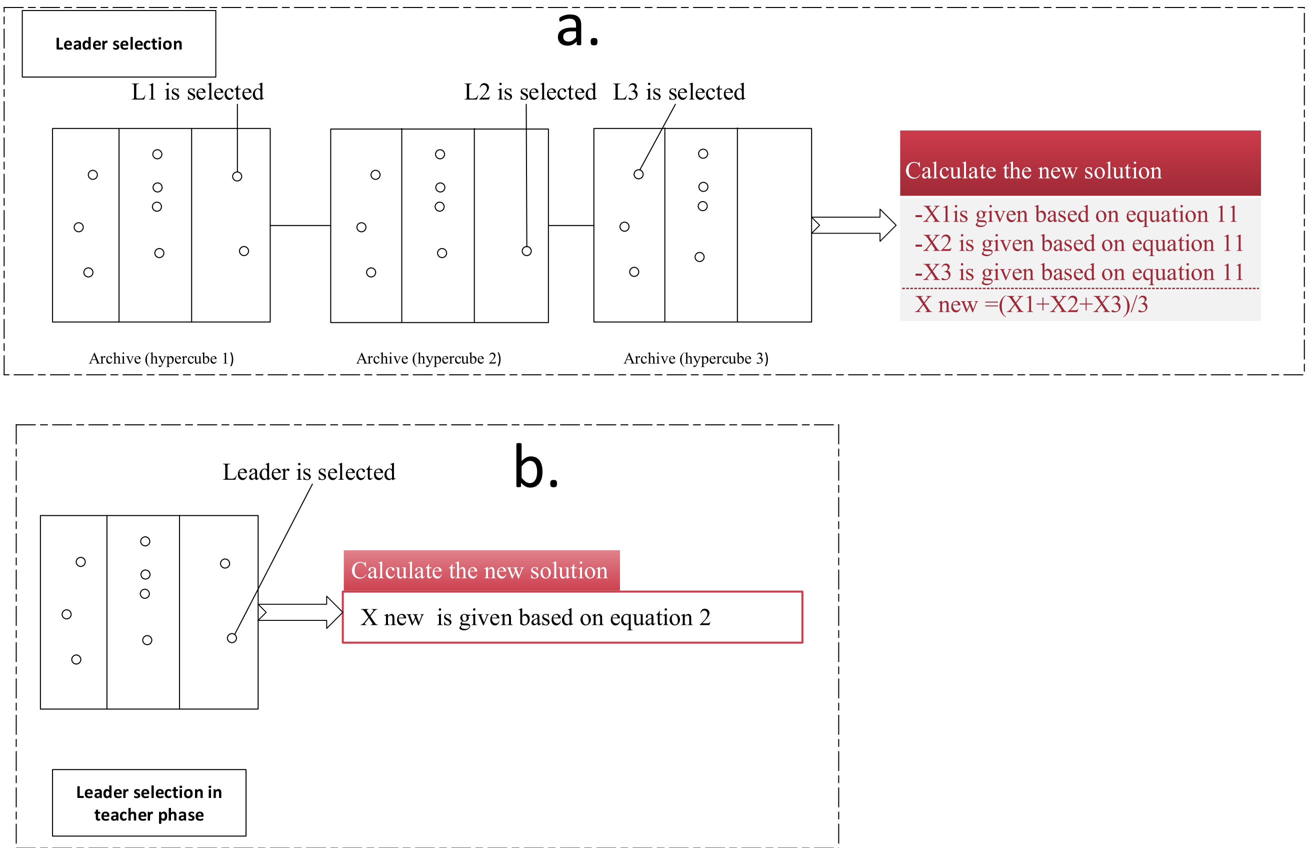

Fig. 3 Leader selection mechanism and new solutions obtained by MOOTLBO algorithm. **a**. Main loop of algorithm, **b**. teacher phase

Maintaining the diversity of solutions is a top priority in leader selection, and the mechanism for selecting leaders has been specifically designed to address it. Once the number of solutions obtained in the hypercube is decreased, the possibility of selecting a hypercube for the selection of leaders among them is enhanced. The leader selection component is performed with the following probability for each hypercube via the roulette wheel method:

$$P_i = \frac{c}{N_i} \tag{13}$$

where c is a constant number bigger than one and N is the total number of Pareto-optimal solutions found in the ith segment (hypercube).

Given that we are required to choose three leaders, it is important to note that there may be some exceptional circumstances. If there are three solutions in the low-population segmentation, three solutions are selected randomly according to Eq. (11) to calculate X_1^{It}, X_2^{It} and X_3^{It}. The second low-population hypercube is also discovered for the selection of other leaders when there are fewer than three solutions in the hypercube. If the second low-population hypercube has one solution, there is an identical scenario; hence, the third leader L_3 must be selected from the third low-population hypercube. Here, we prevented the selection of similar leaders using this method. Therefore, the search is always performed in the unexplored/unexposed areas of the search space since the leader selection mechanism supports the least crowded hypercubes and increases solution diversity.

In addition, since a non-dominated solution is selected by the teacher selection operator among the total population (students) in the teacher phase, the leader selection mechanism is applied, and the term X^{Teacher} in Eq. (2) is selected via this mechanism. Also, Fig. 3b shows the leader selection mechanism and the new solution obtained in the teacher phase.

According to the above, three leaders are selected in each iteration, and a new solution is obtained. If the teacher's phase, which is random, is applied, another leader is selected, and another new solution is obtained based on the leader selection mechanism.

Implementing the proposed MOOTLBO consists of the following steps:

Step 1 In solution space (S), first, generate a random population (P_o).

Step 2 For the initial population (P_o), evaluate objective function (F).

Step 3 Determine non-dominated solutions for the generated population (P_o), save them to the external repository, and create the grid.

Step 4 Apply the leader selection mechanism, create a new solution, and assess domination for the *i*th population (P_i).

Step 5 Apply the leader selection mechanism, create a new solution, and assess domination for the *i*th population (P_i).

Step 6 Select a teacher or observed phase randomly, create a new solution, and consider domination for *i*th population (P_i).

Step 7 Apply the learner phase, create a new solution, and consider domination for the *i*th population (P_i).

Step 8 Repeat steps 4–7 for the number of students (population), then save the non-dominated population to the repository.

Step 9 Determine the domination of new repository members and update the grid.

Step 10 Whenever the external repository becomes full, remove the dominant solutions in the external repository.

Step 11 The steps mentioned earlier are repeated to satisfy the termination criterion.

2.3 Metrics

The performance of the proposed algorithm must be investigated from several perspectives, such as convergence (accuracy), coverage (distribution), spread and uniformity (Branke et al. 2001). The main challenge in multi-objective optimization by random algorithms is the contradiction of all perspectives. If an algorithm is only focused on improving accuracy in non-dominated solutions, the coverage or spread will be weak. On the other hand, merely considering the coverage results in negative effects on the other attributes of non-dominated solutions. The proposed multi-objective algorithm frequently balances the contrasting attributes to find the exact approximation of Pareto-optimal solutions with an even distribution in all objectives. Generally, the main convergence mechanism in the single-objective version of an algorithm suffices convergence (Eberhart and Kennedy 1995; Kennedy 2010). The mechanism of selecting archive, leader and mutation operator in the proposed multi-objective algorithm ensures the diversity of solutions and, as a result, improvements in coverage, spread and uniformity (Coello et al. 2007; Mirjalili et al. 2017).

In our research, several metrics are discussed to examine the performance of the proposed multi-objective algorithm. In terms of faster convergence metrics, such as GD, coverage, radial coverage metric, MS, combined uniformity–convergence–spread—e.g. HV, IGD, Delta-P—and combined diversity spread—e.g. spacing. Each of the matrices is described below:

- Inverted Generational Distance (IGD) (Sierra and Coello Coello 2005).

$$\mathrm{IGD} = \sqrt{\frac{\sum_{i=1}^{n} (d_i')^2}{n}} \tag{14}$$

n is the number of true Pareto-optimal solutions. d_i' is the Euclidean distance between each true Pareto-optimal solution and the closest member of the obtained solutions in the objective space.

Radial coverage is a metric for the performance of multi-objective models in quantity determination and coverage measurement. The mathematical phrase of this metric is as follows (Lewis et al. 2009):

$$\begin{aligned} &\Psi = \frac{1}{N}\sum_{n=1}^{N}\Psi_i \\ &\Psi_i = \begin{cases} 1 & \text{if} \quad P_i \in \mathrm{PF} \cdot \alpha_{i-1} \le \tan\left(\dfrac{f_1(x)}{f_2(x)}\right) \le \alpha_n \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{15}$$

Ψ is the radial metric with a value in the range [0, 1], which is equal to 1 at its best case. N is the number of radial sectors, PF denotes the obtained Pareto-optimal front, and α refers to the tangent of radial sectors.

- Spread (MS) (Tan et al. 2002)

$$\mathrm{MS} = \sum_{i=1}^{O} \max(d(a_i \cdot b_i)) \tag{16}$$

where d is a function to calculate the Euclidean distance, a_i is the maximum value in the *i*th objective, b_i is the minimum in the ith objective and o is the number of objectives.

- Generational Distance (GD) (Sierra and Coello Coello 2005)

$$\mathrm{GD} = \sqrt{\frac{\sum_{i=1}^{\mathrm{no}} d_i^2}{n}} \tag{17}$$

no is the number of Pareto-optimal obtained solutions and d_i is Euclidean distance (in the objective space) between obtained solution and the nearest member of the true Pareto front.

- Spacing metric (SP) (Coello et al. 2004)

$$\mathrm{spacing(sp)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\overline{d} - d_i\right)^2} \tag{18}$$

$\overline{d}$ is the mean of all d_i.

- Hypervolume (HV) (Srinivas and Deb 1994)

$$HV = \wedge\left(U_{s\in PF}\left\{\left(S'|S \prec S' \prec S^{\text{nadir}}\right)\right\}\right) \tag{19}$$

- Average Hausdorff Distance (Delta - P) (Srinivas and Deb 1994)

$$\text{Delta} - P = \max(\text{mean IGD} \cdot \text{mean GD}) \tag{20}$$

2.4 Designing parameters of multi-objective models

The effectiveness of meta-heuristic algorithms is highly dependent on the correct selection of parameters (Meyr 2009). The method used in most studies to overcome this issue is to experiment with all possible combinations of the multi-objective model parameters. The number of experiments is equal to the full factorial of the number of parameters (Al-Aomar and Al-Okaily 2006). However, in cases where the number of parameters is significantly increased, a full factorial design can become excessively time-consuming and expensive (Mozdgir et al. 2013). Therefore, here, the Taguchi method was applied to the design of the parameters (Taguchi 1987) to implement an orthogonal array for the classification of experimental results (Roy 2010).

The Taguchi optimization process only involves the optimization of a single response character; it is simpler to optimize than a multi-objective optimization (Elsayed and Chen 1993; Logothetis and Haigh 1988). To overcome this issue, (Durairaj et al. 2013; Muthuramalingam and Mohan 2014; Unnikrishna Pillai et al. 2018) employed the Taguchi–grey relational method for optimization with multi-response characteristics and concluded that the Taguchi–grey relational method is efficient in optimization of multi-response characteristics. Hence, the Taguchi–grey relational method has been employed in this study to adjust multi-objective model parameters.

The Taguchi–grey relational method is as follows:

Step 1 A proper orthogonal array is first selected using the Taguchi method based on the number of parameters in MOOTLBO, MOGWO and MOPSO. In this study, an array with 27 rows, L_{27}, was considered, hence 27 trials. The controlling factors (parameters of MOOTLBO, MOGWO and MOPSO) were evaluated with three levels for each controlling factor (Table 1). Also, Table 2 shows the orthogonal array L27 with control factor levels (parameters of multi-objective models) in each trial for each of the algorithms.

Step 2 In order to determine the quality characteristics, each trial response must be converted into the *S/N* ratio, according to Eqs. (21) and (22). It must be noted that the responses involve the IGD convergence metric and radial coverage metric at each trial. Since IGD, DG, MS, SP and Delta-P must be minimized, S/Nratio is calculated via the following equation with the smaller the better type condition.

$$S/_N\text{ratio} = -10 * \log\left(\left(1/_r\right)\sum Y_{nj}^2\right) \tag{21}$$

Since the bigger the radial and HV are better, S/Nratio is calculated from the following equation with the condition of bigger is better.

$$S/_N\text{ratio} = -10 * \log\left((1/r)\sum 1/Y_{nj}^2\right) \tag{22}$$

r is the number of replicated experiments, and Y_{nj} is the observed response of nth trial for jth metrics.

Step 3: the S/Nratio is normalized according to Eq. (23):

$$Z_{nj} = \left(S_{nj} - \min\left(S_{nj}\right)\right)/\left(\max\left(S_{nj}\right) - \min\left(S_{nj}\right)\right) \tag{23}$$

S_{nj} is the S/Nratio in the nth trial for jth metrics, and Z_{nj} is the normalized S/Nratio in the nth trial for the jth metrics.

