



# Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization

Rizk M. Rizk-Allah<sup>1</sup> · Aboul Ella Hassanien<sup>2</sup> · Adam Slowik<sup>3</sup>

Received: 11 June 2019 / Accepted: 7 February 2020 / Published online: 13 March 2020  
© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

Many engineering optimization problems are typically multi-objective in their natures and multidisciplinary with a large number of decision variables. Furthermore, Pareto dominance loses its effectiveness in such situations. Thus, developing a robust optimization algorithm undoubtedly becomes a true challenge. This paper proposes a multi-objective orthogonal opposition-based crow search algorithm (M2O-CSA) for solving large-scale multi-objective optimization problems (LSMOPs). In the M2O-CSA, a multi-orthogonal opposition strategy is employed to mitigate the conflicts among the convergence and distribution of solutions. First, two individuals are randomly chosen to undergo the crossover stage and then orthogonal array is presented to obtain nine individuals. Then individuals are used in the opposition stage to improve the diversity of solutions. The effectiveness of the proposed M2O-CSA is investigated by implementing it on different dimensions of multi-objective optimization problems (MOPs). The Pareto front solutions of these MOPs have various characteristics such as convex, non-convex and discrete. It is also applied to solve multi-objective design applications with distinctive features such as four bar truss (FBT) design, welded beam (WB) design, disk brake (DB) design, and speed reduced (SR) design, where they involve different characteristics. In this context, a new decision making tool based on multi-objective optimization on the basis of ratio analysis (MOORA) technique is employed to help the designer for extracting the operating point as the best compromise or satisfactory solution to execute the candidate engineering design. Simulation results affirm that the proposed M2O-CSA works efficiently and effectively.

**Keywords** Crow search algorithm · Orthogonal · Opposition · Multi-objective optimization · Metaheuristic · Engineering designs · MOORA

---

Rizk M. Rizk-Allah and Aboul Ella Hassanien: Scientific Research Group in Egypt.

✉ Aboul Ella Hassanien  
aboitcairo@fci-cu.edu.eg  
<http://www.egyptscience.net>

Rizk M. Rizk-Allah  
<http://www.egyptscience.net>

<sup>1</sup> Faculty of Engineering, Menoufia University, Shebin El-Kom, Egypt

<sup>2</sup> Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt

<sup>3</sup> Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

## 1 Introduction

In almost real design applications, the designer often faces the problem of achieving many design targets. These targets are often conflicting and incommensurable and needed to be optimized simultaneously. The process of optimizing multiple targets (objectives) is denoted by a multi-objective optimization problem (MOP). In this regard, there is no single optimal solution but a set of solutions that optimize all the objective functions simultaneously. The set of solutions is denoted as Pareto-optimal solutions, non-inferior, non-dominated or efficient solutions [1]. According to Pareto optimality, a solution is identified as Pareto-optimal solution (or non-dominated, efficient, non-inferior), if no objective can be improved without deteriorating one other objective at least.

The traditional approaches for solving multi-objective optimization problems (MOPs) are classified into three

categories according to the preferences that are provided by expert or decision maker (DM), namely a priori, a posteriori or generation, and progressive approaches. In a priori approach (decide  $\Rightarrow$  search), the expert or the DM set his/her preferences before the solution process (e.g., weight for each objective function that reflects the significance of the objective). Afterward, the MOP is converted into a single-objective problem (SOP) using the set of weights and then it can be solved as a single-objective optimizer. The disadvantages are as follows: there is a difficulty in quantifying (i.e., weights) his/her preferences. They give one solution for certain weights, so they required multiple runs to obtain the Pareto-optimal set (POS). In addition, they fail to deal with non-convex problems. The prominent methods of this category are the weighted sum method and  $\varepsilon$ -constraint method [1–4]. In a posteriori approach (search  $\Rightarrow$  decide), the expert or the DM engages eventually to choose one of the obtained solutions according to preferences. However, they require higher computational cost, the advantage of such approaches is that the Pareto-optimal set can be obtained in one single run. For progressive method (decide  $\leftrightarrow$  search), the decision maker is integrated with an interactive form to set his/her preferences during the optimization process until a satisfying solution is obtained or no further improvement is possible [5]. Since the conventional approaches are occupied by some drawbacks, such as they suffer from stagnation in local optima, they rely on initial guess and are derivative based-algorithms. Thus, such techniques are not suitable for solving a large variety of optimization problems [6].

In 1985, Schaffer [7] proposed a revolutionary idea to handle multi-objective optimization using an evolutionary optimization technique. The advantages occupied by this technique are: it can avoid local optima and gradient-free mechanism, which made it readily applicable to the real problems as well. Since this work was proposed, a significant number of studies were flourished in this regard such as vector evaluated genetic algorithms (VEGA) [8], non-dominated sorting genetic algorithm (NSGA) [9], niched Pareto genetic algorithm (NPGA) [10], multi-objective genetic algorithm (MOGA) [11], strength Pareto evolutionary algorithm (SPEA) [12]. Also, some recent population-based search algorithms for multi-objective optimization [13–17] have been proposed in the literature and applied to a variety of MOPs among them multi-objective bee algorithm [18], multi-objective salp swarm [19], multi-objective ant lion optimizer [20], and multi-objective grey wolf optimizer (MOGWO) [21], etc.

