



MOBRO: multi-objective battle royale optimizer

Sait Alp¹ · Rahim Dehkharghani² · Taymaz Akan^{3,4} · Mohammad A. N. Bhuiyan³

Accepted: 18 September 2023 / Published online: 16 October 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Battle Royale Optimizer (BRO) is a recently proposed optimization algorithm that has added a new category named game-based optimization algorithms to the existing categorization of optimization algorithms. Both continuous and binary versions of this algorithm have already been proposed. Generally, optimization problems can be divided into single-objective and multi-objective problems. Although BRO has successfully solved single-objective optimization problems, no multi-objective version has been proposed for it yet. This gap motivated us to design and implement the multi-objective version of BRO (MOBRO). Although there are some multi-objective optimization algorithms in the literature, according to the no-free-lunch theorem, no optimization algorithm can efficiently solve all optimization problems. We applied the proposed algorithm to four benchmark datasets: CEC 2009, CEC 2018, ZDT, and DTLZ. We measured the performance of MOBRO based on three aspects: convergence, spread, and distribution, using three performance criteria: inverted generational distance, maximum spread, and spacing. We also compared its obtained results with those of three state-of-the-art optimization algorithms: the multi-objective Gray Wolf optimization algorithm (MOGWO), the multi-objective particle swarm optimization algorithm (MOPSO), the multi-objective artificial vulture's optimization algorithm (MOAVAO), the optimization algorithm for multi-objective problems (MAOA), and the multi-objective non-dominated sorting genetic algorithm III (NSGA-III). The obtained results approve that MOBRO outperforms the existing optimization algorithms in most of the benchmark suites and operates competitively with them in the others.

Keywords Optimization · Battle-royale-game-based optimization algorithms · Battle royale optimization algorithm · Multi-objective problems

1 Introduction

Many optimization algorithms have been proposed for different types of problems; however, according to the “No free lunch” theorem [1], no optimization algorithm can be appropriate for solving all types of problems. That is why new algorithms are still being proposed in the literature. Until 2021, optimization algorithms could be classified into three main categories: Evolutionary Algorithms (EA) such as Genetic algorithm [2], and Bird mating optimizer [3], Swarm Intelligence (SI) algorithms such as Particle Swarm Optimization (PSO) algorithm [4], Artificial Bee Colony algorithm [5], and Chimp optimization algorithm, and Physical Phenomena (PP) algorithms such as Passing Vehicle search [6], and Gravitational Search algorithm [7].

In 2021, a new optimization algorithm named Battle Royale Optimization (BRO) [8] added a new category named “Game-based” optimization algorithms [9] to the literature. The original version of the BRO algorithm was designed only for solving problems of a continuous nature, and it was incapable of solving problems of a discrete nature. Later, the Binary Battle Royale algorithm (BinBRO) was proposed for solving binary optimization problems. Still, all those algorithms can only solve single-objective problems. However, some problems have a multi-objective nature, in which there is more than one objective to be satisfied (minimized or maximized). As most real-world problems are multi-objective, tackling multi-objective problems is essential. For example, when making a trade, the trader is usually interested in low prices and high quality, while these two objectives are conflicting, i.e., achieving one goal may compromise another. Therefore, there is no universal solution that can minimize/maximize all objective functions for all problems; instead, there are Pareto optimal solutions, which offer the best trade-offs between conflicting objectives [10]. Multi-objective problems may have two or more objectives. Multi-objective problems with three or more objectives are known as “many-objective problems” [11].

In this paper, we propose the multi-objective version of the Battle Royale Optimization algorithm for the first time in the literature and compare its performance with the multi-objective version of five state-of-the-art algorithms: the Gray Wolf optimization algorithm (MOGWO), the Multi-Objective Particle Swarm Optimization algorithm (MOPSO), the Multi-Objective Artificial Vultures Optimization algorithm (MOAVAO), the Arithmetic Optimization Algorithm for Multi-Objective Problems (MAOA), and the Multi-Objective Non-dominated Sorting Genetic Algorithm III (NSGA-III).

Multi-objective problems can be divided into constrained and unconstrained groups. In constrained problems, some constraints restrict the values that can be assigned to variables, while in unconstrained problems, no such restriction exists. The terminology for multi-objective problems is provided as follows:

- *Multi-objective problem* A quadruple $\langle V, D, F, S \rangle$ represents a multi-objective problem, where V is a set of variables $V = \{v_1, v_2, \dots, v_m\}$ to be initialized by values taken from a set of domains $D = \{d_1, d_2, \dots, d_m\}$, so that objectives $F = \{f_1, f_2, \dots, f_p\}$ are minimized (or maximized).
- *Constrained Multi-objective problem (CMOP)* A CMOP is a multi-objective problem, in which, the goal is to find a vector of solutions, X , which optimizes

$f(X)$ subject to a set of inequality and equality constraints while satisfying some conflicting objectives:

$$g_i(X) \leq 0, \quad i = 1, \dots, m$$

$$h_i(X) = 0, \quad i = 1, \dots, k$$

Where m and k are, respectively, the number of inequality and equality constraints. These constraints can be linear or nonlinear [12].

The feasible and infeasible solution A solution for CMOPs would be named as feasible if it satisfies all constraints, or infeasible otherwise.

On the other hand, $S = \{s_1, s_2, \dots, s_p\}$ as a set of candidate solutions can be divided into two subsets: population (dominated) and repository (non-dominated) candidate solutions. The non-dominated solutions are placed on a border so-called the Pareto front (PF).

- *Domination Solution* x dominates solution y if x is as good as y for all objectives and better than it at least in one objective: $x < y \Leftrightarrow \forall i \in \{1, \dots, p\} f_i(x) \leq f_i(y)$ and $\exists_i \in \{1, \dots, p\} f_i(x) < f_i(y)$.
- *Pareto optimal* A candidate solution x is called Pareto optimal (PO) if there exists no other candidate solution y that can dominate x : x is *PO* $\Leftrightarrow \exists y \in S$, such that $y < x$.
- *Pareto Optimal set (PS)* This set includes all Pareto optimal solutions: $PS^* = \{s_i | s_i \text{ is PO}\}$.
- *Pareto optimal front* Given the Pareto Optimal set and objective function F , Pareto optimal front (*POF*) is the output values obtained by applying the objective functions to solutions in Pareto Optimal set: $POF = \{F(x) = (f_1(x), f_2(x), \dots, f_p(x)) \mid x \in PS^*\}$.

There are different categorizations of methods for solving multi-objective problems. Those categorizations generally differ in the way they choose the next move to achieve the global optima, i.e., they use different preference methods to find the preferred solutions. So, preference information is required to guide the search to the Pareto front area. The ultimate goal is to find a solution that is approved by a decision-maker. The decision-makers are either individual experts regarding the matter at hand or methods automatically applied to the selection procedure for choosing the next move. Hwang and Masud [13] classify the existing approaches for solving multi-objective problems into four different groups based on preference methods:

- *No-preference method* In this approach, the MO problem is solved by employing a relatively straightforward method (classical single-objective optimization algorithms), and the obtained solution is delivered to the decision-maker; however, the suggestions of the decision-maker are not taken into consideration.
- *Posteriori methods* In this approach, a decision-maker indicates its preferences a posteriori after receiving information about the trade-offs among the non-dominated solutions. Specifically, the decision-maker indicates which solution is the optimal one found so far. Our proposed algorithm, MOBRO, belongs to

this category and determines the optimal solution in each iteration using the Roulette Wheel selection method. Like MOPSO [14], we use a grid mechanism for measuring the density of the PF area. The sparser areas in the PF have a higher probability of including the optimal solution(s). If more than one solution exists in the sparsest grid, one of them is randomly chosen as optimal.

- *Priori methods* The objectives may be converted into a single objective by assigning a weight to each objective and then combining them. The associated weights should be defined in advance by the decision-makers.
- *Interactive methods* In this approach, the objective functions, constraints, and their respective priorities are updated by getting user feedback on preferences several times throughout the optimization process.

Several research works have solved multi-objective problems using different methods. According to [15], multi-objective optimization algorithms generally rely either on non-dominated sorting methods or archive-based methods when searching for optimal solutions [14, 16]. The proposed approach, MOBRO, lies in the latter group (archive-based methods). One of the archive-based approaches has been proposed in [16], in which the authors propose a multi-objective version of the Gray Wolf optimization (GWO) algorithm, so-called MOGWO, by using a grid mechanism to detect and store the non-dominated solutions (in the archive) and choosing three solutions as leaders—global optima found so far—to guide other solutions in the population toward regions that may contain the global optimum. Social leadership was the main inspiration for GWO. As mentioned already, in archive-based multi-objective optimization algorithms, those solutions that are closer to the global optima are also stored in a separate set named a repository or archive. Despite its success in solving some single-objective problems, due to the nature of the original GWO algorithm, MOGWO suffers from a shortage of candidate leaders in the Pareto front as it requires three leaders (alpha, beta, and gamma), instead of one to solve the optimization problems. In other words, in some cases, the GWO algorithm is unable to provide more than one or two leaders. The most popular archive-based algorithm in the literature, a multi-objective version of the PSO algorithm (MOPSO), was proposed in [14]. In this work, the authors add a constraint-handling mechanism and a mutation operator to the original PSO algorithm to achieve a better exploration mechanism. ϵ -MOABC [11], as another archive-based algorithm, a multi-objective version of the Artificial Bee Colony algorithm uses a relaxed form of Pareto-dominance, named ϵ -dominance. A combined mutation operator is suggested in MOABC to prevent the basic ABC algorithm from being trapped in a local Pareto front. In [17], Mirjalili et al. adapt the Grasshopper optimization algorithm (GOA) for multi-objective optimization problems by using an archive and target selection technique to estimate the optimal solutions. The authors believe that GOA benefits from high exploration while showing a very fast convergence speed, and its balance between exploration and exploitation makes it an appropriate algorithm for multi-objective problems. The Ant colony algorithm has also been adapted to solve multi-objective problems [18]. This algorithm creates a pheromone matrix as an archive and stores the best N solutions in it. In [19], the authors propose a multi-objective version of the artificial

vultures optimization algorithm (AVOA) for solving multi-objective optimization problems, so-called MOAVOA. The archive and grid mechanisms are also used in this algorithm. Another archive-based multi-objective arithmetic optimization algorithm (MAOA) is proposed in [20] for solving industrial engineering problems. The original AVOA is based on the distribution behaviour of four arithmetic operators. The most popular multi-objective algorithm among the sorting-based algorithms is probably NSGA III, proposed in [21], which is an evolutionary many-objective optimization algorithm using a reference-point-based non-dominated sorting approach based on the Genetic Algorithm for solving unconstrained multi-objective problems. This algorithm, as an extension of its previous version, NSGA II [22], attempts to preserve diversity among the solutions using a set of reference directions. There exist also other algorithms in the literature proposed for solving multi-objective problems [23–26]

It is very difficult to measure the performance of an optimization algorithm in theory; therefore, researchers apply their proposed optimization algorithms to benchmark datasets and compare them with other algorithms [27]. We evaluated the proposed algorithm, MOBRO, by applying it to four benchmark datasets: the CEC 2009 [28], CEC 2018 [29], ZDT [30], and DTLZ [31] datasets. The obtained results prove that MOBRO can solve unconstrained multi-objective problems; it also competes with other state-of-the-art MO optimization algorithms and, in most of the benchmark suites, surpasses them.

The remaining sections of the paper are designed as follows: Sect. 2 explains the proposed multi-objective version of the Battle Royale Optimization algorithm. Section 3 provides an experimental evaluation of the proposed method, and Sect. 4 provides conclusions and future works.

2 MOBRO: multi-objective battle royale algorithm

The original version of the Battle Royale optimization algorithm was recently proposed by Farshi (2020). This optimization algorithm, which drew inspiration from the Battle Royale video game, added a new category to optimization algorithms: “game-based” optimization algorithms. In this category, each solution competes with its nearest neighbor, and better solutions, according to fitness value (stronger soldiers in the game), attempt to overcome the worse ones (weaker soldiers). If a solution loses a specified number of times when competing with other solutions, it will be removed from the problem space and will be regenerated using Eq. 1. If the number of losses for a solution does not reach a threshold, it will be updated by Eq. 2, hoping that it will approach the best solution found so far in dimension d .

$$x_{\text{dam},d} = r(ub_d - lb_d) + lb_d \quad (1)$$

$$x_{\text{dam},d} = x_{\text{dam},d} + r(x_{\text{best},d} - x_{\text{dam},d}) \quad (2)$$

In these equations, r is a random number generated using a uniform distribution in the range $[0, 1]$; $X_{\text{dam},d}$ and $X_{\text{best},d}$ are the position of the damaged (looser) and the

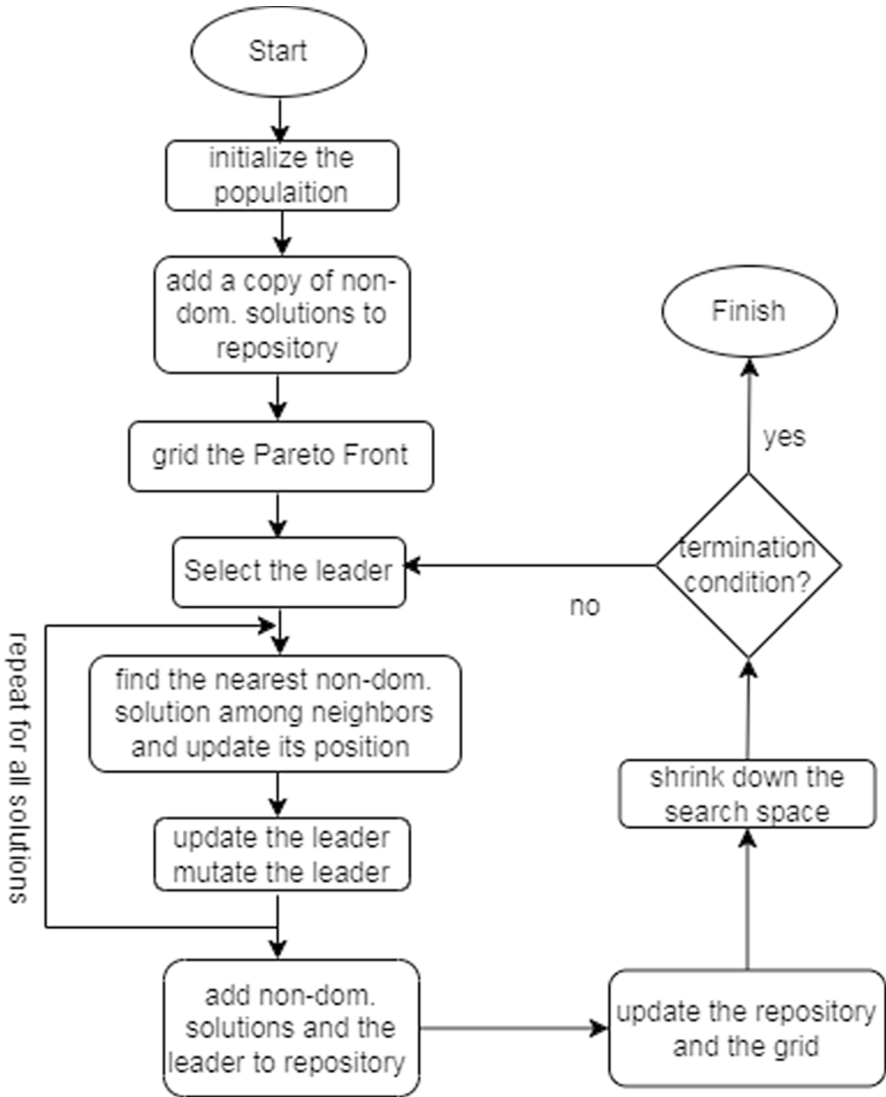


Fig. 1 The proposed algorithm, MOBRO, as a flowchart

best solution (found so far). The lower and the upper bounds of dimension d in the problem space are denoted by lb_d and ub_d .