Step 4 The calculation of the grey relational coefficient (GRC) based on the following equation:

$$\text{GC}_{nj} = (\Delta_{\min} + (\delta * \Delta_{\max}))/\left(\Delta_{nj} + (\delta * \Delta_{\max})\right) \tag{24}$$

in which GC_{nj} is the GRC for the nth trial of the jth dependent response (metrics), δ is the quality loss factor, $\delta \in [0, 1]$ where δ is set at 0.5 for the purposes of this study and Δ is the difference between the response and the reference response

Step 5 The calculation of the average grey relational coefficients related to each control factor according to each control factor level, using Eq. (25):

$$G_{li} = (1/Q)\sum \text{GC}_{nj} \tag{25}$$

where G_{li} is the average grey relational grade for the lth level control factor and ith control factor (parameter), Q is the number of grey relational coefficients for the lth level control factor and ith control factor and GC_{nj} is the GRC for the nth trial for the jth independent response.

The best level of each parameter in each model can be obtained from the average grey coefficients. The results of the three MOOTLBO, MOGWO, MOPSO and MOMPA algorithms will be discussed in the next section.

3 Results and discussion

In this section, test problems, the parameters' setup and test results are represented and discussed in brief.

Table 1 Characteristics of the parameters of each model and the level of each of the parameters

	Parameters (control factor)								
	Mu	Gamma	Beta	NPop	NItr	NRep	C1	C2	W
MOPSO									
Level1	0.2	2	2	100	750	100	0.7	0.7	0.4
Level2	0.5	4	4	150	1000	150	1.0	1.0	0.7
Level3	0.7	8	8	200	1250	200	1.5	1.5	1.0

	Parameters (control factor)					
	Mu	Gamma	Beta	NPop	NItr	NRep
MOOTLBO						
Level1	0.2	2	2	100	750	100
Level2	0.5	4	4	150	1000	150
Level3	0.7	8	8	200	1250	200

	Parameters (control factor)				
	Gamma	Beta	NPop	NItr	NRep
MOGWO					
Level1	2	2	100	750	100
Level2	4	4	150	1000	150
Level3	8	8	200	1250	200

	Parameters (control factor)				
	PLI	P	FADS	NPop	NItr
MOMPA					
Level1	1.2	0.3	0.15	100	750
Level2	1.5	0.5	0.2	150	1000
Level3	1.7	0.7	0.25	200	1250

Mu, mutation rate, *gamma*, leader selection pressure parameter, *beta*, extra (to be deleted) repository member selection Pressure, *NPop*, number of population, *NItr*, number of iteration, *NRep*, number of repository (archive) member, *C1*, personal learning coefficient, *C2*, global learning coefficient, *W*, inertia weight, *PLI*, power law index, *P*, prey movement coefficient, *FADS*, fish aggregating devices effects

3.1 Test problems

The performance of the proposed algorithm in nineteen standard multi-objective test problems was investigated. BI1–BI9 are bi-objective test problems proposed in CEC 2009 by (Liang et al. 2019; Zhang et al. 2009). These test problems are unconstrained. BI1, BI2 and BI3 problems include non-convex and continuous Pareto fronts, while the BI4 problem has a convex and continuous Pareto front. Moreover, BI5 and BI6 problems contain discontinuous Pareto front and BI7 has a linear Pareto front. BI8 has a convex and linear Pareto front, and BI9 includes a convex and nonlinear Pareto front. Table 3 presents the bi-objective test problems used here.

BT1, BT2, BT3 and BT4 are multi-objective test problems proposed by (Deb et al. 2005). In this research, tri-objective functions of these benchmarks are employed. BT1 test problem has a linear Pareto-optimal front in a linear hyperplane and multi-modal space. BT2 and BT3 test problems contains a concave and continuous Pareto-optimal front in a three-dimensional space. BT4 includes disconnected Pareto-optimal regions in the search and multi-modal space (Huband et al. 2006). The features of the problems are shown in Table 4).

BR1 (Ray and Liew 2010), BR2 (Kannan and Kramer 1994), BR3 (Amir and Hasegawa 1989), BR4 (Lampinen and Zelinka 1999) and BR5 (Vaidyanathan et al. 2003) are real-world multi-objective optimization problems. BR1, BR2, BR3 and BR4 are real-world constraint problems. Also, BR6 (Binh and Korn 1997) is a constraint problem. Table 5 specifies the characteristics of these problems.

Therefore, here, test problems that provide various multi-objective search spaces with different non-convex, convex, discontinuous and multi-modal Pareto-optimal

Table 2 Orthogonal array L27 with control factor levels for each algorithms

	Parameters								
	1	2	3	4	5	6	7	8	9
MOPSP	Mu	Gamma	Beta	NPop	NItr	NRep	C1	C2	W
MOOTLBO	Mu	Gamma	Beta	NPop	NItr	NRep			
MOGWO	Gamma	Beta	NPop	NItr	NRep				
MOMPA	PLI	P	FADS	NPop	NItr				
Trail number	Level								
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2
5	1	2	2	2	2	2	2	3	3
6	1	2	2	2	3	3	3	1	1
7	1	3	3	3	1	1	1	3	3
8	1	3	3	3	2	2	2	1	1
9	1	3	3	3	3	3	3	2	2
10	2	1	2	3	1	2	3	1	2
11	2	1	2	3	2	3	1	2	3
12	2	1	2	3	3	1	2	3	1
13	2	2	3	1	1	2	3	2	3
14	2	2	3	1	2	3	1	3	1
15	2	2	3	1	3	1	2	1	2
16	2	3	1	2	1	2	3	3	1
17	2	3	1	2	2	3	1	1	2
18	2	3	1	2	3	1	2	2	3
19	3	1	3	2	1	3	2	1	3
20	3	1	3	2	2	1	3	2	1
21	3	1	3	2	3	2	1	3	2
22	3	2	1	3	1	3	2	2	1
23	3	2	1	3	2	1	3	3	2
24	3	2	1	3	3	2	1	1	3
25	3	3	2	1	1	3	2	3	2
26	3	3	2	1	2	1	3	1	3
27	3	3	2	1	3	2	1	2	1

fronts were employed to evaluate the performance of the proposed algorithm with those of the MOGWO, MOPSO and MOMPA algorithms.

3.2 Designing parameters

As previously mentioned, obtaining the best parameter adjustment is crucial for improving the performance of optimization algorithms. To achieve this, optimal parameters for MOPSO, MOOTLBO, MOGWO and MOMPA algorithms are calculated using the Taguchi–grey relational method. For example, the parameter adjustment steps for the MOOTLBO algorithm on the BI1 problem are outlined below, and the results are presented in Tables 6 and 7.

As an example, the S/N ratio for radial, IGD and GD criteria was calculated using Eqs. (21) and (22), respectively, and the results are tabulated in Table 6. This ratio is then normalized for three criteria using Eq. (23). Table 6 (the sixth, seventh and eighth columns) presents the calculated results of the GRC based on Eq. (24) for three criteria. The highest values of GRC = 1.0 for radial, IGD and GD criteria are related to trials numbers 21,10 and 9, respectively and the highest values are highlighted in bold font. Since the problem has seven answers (criteria), the average GRC for seven criteria has been given in the last

Table 3 Bi-objective test problems

Name	Mathematical formulation
BI1	$f_1 = x_1 + \frac{2}{\|J_1\|}\sum_{j\in J_1}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2, f_2 = 1 - \sqrt{x_1} + \frac{2}{\|J_2\|}\sum_{j\in J_2}\left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2$
	$J_1 = \{(j\|j \text{ is odd and } 2 \le j \le n)\}$, $J_2 = \{(j\|j \text{ is even and } 2 \le j \le n)\}$
BI2	$f_1 = x_1 + \frac{2}{\|J_1\|}\sum_{j\in J_1} y_j^2 \quad f_2 = 1 - \sqrt{x_1} + \frac{2}{\|J_2\|}\sum_{j\in J_2} y_j^2$
	$J_1 = \{(j\|j \text{ is odd and } 2 \le j \le n)\}$, $J_2 = \{(j\|j \text{ is even and } 2 \le j \le n)\}$
	$y_j = \begin{cases} x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]\cos\left(6\pi x_1 + \frac{j\pi}{n}\right) & if\ j \in J_1 \\ x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]\sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & if\ j \in J_2 \end{cases}$
BI3	$f_1 = x_1 + \frac{2}{\|J_1\|}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)\cdot f_2 = 1 - \sqrt{x_1} + \frac{2}{\|J_2\|}\left(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2}\cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$
	where J_1 and J_2 are the same as those of BI1, and $y_j = x_j - x_1^{0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)}$, $j = 2,3,\ldots,n$
BI4	$f_1 = x_1 + \frac{2}{\|J_1\|}\sum_{j\in J_1} h(y_j)\cdot f_2 = 1 - x_1^2 + \frac{2}{\|J_2\|}\sum_{j\in J_2} h(y_j)$
	where J_1 and J_2 are the same as those of BI1 and $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)$, $j = 2,3,\ldots,n$
BI5	$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right)\|\sin 2N\pi x_1\| + \frac{2}{\|J_1\|}\sum_{j\in J_1} h(y_j) f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right)\|\sin 2N\pi x_1\| + \frac{2}{\|J_2\|}\sum_{j\in J_2} h(y_j)$
	J_1amd $J_2$are identical to those of BI1, $\varepsilon > 0$, $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)$, $j = 2,3,\ldots,n$
BI6	$f_1 = x_1 + \max\left\{0.2\left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1)\right\} + \frac{2}{\|J_1\|}\left(44\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos\left(\frac{20y_j}{j}\right) + 2\right)$
	$f_2 = 1 - x_1 + \max\left\{0.2\left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1)\right\} + \frac{2}{\|J_2\|}\left(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2}\cos\left(\frac{20y_j}{j}\right) + 2\right)$
	J_1amd $J_2$are identical to those of BI1, $\varepsilon > 0$, $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)$, $j = 2,3,\ldots,n$
BI7	$f_1 = \sqrt[5]{x_1} + \frac{2}{\|J_1\|}\sum_{j\in J_1} y_j^2 . f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{\|J_2\|}\sum_{j\in J_2} y_j^2$
	J_1 and J_2 are identical to those of BI1, $\varepsilon > 0$, $\varepsilon > 0$, $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)$, $j = 2,3,\ldots,n$
BI8	$f_1 = (p_1 + a)^2 + p_2^2 . f_2 = (p_1 - a)^2 + p_2^2$ where $p_1 = x_1 - t_1(c + 2a)$, $p_2 = x_2 - t_2 b$
	$t_i = \mathrm{sgn}(\hat{t}_i) \times \min\{\|\hat{t}_i\|.1\}$, $i = 1,2$ $\begin{cases} \hat{t}_1 = \mathrm{sgn}(r_1) \times \frac{\|r_1\| - \left(a + \frac{c}{2}\right)}{2a + c} \\ \hat{t}_2 = \mathrm{sgn}(r_2) \times \frac{\|r_2\| - \frac{b}{2}}{b} \end{cases}$ $\begin{cases} r_1 = (\cos\omega) \times x_1 - (\sin\omega) \times x_2 \\ r_2 = (\sin\omega) \times x_1 - (\cos\omega) \times x_2 \end{cases}$
BI9	$f_1 = x_1, f_2 = \frac{g(t)}{x_1}$ where $g(t) = 2 - \exp\left[-2\log(2)\left(\frac{t - 0.1}{0.8}\right)^2\right]\sin^6(n_p\pi(t)), t = x_2 + \sqrt{x_3}$

f_1, first objective function, f_2, second objective function; n, number of variables, x_1, the first decision variable, x_M, the last k variables; M, number of objective function

column. As can be seen, the highest value of the grey relational grade = 0.7 is related to trial number 10.

The best trial and the respective parameters have been obtained so far. However, the Taguchi–grey relational method contains a combination of parameters in each trial; hence, the average grey relational grade for each parameter (factor) level must be determined. According to Eq. (25) and parameter (factor) level at each trial (Table 2), the average grey relational grade for each control factor level was determined. After that, optimal parameters for the MOOTLBO algorithm on BI1 were identified according to Table 1 (see Table 7 for BI1).

According to Table 7 that the highest values are highlighted in bold font, among the three levels considered for the Mu factor (parameter), the highest average grey relational grade is related to level 2. Consequently, Mu is equal to 0.5, according to Table 1. The optimal parameters results are tabulated in the last column of Table 7 and are as follows: gamma = 8, beta = 4, Npop = 200, NItr = 750, and NRep = 150.

Table 8 presents the optimal parameters of the algorithms on benchmark problems using the same method. For all test problems, the following optimal parameters for the MOPSO algorithm were obtained: Mu = 0.2 for twelve tests, gamma = 2 for nine tests, beta = 4 for seven tests, NItr = 1250 for fourteen tests, NRep = 200 for fourteen tests, NPop = 200 for ten tests, C1 = 1.5 for ten tests, C2 = 1.5 for ten tests, and w = 0.7 for nine tests, indicating that a better algorithm performance is seen if the number of iterations and members of the archive, as well as population individuals, are of level 2 and level 3, except for the mutation operator (Mu). In addition, the following optimal parameters for the MOOTLBO algorithm were obtained: each Mu = 0.2 or 0.5 for seven tests, gamma = 8 for nine tests, beta = 2 for seven tests, NItr = 1250 for fifteen tests, NRep = 200 for fifteen tests, and each NPop = 150 for nine tests, suggesting that a better algorithm performance is generally expected if the number of iterations and members of the archive, as well as population individuals, are of level 2 and level 3, except for the mutation operator (Mu).

In addition, the following optimal parameters for the MOGWO algorithm were obtained: gamma = 4 for eight tests, beta = 2 for eight tests, NItr = 1000 for eight tests, NRep = 200 for twelve tests, and NPop = 200 for ten tests, indicating that an almost better algorithm performance is observable if the number of iterations and members of the archive as well as population individuals are of level 2 and level 3. Also, according to Table 8, the parameters of the MOMPA algorithm are at levels 2 and 3, apart from PLI.

After designing optimal parameters, all algorithms were run 10 times for benchmark problems. First, the performance of the proposed MOOTLBO algorithm and other algorithms, such as MOPSO, MOGWO and MOMPA, are evaluated for unconstrained benchmark bi-objective problems. The various results obtained by all the multi-objective algorithms on the unconstrained multi-objective test sets are given in Tables 9, 10, 11, 12, 13, 14 and 15. The optimal results are highlighted in bold font. Figures 4 and 5 illustrate the qualitative outcomes of the proposed MOOTLBO algorithm. The performance of the MOOTLBO algorithm and other algorithms is evaluated on tri-objective, real-world and constraint test problems of multi-metric problems. Tables 16, 17, 18, 19, 20, 21 and 22 present the diverse outcomes obtained by all algorithms. As in the previous tables in this section, in tables 15–22, the optimal results are highlighted in bold font. Additionally, Figs. 6, 7, 8 display the obtained PF and qualitative results of the proposed MOOTLBO algorithm.

3.3 Unconstraint benchmark problems bi-objective

The test problems include BI1–BI9. The best mean and STD values of the metrics for all algorithms are presented in Tables 9, 10, 11, 12, 13, 14 and 15.

The evaluation of the quality of the Pareto fronts obtained is given in Table 9. The best IGD metric values for 7 out of 9 problems belong to MOOTLBO, and the

Table 4 Tri-objective test problems

Name	Mathematical formulation
BT1	$f_1 = \frac{1}{2}x_1x_2(1+g(x_M)), f_2 = \frac{1}{2}x_1x_2(1-x_2)(1+g(x_M)), f_3 = \frac{1}{2}(1-x_1)(1+g(x_M))$
	$g(x_M) = 100\left[\lvert x_M\rvert + \sum_{x_i \in x_M}\left((x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right)\right]$
	$0 \le x_i \le 1$ for i = 1,2,..,n $n = M + k - 1$, $k = 5$, $M = 3$
BT2	$f_1 = (1+g(x_M))\cos\left(x_1{}^{\pi}/_2\right)\cos\left(x_2{}^{\pi}/_2\right),$ $f_2 = (1+g(x_M))\cos\left(x_1{}^{\pi}/_2\right)\sin\left(x_2{}^{\pi}/_2\right) f_3 = (1+g(x_M))\sin\left(x_1{}^{\pi}/_2\right)$
	$g(x_M) = \sum_{x_i \in x_M}(x_i-0.5)^2 0 \le x_i \le 1$ for i = 1,2,…,n $n = M + k - 1$, $k = 10$, $M = 3$
BT3	$f_1 = (1+g(x_M))\cos\left(x_1^{\alpha}{}^{\pi}/_2\right)\cos\left(x_2^{\alpha}{}^{\pi}/_2\right),$ $f_2 = (1+g(x_M))\cos\left(x_1^{\alpha}{}^{\pi}/_2\right)\sin\left(x_2^{\alpha}{}^{\pi}/_2\right), f_3 = (1+g(x_M)\sin\left(x_1^{\alpha}{}^{\pi}/_2\right)$
	$g(x_M) = \sum_{x_i \in x_M}(x_i-0.5)^2 \cdot 0 \le x_i \le 1$ for i = 1,2,…,n $n = M + k - 1$, $k = 10$, $M = 3, \alpha = 100$
BT4	$f_1 = x_1, f_2 = x_2, f_3 = (1+g(x_M)h(f_1.f_2.\ldots.f_{M-1}\cdot g)$
	$g(x_M) = 1 + \frac{9}{\lvert x_M\rvert}\sum_{x_i \in x_M} x_i,\ h(f_1.f_2.\ldots.f_{M-1}.g) = M - \sum_{i=1}^{M-1}\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right]$
	$0 \le x_i \le 1$ for i = 1,2,…,n $n = M + k - 1$, $k = 20$, $M = 3$

f_1, first objective function, f_2, second objective function; f_3, the third objective function n, number of variables, x_1, the first decision variable, x_M, the last k variables; M, number of objective function

Table 5 Real-world test problems

Name	Original name	M	D	Pareto front
BR1	Welded beam design	2	4	Unknown, constraint
BR2	Pressure vessel design	2	4	Mixed, Disconnected, constraint
BR3	Hatch cover design	2	2	Convex, constraint
BR4	Coil compression spring design	2	3	Mixed, Disconnected, constraint
BR5	Rocket injector design	3	4	Unknown
BR6	BNH	2	2	Disconnected, constraint

M, number of objective; *D*, number of decision variable

Table 6 Signal-to-noise ratio and normalized signal-to-noise ratio on BI1 problem for MOOTLBO

Trial No:	Radial		IGD		GD, …		Grey relational coefficient			Grey relational grade
	S/N ratio	Normalized S/N ratio	S/N ratio	Normalized S/N ratio	S/N ratio	Normalized S/N ratio	radial	IGD	GD…	
1	− 6.52	0.00	24.00	0.19	49.20	0.62	0.33	0.38	0.57	0.49
2	− 3.52	0.65	25.08	0.36	36.38	0.00	0.59	0.44	0.33	0.53
3	− 3.52	0.65	25.74	0.46	53.09	0.81	0.59	0.48	0.72	0.51
4	− 2.83	0.80	24.95	0.34	46.64	0.50	0.71	0.43	0.50	0.54
5	− 5.11	0.30	26.06	0.51	53.13	0.81	0.42	0.51	0.72	0.50
6	− 4.28	0.48	26.83	0.63	55.35	0.92	0.49	0.58	0.86	0.53
7	− 4.28	0.48	25.88	0.49	48.01	0.56	0.49	0.49	0.53	0.50
8	− 2.18	0.93	27.67	0.77	52.48	0.78	0.88	0.68	0.69	0.60
9	− 4.28	0.48	26.67	0.61	57.10	1.00	0.49	0.56	**1.00**	0.55
10	− 2.83	0.80	29.18	1.00	55.71	0.93	0.71	**1.00**	0.88	**0.70**
11	− 4.28	0.48	26.54	0.59	54.15	0.86	0.49	0.55	0.78	0.52
12	− 5.11	0.30	25.32	0.40	49.00	0.61	0.42	0.45	0.56	0.49
13	− 4.68	0.40	22.75	0.00	51.76	0.74	0.45	0.33	0.66	0.54
14	− 5.11	0.30	25.32	0.40	49.00	0.61	0.42	0.45	0.56	0.49
15	− 4.68	0.40	25.74	0.47	39.95	0.17	0.45	0.48	0.38	0.54
16	− 2.50	0.87	27.61	0.76	52.39	0.77	0.79	0.67	0.69	0.59
17	− 3.17	0.72	26.25	0.54	48.44	0.58	0.64	0.52	0.54	0.56
18	− 3.52	0.65	26.21	0.54	49.63	0.64	0.59	0.52	0.58	0.54
19	− 3.89	0.57	25.01	0.35	50.08	0.66	0.54	0.44	0.60	0.50
20	− 6.52	0.00	23.71	0.15	49.00	0.61	0.33	0.37	0.56	0.49
21	− 1.88	1.00	27.15	0.68	52.75	0.79	**1.00**	0.61	0.70	0.64
22	− 5.11	0.30	24.83	0.32	49.24	0.62	0.42	0.43	0.57	0.52
23	− 3.17	0.72	26.45	0.57	49.62	0.64	0.64	0.54	0.58	0.54
24	− 2.50	0.87	27.52	0.74	47.41	0.53	0.79	0.66	0.52	0.59
25	− 3.89	0.57	25.18	0.38	47.38	0.53	0.54	0.45	0.52	0.57
26	− 4.28	0.48	24.98	0.35	46.85	0.51	0.49	0.43	0.50	0.53
27	− 6.02	0.11	23.60	0.13	51.18	0.71	0.36	0.37	0.64	0.55

other three problems belong to MOMPA. Table 10 lists the radial metric values. The proposed MOOTLBO obtains the best results for 4 problems out of 9, and MOGWO and MOMPA have the best results for 3 and 2 problems, respectively. Table 11 lists the results of the GD metric. The best results for 6 out of 9 items are related to MOOTLBO. Three other items belong to MOMPA. To evaluate how the non-dominated solutions are distributed, Table 12 lists the values of the SPEARD metric (MS). The proposed MOOTLBO has the best results for 5 out of 9

Table 7 Average grey relational grade for each level in MOOTLBO on BI1

Input factor	Average grey relational grade by factor level			Optimized parameter
	Level1	Level2	Level3	
Mu	0.5279	**0.5514**	0.5487	0.5
Gamma	0.5432	0.5322	**0.5526**	8
Beta	0.5426	**0.5466**	0.5387	4
NPop	0.5263	0.5448	**0.5569**	200
NItr	**0.5494**	0.5297	0.5197	750
NRep	0.4898	**0.5812**	0.5141	150

problems; MOMPA, MOGWO and MOPSO have the best results for 2, 1 and 1 problems, respectively. To show the spacing quality of non-dominated solutions, Table 13 shows SPACING metric values. MOOTLBO has 5 cases, MOPSO, MOGWO and MOMPA Pareto-optimal front(PF) have 2, 1 and 1 cases, respectively, out of 9 problems. To show the high volume enclosed near PF, in Table 14, HV metric values are given. The MOOTLBO algorithm has obtained the best results for six out of nine problems. MOMPA has the best results for three problems. To show the uniformity–convergence–spread attribute, Delta-P metric values are given in Table 15. The best results (5 out of 9 cases) are related to the MOOTLBO algorithm. and the next one is the MOMPA algorithm with four cases.

To be more clear, as an example, the convergence metric (IGD) boxplot of all algorithms is given in Fig. 4. As can be seen, the boxplots of MOOTLBO are considerably below those of MOGWO, MOPSO and MOMPA in the BI3, BI4, BI5, BI6, BI8 and BI9 test problems. In the instances of BI1, BI2 and BI7, the boxplots of MOMPA are narrower and further below the other algorithms. The obtained Pareto-optimal solution of each algorithm for BIs test problems is also shown in Fig. 5. As can be seen, MOOTLBO convergence is better than other algorithms for six test problems. Also, MOOTLBO algorithm coverage has better performance for four test problems than other algorithms.

In general, based on the results of Tables 9, 10, 11, 12, 13, 14 and 15, it can be acknowledged that the MOOTLBO algorithm has the best results with 7 out of the 7 metrics used in the BI3 and BI4 problems. In the case of the BI5 problem, it has the best results with 5 out of 7 metrics; in the BI6 problem, it has 4 out of 7, and in the BI8 and BI9 problems, it has the best results with 6 out of 7. Therefore, the MOOTLBO algorithm has superior performance in six out of nine unconstrained bi-objective benchmark problems.

3.4 The tri-objective, real-world test and constraint problems

BT1, BT2, BT3 and BT4 are tri-objective problems that are more challenging than bi-objective problems. BR1, BR2, BR3 and BR4 are constraint real-world test problems and BR5 is unconstraint real-world test problems. BR6 is the constraint test problem whose specifications are given in Table 5. In this section, the proposed MOOTLBO and other algorithms are evaluated on these issues. Tables 16, 17, 18, 19, 20, 21 and 22 list the best mean and standard deviation values. Table 16 shows the IGD metric values that verify the quality of the Pareto-optimal front obtained. The MOOTLBO algorithm has obtained the best results for 8 out of 10 test problems. MOMPA and MOPSO algorithms have won one top case each. Table 17 lists the COVERAGE metric (radial) values. This quality metric shows the ratio of points in a Pareto set estimate, and MOOTLBO achieves the best results in five cases. The MOMPA algorithm also obtained 5 out of 10 cases. From Table 18, it can be seen that MOOTLBO obtained the best GD metric values for 6 out of 10 cases, proving the convergence quality of MOEO. Table 19 lists the values of SPREAD metric (MS) and MOOTLBO that obtain the best results for five cases. MOMPA, MOGWO and MOPSO algorithms have the best results for 1, 3 and 1 test problems, respectively. Table 20 lists the SPACING metric values, and MOOTLBO has the best results for 7 cases, which shows the high quality of the spacing quality of the non-dominated solutions obtained by MOOTLBO. Table 21 lists the HV metric values. The MOOTLBO algorithm has obtained the best results in 8 out of 10 cases. Table 22 lists the Delta-P values. MOOTLBO scores 9 out of 10 cases, which proves the high quality of this algorithm in the uniformity–convergence–spread attribute.