To move a solution toward the best one, Akan and Akan [32] updated Eq. 2 as follows:

$$\begin{aligned}
 \lambda_{\text{dam}, d} &= r_1 x_{\text{best}, d} + r_2 (\lambda_{\text{dam}, d} - x_{\text{dam}, d}), \\
 x_{\text{dam}, d} &= \lambda_{\text{dam}, d} + x_{\text{dam}, d}
 \end{aligned}
 \tag{3}$$

where r_1 and r_2 are independent numerical values randomly chosen from the interval $[0, 1]$, and $\lambda_{dam,d}$ is also a combination of randomly generated values taken from a uniform distribution $[0, 1]$ that will be updated in each iteration. The updated movement strategy is more stochastic in exploration and exploitation.

The key point of this algorithm is that the safe zone (game field) shrinks down by $\Delta = \Delta + \text{round}(\Delta/2)$, where Δ is initialized by $\log_{10}(\text{MaxCircle})$; MaxCircle is the maximum number of generations. This space restriction will force all solutions to move toward the best one found so far. Diminishing the space is done based on Eq. 4, in which, $SD(x_d)$ is the standard deviation of the population in dimension d .

$$\begin{aligned} lb_d &= x_{best,d} - SD(\bar{x}_d) \\ ub_d &= x_{best,d} + SD(\bar{x}_d) \end{aligned} \tag{4}$$

The overall methodology of MOBRO is provided as a pseudo-code in Algorithm 1 and also as a flowchart in Fig. 1. As seen in this algorithm, in the beginning, some solutions are randomly generated (initialized). As mentioned in the introduction, an attribute of the multi-objective version of BRO is using a set named archive (or repository). This set stores copies of non-dominated solutions. If a solution is not dominated by any other solution, it should be added to the archive. If any candidate solution inside the archive gets dominated by a new solution, the dominated solution will be replaced by a new (dominating) one. Solutions can remain in the archive as long as they are non-dominated [16]. In the MOBRO algorithm, the non-dominated solutions are determined, and a copy of them is added to the archive after evaluating the solutions.

Algorithm 1 MOBRO for unconstraint problems

```

1: Initialize population
2: Evaluate all solutions in the population
3: Determine non-dominated solutions
4: Add a copy of non-dominated solutions to the Repository (Archive)
5: Grid the Pareto front area
6: while not terminate do
7:   Select the Leader (Flag)
8:   for all solutions do
9:     check the neighbours to determine a non-dominated solution in order of distance
10:    Update the position of the damaged (looser) solution based on Eq.1 or Eq.3
11:    Evaluate the damaged solution
12:    Apply the mutation on the leader and update the leader if the mutation could dominate
13:    the leader (Flag)[14,34]
14:    Shrink the search space based on Eq. 4
15:    the leader will be replaced by the damaged solution if the damaged solution dominates
16:    the leader.
17:    Choose a candidate leader from the repository and check if it dominates the current leader. If so,
18:    update the leader
19:   End for
20:   Determine non-dominated solutions in the population
21:   Add the leader and non-Dominated solutions to the repository
22:   Determine non-dominated solutions in the repository
23:   Update repository (Keep non-dominated solutions & remove dominated ones)
24:   Update Grid
25:   Check the repository population limit
26: End while

```

Then, the whole process is repeated for a predefined number of iterations. As the first task in each iteration, a flag solution (assumed to be the global optimum)

is chosen out of solutions in the repository using a selection strategy based on the roulette wheel selection method [14]. Concretely, the Pareto Front region is divided into grids using a gridding mechanism, and one of the solutions in the least crowded region of the Pareto Front is selected as the flag (leader). The gridding mechanism keeps the candidate solutions in the archive as diverse as possible. Gridding recursively divides the Pareto front space into grids and covers all solutions. If any solution lies outside the gridding region, gridding should be repeated. The solutions belonging to each grid are recorded, so the number of solutions in each grid can be easily calculated [16]. The advantages of this mechanism are twofold: (1) low computational cost; and (2) no need for a niche-size parameter setting [34].

Then the following process is repeated for all solutions in the population: Each solution is compared to its neighbors, starting from the nearest to the farthest. This comparison stops when one solution (the winner) dominates the other (the damaged one). If no solution could dominate the other, the winner and the loser tags are randomly assigned to a solution and its nearest neighbor. When a solution gets damaged, its fault counter will be incremented by one, while the same counter will be set to zero for the winner. If the fault counter does not reach a threshold value, a crossover task [32], given in Eq. 3, will be applied to the loser, hoping to make it closer to the best solution. The intuition behind this policy is that the loser is not in an appropriate state (the soldier is in danger) and should be updated to move to a better position. If this cross-over causes the loser to leave the safe zone, it will be brought back to the edge of the safe zone. However, if the fault counter of the loser reaches a threshold (one of the parameters of MOBRO), this solution will be deleted and regenerated as another one with new values (as a soldier is respawned in a random area inside the safe zone). In each iteration, a mutation task [14, 33] is also applied to the flag. The flag will be updated if this mutation improves it or will remain unchanged, otherwise. Note that, similar to the original BRO algorithm, the search space shrinks once in Δ iterations based on Eq. 4. In all cases, the leader will be updated if the recently updated damaged solution dominates it. At the end of each iteration, a flag is chosen from the repository and is compared to the current flag. The current flag will be replaced by a new one if it dominates the current one.

As mentioned already, the search strategy explained above is repeated for all n solutions in the population, and again, the newly found non-dominated solutions, as well as the flag solution, are added to the repository (archive). If the repository becomes full, one of the existing solutions in its most crowded region will be removed. There are two reasons for removing the surplus solution from the repository: (1) newly added solutions might dominate the existing ones, and (2) the capacity of the repository becomes full. Finally, at the end of each iteration, a selection mechanism will be performed on the repository to update the flag.

3 Experimental results and performance evaluation

In this section, we first introduce the datasets and performance metrics used for evaluating the proposed algorithm and then provide the obtained results as well as a detailed discussion on them.

3.1 Dataset

As the theoretical evaluation of optimization algorithms in terms of their performance is difficult, benchmark datasets are used for this purpose. The benchmark multi-objective optimization problems of the competitions on evolutionary computation (CEC), CEC 2009 [28], and CEC 2018 [29] as well as two other datasets, ZDT [30] and DTLZ [31] have been used for the evaluation of the proposed algorithm and its comparison with five state-of-the-art algorithms. Tables 1 and 2, respectively, list the mathematical presentation of different test problems in the CEC 2009, ZDT, and DTLZ datasets, including 10, 6, and 7 test problems. Due to the long equations, the mathematical presentation of CEC 2018 is provided in Table S1 in the Appendix file. It is worth noting that the crossover task for the DTLZ dataset is performed by Eq. 2, but the same task is performed by Eq. 3 for other datasets.

3.2 Performance metrics

Different performance metrics have been proposed in the literature for evaluating different aspects of a multi-objective optimization algorithm [35]. These aspects include convergence, spread, and distribution. The proposed metrics in the literature do not directly measure one aspect in isolation; instead, they measure different aspects while assigning different weights to each one. We evaluated the proposed algorithm using three metrics: Inverted Generational Distance (IGD) [35], Spacing (SP) [36], and Maximum spread [21] or spread in short. These metrics are provided in Eqs. 5–7, respectively.

$$\text{IGD} = \frac{\sqrt{\sum_{i=1}^m (r_i - e_i)}}{m} \quad (5)$$

where m is the number of true solutions in the Pareto optimal set, r_i and e_i are, respectively, the real and closest found solutions to the true one. This metric is calculated for m times, which is the number of true solutions. IGD is sensitive to the number of points found by the algorithm [37]. This metric favors those algorithms providing a denser area for the solutions than those providing a more distributed version of the Pareto front. IGD is the most commonly used metric for evaluating the performance of many-objective optimization algorithms [35]. Furthermore, it is cheaper to compute, i.e., its time complexity is lower than some other metric's such as hyper volume [38].

Table 1 The mathematical representation of different test cases in CEC 2009 dataset

Name	Mathematical formulation
UF1	$f1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} \left[x_j - \sin \left(6\pi x_1 \right) + \frac{i\pi}{n} \right]^2, f2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} \left[x_j - \sin \left(6\pi x_1 \right) + \frac{i\pi}{n} \right]^2$ $j1 = \{j j \text{ is odd and } 2 \leq j \leq n\} \text{ and } j2 = \{j j \text{ is even and } 2 \leq j \leq n\}$
UF2	$f1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f2 = \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $1 = \{j j \text{ is odd and } 2 \leq j \leq n\} \text{ and } j2 = \{j j \text{ is even and } 2 \leq j \leq n\}, \text{ and}$ $y_j = \begin{cases} x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4i\pi}{n}) + 0.6x_1] \cos(6\pi x_1 + \frac{i\pi}{n}) & j \in J_1 \\ x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4i\pi}{n}) + 0.6x_1] \sin(6\pi x_1 + \frac{i\pi}{n}) & j \in J_2 \end{cases}$
UF3	$f1 = x_1 + \frac{2}{ J_1 } (4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \left(\frac{20y_j \pi}{\sqrt{j}} \right) + 2)$ $f2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } (4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos \left(\frac{20y_j \pi}{\sqrt{j}} \right) + 2)$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1, and } y_j = x_j - x_1^{0.5(1.0 + \frac{3j-2}{n-2})}, j = 2, \dots, n$
UF4	$f1 = x_1 + \frac{2}{ J_1 } (\sum_{j \in J_1} h(y_j)), f2 = 1 - x_2 + \frac{2}{ J_2 } (\sum_{j \in J_2} h(y_j))$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1, and } y_i = x_j \left(6\pi x_1 + \frac{i\pi}{n} \right), j = 2, \dots, n$
UF5	$f1 = x_1 + \left(\frac{1}{2n} + \epsilon \right) \sin(2N\pi x_1) + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j),$ $f2 = 1 - x_1 + \left(\frac{1}{2n} + \epsilon \right) \sin(2N\pi x_1) + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $J_1 = \{j j \text{ is odd and } 2 \leq j \leq n\} \text{ and } J_2 = \{j j \text{ is even and } 2 \leq j \leq n\}. N \text{ is an integer, } \epsilon < 0,$ $y_j = x_j - \sin \left(6\pi x_1 + \frac{i\pi}{n} \right), j = 2, \dots, n \text{ and } h(t) = 2t^2 - \cos(4\pi t) + 1$
UF6	$f1 = x_1 + \max \left\{ 0, 2 \left(\frac{1}{2N} + \epsilon \right) \sin(2N\pi x_1) \right\} + \frac{2}{ J_1 } (4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos \left(\frac{20y_j \pi}{\sqrt{j}} \right) + 2)$ $f2 = 1 - x_1 + \max \left\{ 0, 2 \left(\frac{1}{2N} + \epsilon \right) \sin(2N\pi x_1) \right\} + \frac{2}{ J_2 } (4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos \left(\frac{20y_j \pi}{\sqrt{j}} \right) + 2)$ $J_1 = \{j j \text{ is odd and } 2 \leq j \leq n\} \text{ and } J_2 = \{j j \text{ is even and } 2 \leq j \leq n\}. y_j = x_j - \sin \left(6\pi x_1 + \frac{i\pi}{n} \right),$ $j = 2, \dots, n$
UF7	$f1 = \sqrt[3]{x_1} + \frac{2}{\sqrt{J_1}} \sum_{j \in J_1} y_j^2, f2 = \sqrt[3]{x_1} + \frac{2}{\sqrt{J_2}} \sum_{j \in J_2} y_j^2$ $J_1 = \{j j \text{ is odd and } 2 \leq j \leq n\} \text{ and } J_2 = \{j j \text{ is even and } 2 \leq j \leq n\}. y_j = x_j - \sin \left(6\pi x_1 + \frac{i\pi}{n} \right),$ $j = 2, \dots, n$
UF8	$f1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{i\pi}{n}))^2$ $f2 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{i\pi}{n}))^2$ $f3 = \cos(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{i\pi}{n}))^2$ $J_1 = \{j 3 \leq j \leq n, \text{ and } j - 1 \text{ is a multiplication of } 3\},$ $J_2 = \{j 3 \leq j \leq n, \text{ and } j - 2 \text{ is a multiplication of } 3\},$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}.$

Table 1 (continued)

Name	Mathematical formulation
UF9	$f1 = 0.5 \left[\max \left\{ 0, (1 + \epsilon)(1 - 4(2x_1 - 1)^2) \right\} + 2x_1 \right] x_2 + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f2 = 0.5 \left[\max \left\{ 0, (1 + \epsilon)(1 - 4(2x_1 - 1)^2) \right\} + 2x_1 \right] x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f3 = 1 - x_2 + \frac{2}{ J_3 } \sum_{j \in J_3} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ <p> $J_1 = \{j 3 \leq j \leq n, \text{ and } j - 1 \text{ is a multiplication of } 3\}, J_2 = \{j 3 \leq j \leq n, \text{ and } j - 2 \text{ is a multiplication of } 3\},$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}.$ </p>
F10	$f1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} [4y_j^2 - \cos(8\pi y_j) + 1]$ $f1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} [4y_j^2 - \cos(8\pi y_j) + 1]$ $f1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} [4y_j^2 - \cos(8\pi y_j) + 1]$ <p> $J_1 = \{j 3 \leq j \leq n, \text{ and } j - 1 \text{ is a multiplication of } 3\}, J_2 = \{j 3 \leq j \leq n, \text{ and } j - 2 \text{ is a multiplication of } 3\},$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}.$ </p>

$$SP(S) = \sqrt{\frac{1}{|S| - 1} \sum_{i=1}^{|S|} (\bar{d} - d_i)^2} \tag{6}$$

where $|S|$ is the number of solutions in the Pareto optimal set (S), \bar{d} is the mean of all d_s , and.