For more clarification of the results, as an example, the convergence metric (IGD) boxplot of all algorithms is given in Fig. 6. It can be seen in Fig. 6 that the boxplots of MOOTLBO for BT1, BT4, BR2, BR3, BR4, BR5 and BR6 are further below in comparison with other algorithms. According to Table 16 and Fig. 6, MOOTLBO outperforms other algorithms in 8 out of 10 test problems.

Table 8 Optimal parameters for algorithms

	Parameters											
	Mu	Gamma	Beta	NPop	NItr	NRep	C1	C2	W	PLI	P	FADS
BI1												
MOPSO	0.2	8	4	200	1000	100	1.0	1.5	0.7			
MOOTLBO	0.5	8	4	200	750	150						
MOGWO		4	2	150	1250	100						
MOMPA				200	1250					1.7	0.3	0.25
BI2												
MOPSO	1	2	8	200	1250	200	1.5	1.5	0.7			
MOOTLBO	0.5	2	8	200	1250	150						
MOGWO		2	2	200	1250	200						
MOMPA				200	1250					1.7	0.3	0.25
BI3												
MOPSO	0.2	8	4	150	1250	200	1.5	1.0	0.7			
MOOTLBO	0.2	8	2	150	1250	200						
MOGWO		4	4	200	1000	100						
MOMPA				150	1250					1.5	0.5	0.2
BI4												
MOPSO	0.5	2	4	150	1250	200	1.5	1.5	0.9			
MOOTLBO	0.5	2	2	100	1250	200						
MOGWO		2	2	200	1000	200						
MOMPA				200	1250					1.7	0.3	0.25
BI5												
MOPSO	0.5	2	2	150	1250	150	1	1	0.9			
MOOTLBO	0.2	2	2	150	750	100						
MOGWO		2	4	150	1000	200						
MOMPA				200	1250					1.7	0.3	0.25
BI6												
MOPSO	1	2	2	150	1000	150	0.7	1.5	0.7			
MOOTLBO	1	8	2	100	1250	100						
MOGWO		8	8	150	750	150						
MOMPA				200	1250					1.2	0.5	0.25
BI7												
MOPSO	0.2	2	8	200	1250	200	1.5	1.5	0.9			
MOOTLBO	0.5	4	2	150	1250	200						
MOGWO		8	2	200	1250	150						
MOMPA				200	1250					1.2	0.3	0.25
BI8												
MOPSO	0.2	2	4	200	1000	200	0.7	1	0.9			
MOOTLBO	0.2	4	4	150	1250	200						
MOGWO		2	4	150	1000	200						
MOMPA				200	1250					1.2	0.7	0.25
BI9												
MOPSO	1	8	4	150	1000	200	1	1.5	0.7			
MOOTLBO	0.2	4	4	150	1250	200						
MOGWO		4	4	200	1000	200						
MOMPA				200	1250					1.2	0.3	0.15
BT1												
MOPSO	0.2	2	8	200	1250	200	1.5	1.5	0.9			
MOOTLBO	1	8	4	200	1250	200						
MOGWO		8	2	200	1000	150						
MOMPA				200	1250					1.2	0.7	0.25
BT2												

Table 8 (continued)

	Parameters											
	Mu	Gamma	Beta	NPop	NItr	NRep	C1	C2	W	PLI	P	FADS
MOPSO	0.2	4	8	150	1250	200	0.7	1	0.7			
MOOTLBO	0.5	8	8	200	1250	200						
MOGWO		8	4	200	750	200						
MOMPA				200	1250					1.5	0.3	0.25
BT3												
MOPSO	1	8	2	150	1250	150	1.5	1.5	0.4			
MOOTLBO	0.2	4	2	200	1250	200						
MOGWO		4	2	100	750	200						
MOMPA				150	1000					1.5	0.3	0.25
BT4												
MOPSO	0.2	2	4	150	1250	200	1.5	1.5	0.9			
MOOTLBO	0.5	2	4	150	1250	200						
MOGWO		8	8	150	1000	150						
MOMPA				150	1000					1.7	0.3	0.25
BR1												
MOPSO	0.5	8	2	150	1000	200	0.7	0.7	0.7			
MOOTLBO	1	2	8	150	1000	200						
MOGWO		4	2	100	750	200						
MOMPA				150	1250					1.7	0.5	0.25
BR2												
MOPSO	0.2	8	8	150	1250	150	1.5	1.5	0.7			
MOOTLBO	0.5	8	4	100	1250	200						
MOGWO		8	8	200	750	200						
MOMPA				200	1000					1.7	0.5	0.25
BR3												
MOPSO	0.2	8	2	200	750	200	0.7	1.0	0.4			
MOOTLBO	0.5	8	4	100	1250	200						
MOGWO		4	4	150	1000	150						
MOMPA				200	1250					1.5	0.7	0.2
BR4												
MOPSO	0.2	2	2	200	1250	200	1.0	1.0	0.7			
MOOTLBO	1	4	8	150	1250	200						
MOGWO		8	8	200	1250	200						
MOMPA				200	1250					1.7	0.7	0.2
BR5												
MOPSO	0.2	4	4	150	1250	200	1.5	1.0	0.4			
MOOTLBO	0.2	8	8	200	1250	200						
MOGWO		4	8	150	1250	200						
MOMPA				200	1250					1.5	0.5	0.25
BR6												
MOPSO	0.2	4	2	200	1250	200	0.7	0.7	0.9			
MOOTLBO	0.2	8	2	150	750	200						
MOGWO		4	2	200	750	200						
MOMPA				150	1000					1.7	0.7	0.25

Additionally, Figs. 7 and 8 illustrate the solutions obtained from all algorithms and the true Pareto-optimal fronts. In terms of the extent and convergence points of the solutions, the MOOTLBO algorithm exhibits better performance than other algorithms for the experimental problems mentioned.

In general, based on the results of Tables 16, 17, 18, 19, 20, 21 and 22, it can be acknowledged that the MOOTLBO algorithm has the best results with 7 out of the 7 metrics used in the BT1 and BR2 problems. In the case of problems BT4, BR3 and BR4, it has the best results with 6 out of 7 metrics; in problems BR5 and BR6, it has the best results with 5 out of 7, and in BR1 problem, it has the best results with 4 out of 7, so the MOOTLBO algorithm has superior performance in 8 out of 10 problems.

Table 9 Mean and STD values (IGD) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BI1	0.08549	0.01372	0.06407	0.01497	0.04621	0.00788	**0.00585**	0.00076
BI2	0.07605	0.00891	0.02538	0.00287	0.02313	0.00179	**0.00549**	0.00027
BI3	0.31555	0.04018	0.25432	0.06602	**0.02486**	0.00180	0.69425	9.5217E−06
BI4	0.04709	0.00209	0.04975	0.00179	**0.02926**	0.00133	0.03072	0.00080
BI5	0.40088	0.13794	0.48008	0.17355	**0.139608**	0.011954	0.17497	0.00232
BI6	0.37220	0.08844	0.62460	0.12792	**0.105764**	0.009459	0.11635	0.02686
BI7	0.03918	0.00664	0.06297	0.06019	0.02848	0.00584	**0.00635**	0.00184
BI8	0.02318	0.00344	0.06918	0.04606	**0.02072**	0.00189	0.02333	9.7551 E−05
BI9	0.02113	0.00482	0.02515	0.00561	**0.01958**	0.00306	0.02181	2.6077 E−05

Table 10 Mean and STD values (radial) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BI1	0.40000	0.15426	0.54167	0.05990	0.68056	0.06117	**0.98333**	0.02546
BI2	0.81667	0.05300	0.94167	0.04722	0.98889	0.01361	**1.00000**	0.00000
BI3	0.13889	0.11719	0.22222	0.06086	**0.26078**	0.04212	0.02778	3.4694 E−18
BI4	0.94167	0.03819	0.88333	0.03469	**0.96944**	0.01944	0.96667	0.01667
BI5	0.11111	0.10244	**0.26944**	0.08962	0.10278	0.01273	0.15278	0.03106
BI6	0.15278	0.09722	**0.40278**	0.10111	0.14167	0.05479	0.23333	0.05583
BI7	0.73611	0.10989	0.76111	0.11667	0.76789	0.05652	**0.95556**	0.04157
BI8	0.96111	0.02546	0.92500	0.04488	**0.96389**	0.03525	0.95833	2.2204 E−16
BI9	0.93846	0.02551	0.93846	0.03077	**0.94231**	0.03546	0.93103	0.00000

Table 11 Mean and STD values (GD) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BI1	0.01176	0.01458	0.00550	0.00347	0.00374	0.00113	**0.00177**	0.00176
BI2	0.00633	0.00149	0.00281	0.00092	0.00316	0.00159	**0.00059**	0.00043
BI3	0.06402	0.09159	0.00886	0.00244	**0.00394**	0.00182	0.03803	1.0938 E−08
BI4	0.00371	0.00021	0.00361	0.00009	**0.00234**	0.00030	0.00239	9.5596 E−05
BI5	0.06656	0.06048	0.13722	0.10791	**0.00301**	0.00076	0.01127	0.00078
BI6	0.06015	0.04782	0.20315	0.04195	**0.01059**	0.01052	0.01227	0.00835
BI7	0.00209	0.00122	0.00425	0.00178	0.00132	0.00035	**0.00036**	0.00015
BI8	0.00348	0.00031	0.00292	0.00008	**0.00225**	0.00026	0.00281	1.8658 E−06
BI9	0.03390	0.00132	0.03367	0.00247	**0.03318**	0.00185	0.03671	4.2165 E−06

Table 12 Mean and STD values (MS) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BI1	**0.94686**	0.06517	0.96704	0.09813	1.58464	0.17601	0.96725	0.17164
BI2	0.66694	0.04398	0.67301	0.03543	1.22605	0.06907	**0.52641**	0.11456
BI3	0.97259	0.03414	1.72830	0.25791	**0.94674**	0.01587	1.00000	0.00000
BI4	0.75865	0.05801	0.87810	0.05643	**0.70926**	0.05538	0.77175	0.06815
BI5	1.08155	0.13607	1.08972	0.23082	**1.02021**	0.00055	1.78359	0.12706
BI6	1.04822	0.12529	**0.83165**	0.07791	1.64070	0.39562	1.71537	0.20603
BI7	0.79311	0.03721	0.88058	0.08337	1.32418	0.07622	**0.60010**	0.06355
BI8	0.65211	0.03524	1.05007	0.19453	**0.65056**	0.03182	0.70120	0.00226
BI9	0.92719	0.03628	0.98106	0.08268	**0.72506**	0.02430	0.74443	0.00002

Table 13 Mean and STD values (SP) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BI1	**0.00869**	0.00566	0.01577	0.00585	0.01165	0.00525	0.02581	0.02418
BI2	0.00967	0.00320	0.00600	0.00076	**0.00578**	0.00051	0.01012	0.00575
BI3	**0.00526**	0.00595	0.02386	0.01617	0.01141	0.00788	1.00000	0.00000
BI4	0.00580	**0.00055**	0.00870	0.00272	0.00734	0.00087	0.00608	0.00196
BI5	0.06815	0.13592	0.11492	0.11315	**0.05980**	0.01993	0.01352	0.02797
BI6	0.04644	0.10531	0.15975	0.05711	**0.01956**	0.03069	0.09748	0.14702
BI7	0.00680	0.00367	0.01126	0.00519	0.00837	0.00271	**0.00469**	0.00060
BI8	0.02437	0.00615	0.02761	0.00501	**0.02335**	0.00131	0.05598	9.8525 E−05
BI9	0.03723	0.00258	0.04319	0.00824	**0.03673**	0.00441	0.16162	1.4870 E−05

Table 14 Mean and STD values (HV) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	mean	STD.Dev	Mean	STD.Dev
BI1	0.59989	0.01403	0.63153	0.02452	0.65987	0.01027	**0.71678**	0.00091
BI2	0.62872	0.01001	0.69201	0.00372	0.69405	0.00118	**0.71775**	0.00029
BI3	0.32193	0.03103	0.35352	0.07738	**0.39418**	0.02063	0.00000	0.00000
BI4	0.38191	0.00265	0.37544	0.00257	**0.43739**	0.00087	0.40582	0.00068
BI5	0.19620	0.06532	0.09178	0.09464	**0.35643**	0.00303	0.32570	0.00291
BI6	0.14300	0.07093	0.01681	0.02416	**0.34902**	0.01429	0.30034	0.04418
BI7	0.53097	0.00805	0.50109	0.05843	0.54711	0.00860	**0.57700**	0.00223
BI8	0.85813	0.00064	0.85619	0.00231	**0.88801**	0.00036	0.86015	7.370 E−06
BI9	0.72773	0.00030	0.72712	0.00043	**0.74728**	0.00012	0.72687	6.441 E−07

3.5 Statistical tests

In order to further compare the performance of the MOOTLBO algorithm with other algorithms, Wilcoxon's signed-rank test and Friedman's statistical tests were performed.

3.5.1 Assessment test problems by Wilcoxon signed-rank test

In this paper, the Wilcoxon signed-rank test (Wilcoxon 1945) is used to evaluate the performance of MOOTLBO against MOMPA, MOGWO and MOPSO. H0 is the null hypothesis, which indicates the superiority of other algorithms to the MOOTLBO algorithm. While H1 is the alternative hypothesis, both are presented in this test. The level of significance associated with the null hypothesis has a pre-defined degree of statistical significance. When the

Table 15 Mean and STD values (Delta-P) of all algorithms on the unconstrained bi-objective test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BI1	0.10836	0.05487	0.06651	0.01966	0.04661	0.00796	**0.00668**	0.00148
BI2	0.07605	0.00891	0.02665	0.00401	0.02744	0.00361	**0.00549**	0.00027
BI3	0.43493	0.19241	0.25432	0.06602	**0.24865**	0.01801	0.69426	8.5596 E−06
BI4	0.05068	0.00200	0.04997	0.00169	**0.03018**	0.00338	0.03279	0.00075
BI5	0.42296	0.14096	0.69840	0.44737	**0.16661**	0.02264	0.17497	0.00232
BI6	0.40640	0.10121	0.78659	0.13693	**0.11422**	0.01346	0.11928	0.02669
BI7	0.03918	0.00664	0.06326	0.06006	0.02848	0.00584	**0.00635**	0.00184
BI8	0.03887	0.00177	0.07443	0.04049	**0.03275**	0.00161	0.03416	2.9833 E−05
BI9	0.28410	0.00975	0.27010	0.02434	0.29429	0.01756	**0.20952**	1.2917 E−05

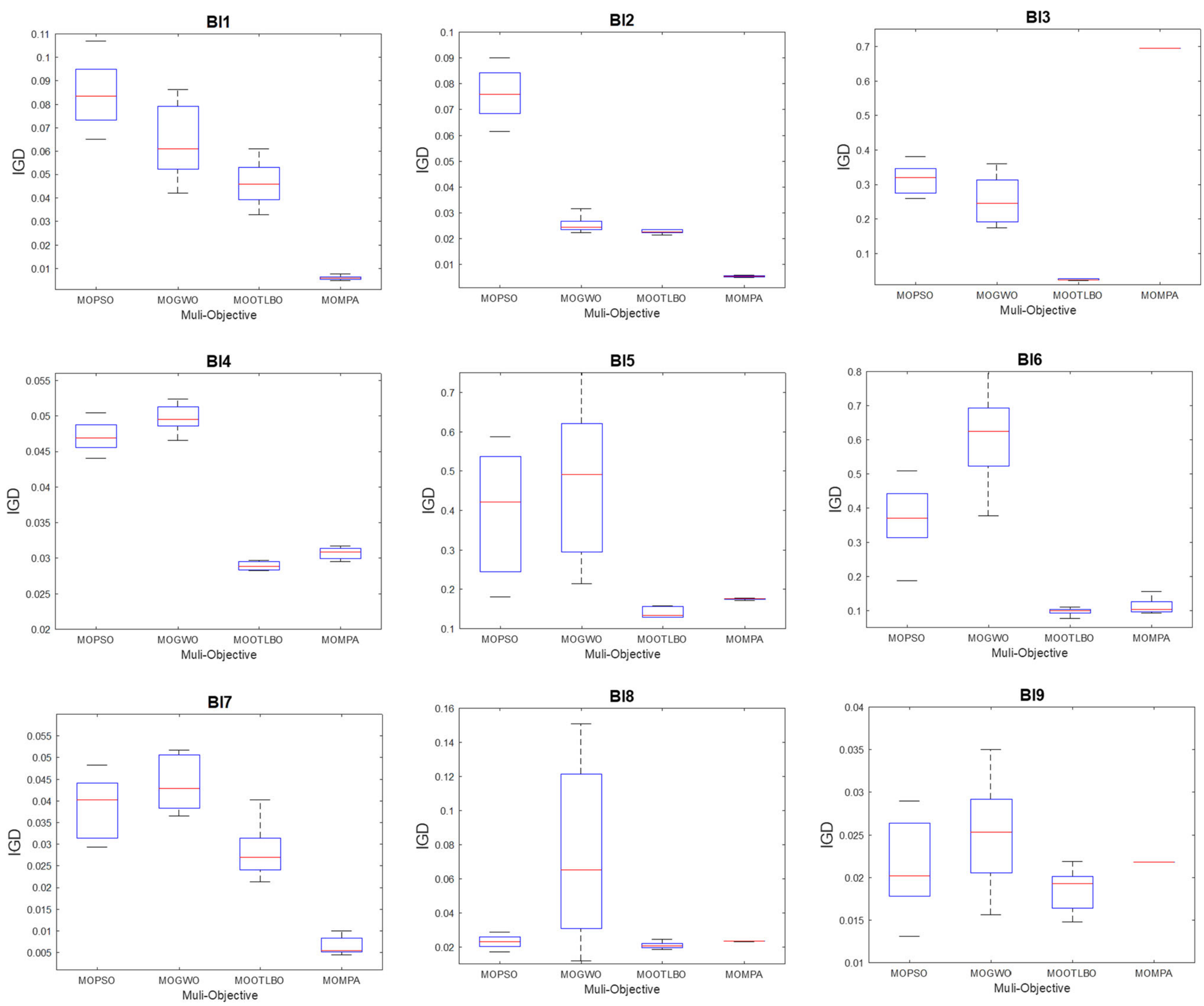

Fig. 4 Boxplot of the statistical results for IGD on BI1–BI9

null hypothesis is rejected, the value required to reject it is measured and compared with the significance threshold (10%). Findings are inferred based on the rejection or acceptance of the null hypothesis (or both). For example, IGD and HV metrics are investigated based on Wilcoxon

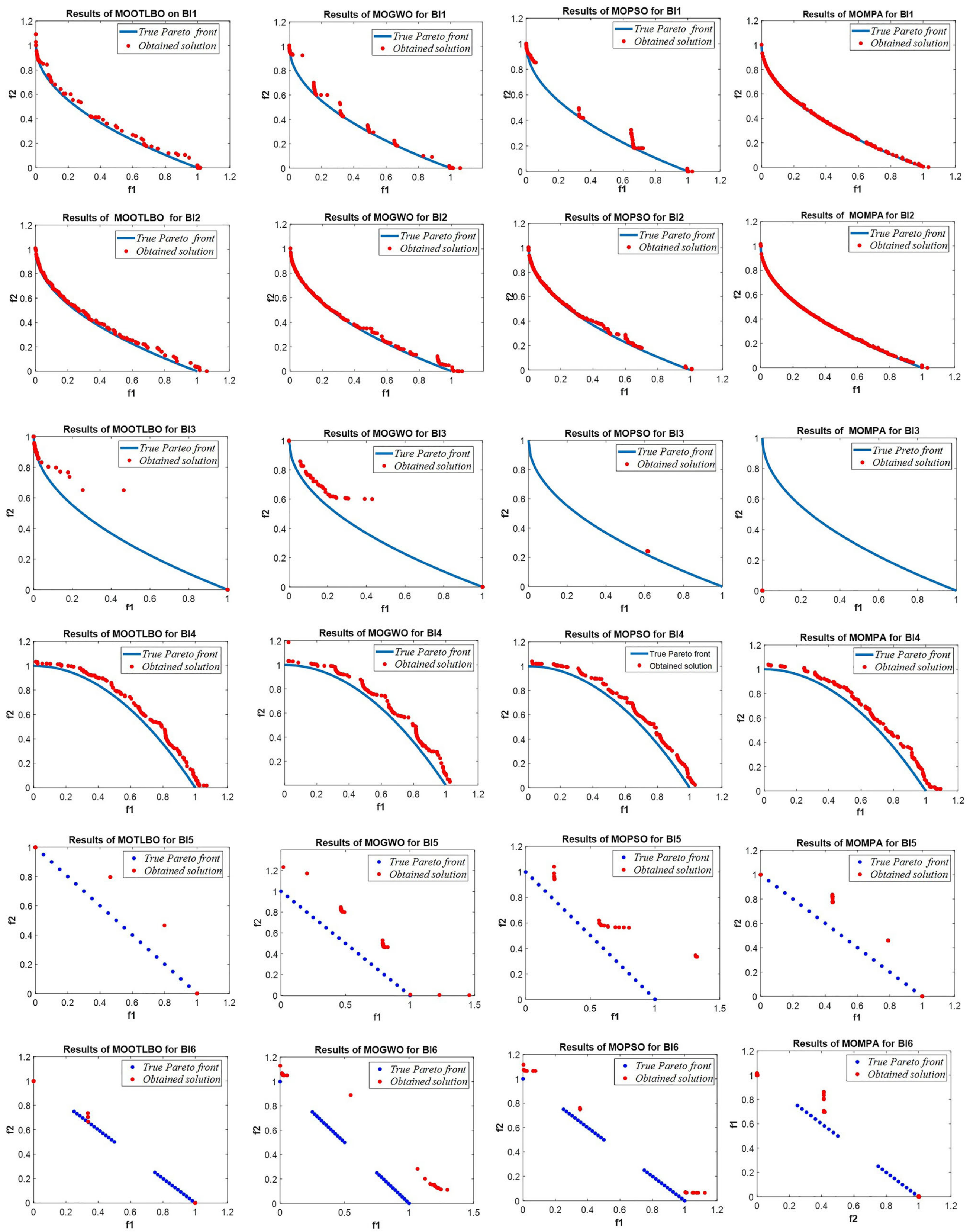

Fig. 5 Obtained Pareto-optimal solutions by MOOTLBO, MOGWO, MOPSO and MOMPA for BI1–BI9

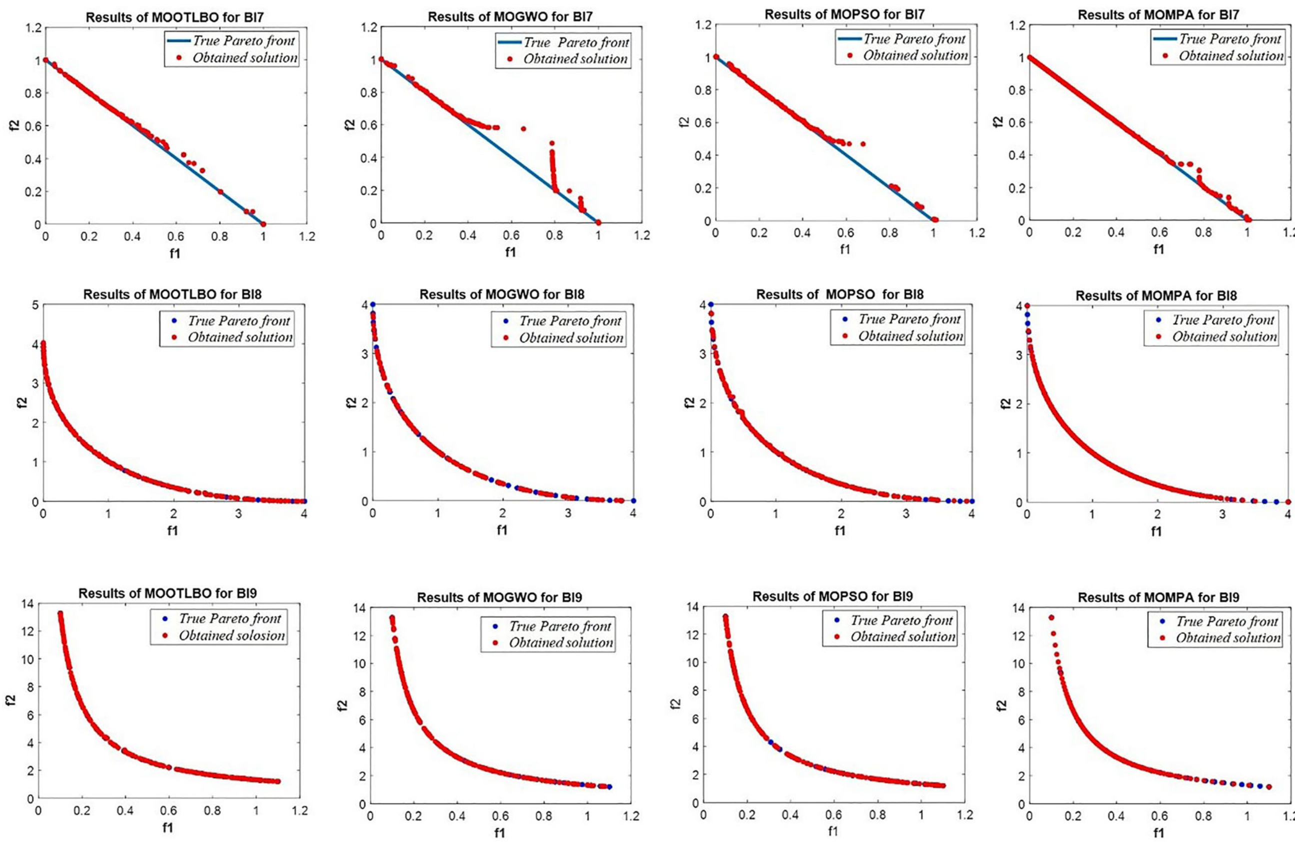

Fig. 5 continued

signed-rank test which is given in Table 23. The steps of the test are as follows:

Step 1 IGD or HV metrics values are obtained by MOOTLBO and other algorithms (Say Y), are collected.