$(d = \min(|f_1^i(\bar{x}) - f_1^j(\bar{x})| + |f_2^i(\bar{x}) - f_2^j(\bar{x})|)), i, j = 1, 2, \dots, |S|$, and $\bar{x} \in |S|$. The spacing metric considers the distance between a point and its closest neighbor in the Pareto front generated by the optimization algorithm to measure how evenly the estimated solutions are distributed. The value 0 for this metric indicates that the estimated solutions are spaced at equal distances [37].

$$MS = \sqrt{\sum_{k=1}^{|F|} \max(\text{ED}(a_i - b_i))} \tag{7}$$

where ED is the Euclidean distance; a_i and b_i are, respectively, the maximum and minimum values in the i_{th} objective; and $|F|$ is the number of objectives. This metric measures the extent of the points along the Pareto front; it calculates the diversity of estimated points. In other words, it indicates the range of values covered by the solutions. Generally, a wider area including the estimated solutions in the Pareto front is desired, i.e., the larger the value of MS, the better it is. This metric is also known as the ‘‘extent metric’’ in literature.

Table 2 The mathematical representation of different test cases in the ZDT and DTLZ datasets

Name	Mathematical formulation	<i>d</i>
ZDT1	$f_1 = x_1, f_2 = g\left(1 - \sqrt{F_1/g}\right), g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30
ZDT2	$f_1 = x_1, f_2 = g\left(1 - (f_1/g)^2\right), g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30
ZDT3	$f_1 = x_1, f_2 = g\left(1 - \sqrt{f_1/g} - f_1/g \sin(10\pi f_1)\right), g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30
ZDT4	$f_1 = x_1, f_2 = g\left(1 - \sqrt{f_1/g}\right), g = 1 + 10(d-1) + \sum_{i=2}^d (x_i^2 - 10\cos(4\pi x_i))$	10
ZDT6	$f_1 = 1 - \exp(-4x_1)\sin^6(6\pi x_1), f_2 = g\left(1 - (f_1/g)^2\right), -g = 1 + 9\left(\frac{\sum_{i=2}^d x_i}{d-1}\right)$	10
DTLZ1	$f_1(\mathbf{x}) = \frac{1}{2}x_1x_2 \cdots x_{M-1}(1 + g(\mathbf{x}_M)),$ $f_2(\mathbf{x}) = \frac{1}{2}x_1x_2 \cdots (1 - x_{M-1})(1 + g(\mathbf{x}_M)), \dots,$ $f_{M-1}(\mathbf{x}) = \frac{1}{2}x_1(1 - x_2)(1 + g(\mathbf{x}_M)),$ $f_M(\mathbf{x}) = \frac{1}{2}(1 - x_1)(1 + g(\mathbf{x}_M))$ $g(\mathbf{x}_M) = 100\left[\mathbf{x}_M + \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]$	30
DTLZ2	$f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M))\cos\left(\frac{x_1\pi}{2}\right) \cdots \cos\left(\frac{x_{M-2}\pi}{2}\right)\cos\left(\frac{x_{M-1}\pi}{2}\right),$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M))\cos\left(\frac{x_1\pi}{2}\right) \cdots \cos\left(\frac{x_{M-2}\pi}{2}\right)\sin\left(\frac{x_{M-1}\pi}{2}\right),$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M))\cos\left(\frac{x_1\pi}{2}\right) \cdots \sin\left(\frac{x_{M-2}\pi}{2}\right), \dots,$ $f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M))\sin(x_1\pi/2),$ $g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2$	30
DTLZ3	DTLZ2 can be made harder by making a modification to the <i>g</i> function	
	$g(\mathbf{x}_M) = 100\left[\mathbf{x}_M + \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]$	30
DTLZ4	$f_1(x) = (1 + g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2) \cdots \cos(x_{M-2}^\alpha\pi/2)\cos(x_{M-1}^\alpha\pi/2)$ $f_2(x) = (1 + g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2) \cdots \cos(x_{M-2}^\alpha\pi/2)\sin(x_{M-1}^\alpha\pi/2)$ $f_3(x) = (1 + g(\mathbf{x}_M))\cos(x_1^\alpha\pi/2) \cdots \sin(x_{M-2}^\alpha\pi/2)$ \vdots $f_M(x) = (1 + g(\mathbf{x}_M))\sin(x_1^\alpha\pi/2)$ $g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2$	30
DTLZ5	$f_1(x) = (1 + g(\mathbf{x}_M))\cos(\theta_1\pi/2) \cdots \cos(\theta_{M-2}\pi/2)\cos(\theta_{M-1}\pi/2),$ $f_2(x) = (1 + g(\mathbf{x}_M))\cos(\theta_1\pi/2) \cdots \cos(\theta_{M-2}\pi/2)\sin(\theta_{M-1}\pi/2),$ $f_3(x) = (1 + g(\mathbf{x}_M))\cos(\theta_1\pi/2) \cdots \sin(\theta_{M-2}\pi/2)$ \vdots $f_M(x) = (1 + g(\mathbf{x}_M))\sin(\theta_1\pi/2),$ $\theta_i = \frac{\pi}{4(1+g(\mathbf{x}_M))} (1 + 2g(\mathbf{x}_M)x_i), \text{ for } i = 2, 3, \dots, (M-1)$ $g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2,$	30
DTLZ6	DTLZ5 can be made harder by making a modification to the <i>g</i> function	
	$g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} x_i^{0.1}$	30

Table 2 (continued)

Name	Mathematical formulation	<i>d</i>
DTLZ7	$f_1(\mathbf{x}_1) = x_1$ $f_2(\mathbf{x}_2) = x_2$ \vdots $f_{M-1}(\mathbf{x}_{M-1}) = x_{M-1}$ $f_M(x) = (1 + g(\mathbf{x}_M))h(f_1, f_2, \dots, f_{M-1}, g)$ $g(\mathbf{x}_M) = 1 + \frac{9}{ \mathbf{x}_M } \sum_{x_i \in \mathbf{x}_M} x_i$ $h(f_1, f_2, \dots, f_{M-1}, g) = M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right]$	30

Note that in all experiments on the CEC 2009, ZDT, and DTLZ benchmarks, we have utilized 100 solutions and a maximum of 1000 iterations. The main criterion in these benchmarks was the number of iterations. However, we used the Maximum Number of Function Evaluations (MNFE) as the termination criterion rather than relying on a 'Maximum Iteration Number for the CEC 2018 dataset. In CEC 2108 experiments, the number of objectives (*M*) in MaOP1-MaOP10 is set to 3. The number of variables (*N*) in all ten test problems is 20. The population size in each algorithm is pop = 100 × *M* and finally, the total number of function evaluations is set to pop × 500. Besides, the general parameters of the proposed algorithm as well as five other algorithms are listed in Table 3. We have run experiments with different values for the parameters used in MOBRO and concluded that MOBRO achieves its highest performance when using the values shown in Table 3. The parameter values of other algorithms have been taken from their publications. Note that four optimization algorithms (MOBRO, MOGWO, MOPOS, and NSGA-III) have been applied to the CEC 2009, ZDT, and DTLZ datasets, but all algorithms, including MOAVOA and MAOA, have been tested on the CEC 2018 dataset.

3.3 Results

All the experiments have been conducted on MATLAB R2019b, installed on Windows 10, on a computer with a Core i7-7700HQ processor running at 2.80 GHz and 32 GB of RAM. Since heuristic algorithms are inherently stochastic, they may generate relatively diverse results. Therefore, for each dataset, an average of 25 separate runs have been used. As the theoretical evaluation of optimization algorithms is difficult, benchmark datasets are used for this purpose. We calculated the value of three performance metrics explained in Sect. 3.1 and provided them in Tables 4, 6 and 8 separately for different datasets. For the sake of fair comparison, the obtained results are compared with five state-of-the-art algorithms: the non-dominated sorting genetic algorithm (NSGA-III), multi-objective particle swarm optimization (MOPSO), multi-objective gray wolf optimization (MOGWO), multi-objective arithmetic optimization algorithm (MAOA), and the multi-objective version of the artificial vulture optimization algorithm (MOAVOA). Note that the first three

Table 3 The initial values of the parameters in optimization algorithms

Algorithms	Parameter	Value
MOBRO	Maximum Damage	6
	Number of Grid	10
	Archive size	100
	Alpha	0.1
	Beta	1
	Gamma	2
NSGA-III	Mutation rate	$(1 - (\text{iteration} - 1) / (\text{maximum number of iteration} - 1))$
	Crossover Percentage	0.5
	Number of crossovers	$2 \times \text{round}(\text{Crossover Percentage} \times \text{Population size} / 2)$
	Mutation Percentage	0.5
	Number of Mutation	$\text{round}(\text{Mutation Percentage} \times \text{Population size})$
MOPSO	Mutation Rate	0.02
	Selection mechanism	Roulette wheel
	C_1	1
	C_2	2
	Inertia weights range	[0.1, 0.5]
	acceleration coefficients	2 and 2
	Number of Grid	10
	Archive size	100
	Alpha	0.1
MOGWO	Beta	2
	Gamma	2
	Mutation rate	0.1
	Number of Grid	10
	Archive size	100
MAOA	Alpha	0.1
	Beta	4
	Gamma	2
	\bar{a}	linearly decreased from 2 to 0 over the iterations
MOAVOA	Archive size	100
	Number of adaptive Grid	30
	Beta	4
	Gamma	2

algorithms have been applied to three datasets (CEC 2009, ZDT, and DTLZ), but all five algorithms have been tested on the CEC 2018 dataset. As the number of runs for obtaining these results is 25, the best, worst, average, and standard deviations of

these runs are also provided in these tables, i.e., according to, for example, IGD, the average case shows the average IGD value across a set of runs of each algorithm on the problem. The best case shows the lowest IGD value achieved by any run of the algorithm on the problem, and the worst case shows the highest IGD value achieved by any run of the algorithm on the problem. The SD shows the standard deviation of the IGD values across the set of runs for each algorithm. The rank of each optimization algorithm among others is also provided in Tables 5, 7 and 9.

Based on the results in Table 4, UF1 function, the MOBRO algorithm achieves competitive results with state-of-the-art. According to the obtained results on the CEC 2009 dataset, MOBRO has the lowest average IGD value among the four algorithms, followed by NSGA-III, MOGWO, and MOPSO. MOBRO also has the lowest best and worst IGD values and the lowest standard deviation. This suggests that MOBRO may be the most consistent and efficient algorithm among the four in terms of achieving good solutions. All four algorithms have small and consistent average values for the SP indicator, with an average value of around 0.02 and a standard deviation of around 0.03. A notable aspect of Table 4 is the best values for each algorithm, which are all very small. The best value for MOBRO is 0, indicating that the solutions in the Pareto front approximation are evenly spaced and perfectly distributed. This suggests that the algorithms can find a set of solutions that are very diverse and evenly distributed, covering a wide range of the objective space. Also, NSGA-III outperforms others in the worst case; however, other algorithms provide competitive results. The average values for the spread metric for all four algorithms are close to 1, with a standard deviation of around 0.01. This indicates that the algorithms have similar performance in terms of the average spread or dispersion of the solutions. The best values for the spread metric obtained by each algorithm are very close to 1, while MOGWO achieves the best value of 1.115994. Also, MOBRO provides the best value in terms of the worst case. Figure 2 (UF1) shows that MOBRO provides an accurate approximation of the true Pareto Optimal Front with the highest diversity as it is evenly spaced and distributed over the true Pareto Front. For the remaining functions (U2-U10), according to the results given in Table 4, all the algorithms provide satisfactory results. Nonetheless, as shown in Fig. 2 (UF2-UF10), MOBRO provides a more accurate approximation of the true Pareto optimal front with the highest diversity where its solutions are equally spaced and dispersed across the true Pareto front.

By inspecting the obtained results on the CEC 2009 dataset, which are summarized in Table 5, the MOBRO algorithm ranks first according to the average, best, worst, and SD values of the IGD metric. Meanwhile, MOBRO ranks third, first, third, and third in average, best, worst, and SD, respectively, in terms of the spacing metric. Also, MOBRO ranks first, second, first, and fourth in average, best, worst, and SD, respectively, in terms of the spread metric. Finally, MOBRO ranks first alongside NSGA-III over the average on all metrics' ranks.

Table 6 provides the results obtained by the four different multi-objective optimization algorithms on the ZDT benchmark functions, while the ranks of the algorithms are summarized in Table 7. On the ZDT1 problem, in terms of average IGD, MOPSO has the lowest value, followed by MOBRO, NSGA-III, and MOGWO. Also, the best IGD values show a similar trend. The worst IGD values are somewhat lower for MOPSO and MOBRO compared to NSGA-III and MOGWO, but the

Table 4 Obtained results for the CEC 2009 dataset by MOBRO and three state-of-the-art optimization algorithms (UF)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
UF1	IGD	Average	1.058879e-01	1.190825e-01	1.294765e-01	1.126504e-01
		Best	8.041294e-02	7.938359e-02	1.013040e-01	7.428851e-02
		Worst	1.282778e-01	1.850927e-01	2.419845e-01	1.533307e-01
		SD	1.267987e-02	2.477559e-02	3.158483e-02	1.285874e-02
	Spacing	Average	2.074686e-02	3.228779e-02	2.343677e-02	2.042699e-02
		Best	0	1.310436e-05	9.850322e-03	6.170633e-04
		Worst	1.188528e-01	7.356352e-02	1.189510e-01	1.505613e-01
		SD	3.170980e-02	2.668677e-02	2.330571e-02	3.567157e-02
	Spread	Average	1.008284	1.001394	9.972661e-01	1.006243
		Best	1.039642	1.010989	1.000815	1.115994
		Worst	1	9.999997e-01	9.919703e-01	9.940015e-01
		SD	1.418461e-02	2.273553e-03	2.624179e-03	2.357498e-02
UF2	IGD	Average	3.807625e-02	6.477508e-02	1.439172e-01	6.663130e-02
		Best	3.074160e-02	3.202552e-02	1.072780e-01	5.428564e-02
		Worst	5.506947e-02	1.138059e-01	1.772986e-01	8.090885e-02
		SD	5.846266e-03	2.430804e-02	2.040275e-02	6.814360e-03
	Spacing	Average	2.360691e-02	2.168966e-02	2.093645e-02	1.619863e-02
		Best	8.020962e-03	6.356538e-03	1.306679e-02	9.418330e-03
		Worst	5.183141e-02	5.369421e-02	3.194228e-02	2.637573e-02
		SD	1.033737e-02	1.413747e-02	4.662564e-03	5.438637e-03
	Spread	Average	1.046580	1.001036	9.924647e-01	9.966049e-01
		Best	1.011413	9.884833e-01	9.746364e-01	9.595267e-01
		Worst	2.952298e-02	6.554787e-03	4.671885e-03	1.373376e-02
		SD	1.128760	1.018981	9.979203e-01	1.017377
UF3	IGD	Average	2.702032e-01	3.094566e-01	5.515161e-01	2.833685e-01
		Best	1.288660e-01	2.137218e-01	4.304273e-01	1.463563e-01
		Worst	3.903488e-01	3.318596e-01	7.062751e-01	3.834633e-01
		SD	8.585441e-02	2.638603e-02	6.612060e-02	5.786431e-02
	Spacing	Average	3.787285e-03	2.118774e-03	3.002797e-02	6.575234e-02
		Best	0	3.028966e-06	7.993507e-03	1.619912e-02
		Worst	5.311535e-02	4.409929e-02	6.207261e-02	1.207440e-01
		SD	1.314808e-02	8.811720e-03	1.439819e-02	2.785524e-02
	Spread	Average	1.001759	1.000057	9.983732e-01	1.019559
		Best	1.040307	1.001287	1.000034	1.055983
		Worst	1	9.999996e-01	9.903914e-01	9.908404e-01
		SD	8.064068e-03	2.570035e-04	2.274009e-03	1.691203e-02

Table 4 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
UF4	IGD	Average	6.236254e-02	6.384848e-02	7.870343e-02	6.399682e-02
		Best	5.821814e-02	5.462501e-02	7.228058e-02	4.921678e-02
		Worst	7.482343e-02	8.077696e-02	8.711760e-02	8.473269e-02
		SD	4.843749e-03	6.811379e-03	4.090672e-03	9.822848e-03
	Spacing	Average	5.817968e-02	1.805360e-02	1.024719e-02	1.223679e-02
		Best	2.291433e-02	5.290659e-10	8.610813e-03	7.121168e-03
		Worst	1.352620e-01	3.366050e-02	1.274377e-02	1.928826e-02
		SD	2.425651e-02	9.763574e-03	9.525431e-04	3.764147e-03
	Spread	Average	1.053060	1.004539	9.958688e-01	9.973942e-01
		Best	1.217261	1.016587	9.983493e-01	1.002877
		Worst	9.812560e-01	1	9.929963e-01	9.931065e-01
		SD	4.914117e-02	3.795481e-03	1.244437e-03	2.694213e-03
UF5	IGD	Average	5.148433e-01	2.600057e-01	1.083485	1.032972
		Best	2.921407e-01	1.631642e-01	5.967223e-01	5.460534e-01
		Worst	7.071068e-01	4.337244e-01	2.468638	2.337737
		SD	1.160190e-01	6.335942e-02	5.574357e-01	5.038715e-01
	Spacing	Average	0	1.970091e-02	6.568606e-02	2.609233e-01
		Best	0	4.525583e-03	1.206815e-06	1.354659e-05
		Worst	0	3.460013e-02	3.346846e-01	1.011886
		SD	0	8.519111e-03	1.019496e-01	2.561727e-01
	Spread	Average	1	1.038523	9.931861e-01	1.016944
		Best	1	1.077282	1.011658	1.082921
		Worst	1	1.007663	9.335478e-01	9.691344e-01
		SD	0	1.928604e-02	1.731291e-02	2.593242e-02
UF6	IGD	Average	3.545686e-01	2.988375e-01	6.138773e-01	3.729353e-01
		Best	1.168526e-01	1.697471e-01	3.966720e-01	3.153002e-01
		Worst	6.449754e-01	4.979779e-01	1.002562	5.675313e-01
		SD	1.397700e-01	7.700448e-02	1.695441e-01	5.044658e-02
	Spacing	Average	2.630257e-02	2.115380e-02	5.813419e-02	2.635353e-02
		Best	0	4.703457e-03	0	2.026554e-03
		Worst	6.575644e-01	4.285737e-02	1.185145	1.269807e-01
		SD	1.315129e-01	1.132747e-02	2.351311e-01	3.097434e-02
	Spread	Average	1.002551	1.000649	1.000129	1.000200
		Best	1.063784	1.001707	1.005313	1.002104
		Worst	1	1.000166	9.994161e-01	9.998040e-01
		SD	1.275679e-02	3.925596e-04	1.089432e-03	4.606180e-04

Table 4 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
UF7	IGD	Average	1.935032e-01	2.840046e-01	1.345812e-01	8.829407e-02
		Best	4.491640e-02	2.570611e-02	5.881359e-02	6.221430e-02
		Worst	4.106740e-01	6.183849e-01	3.657388e-01	3.510646e-01
		SD	1.483561e-01	1.925225e-01	7.675563e-02	5.842972e-02
	Spacing	Average	2.124306e-02	1.516826e-02	1.929874e-02	1.535620e-02
		Best	0	7.283484e-07	8.235993e-04	4.478326e-03
		Worst	5.136007e-02	6.477584e-02	3.616026e-02	4.741173e-02
		SD	1.811356e-02	1.502914e-02	8.743771e-03	9.870987e-03
	Spread	Average	1.222007	1.003851	9.927430e-01	9.926709e-01
		Best	1.493219	1.045802	9.999951e-01	1.004355
		Worst	1	9.688524e-01	9.753168e-01	9.669770e-01
		SD	2.204172e-01	1.709751e-02	5.128329e-03	9.361485e-03
UF8	IGD	Average	1.847130e-01	4.305215e-01	7.967083e-01	1.852416
		Best	1.551407e-01	2.530296e-01	4.808991e-01	2.596214e-01
		Worst	2.020404e-01	4.420496e-01	1.281790	3.293377
		SD	7.980226e-03	3.748258e-02	1.561078e-01	1.002490
	Spacing	Average	5.931855e-02	1.325272e-02	3.049455e-01	2.412446e-02
		Best	3.758080e-02	9.009516e-03	1.961534e-01	8.258467e-03
		Worst	8.260748e-02	4.334506e-02	4.403492e-01	5.322328e-02
		SD	1.024589e-02	6.584966e-03	7.373366e-02	1.150028e-02
	Spread	Average	9.798638e-01	9.993291e-01	9.984394e-01	9.999478e-01
		Best	1.006017	9.998775e-01	9.992794e-01	1.000015
		Worst	9.568720e-01	9.976321e-01	9.973884e-01	9.996560e-01
		SD	1.349235e-02	4.766431e-04	4.383336e-04	9.325745e-05
UF9	IGD	Average	1.870647e-01	2.925822e-01	1.094135	2.411257e-01
		Best	9.536503e-02	1.198416e-01	7.240413e-01	1.141366e-01
		Worst	2.978038e-01	5.178909e-01	1.469235	7.388227e-01
		SD	6.913882e-02	7.032754e-02	2.019914e-01	1.631921e-01
	Spacing	Average	7.537386e-02	1.697862e-02	3.824521e-01	3.191370e-02
		Best	4.113810e-02	7.449715e-03	1.718706e-01	4.964488e-03
		Worst	1.447432e-01	4.130173e-02	6.070866e-01	8.786275e-02
		SD	2.689333e-02	8.102589e-03	1.175569e-01	1.715532e-02
	Spread	Average	1.008118	9.989047e-01	9.985149e-01	9.999034e-01
		Best	1.027527	1.000104	9.991865e-01	1.000171
		Worst	1.002171	9.973850e-01	9.972399e-01	9.995532e-01
		SD	5.619450e-03	7.908437e-04	4.504668e-04	1.565576e-04

Table 4 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
UF10	IGD	Average	6.437072e-01	5.291184e-01	5.747524	1.720505
		Best	2.192057e-01	3.188761e-01	3.451154	4.613242e-01
		Worst	1.043141	7.689212e-01	7.860392	3.375046
		SD	2.466995e-01	9.744921e-02	1.013933	8.853796e-01
	Spacing	Average	1.985429e-02	2.968321e-02	1.338813	1.656022e-01
		Best	0	2.878160e-05	8.381262e-01	0
		Worst	3.347792e-01	1.191497e-01	2.938443	4.927139e-01
		SD	6.845717e-02	3.284048e-02	4.239808e-01	1.469929e-01
	Spread	Average	9.985482e-01	1.000048	9.993924e-01	9.997736e-01
		Best	1.002403	1.000217	9.997581e-01	1.000097
		Worst	9.756552e-01	9.999930e-01	9.991681e-01	9.991277e-01
		SD	5.657893e-03	5.875331e-05	1.530178e-04	2.191258e-04

difference is not as large as it is for the average and best values. This suggests that the worst solutions found by MOPSO and MOBRO were not as far from the true Pareto front as the worst solutions found by NSGA-III and MOGWO are. Finally, the standard deviation values show that the IGD values for MOPSO and BRO had smaller variations across the set of runs compared to NSGA-III and MOGWO. This indicates that the performance of MOPSO and MOBRO was more consistent across the runs, while the performance of NSGA-III and MOGWO was more variable. Based on the average spacing values provided in the table, MOBRO has the lowest average spacing value, followed by MOPSO, NSGA-III, and MOGWO. This ranking means that MOBRO outperforms the other algorithms in terms of producing a more focused set of solutions on the Pareto front, followed by MOPSO, NSGA-III, and MOGWO. Based on the best and worst spacing values, MOBRO has the narrowest range of spacing values, followed by MOPSO, NSGA-III, and MOGWO. This could indicate that MOBRO produces a more focused set of solutions on the Pareto front, while the other algorithms produce a more diverse set of solutions. The standard deviation of the spacing values also suggests that MOBRO has the most consistent set of spacing values, followed by MOPSO, NSGA-III, and MOGWO. This could further support the idea that MOBRO produces a more focused set of solutions.