Step 2 The sum of the ranks for cases where MOOTLBO outperforms Y and vice versa is used to calculate R^+ and R^-.

Step 3 To determine the significance of a statistical test based on the hypotheses, a *P* value is calculated. A smaller *P* value indicates stronger evidence against H0.

Step 4 The results are analysed to validate the findings. Table 23 shows that the *P*-value amounts for both metrics are less than 0.1, and it can be concluded that MOOTLBO performs better than MOMPA, MOGWO and MOPSO with a probability of 90%.

3.5.2 Assessment test problems by Friedman aligned test

Friedman's aligned rank test (Friedman 1937) is a statistical test that is used to compare several groups and determine the average ranks of the groups. In this research, the average metric spread and Delta-P of all algorithms were evaluated using this test (see the results in Table 24). As shown, MOOTLBO ranks first and outperforms MOMPA, MOGWO and MOPSO algorithms.

4 Discussion

In brief, both quantitative and qualitative results have shown the ability of MOOTLBO to represent very hopeful results. The statistical results demonstrate that the performance of MOOTLBO is superior to that of 6 out of 9 bi-objective problems, 2 out of 4 tri-objective problems, and 6 out of 6 real-world and constraint test problems.

MOOTLBO's high convergence is attributed to the teacher phase, which enables students to learn from the teacher, the learner phase, which facilitates knowledge exchange between classmates, and the observer phase, which allows students to learn from their classmates' information randomly. MOOTLBO's ability to maintain solution diversity and high coverage of the obtained Pareto-optimal solutions is evident from the radial convergence metric results. This is due to MOOTLBO's updating mechanism, which considers at least three of the best-obtained solutions so far and forces students to change their location with respect to them. Additionally, MOOTLBO

Table 16 Mean and STD values (IGD) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	4.85140	2.54776	7.60244	2.80796	**0.01561**	0.00034	0.29184	0.15610
BT2	0.25308	0.06478	0.25574	0.03236	0.12371	0.02245	**0.03689**	0.00103
BT3	0.04669	0.00417	0.07826	0.01447	**0.04148**	0.00331	0.04288	0.00040
BT4	0.04833	0.00347	0.06077	0.00426	**0.03717**	0.00381	0.06680	0.00287
BR1	**0.10693**	0.01230	0.26658	0.10350	0.14745	0.01396	0.18295	0.01334
BR2	5996.59	581.47	4017.47	562.11	**3512.97**	187.04	7439.71	0.00103
BR3	1.29931	0.06353	1.76440	0.32786	**1.23360**	0.04533	5.42468	0.00134
BR4	0.01747	0.01701	0.09642	0.03914	**0.00840**	0.00110	0.10342	0.03249
BR5	0.08234	0.00560	0.28894	0.02559	**0.04015**	0.00043	0.04926	0.00144
BR6	0.35851	0.01270	0.69309	0.20901	**0.35496**	0.02308	0.67200	0.00505

Table 17 Mean and STD values (radial) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	0.67407	0.12847	0.71111	0.11489	**0.99815**	0.00370	0.98241	0.01339
BT2	0.74167	0.09067	0.74259	0.06439	0.95926	0.02658	**1.00000**	0
BT3	0.98796	0.00593	0.98426	0.00931	0.95278	0.02048	**1.00000**	0
BT4	0.92500	0.03056	0.92315	0.02987	**0.96111**	0.02546	0.79722	0.04292
BR1	0.90833	0.01273	0.73889	0.10773	0.90278	0.02240	**0.92222**	0.01111
BR2	0.84600	0.02375	0.88800	0.01327	**0.89200**	0.00980	0.63000	0
BR3	0.92857	0.01645	0.92857	0.02457	**0.93265**	0.01594	0.65306	1.11022 E−16
BR4	0.84706	0.07059	0.59412	0.07180	**0.95882**	0.03767	0.41176	5.55112 E−17
BR5	0.98611	0.00854	0.63519	0.06089	0.99630	0.00454	**0.99907**	0.00278
BR6	0.97500	0.02621	0.95278	0.01779	0.98333	0.01843	**1.00000**	0

Table 18 Mean and STD values (GD) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	4.35429	3.10020	3.93030	0.58177	**0.00054**	0.00004	0.02898	0.01721
BT2	**0.00038**	0.00001	0.00111	0.00026	0.00785	0.00242	0.00072	0.00059
BT3	0.00483	0.00052	0.00322	0.00133	0.00534	0.00037	**0.00206**	0.00001
BT4	0.00247	0.00030	0.00379	0.00091	**0.00220**	0.00018	0.00384	0.00042
BR1	0.00572	0.00121	0.02106	0.01149	**0.00477**	0.00084	0.00487	0.00030
BR2	28.157	1.89391	28.796	1.46759	**27.5335**	0.82125	50.623	0.00059
BR3	0.02483	0.01623	0.06498	0.00898	0.01423	0.00231	**0.01298**	0.00010
BR4	0.00008	0.00001	0.00008	0.00000	**0.00008**	0.00001	0.00017	0.00030
BR5	**0.00159**	0.00016	0.00236	0.00029	0.00188	0.00039	0.00181	0.00022
BR6	0.05874	0.00331	0.05365	0.00712	**0.05227**	0.00309	0.05510	0.00006

selects a solution with a combination of three leaders to update each student's location, emphasizing exploration.

It is important to note that the teaching factor (TF) determines the decision to change the average in MOOTLBO. Lower TF values allow for accurate search at small steps but result in slow convergence. On the other hand, higher TF values enhance searching speed but decrease exploration capability (Rao and Patel 2013). The solutions obtained from the leader selection strategy are directly related to the mechanisms of the solutions obtained

Table 19 Mean and STD values (MS) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	0.94888	0.24464	0.57071	0.09218	**0.42535**	0.03494	0.55596	0.17227
BT2	0.57787	0.02646	0.56603	0.03022	0.93672	0.07156	**0.17538**	0.00335
BT3	0.37730	0.01958	0.47827	0.08841	1.12794	0.09782	**0.18808**	0.01888
BT4	**0.40513**	0.03300	0.48531	0.03882	0.64147	0.02780	0.90704	0.07238
BR1	0.91035	0.03394	1.03750	0.07266	**0.87204**	0.02760	0.93253	0.03659
BR2	1.15222	0.04359	1.01586	0.10937	**0.97019**	0.02518	1.02811	0.00335
BR3	0.94275	0.02784	0.94989	0.04359	**0.84722**	0.02647	1.20474	0.00034
BR4	1.48373	0.07734	**1.20272**	0.09814	1.74039	0.04053	1.50804	0.25737
BR5	0.43933	0.01480	0.70446	0.02695	0.66422	0.04605	**0.35419**	0.01684
BR6	0.74956	0.03040	1.02871	0.05463	**0.65411**	0.03608	0.66613	0.00763

Table 20 Mean and STD values (SP) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	5.78657	3.05918	3.00298	0.71214	**0.01628**	0.00093	0.04294	0.02048
BT2	**0.02344**	0.00347	0.02537	0.00316	0.06697	0.00903	0.03796	0.00060
BT3	0.04203	0.00351	0.03528	0.01105	**0.02912**	0.00941	0.04781	0.00285
BT4	0.03995	0.00251	0.05703	0.00442	**0.02898**	0.00357	0.06619	0.00477
BR1	0.11551	0.01238	0.24660	0.06839	**0.11378**	0.01266	0.19854	0.02102
BR2	6874.64	700.29	4632.61	325.71	**4343.24**	273.06	8699.67	0.00060
BR3	2.08792	0.15977	2.14897	0.19075	**1.70102**	0.11998	7.02285	0.00129
BR4	0.00607	0.00101	0.00333	0.00106	**0.00320**	0.00088	0.02666	0.01360
BR5	0.02710	0.00238	**0.01277**	0.00223	0.03427	0.00152	0.04673	0.00155
BR6	0.46576	0.02702	0.68691	0.07436	**0.46162**	0.03440	1.38578	0.00821

Table 21 Mean and STD values (HV) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	0.00000	0.00000	0.00000	0.00000	**0.83705**	0.00032	0.26032	0.26267
BT2	0.34689	0.03680	0.30283	0.01580	0.45802	0.02859	**0.57318**	0.00013
BT3	0.53416	0.00327	0.49430	0.04119	**0.56848**	0.02169	0.56440	0.00141
BT4	0.27747	0.00372	0.26672	0.00454	**0.32797**	0.00086	0.27577	0.00107
BR1	0.84164	0.00056	0.83330	0.00297	0.84125	0.00033	**0.84380**	0.00020
BR2	0.95826	0.00015	0.95865	0.00011	0.95875	2.257 E−05	0.95810	0.00013
BR3	0.85581	0.00025	0.85300	0.00042	**0.88653**	0.00020	0.85646	8.845 E−07
BR4	0.81884	0.00002	0.81870	0.00011	0.81888	0.00001	0.81885	6.408 E−09
BR5	0.65457	0.00761	0.54563	0.01253	**0.76110**	0.00240	0.66085	0.00254
BR6	0.78364	0.00013	0.78015	0.00050	0.78508	0.00045	0.78489	0.00001

from the teacher phase. To update the location of each student more than three times (the fourth time being random with a probability of less than 0.5), the teaching factor is activated. This results in a more accurate search and higher exploration capacity due to decreased drawbacks, subsequently, improving the algorithm's exploration and exploitation capacities.

Although the selection and grid mechanisms of MOOTLBO and MOPSO are similar, MOOTLBO results are superior to those of MOPSO because the particles' updating procedure for MOPSO has been performed using

Table 22 Mean and STD values (Delta-P) of all algorithms on the tri-objective, real-world and constrained test problems

Problem	MOPSO		MOGWO		MOTLBO		MOMPA	
	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev	Mean	STD.Dev
BT1	19.49481	12.43862	23.46095	2.98320	**0.01561**	0.00034	0.39464	0.23508
BT2	0.25308	0.06478	0.25574	0.03236	0.12371	0.02245	**0.03689**	0.00103
BT3	0.04937	0.00335	0.07826	0.01447	**0.04068**	0.00457	0.04288	0.00040
BT4	0.04833	0.00347	0.06077	0.00426	**0.04413**	0.00383	0.06680	0.00287
BR1	0.10693	0.01230	0.26658	0.10350	**0.10645**	0.01220	0.18295	0.01334
BR2	5996.594	581.474	4017.473	562.110	**3512.970**	187.044	7439.707	0.00103
BR3	1.29931	0.06353	1.76440	0.32786	**1.23360**	0.04533	5.42468	0.00134
BR4	0.01747	0.01701	0.09642	0.03914	**0.00840**	0.00110	0.10342	0.03249
BR5	0.08234	0.00560	0.28894	0.02559	**0.04015**	0.00043	0.04926	0.00144
BR6	0.56597	0.02225	0.69434	0.20844	**0.54242**	0.03816	0.67200	0.00505

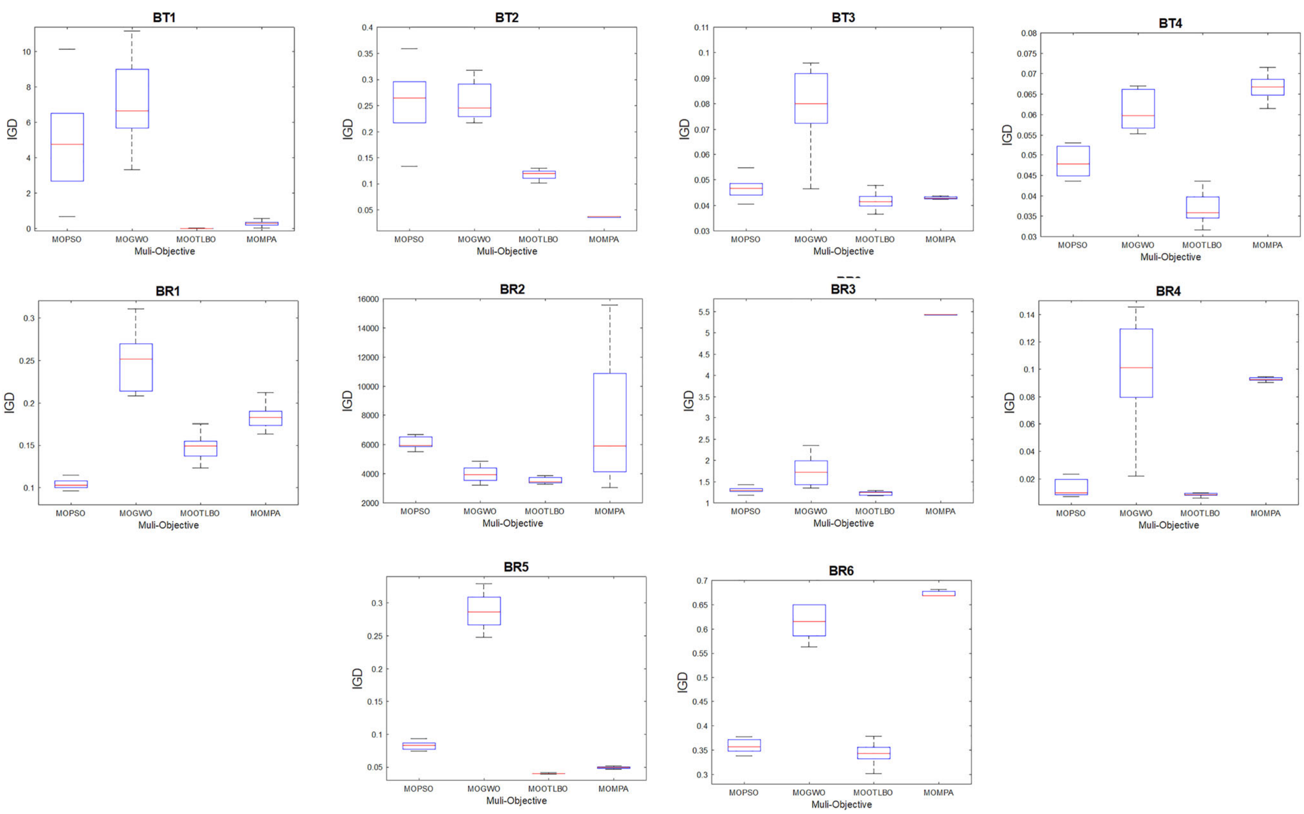

Fig. 6 Boxplot of the statistical results for IGD on BT1–BR6

the non-dominated solution so far (Nebro et al. 2013). But MOOTLBO has employed at least three non-dominated solutions so far. In addition, during each repeat in MOPSO, all particles are absorbed by identical global bests since the global best is only updated once at each iteration. MOOTLBO leaders are updated for every position update. Consequently, search agents are capable of doing a more extensive search in the search space.

From convergence and coverage perspectives, MOOTLBO has performed better than MOGWO. One reason may be due to the mutation operator, which slows early convergence and contributes to a more accurate and extensive search of the search space. The superiority in convergence and maintaining the diversity of solutions represented by MOOTLBO in comparison with MOGWO can be attributed to the employment of the TF operator several times, which is related to the selection mechanism for obtained solutions by leaders and the teacher phase. To sum up, the multi-objective of MOOTLBO can represent a proper and comparative performance in solving linear,

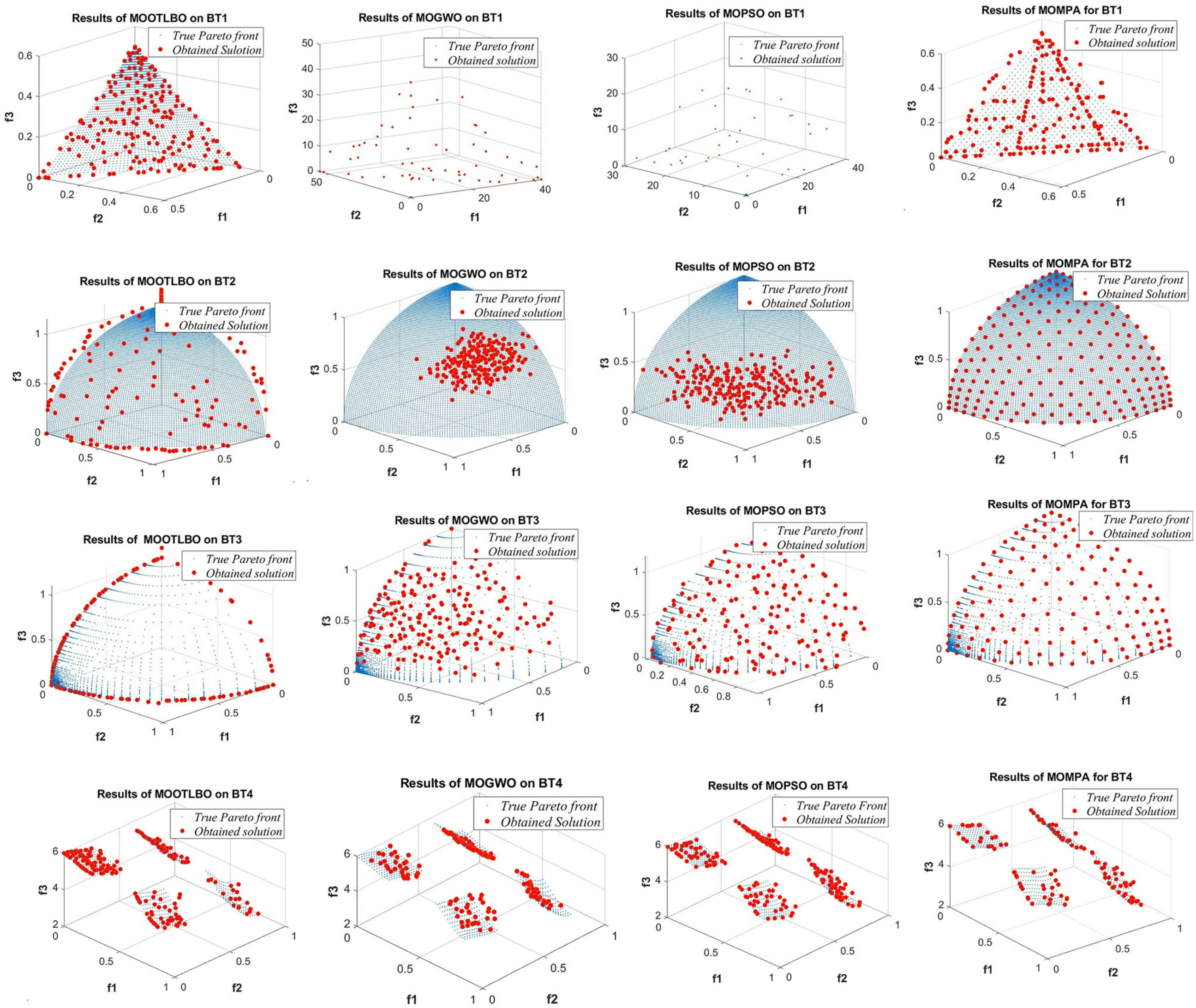

Fig. 7 Obtained Pareto-optimal solutions by MOOTLBO, MOGWO, MOPSO and MOMPA for BT1–BT4

discontinuous, convex, non-convex and multi-modal problems.

5 Conclusion

The MOOTLBO algorithm, a multi-objective version of the OTLBO algorithm, was developed in this article. The OTLBO algorithm's primary mechanism was maintained, and three new components, namely the mutation operator, archive and leader selection, were integrated into the algorithm to enable multi-objective optimization. The MOOTLBO algorithm was tested on nine standard bi-objective test problems, four standard tri-objective test problems and six standard real-world and constraint problems. The algorithm's performance was compared with three well-known algorithms, namely MOPSO, MOGWO and MOMPA. The evaluations were conducted using various metrics, including IGD, radial, GD, MS, SP, HP and Delta-P. The results demonstrated the MOOTLBO algorithm's ability to provide highly comparative results. The algorithm was tested on different problems with diverse Pareto-optimal fronts, and it was found to be capable of finding the Pareto-optimal front of any shape. Based on the study, it can be concluded that the MOOTLBO algorithm has significant advantages in solving multi-objective problems.

Meta-heuristic algorithms typically require common controlling parameters, such as population size and the number of iterations, while various algorithms have their unique controlling parameters. One advantage of the MOOTLBO algorithm is that it does not require any special controlling parameter. However, as previously mentioned, the teaching factor (TF) can impact convergence

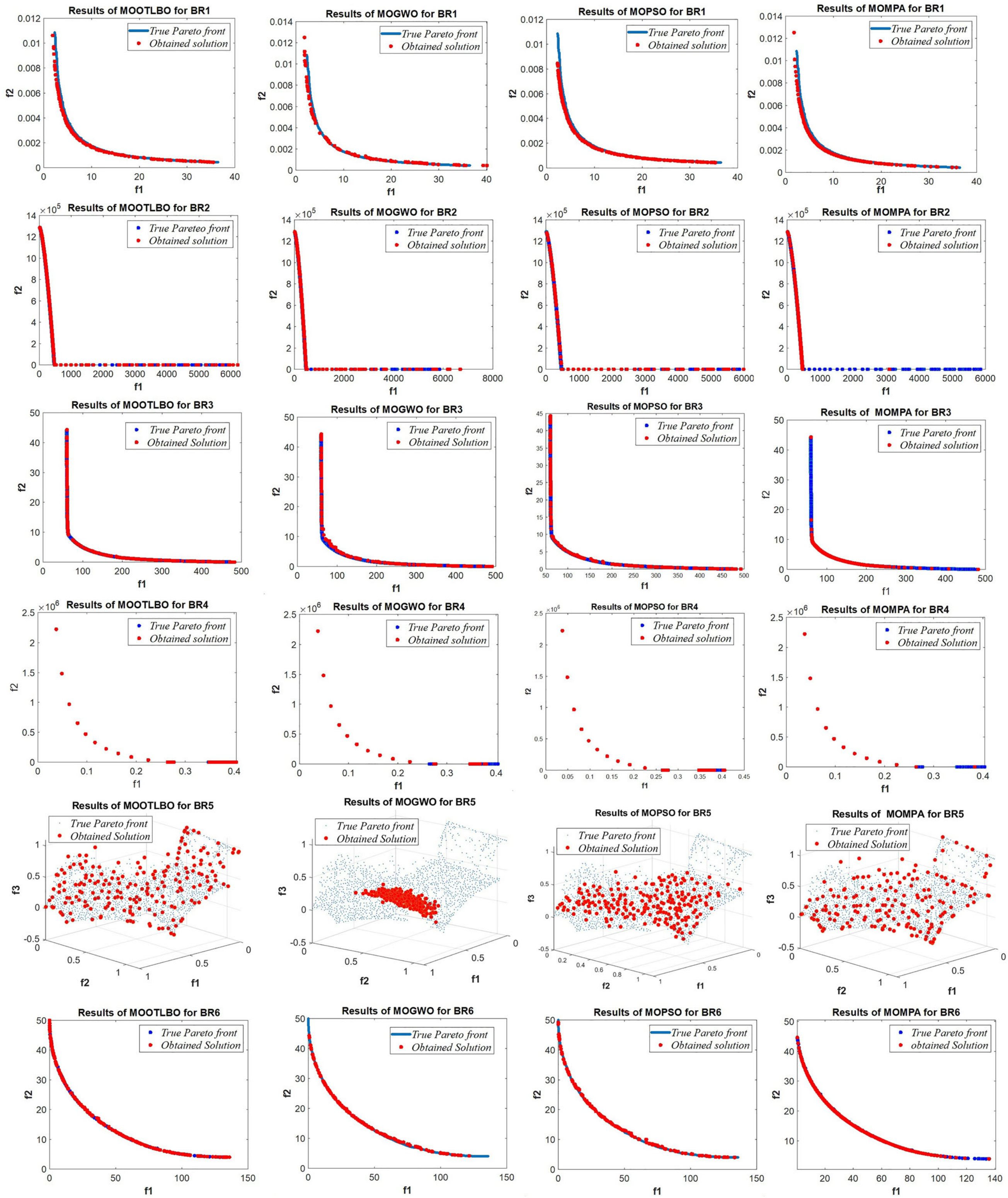

Fig. 8 Obtained Pareto-optimal solutions by MOOTLBO, MOGWO, MOPSO and MOMPA for BR1–BR6

Table 23 Results of Wilcoxon signed-rank test for IGD and HV metrics

Copmarition	IGD			HV		
	R^+	R^-	Exact *P*-value	R^+	R^-	Exact *P*-value
MOOTLBO versos MOMPA	42	148	0.033	138	52	0.084
MOOTLBO versos MOGWO	0	190	0	150	40	0.027
MOOTLBO versos MOPSO	10	180	0.001	188	2	0

Table 24 Results of Friedman aligned test for SP and Delta-P metrics

SP metric			Delta-P metric		
Rank	Algorithm	Ranking	Rank	Algorithm	Ranking
1	MOOTLBO	1.63	1	MOOTLBO	1.42
2	MOPSO	2.16	2	MOMPA	2.37
3	MOGWO	2.89	3	MOPSO	2.89
4	MOMPA	3.32	4	MOGWO	3.32

and search speed, and future studies can consider corrected values of this factor. The authors recommend using MOOTLBO for other engineering design problems.