Based on the values given in Table 6, MOBRO outperforms the other algorithms in terms of the spread metric for the ZDT1 optimization problem, with the greatest average, best, and worst values. However, MOPSO, although ranked 4th in the spread metric, has the lowest standard deviation, indicating that it has a more consistent set of values and potentially a smaller range of solutions. As mentioned already, due to the nature of the original GWO algorithm, in some cases MOGWO suffers from a shortage of candidate leaders in the Pareto front as it requires three leaders to find the optimal solution. For example, MOGWO is unable to provide any

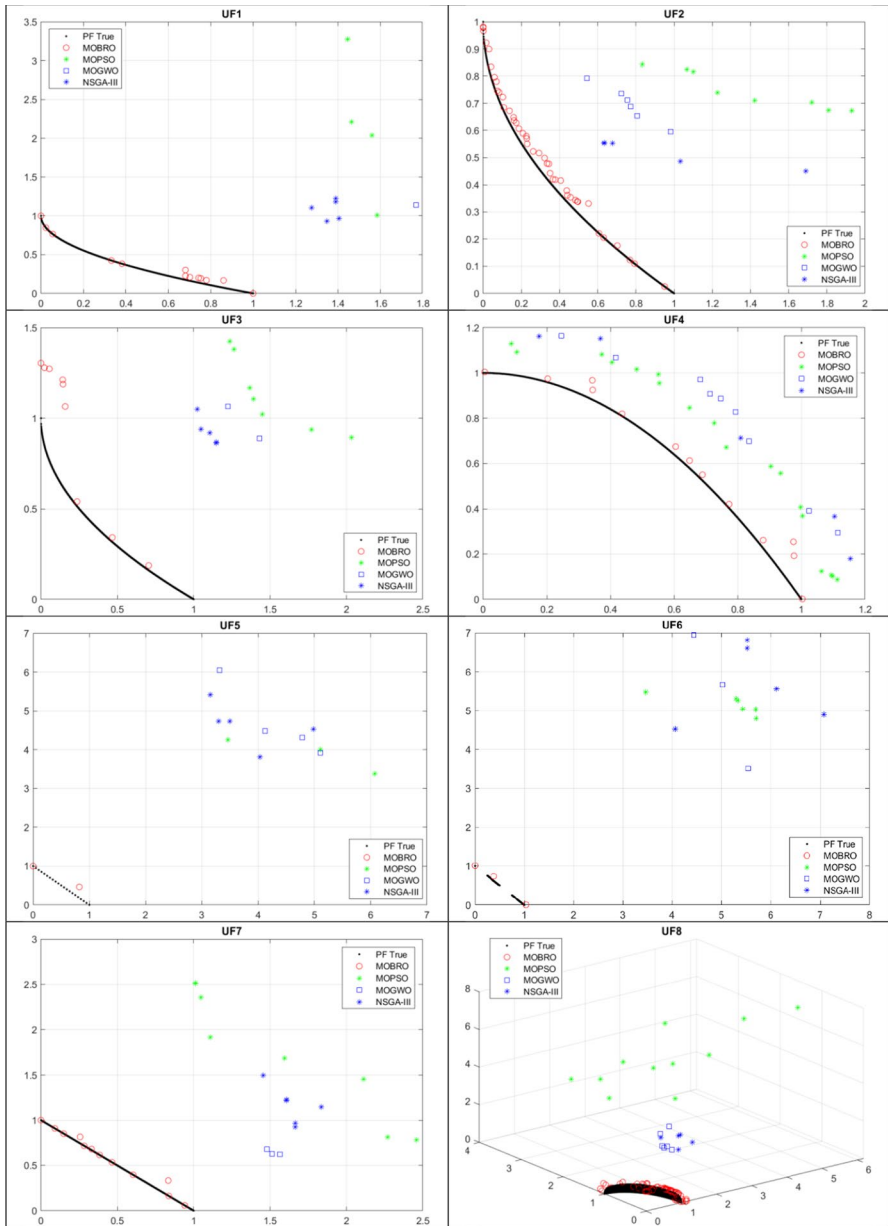


Fig. 2 The Pareto fronts obtained by the applied algorithms on the all test problems

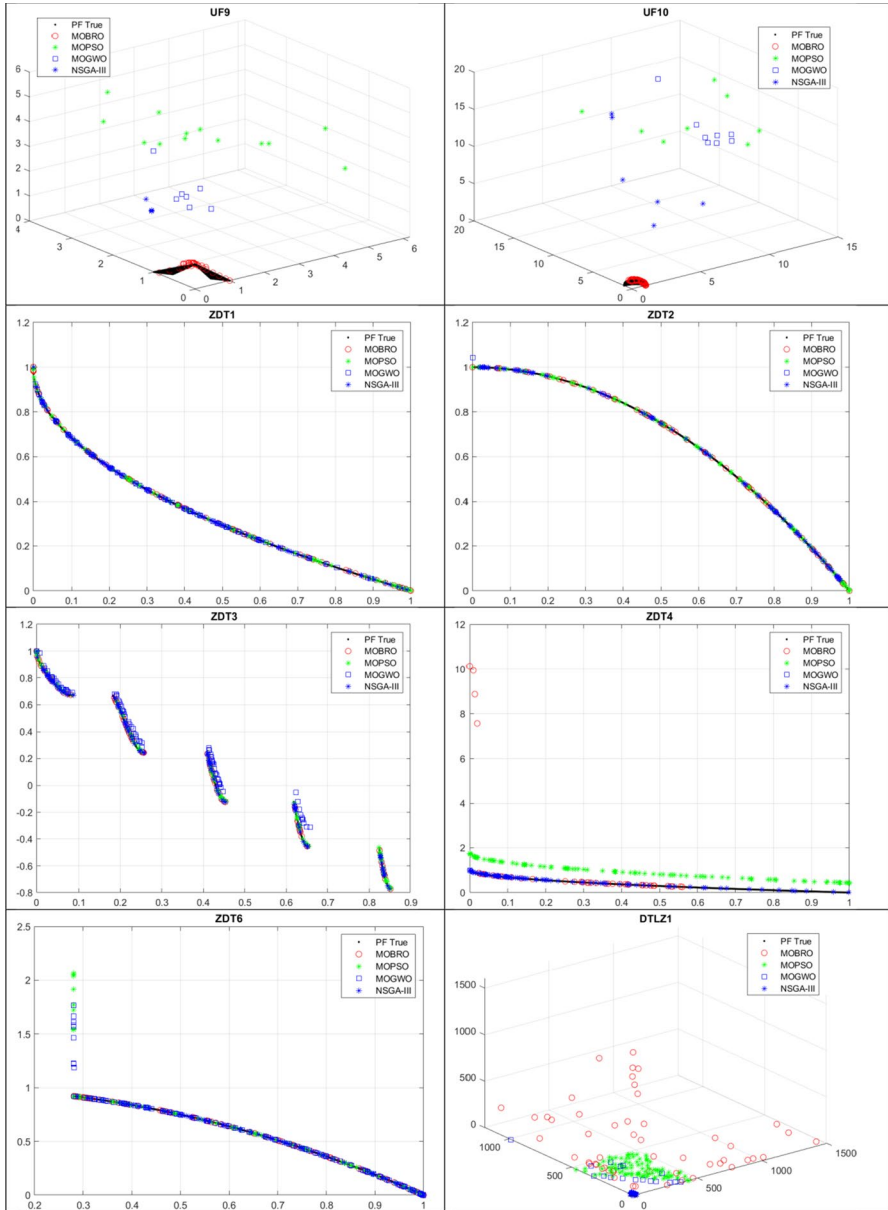


Fig. 2 (continued)

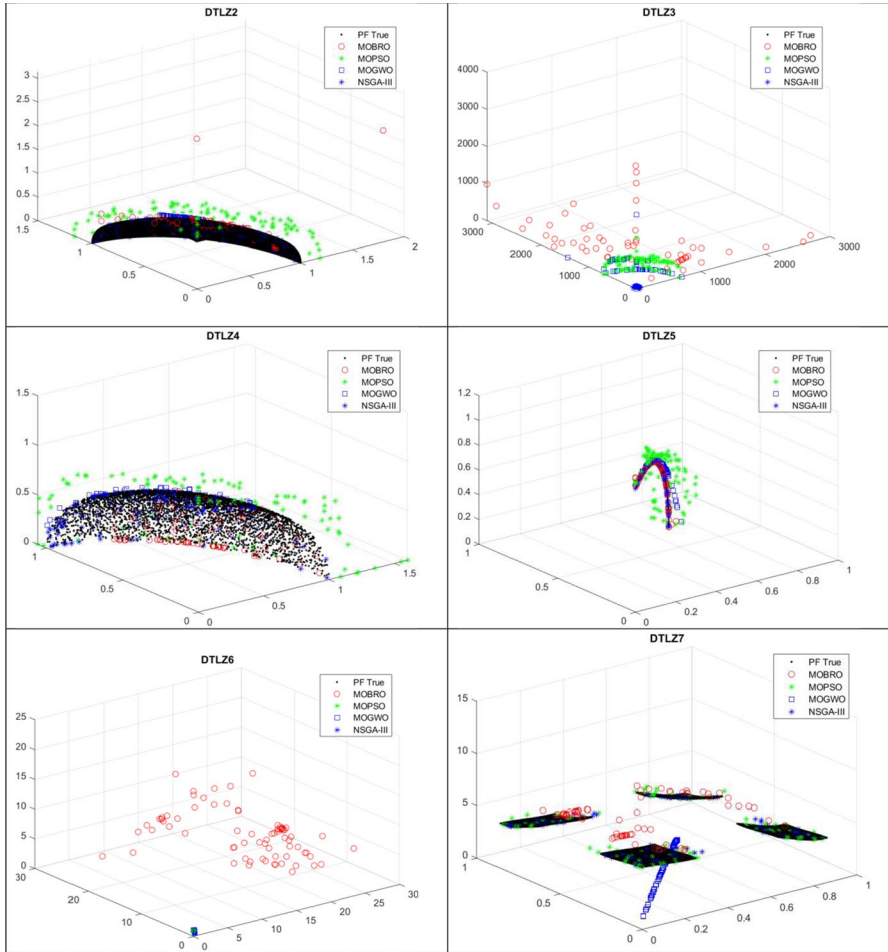


Fig. 2 (continued)

solution for the ZDT2 and ZDT3 test suits as it could not provide more than one or two candidate leaders in the Pareto Front. Moreover, it can be seen in Table 5 that the MOBRO algorithm outperforms its competitors in terms of spacing and spread indicators and ranks first according to average, best, and worst values. Considering all ZDT functions, MOBRO ranks second, third, second, and second in average, best, worst, and SD, respectively, in terms of the IGD metric. Finally, by taking the average over all metrics ranks on all ZDT datasets, MOBRO ranks first, followed by MOPSO, NSGA-III, and MOGWO. Figure 2 for ZDT1-ZDT6 shows that in most cases, the algorithms can find a very accurate approximation of the true Pareto optimal solutions with a good diversity that is evenly spaced and perfectly distributed. However, MOGWO is unable to provide solutions for ZDT2 and ZDT3.

Table 5 The ranks of each algorithm among others according to CEC 2009 dataset

Metrics	MOBRO		NSGA-III		MOPSO		MOGWO		
	Average	Overall	Average	Overall	Average	Overall	Average	Overall	
IGD	Average	1,50	1	2,20	2	3,70	4	2,60	3
	Best	1,60	1	2,00	2	3,90	4	2,50	3
	Worst	1,80	1	2,00	2	3,70	4	2,50	3
	SD	2,00	1	2,30	2	3,20	4	2,50	3
Spacing	Average	2,60	3	1,90	1	3,10	4	2,40	2
	Best	1,80	1	2,30	2	3,20	4	2,50	3
	Worst	2,60	2	1,90	1	2,90	4	2,60	2
	SD	2,70	3	1,90	1	2,60	2	2,80	4
Spread	Average	2,00	1	2,10	2	3,60	4	2,30	3
	Best	1,50	1	2,60	3	3,60	4	2,30	2
	Worst	3,40	4	1,60	1	2,90	3	2,10	2
	SD	3,50	4	2,20	2	1,80	1	2,50	3
Average	1.92	1.75	3.50	2.75					
Rank	2	1	4	3					

Tables 8 and 9, respectively, represent the obtained results and the rankings of the four algorithms on the DTLZ dataset. Based on the average IGD value for the DTLZ1, NSGA-III ranks first by a large margin, and the following algorithms are MOGWO, MOBRO, and MOPSO, respectively, from second to last. Moreover, based on the spacing metric, the NSGA-III algorithm again ranks first in all cases (best, worst, average, and SD) on DTLZ1. However, according to Spread, the MOBRO algorithm ranks first in “average”, “best,” and “worst,” while MOPSO ranks first in “SD”. According to the obtained results, the DTLZ dataset appears to be more challenging than the other two datasets. As is shown in Table 8, MOGWO could not provide any solutions for the DTLZ7 test function in some runs of the algorithm. However, the best Pareto Front of all success runs for MOGWO have been plotted in Fig. 2 (DTLZ1).

In a nutshell, according to all DTLZ functions, MOBRO ranks second, third, fourth, and third in average, best, worst, and SD, respectively, in terms of the IGD metric. Also, it ranks fourth, third, fourth, and fourth in average, best, worst, and SD, correspondingly, in terms of the spacing metric. According to the Spread metric, MOBRO ranks first in average, best, worst and ranks second in SD. Finally, by taking the average over all metrics’ ranks, the overall ranking of the four algorithms is NSGA-III, MOPSO, MOBRO, and MOGWO from best to worst. These results show that the distance between MOBRO’s solutions for DTLZ and the true Pareto Front is not as close as others. Also, a higher spacing value for MOBRO indicates that the solutions are more evenly distributed within the set, while a lower spacing value for the competitors suggests that the solutions are more densely clustered. The spacing value depends on the optimization problem’s context and goals. In some

Table 6 Obtained results for the ZDT dataset by MOBRO and three state-of-the-art optimization algorithms (ZDT)

Indicators		MOBRO	NSGA-III	MOPSO	MOGWO	
ZDT1	IGD	Average	1.327705e-02	1.474600e-02	9.336466e-03	2.995821e-02
		Best	1.056102e-02	6.995229e-03	7.507571e-03	1.542783e-02
		Worst	1.960878e-02	5.095734e-02	1.186242e-02	1.089518e-01
		SD	2.261685e-03	8.693213e-03	1.180501e-03	1.877269e-02
	Spacing	Average	1.151922e-02	1.398349e-02	1.153518e-02	1.578788e-02
		Best	7.405108e-03	8.326846e-03	9.803473e-03	8.971108e-03
		Worst	1.352383e-02	2.033908e-02	1.406406e-02	2.976003e-02
		SD	1.753412e-03	3.300849e-03	1.204771e-03	5.668460e-03
	Spread	Average	1.026903	9.921211e-01	9.850165e-01	1.009012
		Best	1.090888	1.003418	9.991394e-01	1.064891
		Worst	1.003829	9.286323e-01	9.310711e-01	1.000285
		SD	2.134775e-02	1.442679e-02	1.430293e-02	1.451846e-02
ZDT2	IGD	Average	1.570948e-01	3.860709e-01	8.954065e-03	–
		Best	1.152778e-02	1.436772e-02	7.864588e-03	–
		Worst	6.102383e-01	8.168218e-01	1.061456e-02	–
		SD	2.599017e-01	1.773057e-01	6.711798e-04	–
	Spacing	Average	9.300668e-03	8.917949e-03	1.146621e-02	–
		Best	0	1.927686e-05	9.445561e-03	–
		Worst	1.500097e-02	3.144561e-02	1.436123e-02	–
		SD	5.512555e-03	7.124149e-03	1.159716e-03	–
	Spread	Average	1.042487	1.063120	9.585570e-01	–
		Best	1.143705	1.467445	1.015588	–
		Worst	1	9.956852e-01	8.600362e-01	–
		SD	3.561635e-02	1.352227e-01	3.795278e-02	–
ZDT3	IGD	Average	1.237738e-02	1.272082e-02	9.248437e-03	–
		Best	9.626243e-03	8.998238e-03	7.864033e-03	–
		Worst	1.624695e-02	1.726821e-02	1.106055e-02	–
		SD	1.761015e-03	2.544873e-03	8.863464e-04	–
	Spacing	Average	1.238916e-02	1.523259e-02	1.074315e-02	–
		Best	1.001622e-02	7.877873e-03	8.045306e-03	–
		Worst	1.444268e-02	2.306351e-02	1.415532e-02	–
		SD	1.246359e-03	4.309461e-03	1.522289e-03	–
	Spread	Average	1.020829	9.987427e-01	9.174001e-01	–
		Best	1.071257	1.007409	9.749227e-01	–
		Worst	1.001514	9.843139e-01	7.633056e-01	–
		SD	1.643803e-02	5.400559e-03	4.021701e-02	–