Acknowledgements Not applicable.

Author contribution SD carried out the review analysis and modelling and participated in drafting the manuscript. AS proposed the topic, participated in coordination and aided in the interpretation of results and paper editing. SHM carried out the visualization and paper editing. All authors read and approved the final manuscript.

Funding No funding.

Data availability Please contact the corresponding author for data requests.

Code availability Please contact the corresponding author for code requests.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval Not applicable.

References

Al-Aomar R, Al-Okaily A (2006) A GA-based parameter design for single machine turning process with high-volume production. Comput Ind Eng 50(3):317–337. https://doi.org/10.1016/j.cie.2006.02.003

Amir HM, Hasegawa T (1989) Nonlinear mixed discrete structural optimization. J Struct Eng 115(3):626–646. https://doi.org/10.1061/(ASCE)0733-9445(1989)115:3(626)

Binh TT, Korn U (1997) MOBES: a multiobjective evolution strategy for constrained optimization problems. In: Proc Mendel'97, 3rd Int Conf Genetic Algorithms, pp 176–182. https://cir.nii.ac.jp/crid/1571980076303240064

Branke J, Kaußler T, Schmeck H (2001) Guidance in evolutionary multi-objective optimization. Adv Eng Softw 32(6):499–507. https://doi.org/10.1016/S0965-9978(00)00110-1

Buch H, Trivedi IN (2020) A new non-dominated sorting ions motion algorithm: development and applications. Decis Sci Lett 9(1):59–76. https://doi.org/10.5267/J.DSL.2019.8.001

Cabrera JCF, Coello CAC (2010) Micro-MOPSO: a multi-objective particle swarm optimizer that uses a very small population size. Stud Comput Intell 261:83–104. https://doi.org/10.1007/978-3-642-05165-4_4

Coello CA, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279. https://doi.org/10.1109/TEVC.2004.826067

Coello Coello CA (2006) Evolutionary multi-objective optimization: a historical view of the field. IEEE Comput Intell Mag 1(1):28–36. https://doi.org/10.1109/MCI.2006.1597059

Coello CA, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. In: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer US

Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657. https://doi.org/10.1137/S1052623496307510

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017

Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Evolutionary multiobjective optimization, Springer-Verlag, pp 105–145

Deb K (2001) Multi-objective optimization using evolutionary algorithms. John Wiley & Sons. https://books.google.com/books/about/Multi_Objective_Optimization_Using_Evolu.html?id=OSTn4GSy2uQC&printsec=frontcover&source=kp_read_button#v=onepage&q&f=false

Doroudi S, Sharafati A, Mohajeri SH (2021) Estimation of daily suspended sediment load using a novel hybrid support vector regression model incorporated with observer–teacher–learner-based optimization method. Complexity. https://doi.org/10.1155/2021/5540284

Durairaj M, Sudharsun D, Swamynathan N (2013) Analysis of process parameters in wire EDM with stainless steel using single objective Taguchi method and multi objective grey relational grade. Proc Eng 64:868–877. https://doi.org/10.1016/j.proeng.2013.09.163

Eberhart R, Kennedy J (1995) New optimizer using particle swarm theory. Proceedings of the International Symposium on Micro Machine and Human Science, pp 39–43
Eberhart R, Shi Y, Kennedy J (2001) Swarm intelligence—1st Edition. https://www.elsevier.com/books/swarm-intelligence/eberhart/978-1-55860-595-4
Elsayed EA, Chen A (1993) Optimal levels of process parameters for products with multiple characteristics. Int J Prod Res 31(5):1117–1132. https://doi.org/10.1080/00207549308956778
Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701. https://doi.org/10.1080/01621459.1937.10503522
Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506. https://doi.org/10.1109/TEVC.2005.861417
Jangir P, Buch H, Mirjalili S, Manoharan P (2021) MOMPA: multi-objective marine predator algorithm for solving multi-objective optimization problems. Evol Intel 2021:1–27. https://doi.org/10.1007/S12065-021-00649-Z
Kannan BK, Kramer SN (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116(2):405–411. https://doi.org/10.1115/1.2919393
Kennedy J (2010) Encyclopedia of machine learning. Encycl Mach Learn. https://doi.org/10.1007/978-0-387-30164-8
Kim IY, De Weck OL (2005) Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Struct Multidiscip Optim 29(2):149–158. https://doi.org/10.1007/s00158-004-0465-1
Knowles J, Corne D (1999) The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC, vol 1, pp 98–105
Lampinen J, Zelinka I (1999) Mixed integer-discrete-continuous optimization by differential evolution part 1: the optimization method.
Lavasani SHH, Doroudi R (2020) Meta heuristic active and semi-active control systems of high-rise building. Int J Struct Eng 10(3):232–253. https://doi.org/10.1504/IJSTRUCTE.2020.108529
Lewis A, Mostaghim S, Scriven I (2009) Asynchronous multi-objective optimisation in unreliable distributed environments. Stud Comput Intell 210:51–78. https://doi.org/10.1007/978-3-642-01262-4_3
Liang J, Qu B, Gong D, Yue C (2019) Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization. Special Session & competition in IEEE congress on evolutionary computation 2019. https://www.researchgate.net/publication/331815925_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2019_Special_Session_on_Multimodal_Multiobjective_Optimization?channel=doi&linkId=5c8da5f092851c1df94630b4&showFulltext=true
Logothetis N, Haigh A (1988) Characterizing and optimizing multi-response processes by the taguchi method. Qual Reliab Eng Int 4(2):159–169. https://doi.org/10.1002/qre.4680040211
Meyr H (2009) Supply chain planning in the german automotive industry. In: Supply chain planning: quantitative decision support and advanced planning solutions, Springer Berlin Heidelberg, pp 343–365
Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073. https://doi.org/10.1007/S00521-015-1920-1/TABLES/12
Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
Mirjalili S, Saremi S, Mirjalili SM, Coelho LDS (2016) Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl 47:106–119. https://doi.org/10.1016/j.eswa.2015.10.039
Mirjalili S, Jangir P, Saremi S (2017) Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Appl Intell 46(1):79–95. https://doi.org/10.1007/s10489-016-0825-8
Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. Appl Intell 48(4):805–820. https://doi.org/10.1007/S10489-017-1019-8/TABLES/9
Mozdgir A, Mahdavi I, Badeleh IS, Solimanpur M (2013) Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing. Math Comput Model 57(1–2):137–151. https://doi.org/10.1016/j.mcm.2011.06.056
Muthuramalingam T, Mohan B (2014) Application of Taguchi-grey multi responses optimization on process parameters in electro erosion. Meas J Int Meas Confed 58:495–502. https://doi.org/10.1016/j.measurement.2014.09.029
Nebro AJ, Durillo JJ, Coello CAC (2013) Analysis of leader selection strategies in a multi-objective particle swarm optimizer. In: 2013 IEEE Congress on Evolutionary Computation, CEC, pp 3153–3160
OuYang J, Yang F, Yang SW, Nie ZP (2008) The Improved NSGA-II Approach. J Electromagn Waves Appl 22(2–3):163–172. https://doi.org/10.1163/156939308784160703
Pradhan PM, Panda G (2012) Solving multiobjective problems using cat swarm optimization. Expert Syst Appl 39(3):2956–2964. https://doi.org/10.1016/j.eswa.2011.08.157
Premkumar M, Jangir P, Kumar BS, Sowmya R, Alhelou HH, Abualigah L, Yildiz AR, Mirjalili S (2021) A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations. IEEE Access 9:84263–84295. https://doi.org/10.1109/ACCESS.2021.3085529
Premkumar M, Jangir P, Sowmya R, Alhelou HH, Mirjalili S, Kumar BS (2022) Multi-objective equilibrium optimizer: framework and development for solving multi-objective optimization problems. J Comput Des Eng 9(1):24–50. https://doi.org/10.1093/JCDE/QWAB065
Rao RV, Patel V (2013) An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. Sci Iran 20(3):710–720. https://doi.org/10.1016/j.scient.2012.12.005
Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. CAD Comput Aided Des 43(3):303–315. https://doi.org/10.1016/j.cad.2010.12.015
Ray T, Liew KM (2010) A swarm metaphor for multiobjective design optimization. Eng Optim 34(2):141–153. https://doi.org/10.1080/03052150210915
Reyes-Sierra M, Coello CAC (2006) Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int J Comput Intell Res 2(3):287–308
Roy RK (2010) A primer on the taguchi method. Soc Manuf Eng
Sadollah A, Eskandar H, Kim JH (2015) Water cycle algorithm for solving constrained multi-objective optimization problems. Appl Soft Comput 27:279–298. https://doi.org/10.1016/J.ASOC.2014.10.042

Shahrouzi M, Aghabagloua M, Rafiee F (2017) Observer–teacher–learner-based optimization: an enhanced meta-heuristic for structural sizing design. Struct Eng Mech 62(5):537–550. https://doi.org/10.12989/sem.2017.62.5.537

Shahrouzi M (2011) A new hybrid genetic and swarm optimization for earthquake accelerogram scaling. In: Iran University of Science & Technology, vol 1, Iran University of Science & Technology. http://ijoce.iust.ac.ir/article-1-11-en.html

Sierra MR, Coello Coello CA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. Lect Notes Comput Sci 3410:505–519. https://doi.org/10.1007/978-3-540-31880-4_35

Srinivas N, Deb K (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248. https://doi.org/10.1162/EVCO.1994.2.3.221

Taguchi G (1987) System of experimental design. Response. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:System+of+experimental+design#0

Tan KC, Lee TH, Khor EF (2002) Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. Artif Intell Rev 17(4):251–290. https://doi.org/10.1023/A:1015516501242/METRICS

Tran DH, Cheng MY, Prayogo D (2016) A novel multiple objective symbiotic organisms search (MOSOS) for time–cost–labor utilization tradeoff problem. Knowl Based Syst 94:132–145. https://doi.org/10.1016/J.KNOSYS.2015.11.016

Unnikrishna Pillai J, Sanghrajka I, Shunmugavel M, Muthuramalingam T, Goldberg M, Littlefair G (2018) Optimisation of multiple response characteristics on end milling of aluminium alloy using Taguchi-Grey relational approach. Meas J Int Meas Confed 124:291–298. https://doi.org/10.1016/j.measurement.2018.04.052

Vaidyanathan R, Kevin Tucker P, Papila N, Shyy W (2003) CFD-based design optimization for a single element rocket injector. In: 41st Aerospace Sciences Meeting and Exhibit. https://doi.org/10.2514/6.2003-296

Vikas, Nanda SJ (2016) Multi-objective moth flame optimization. In: 2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI, pp 2470–2476. https://doi.org/10.1109/ICACCI.2016.7732428

Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80. https://doi.org/10.2307/3001968

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. https://doi.org/10.1109/4235.585893

Yang X-S (2012) Bat algorithm for multi-objective optimisation. Int J Bio-Inspired Comput 3(5):267–274

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731. https://doi.org/10.1109/TEVC.2007.892759

Zhang M, Wang H, Cui Z, Chen J (2018) Hybrid multi-objective cuckoo search with dynamical local search. Memet Comput 10(2):199–208. https://doi.org/10.1007/S12293-017-0237-2/METRICS

Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S (2009) Rescheduling view project evolutionary algorithms for power system optimisation view project multiobjective optimization test instances for the CEC 2009 special session and competition. https://www.researchgate.net/publication/265432807

Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and applications (Berichte Aus Der Informatik): Zitzler, Eckart: 9783826568312: Amazon.com: Books. Shaker Verlag GmbH, Germany. https://www.amazon.com/Evolutionary-Algorithms-Multiobjective-Optimization-Applications/dp/3826568311

Zou F, Wang L, Hei X, Chen D, Wang B (2013) Multi-objective optimization using teaching-learning-based optimization algorithm. Eng Appl Artif Intell 26(4):1291–1300. https://doi.org/10.1016/J.ENGAPPAI.2012.11.006