Table 6 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
ZDT4	IGD	Average	2.260567	2.848878e-01	4.069619	3.248467
		Best	1.220899e-01	1.252948e-02	3.907178e-01	8.392923e-01
		Worst	2.232986e+01	5.955614e-01	1.414372e+01	1.098995e+01
		SD	4.802797	1.893033e-01	3.420569	2.220592
	Spacing	Average	5.743109e-04	1.563621e-02	8.300782e-02	2.655381e-01
		Best	0	3.105363e-03	1.097073e-02	0
		Worst	8.097027e-03	4.703317e-02	1.624314	4.504345
		SD	1.573267e-03	9.827947e-03	3.211379e-01	9.382259e-01
	Spread	Average	1.010865	1.002466	9.997825e-01	1.000196
		Best	1.079330	1.043152	1.000206	1.004563
		Worst	9.895047e-01	9.885998e-01	9.988924e-01	1
		SD	1.989874e-02	1.009507e-02	2.567855e-04	9.105199e-04
ZDT6	IGD	Average	1.388264e-02	1.084263e-01	7.622778e-03	2.020376e-02
		Best	7.282946e-03	6.998088e-03	6.120972e-03	5.691615e-03
		Worst	2.785515e-02	5.210060e-01	9.095217e-03	4.747065e-02
		SD	5.033109e-03	1.589206e-01	7.734070e-04	1.488173e-02
	Spacing	Average	1.383537e-02	2.868863e-02	1.518930e-02	2.295072e-02
		Best	7.949431e-03	1.599477e-09	8.154275e-03	4.925987e-03
		Worst	4.592198e-02	1.313652e-01	5.116604e-02	7.237843e-02
		SD	7.545012e-03	2.922671e-02	1.132442e-02	2.325584e-02
	Spread	Average	1.220917	1.326404	9.763627e-01	1.041479
		Best	1.533949	1.882441	1.177151	1.339947
		Worst	1.018183	9.739449e-01	8.248366e-01	8.424521e-01
		SD	1.348425e-01	3.173127e-01	8.605875e-02	1.207655e-01

cases, a diverse or evenly distributed set of solutions is desired, while in others, a tightly clustered set is more appropriate.

Also, MOBRO ranks first based on the Spread metric in all the datasets. It proves that the obtained results by MOBRO can cover the true Pareto Front through hyper cubes better than its competitors.

Unlike other datasets (CEC 2009 ZDT and DTLZ), we used the number of function evaluation criterion instead of the number of iterations for measuring the performance of MOBRO on the CEC 2018 dataset. Tables 10 and 11, respectively, provide the obtained results for four different algorithms according to performance metrics and the rank of each algorithm, among others. The reason for not listing MOGWO in these tables is that MOGWO is unable to find an optimal solution for any of the datasets in CEC2018, as it requires three solutions in the

Table 7 The ranks of each algorithm among others according to the ZDT dataset

Indicators		MOBRO		NSGA-III		MOPSO		MOGWO	
		Average	Overall	Average	Overall	Average	Overall	Average	Overall
IGD	Average	2.00	2	2.80	3	1.60	1	3.60	4
	Best	2.80	3	2.00	2	1.80	1	3.40	4
	Worst	2.40	2	2.80	3	1.40	1	3.40	4
	SD	2.60	2	2.60	2	1.40	1	3.40	4
Spacing	Average	1.40	1	2.60	3	2.20	2	3.80	4
	Best	1.80	1	1.80	1	3.40	4	2.80	3
	Worst	1.40	1	3.00	3	1.80	2	3.80	4
	SD	1.40	1	3.00	3	1.80	2	3.80	4
Spread	Average	1.40	1	1.80	2	3.60	4	3.20	3
	Best	1.40	1	1.80	2	3.60	4	3.20	3
	Worst	1.60	1	2.40	2	2.60	3	3.40	4
	SD	2.80	3	2.60	2	1.60	1	3.00	4
Average	1.58	2.33	2.17	3.75					
Rank	1	3	2	4					

Pareto Front, and when it cannot find these three, a runtime error occurs in this algorithm.

According to Table 10 and consequently Table 11, NSGA-III ranks first and MOBRO ranks second in terms of the IGD measure, followed by MOPSO, MOAVOA, and MAOA ranking from third to fifth. This superiority suggests that MOBRO and NSGA-III offer a better balance between convergence and diversity. This balance can be seen in their ability to approximate the true Pareto Front effectively. When looking at the Spacing measure, again, NSGA-III and MOBRO respectively rank the first and second, but this time, MOBRO obtains very competitive results to NSGA-III. The MOPSO, MAOA, and MOAVOA respectively rank from third to fifth. These ranks indicate that the estimated solutions by MOBRO and NSGA-III are spaced at almost equal distances. In terms of the spread, MOBRO surpasses all other algorithms and ranks first. The second to fifth ranks are respectively achieved by NSGA-III, MOAVOA, MAOA, and MOPSO. Obtained results on this measure suggest that MOBRO and NSGA-III spread the obtained solutions with a higher distance over the Pareto Front. Note that for the first two factors, IGD and SP, lower values but for the third one, spread, higher values are desired. In summary, MOBRO and NSGA-III emerged as prominent candidates, providing a high solution quality and diversity, which can be chosen as trustable algorithms for solving multi-objective optimization problems in a specific domain.

We also measured the runtime for each algorithm when applied to the CEC 2018 dataset, provided in Table 12. According to this table, the fastest algorithm among the five is MAOA, while the slowest one is NSGA-III. This low speed can be assumed to be a disadvantage for the NSGA-III algorithm, despite its success in finding optimal solutions for many multi-objective problems. Finally, the Wilcoxon

Table 8 Obtained results for MOBRO as well as state-of-the-art optimization algorithms (DTLZ)

Indicators		MOBRO	NSGA-III	MOPSO	MOGWO	
DTLZ1	IGD	Average	2.178377e+02	1.989607e+01	2.248275e+02	1.482031e+02
		Best	8.867041e+01	1.448727e+01	1.943567e+02	1.155510
		Worst	3.397415e+02	2.745007e+01	2.801124e+02	2.520651e+02
	Spacing	SD	6.530199e+01	6.170282	1.897980e+01	9.929580e+01
		Average	8.573991e+01	1.838074	1.991931e+01	4.154402e+01
		Best	5.531672e+01	1.199141	1.399390e+01	1.827708
	Spread	Worst	1.255793e+02	2.593933	2.562060e+01	1.874294e+02
		SD	1.829162e+01	6.068766e-01	3.390659	4.456018e+01
		Average	1.002699	9.984609e-01	9.968741e-01	1.002316
		Best	1.006805	9.988149e-01	9.977757e-01	1.025457
		Worst	1.000580	9.982124e-01	9.959078e-01	9.947683e-01
		SD	1.398521e-03	2.874206e-04	3.938505e-04	7.565210e-03
DTLZ2	IGD	Average	1.005500e-01	6.476109e-02	2.073697e-01	5.713183e-01
		Best	7.937740e-02	6.177356e-02	1.826352e-01	5.106208e-01
		Worst	1.477560e-01	6.780235e-02	2.311046e-01	6.389609e-01
	Spacing	SD	1.537643e-02	2.479877e-03	1.454946e-02	4.417254e-02
		Average	1.789629e-01	5.826382e-02	6.913781e-02	6.475407e-03
		Best	7.584949e-02	4.402854e-02	5.567645e-02	2.509956e-03
	Spread	Worst	3.352709e-01	6.712978e-02	8.871255e-02	1.725288e-02
		SD	8.273272e-02	1.027538e-02	8.273844e-03	3.319916e-03
		Average	9.999725e-01	9.986632e-01	9.991850e-01	9.999950e-01
		Best	1.000339	9.994767e-01	9.995129e-01	1.000001
		Worst	9.988879e-01	9.980401e-01	9.987365e-01	9.999899e-01
		SD	2.426292e-04	6.292910e-04	2.000685e-04	2.378849e-06
DTLZ3	IGD	Average	4.921215e+02	5.898786e+01	6.829964e+02	5.060796e+02
		Best	2.981213e+02	5.102605e+01	6.285669e+02	1.112847
		Worst	7.401399e+02	6.763263e+01	7.239966e+02	7.005583e+02
	Spacing	SD	1.138689e+02	6.798626	3.128210e+01	3.023669e+02
		Average	1.970807e+02	8.596885	4.824420e+01	7.813303e+01
		Best	1.003343e+02	5.025303	3.594191e+01	1.984952
	Spread	Worst	3.488462e+02	1.910082e+01	9.894461e+01	2.915591e+02
		SD	5.302519e+01	7.003038	1.660402e+01	7.666201e+01
		Average	1.001624	9.995634e-01	9.988652e-01	1.005875
		Best	1.002761	1.000111	9.995921e-01	1.122472
		Worst	1.000756	9.992329e-01	9.986239e-01	9.919322e-01
		SD	5.817455e-04	3.985468e-04	2.436799e-04	2.567592e-02

Table 8 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
DTLZ4	IGD	Average	3.335352e-01	6.311718e-01	2.346459e-01	1.743404e-01
		Best	1.226681e-01	1.031362e-01	1.557574e-01	8.031445e-02
		Worst	5.565818e-01	9.311345e-01	3.636732e-01	5.611295e-01
		SD	7.217549e-02	3.932534e-01	5.555505e-02	1.389910e-01
	Spacing	Average	3.642655e-02	2.314514e-02	8.112420e-02	5.141468e-02
		Best	1.243270e-02	1.219790e-11	5.968847e-02	2.073659e-02
		Worst	8.447894e-02	8.155942e-02	1.221101e-01	8.311583e-02
		SD	1.769809e-02	3.928789e-02	1.417447e-02	1.355924e-02
	Spread	Average	9.996940e-01	9.988649e-01	9.958179e-01	9.947957e-01
		Best	1.000431	1.000000	9.981170e-01	1.000133
		Worst	9.948489e-01	9.954818e-01	9.938403e-01	9.870560e-01
		SD	1.297369e-03	2.255460e-03	1.173813e-03	3.509239e-03
DTLZ5	IGD	Average	1.641832e-02	1.343846e-01	9.823208e-02	5.789609e-02
		Best	1.080152e-02	8.110197e-03	6.073218e-02	3.501886e-02
		Worst	2.427658e-02	4.968867e-01	1.482071e-01	9.632784e-02
		SD	3.308058e-03	2.417796e-01	1.895655e-02	1.681788e-02
	Spacing	Average	1.442703e-02	1.528519e-02	2.970802e-02	1.882588e-02
		Best	8.057921e-03	1.327747e-02	2.403535e-02	9.132885e-03
		Worst	1.886809e-02	1.950345e-02	5.444988e-02	3.396901e-02
		SD	2.249722e-03	2.846769e-03	7.476184e-03	6.090413e-03
	Spread	Average	1.002221	9.823109e-01	9.978601e-01	1.000093
		Best	1.011890	1.000013	9.986370e-01	1.000363
		Worst	9.987108e-01	9.721332e-01	9.966008e-01	9.999465e-01
		SD	2.973230e-03	1.221257e-02	4.178853e-04	9.262710e-05
DTLZ6	IGD	Average	1.817637e +01	5.110904e-01	8.672638e-03	2.044580e-02
		Best	1.714688e +01	1.125814e-02	7.190239e-03	1.053539e-02
		Worst	1.916421e +01	1.773498	1.032295e-02	4.643289e-02
		SD	5.234625e-01	8.480715e-01	7.846861e-04	1.187041e-02
	Spacing	Average	2.127344	1.092193e-01	1.310135e-02	1.683930e-02
		Best	1.651504	1.328278e-02	9.638915e-03	9.812509e-03
		Worst	2.378556	3.033813e-01	1.563092e-02	3.437028e-02
		SD	1.775371e-01	1.317103e-01	1.355862e-03	6.051766e-03
	Spread	Average	9.999731e-01	1.000368	8.807755e-01	1.000399
		Best	1.000073	1.001207	1.003490	1.006908
		Worst	9.998856e-01	9.999731e-01	7.533910e-01	9.996811e-01
		SD	5.136830e-05	5.651200e-04	5.791286e-02	1.394143e-03

Table 8 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MOGWO
DTLZ7	IGD	Average	8.926492e-01	2.152397e-01	9.468594e-02	–
		Best	2.970584e-01	7.169967e-02	6.847490e-02	–
		Worst	1.329984	3.588205e-01	1.195888e-01	–
		SD	2.833406e-01	1.634459e-01	1.184403e-02	–
	Spacing	Average	3.796102e-01	5.280027e-02	6.790844e-02	–
		Best	2.109674e-02	3.700353e-02	5.463724e-02	–
		Worst	1.008876	7.142480e-02	7.983412e-02	–
		SD	2.666095e-01	1.811141e-02	5.835539e-03	–
	Spread	Average	1.001936	9.894380e-01	9.906941e-01	–
		Best	1.007610	9.969709e-01	9.955179e-01	–
		Worst	1.000084	9.713501e-01	9.339576e-01	–
		SD	1.820868e-03	1.210718e-02	1.202882e-02	–

Table 9 The ranks of all algorithms among others according to DTLZ dataset

Indicators	MOBRO		NSGA-III		MOPSO		MOGWO		
	Average	Overall	Average	Overall	Average	Overall	Average	Overall	
IGD	Average	2.57	2	2.29	1	2.57	2	2.57	2
	Best	2.86	3	1.86	1	3.00	4	2.29	2
	Worst	2.86	4	2.29	2	2.14	1	2.71	3
	SD	2.57	3	2.43	2	1.71	1	3.29	4
Spacing	Average	3.14	4	1.57	1	2.57	2	2.71	3
	Best	2.86	3	2.00	1	3.00	4	2.14	2
	Worst	3.14	4	1.57	1	2.57	2	2.71	3
	SD	3.00	4	2.29	2	2.00	1	2.71	3
Spread	Average	1.57	1	3.00	3	3.29	4	2.14	2
	Best	1.71	1	3.00	2	3.43	4	1.86	2
	Worst	1.57	1	2.59	3	3.14	4	3.00	3
	SD	2.29	2	2.71	3	2.00	1	3.00	4
Average	2.67	1.83	2.50	2.75					
Rank	3	1	2	4					

signed-rank test, which is widely used in the literature [39, 40], was used for a more accurate comparison of five optimization algorithms. Table 13 lists the statistical results obtained by this test. As can be seen in this table, MBRO overcomes MOPSO, MAOA, and MOAVOA in all benchmarks, but it could overcome NSGA-III in only 7 out of 10 datasets.

Table 10 Obtained results for the CEC 2018 dataset by MOBRO and four state-of-the-art optimization algorithms (UF)

Indicators		MOBRO	NSGA-III	MOPSO	MAOA	MOAVOA	
MaOP 1	IGD	Average	1.280689e+01	1.179722e+01	2.315855e+01	2.300571e+01	2.285252e+01
		Best	1.135367e+01	1.123013e+01	1.522737e+01	1.811016e+01	1.472778e+01
		Worst	1.504236e+01	1.283912e+01	2.562928e+01	2.494138e+01	2.474117e+01
		SD	8.770155e-01	3.739899e-01	1.894522	1.330027	1.891267
	Spacing	Average	1.947458	5.651070e-01	2.246525	1.908771e+02	1.982981e+02
		Best	8.591850e-01	1.601317e-01	1.320829	2.104541e+01	5.554046e+01
		Worst	3.884274	1.579748	3.639818	3.909873e+02	5.783126e+02
		SD	7.862528e-01	2.463941e-01	4.814220e-01	9.620660e+01	1.031690e+02
	Spread	Average	1.003457	9.939078e-01	9.911545e-01	9.916539e-01	9.918152e-01
		Best	1.005403	9.990573e-01	9.950086e-01	9.940113e-01	9.999442e-01
		Worst	1.001724	9.909068e-01	9.892068e-01	9.892618e-01	9.883249e-01
		SD	9.130956e-04	1.931461e-03	1.442245e-03	1.339972e-03	2.077654e-03
MaOP 2	IGD	Average	1.722710e-01	1.540622e-01	5.773744e+01	5.675091e+01	5.849365e+01
		Best	1.158435e-01	9.993969e-02	1.390374e-01	4.042861e+01	9.986961
		Worst	2.634274e-01	2.715598e-01	7.425907e+01	7.841124e+01	7.607871e+01
		SD	3.629104e-02	3.781798e-02	1.521253e+01	8.757226	1.396356e+01
	Spacing	Average	8.934147e-01	1.469772e-01	2.701819e+01	6.911821e+02	9.280110e+02
		Best	2.685580e-02	2.490764e-02	2.003925e-02	2.622005e+01	8.249202e+01
		Worst	1.276578e+01	5.766927e-01	3.633424e+01	2.757809e+03	2.331228e+03
		SD	2.942640	1.124026e-01	7.397986	5.882307e+02	6.332044e+02
	Spread	Average	1.068932	1.006759	9.764803e-01	9.728797e-01	9.770036e-01
		Best	1.772817	1.115801	9.840966e-01	9.808995e-01	9.878845e-01
		Worst	1.005587	9.306922e-01	9.682336e-01	9.642990e-01	9.662391e-01
		SD	1.892623e-01	4.398047e-02	3.990531e-03	5.018455e-03	4.575877e-03
MaOP 3	IGD	Average	8.188333e+01	1.105402e+02	1.583787e+02	1.617477e+02	1.608237e+02
		Best	7.174101e+01	9.872514e+01	5.715911e+01	1.356553e+02	1.200379e+02
		Worst	9.087118e+01	1.190909e+02	1.665655e+02	1.661024e+02	1.670917e+02
		SD	4.794676	4.755184	2.119335e+01	5.806676	8.849519
	Spacing	Average	1.444560e+01	4.000420	6.941603	5.553844e+02	6.533492e+02
		Best	8.587770	3.270431	3.493016	3.617218e+01	6.825776e+01
		Worst	1.817662e+01	4.544279	7.945202	1.132713e+03	1.495855e+03
		SD	2.107314	2.754888e-01	8.193652e-01	3.746089e+02	3.703341e+02
	Spread	Average	1.002001	9.879779e-01	9.875193e-01	9.874067e-01	9.870912e-01
		Best	1.004999	9.897041e-01	9.996217e-01	9.966737e-01	9.979417e-01
		Worst	9.965106e-01	9.858885e-01	9.848358e-01	9.855781e-01	9.847030e-01
		SD	1.755562e-03	8.972692e-04	2.656973e-03	2.061380e-03	2.380726e-03

Table 10 (continued)

Indicators		MOBRO	NSGA-III	MOPSO	MAOA	MOAVOA	
MaOP 4	IGD	Average	4.329582e-01	4.085190e-01	1.523572	1.720857	2.114253
		Best	4.107510e-01	4.084269e-01	4.089634e-01	4.089219e-01	4.104431e-01
		Worst	5.274832e-01	4.086168e-01	3.652429	3.631260	7.110467
		SD	3.188261e-02	5.130346e-05	7.637154e-01	8.649834e-01	1.775775
	Spacing	Average	9.588202e-03	4.166058e-03	3.883692	1.264878e+01	9.742255
		Best	3.367319e-03	3.592775e-03	9.728628e-03	2.966842e-01	6.674813e-02
		Worst	1.348216e-02	5.020481e-03	1.379721e+01	5.345847e+01	4.088302e+01
		SD	2.605233e-03	4.424473e-04	3.410873	1.410021e+01	9.477800
	Spread	Average	1.000686	9.995028e-01	1.008391	1.029228	1.005893
		Best	1.001472	1.000122	1.037784	1.235826	1.018350
		Worst	9.997030e-01	9.989275e-01	9.904983e-01	9.940680e-01	9.913539e-01
		SD	4.227383e-04	2.924278e-04	1.094725e-02	5.392740e-02	6.597454e-03
MaOP 5	IGD	Average	2.826809e-01	2.402887e-01	1.027051e+01	1.015983e+01	9.563648
		Best	1.496619e-01	1.483356e-01	2.664957e-01	1.108681	2.809953e-01
		Worst	8.385730e-01	7.616025e-01	1.349210e+01	1.494494e+01	1.282369e+01
		SD	1.414081e-01	1.481920e-01	2.780095	2.585477	2.555060
	Spacing	Average	7.531275e-02	1.490636e-01	1.541312e+01	5.578227e+02	5.045477e+02
		Best	3.049589e-02	2.301076e-02	5.269646e-02	1.648735e+01	5.705255e+01
		Worst	1.779381e-01	1.944989	2.555967e+01	8.344497e+02	1.081950e+03
		SD	3.503398e-02	3.767966e-01	5.472264	1.990904e+02	2.276011e+02
	Spread	Average	1.003370	9.627138e-01	9.860944e-01	9.835977e-01	9.832198e-01
		Best	1.014274	1.192530	9.991991e-01	1.005753	1.007057
		Worst	9.927690e-01	9.042997e-01	9.660495e-01	9.644006e-01	9.532344e-01
		SD	4.880223e-03	5.355973e-02	8.653666e-03	1.049378e-02	1.179231e-02
MaOP 6	IGD	Average	3.510181e-01	3.758024e-01	9.418994	1.020563e+01	1.078723e+01
		Best	2.509052e-01	2.546738e-01	5.904165e-01	2.426727	6.058222e-01
		Worst	5.506383e-01	5.530022e-01	1.228594e+01	1.412972e+01	1.485629e+01
		SD	8.676948e-02	7.410907e-02	2.409156	2.233431	2.896053
	Spacing	Average	1.613340e-01	5.468309e-01	1.516025e+01	1.978512e+03	1.816432e+03
		Best	3.522512e-02	7.879879e-02	3.550197e-02	4.009158e+01	1.302055e+01
		Worst	1.303564	2.652046	2.799339e+01	3.416431e+03	3.318927e+03
		SD	2.499937e-01	5.911952e-01	5.326254	6.984984e+02	7.120382e+02
	Spread	Average	1.004144	1.002301	9.768345e-01	9.788488e-01	9.804026e-01
		Best	1.016549	1.113324	1.000793	1.016835	1.004567
		Worst	9.994245e-01	9.781679e-01	9.540067e-01	9.574256e-01	9.210042e-01
		SD	4.170024e-03	2.907394e-02	1.245435e-02	1.406736e-02	1.668111e-02

Table 10 (continued)

Indicators			MOBRO	NSGA-III	MOPSO	MAOA	MOAVOA
MaOP 7	IGD	Average	6.132975e-01	4.378830e-01	1.558657e+02	1.512422e+02	1.641558e+02
		Best	3.369621e-01	2.580641e-01	6.349206e-01	4.362942e+01	1.002890e+02
		Worst	9.779144e-01	7.763284e-01	2.027908e+02	1.952029e+02	2.184096e+02
		SD	1.888476e-01	1.707227e-01	3.944183e+01	3.120036e+01	2.428727e+01
	Spacing	Average	3.027074e-03	4.732907e-03	6.093181e+01	4.682714e+02	3.788739e+02
		Best	7.488660e-06	6.501310e-07	1.632292e-01	1.206568	5.526648e+01
		Worst	1.184185e-02	1.498541e-02	1.493050e+02	9.081836e+02	9.760978e+02
		SD	3.397072e-03	3.464717e-03	4.159527e+01	2.436538e+02	2.496942e+02
	Spread	Average	1.000063	9.988089e-01	9.988031e-01	9.987610e-01	9.988655e-01
		Best	1.000501	1.000102	1.001805	1.004583	1.004483
		Worst	9.998935e-01	9.962559e-01	9.952689e-01	9.953489e-01	9.960822e-01
		SD	1.276309e-04	1.131309e-03	1.672478e-03	2.123810e-03	1.784663e-03
MaOP 8	IGD	Average	6.349356e-01	4.966485e-01	1.540216e+02	1.660478e+02	1.523310e+02
		Best	2.834544e-01	2.930514e-01	4.781371e-01	1.265571e+02	8.994681e+01
		Worst	1.106212	7.751063e-01	1.910515e+02	2.186181e+02	1.899878e+02
		SD	1.724965e-01	1.903707e-01	3.503142e+01	2.167131e+01	2.687339e+01
	Spacing	Average	3.359544e-03	2.361138e-02	6.461466e+01	4.804220e+02	4.842002e+02
		Best	2.870490e-04	2.259029e-07	0	3.272022e+01	1.857777e+02
		Worst	1.154178e-02	9.334591e-02	2.256020e+02	9.695265e+02	1.113995e+03
		SD	2.585006e-03	2.820890e-02	5.895722e+01	2.343838e+02	2.164222e+02
	Spread	Average	1.000110	1.000396	9.985036e-01	9.989156e-01	9.994343e-01
		Best	1.000966	1.006346	1.004028	1.002674	1.005704
		Worst	9.999088e-01	9.973604e-01	9.950597e-01	9.954564e-01	9.957529e-01
		SD	2.093556e-04	2.186363e-03	2.217301e-03	1.913629e-03	2.496393e-03
MaOP 9	IGD	Average	8.193997e-01	7.227372e-01	1.605672e+02	1.747336e+02	1.713531e+02
		Best	4.494889e-01	4.215775e-01	6.186603e-01	7.898503e+01	1.107470e+02
		Worst	1.599292	1.132909	2.222351e+02	2.306766e+02	2.227265e+02
		SD	2.165541e-01	1.950350e-01	4.290435e+01	3.618456e+01	2.737988e+01
	Spacing	Average	3.382401e-03	4.110654e-03	6.418518e+01	3.873672e+02	3.752915e+02
		Best	7.759879e-06	1.078429e-06	1.749655e-02	1.726584e+01	1.244289e+01
		Worst	1.594789e-02	1.439092e-02	1.738297e+02	8.379803e+02	6.500526e+02
		SD	4.043935e-03	4.877706e-03	3.904523e+01	2.254150e+02	1.843773e+02
	Spread	Average	1.000124	9.997425e-01	9.981249e-01	9.985187e-01	9.983879e-01
		Best	1.001644	1.000496	1.000487	1.002704	1.002739
		Worst	9.999915e-01	9.982633e-01	9.938710e-01	9.951644e-01	9.967334e-01
		SD	3.454144e-04	4.117493e-04	1.632783e-03	1.933038e-03	1.365260e-03

Table 10 (continued)

Indicators		MOBRO	NSGA-III	MOPSO	MAOA	MOAVOA	
MaOP 10	IGD	Average	1.038715	8.173226e-01	1.565097e+02	1.697243e+02	1.623744e+02
		Best	3.205402e-01	3.551969e-01	7.747941e-01	1.147632e+02	9.775422e+01
		Worst	1.666920	1.636740	2.024902e+02	2.019386e+02	2.037558e+02
		SD	2.419364e-01	2.910690e-01	3.946067e+01	2.330661e+01	2.516184e+01
	Spacing	Average	5.799670e-03	5.730607e-03	8.191059e+01	3.311193e+02	4.434813e+02
		Best	1.404513e-05	4.485781e-07	4.742880e-02	9.124571	5.684993e+01
		Worst	1.961105e-02	2.013774e-02	2.804029e+02	8.254944e+02	1.190714e+03
		SD	5.712601e-03	6.184953e-03	6.706738e+01	2.301720e+02	2.485326e+02
	Spread	Average	1.000125	9.998092e-01	9.983480e-01	9.984051e-01	9.980910e-01
		Best	1.001246	1.000329	1.003685	1.008174	1.004115
		Worst	9.999696e-01	9.983956e-01	9.953869e-01	9.953536e-01	9.954202e-01
		SD	2.825960e-04	7.227372e-01	2.258001e-03	2.674575e-03	2.092946e-03

4 Conclusions

We proposed the multi-objective version of the Battle Royale optimization algorithm for solving unconstrained multi-objective optimization problems. The MOBRO algorithm is based on the Battle Royale optimization algorithm proposed in 2021, which is inspired by the Battle Royale games.

MOBRO uses an archive to store non-dominated solutions, and after finishing its search over the population, it provides several candidate solutions in the Pareto front. All solutions in the Pareto front are better than or at least as good as all other solutions in the population in satisfying the objectives. Although MOBRO is an archive-based optimization algorithm, we compared it with both evolutionary and swarm-based algorithms, where the former group uses the sorting scheme, but the latter group uses the archive-based scheme.

We compared the proposed algorithm with five state-of-the-art multi-objective optimization algorithms, MOPSO, NSGA-III, MOAVOA, MAOA, and MOGWO, based on three performance metrics: inverted generational distance (IGD) for measuring convergence, spacing (SP) for measuring how evenly the estimated solutions are distributed, and maximum spread for measuring the diversity of estimated points. These algorithms have been tested on four benchmark datasets: CEC 2009, CEC 2018, ZDT, and DTLZ. According to the obtained results, MOBRO ranked first in the ZDT and CEC 2009 datasets. In the CEC 2009 and DTLZ datasets, MOBRO ranked second and third, respectively. In terms of performance metrics, MOBRO is more efficient according to IGD and almost always ranks first based on this metric. This indicates that MOBRO has the best convergence among its competitors, but it ranks from first to last in different datasets based on the spacing metric. This suggests that according to the distribution of the estimated solutions and their diversity, MOBRO ranks first, second, third,

Table 11 The ranks of each algorithm among others according to CEC 2018 dataset

Metrics	MOBRO		NSGA-III		MOPSO		MAOA		MOAVOA	
	Average	Overall	Average	Overall	Average	Overall	Average	Overall	Average	Overall
IGD	Average	2	1.2	1	3.7	3	4.2	5	4.1	4
	Best	2	1.5	1	2.9	3	4.5	5	4.1	4
	Worst	2	1.3	1	3.8	3	4	4	4.2	5
	SD	1.6	2	1.4	1	4.7	5	3.4	3	3.9
Spacing	Average	2	1.5	1	2.9	3	4.5	4	4.5	4
	Best	2.1	2	1.5	1	2.4	3	4.3	4.7	5
	Worst	1.7	2	1.5	1	2.8	3	4.4	4.6	5
	SD	1.6	1	1.6	1	2.8	3	4.4	4.6	5
Spread	Average	1.4	1	2.6	3	2.1	2	3.6	3.5	4
	Best	2.8	3	3.2	5	2.4	1	2.9	2.5	2
	Worst	1	1	2.6	3	2	2	3.6	3.8	5
	SD	1.6	1	3.2	3	3.1	2	3.5	3.6	5
Average	1.75	2.75	4,166,667	4,333,333						
Rank	2	3	4	5						

Table 12 Required CPU time for running each algorithm when applied to the CEC 2018 dataset

Functions	MBRO	NSGA	MOPSO	MAOA	MOAVOA
MaOP 1	6.016345e+01	2.683983e+03	6.057002	2.205209e-01	1.385860
MaOP 2	6.506757e+01	3.126875e+03	6.753145	2.397298e-01	1.342538
MaOP 3	5.819457e+01	2.883762e+03	4.377854	1.232011	1.655284
MaOP 4	6.074567e+01	3.097571e+03	8.877737e-01	1.376394	3.779367
MaOP 5	7.178642e+01	2.220301e+03	2.925692	2.995922e-01	5.830067
MaOP 6	8.664028e+01	2.822965e+03	5.619126	3.896699e-01	5.676
MaOP 7	8.885593e+01	2.966547e+03	1.644399	2.907415e-01	5.184917e-01
MaOP 8	6.443893e+01	2.987555e+03	1.686123	3.595861e-01	7.887001e-01
MaOP 9	6.212550e+01	2.849358e+03	2.283527	2.887523e-01	5.442233e-01
MaOP 10	5.664245e+01	2.953737e+03	2.119449	3.448142e-01	7.435091e-01

Table 13 Pair-wise statistical comparison between MBRO and all competitors by Wilcoxon signed-rank test ($\alpha=0.05$)

Functions	NSGA		MOPSO		MAOA		MOAVOA	
	P	H	P	H	P	H	P	H
MaOP 1	0.000032	1	0.000012	1	0.000012	1	0.000012	1
MaOP 2	0.061480	0	0.000014	1	0.000012	1	0.000012	1
MaOP 3	0.000012	1	0.000014	1	0.000012	1	0.000012	1
MaOP 4	0.000012	1	0.000014	1	0.000014	1	0.000014	1
MaOP 5	0.028314	1	0.000012	1	0.000012	1	0.000012	1
MaOP 6	0.150003	0	0.000012	1	0.000012	1	0.000012	1
MaOP 7	0.000493	1	0.000014	1	0.000012	1	0.000012	1
MaOP 8	0.006848	1	0.000014	1	0.000012	1	0.000012	1
MaOP 9	0.051087	0	0.000014	1	0.000012	1	0.000012	1
MaOP 10	0.011876	1	0.000014	1	0.000012	1	0.000012	1

or last in different test suits. Finally, according to the spread metric, MOBRO ranks first or second, indicating that its solutions are well distributed along the Pareto Front border. The limitation of the proposed approach is that it is designed for solving only the unconstrained multi-objective optimization problems, while some of the real-world problems are constrained. We aim to address this issue in our future work.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11227-023-05676-4>.

Author contributions SA and TA provided core concepts and carried out implementations. RD, SA, and TA drafted the manuscript. RD and MANB proofread the manuscript and approved the final manuscript.

Funding No funding for this study.

Data availability The utilized codes and data are available on request to enable the method proposed in the manuscript to be replicated by readers.

Declarations

Competing Interests The authors declare no potential competing interests.

References

1. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82. <https://doi.org/10.1109/4235.585893>
2. Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, London
3. Askarzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Commun Nonlinear Sci Numer Simul* 19:1213–1228. <https://doi.org/10.1016/J.CNSNS.2013.08.027>
4. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: an overview. *Swarm Intel* 1:33–57. <https://doi.org/10.1007/S11721-007-0002-0>
5. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global Opt.* <https://doi.org/10.1007/S10898-007-9149-X>
6. Savsani P, Savsani V (2016) Passing vehicle search (PVS): a novel metaheuristic algorithm. *Appl Math Model* 40:3951–3978. <https://doi.org/10.1016/J.APM.2015.10.040>
7. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: A gravitational search algorithm. *Inf Sci (N Y)* 179:2232–2248. <https://doi.org/10.1016/J.INS.2009.03.004>
8. Rahkar Farshi T (2021) Battle royale optimization algorithm. *Neural Comput Appl* 33:1139–1157. <https://doi.org/10.1007/S00521-020-05004-4/TABLES/10>
9. Rahkar Farshi TA, Agahian S, Dehkharghani R (2022) BinBRO: binary battle royale optimizer algorithm. *Expert Syst Appl* 195:116599. <https://doi.org/10.1016/J.ESWA.2022.116599>
10. Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhangd Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm. Evol Comput* 1:32–49. <https://doi.org/10.1016/J.SWEVO.2011.03.001>
11. Luo J, Liu Q, Yang Y, Li X, Rong Chen M, Cao W (2017) An artificial bee colony algorithm for multi-objective optimisation. *Appl Soft Comput* 50:235–251. <https://doi.org/10.1016/J.ASOC.2016.11.014>
12. Coello Coello C (1999) *A survey of constraint handling techniques used with evolutionary algorithms*. Laboratorio Nacional de Informatica Avanzada, Veracruz, Mexico, Technical report Lania-RI-99-04.
13. Hwang C-L, Masud ASMd (1979) *Multiple objective decision making —methods and applications lecture notes in economics and mathematical systems*. Springer, Berlin
14. Coello Coello CA, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8:256–279. <https://doi.org/10.1109/TEVC.2004.826067>
15. Zhong K, Zhou G, Deng W, Zhou Y, Luo Q (2021) MOMPA: multi-objective marine predator algorithm. *Comput Methods Appl Mech Eng* 385:114029. <https://doi.org/10.1016/J.CMA.2021.114029>
16. 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>
17. Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. *Appl Intell* 48:805–820. <https://doi.org/10.1007/S10489-017-1019-8/TABLES/9>
18. Alaya I, Solnon C, Ghédira K (2007) Ant colony optimization for multi-objective optimization problems. *Proc Int Conf Tools Artif Intel ICTAI* 1:450–457. <https://doi.org/10.1109/ICTAI.2007.108>
19. Khodadadi N, Soleimanian Gharehchopogh F, Mirjalili S (2022) MOAVOA: a new multi-objective artificial vultures optimization algorithm. *Neural Comput Appl* 34:20791–20829. <https://doi.org/10.1007/S00521-022-07557-Y/FIGURES/14>

20. Khodadadi N, Abualigah L, El-Kenawy ESM, Snasel V, Mirjalili S (2022) An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems. *IEEE Access* 10:106673–106698. <https://doi.org/10.1109/ACCESS.2022.3212081>
21. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Trans Evol Comput* 18:577–601. <https://doi.org/10.1109/TEVC.2013.2281535>
22. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197. <https://doi.org/10.1109/4235.996017>
23. Wang Z, Zhang W, Guo Y, Han M, Wan B, Liang S (2022) A multi-objective chicken swarm optimization algorithm based on dual external archive with various elites. *Appl Soft Comput*. <https://doi.org/10.1016/J.ASOC.2022.109920>
24. Cui Y, Meng X, Qiao J (2022) A multi-objective particle swarm optimization algorithm based on two-archive mechanism. *Appl Soft Comput* 119:108532. <https://doi.org/10.1016/J.ASOC.2022.108532>
25. Zhang W, Wang S, Zhou A, Zhang H (2022) A practical regularity model based evolutionary algorithm for multiobjective optimization. *Appl Soft Comput* 129:109614. <https://doi.org/10.1016/J.ASOC.2022.109614>
26. Zhou X, Gao Y, Yang S, Yang C, Zhou J (2022) A multiobjective state transition algorithm based on modified decomposition method. *Appl Soft Comput* 119:108553. <https://doi.org/10.1016/J.ASOC.2022.108553>
27. Kumar A, Wu G, Ali MZ, Luo Q, Mallipeddi R, Suganthan PN, Das S (2021) A Benchmark-Suite of real-World constrained multi-objective optimization problems and some baseline results. *Swarm Evol Comput* 67:100961. <https://doi.org/10.1016/J.SWEVO.2021.100961>
28. Q Zhang, A Zhou, S Zhao, PN Suganthan, W Liu, S Tiwari (2009) Multi-objective optimization test instances for the congress on evolutionary computation (CEC 2009) special session & competition
29. H Li, K Deb, Q Zhang, PN Suganthan (2018) Challenging novel many and multi-objective bound constrained benchmark problems, In: Technical Report, Technical Report
30. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8:173–195. <https://doi.org/10.1162/106365600568202>
31. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. *Evol Multiobj Opt*. https://doi.org/10.1007/1-84628-137-7_6
32. Akan S, Akan T (2022) Battle royale optimizer with a new movement strategy. *Stud Syst Decis Control* 212:265–279. https://doi.org/10.1007/978-3-031-07512-4_10/COVER
33. Van Den Bergh F (2001) An Analysis of Particle Swarm Optimizers (PSO). University of Pretoria, Pretoria, pp 78–85
34. Knowles JD, Corne DW (2000) Approximating the nondominated front using the pareto archived evolution strategy. *Evol Comput* 8:149–172. <https://doi.org/10.1162/106365600568167>
35. Bezerra LCT, López-Ibáñez M, Stützle T (2017) An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. *Lect Notes Comput Sci*. https://doi.org/10.1007/978-3-319-54157-0_3/FIGURES/9
36. JR Schott (1995) Fault tolerant design using single and multicriteria genetic algorithm optimization
37. N Riquelme, C Von Lüken, B Barán (2015) Performance metrics in multi-objective optimization. In: Proceedings 2015 41st Latin American Computing Conference, CLEI 2015. <https://doi.org/10.1109/CLEI.2015.7360024>.
38. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3:257–271. <https://doi.org/10.1109/4235.797969>
39. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol Comput* 1:3–18. <https://doi.org/10.1016/J.SWEVO.2011.02.002>
40. Chen MR, Zeng GQ, Di Lu K (2019) A many-objective population extremal optimization algorithm with an adaptive hybrid mutation operation. *Inf Sci (N Y)* 498:62–90. <https://doi.org/10.1016/J.IJNS.2019.05.048>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Sait Alp¹ · **Rahim Dehkharghani**² · **Taymaz Akan**^{3,4} · **Mohammad A. N. Bhuiyan**³

✉ Mohammad A. N. Bhuiyan
nobel.bhuiyan@lsuhs.edu

Sait Alp
sait.alp@erzurum.edu.tr

Rahim Dehkharghani
rahim.dehkharghani@khas.edu.tr

Taymaz Akan
taymazakan@topkapi.edu.tr; taymaz.akan@lsuhs.edu

¹ Department of Computer Engineering, Erzurum Technical University, Erzurum, Turkey

² Department of Computer Engineering, Department of Management Information Systems, Kadirhas University, Istanbul, Turkey

³ Department of Medicine, Louisiana State University Health Sciences Center, Shreveport 71103, USA

⁴ Istanbul Topkapi University, Istanbul, Turkey