Apart from the previously developed optimization algorithms, the literature becomes very rich with several recent methodologies that presented to solve different aspects of multi-objective optimization applications

[22–31]. Tian et al. [22] developed an enhanced multi-objective evolutionary algorithm (MOEA) based on distance indicator to address the versatility. Rong et al. [23] solved the dynamic MOPs through the multidirectional prediction strategy, where the multiple directions are induced by multiple representative solutions acquiring from the previous environments to predict the new set of the Pareto-optimal solutions. Zhang et al. [24] proposed a novel multi-objective particle swarm algorithm based on the competitive mechanism as a learning strategy to guide the search of particles and enhance the robustness while handling the MOPs. The MOPs that contain more than three objectives are usually denoted as many-objective optimization problems (Ma-OPs). In this sense, Liu et al. [25] proposed a many-objective EA based on a one-by-one selection mechanism to maintain the balance among the convergence and diversity of solution while solving the Ma-OPs. Additionally, Liu et al. [26] introduced a reference point-based EA to maintain the spread and uniform distribution of solutions while tackling the Ma-OPs. Gong Liu et al. [27] presented a set-based GA to solve interval Ma-OPs, where an evolutionary scheme and set-based Pareto dominance concepts are introduced to improve the performance of this approach. Yue et al. [28] exhibited a novel multi-objective particle swarm algorithm based on ring topology and crowding distance concept. The crowding distance aims to improve the distribution of solutions in both decision and objective spaces while the ring topology technique helps in generating stable niches. Gu et al. [29] introduced a self-organizing multi-objective PSO algorithm to deal with multimodal MOPs. This approach can map the solutions of the population to a latent space in which a neighboring relation is building to emphasize the distribution of solutions. Adel et al. [30] proposed a new multi-objective whale optimization algorithm (WOA) for MOPs. It is established based on the external archive to store the non-dominated, where an external archive is to guide the population toward promising areas in the search space. In addition, crowding distance strategy is introduced to maintain the diversity of solutions. Wenjun et al. [31] developed an effective ensemble algorithm for solving MOPs. This algorithm embeds various evolutionary operators and different selection criteria which are performed on multiple populations to achieve more distribution of solutions. Additionally, there is an increasing interest in mathematical MOPs using recent MOEAs as it is presented in the papers [32, 33].

Despite the notable number of optimization algorithms which are proposed in this field, there is an important question: why we create more optimization techniques? The answer is referring to the No Free Lunch (NFL) theorem [34] which proves that no optimization algorithm copes all optimization problems; when the algorithm

successes in solving a specific set of problems, it does not mean that it succeeds with all natures and types of optimization problems. Meanwhile, all the optimization techniques are supposed to equal on average when considering all optimization problems, despite the superior performance on a subset of optimization problems. Thus NFL theorem permits the researchers to develop new algorithms or enhance/modify the current ones to deal with different natures of optimization fields whether for single optimization (SO) or multi-objective optimization (MOO) techniques. In an effort to maintain the diversity and convergence of the Pareto front solutions for the multiple objectives, this work proposes a modified version of the recently proposed crow search algorithm (CSA) [35]. The proposed is named multi-objective orthogonal opposition-based crow search algorithm (M2O-CSA) for solving large-scale multi-objective optimization problems (LSMOPs). The M2O-CSA comprises a multi-orthogonal-opposition strategy to maintain the convergence and distribution of solutions. It works by making a crossover among two individuals chosen randomly to generate a third one. These three individuals are arranged as orthogonal arrays and also their opposition arrays are obtained, where the former enhances the convergence and the later enables the diversity of solutions. The validation of the proposed M2O-CSA is investigated by employing different dimensions for multi-objective optimization problems (MOPs). Also, multi-objective design applications are solved such as four bar truss (FBT) design, welded beam (WB) design, disk brake (DB) design, and speed reduced (SR) design. Simulation results affirm that the proposed M2O-CSA works efficiently and effectively.

The contributions of this research are as follows:

1. A multi-objective orthogonal opposition-based crow search algorithm (M2O-CSA) is proposed for large-scale multi-objective optimization.
2. The crows' memory is updated using the domination concepts by using a parallel-orthogonal-opposition (P2O) strategy.
3. An archive is integrated into the M2O-CSA to maintain Pareto (non-dominated) solutions.
4. A grid mechanism based on P2O is employed to improve the stored solutions in the archive.
5. The effectiveness of M2O-CSA is validated by comprehensive simulation for different dimensions of MOPs and real engineering applications.
6. A new strategy based on the MOORA technique to extract the satisfactory or compromise solution which helps the designers to perform prudent designs.

The novelty and strength of the M2O-CSA lie in the amalgamation of the orthogonal arrays-based crossover scheme and opposition arrays-based strategy that is

dynamically evolved during the optimization process, thereby maintaining the balance among the convergence pattern and distribution of solutions. In addition, the MOORA technique is suggested for automatically determining the best compromise solution from Pareto-optimal solutions, which practically help the inexperienced designer to meet the diverse needs regarding different operating conditions and then provide a more realistic decision. Besides, to the best of our information, no endeavors have been suggested in the literature to innovate the M2O-CSA algorithm in solving the multi-objective optimization applications as well as the integration of the MOORA technique for solving MOPs.

Following the introduction in Sect. 1, the rest of the paper is organized as follows. Section 2 provides the preliminaries of multi-objective optimization with some performance indices. Section 3 briefly overviews the basics of CSA and then presents the M2O-CSA algorithm. The simulations of the obtained results as well as discussion are developed in Sect. 4. Lastly, Sect. 5 concludes the present work and states some recommendations for future works.

## 2 Preliminaries

This section gives the fundamental concepts of multi-objective optimization and performance assessments of the optimization techniques.

### 2.1 Statements of the multi-objective optimization

A multi-objective optimization problem refers to optimize more than one objective function. This means that set conflicted objective functions are optimized simultaneously in the presence of some equality and inequality constraints. It can be formulated as a minimization problem as follows:

$$\begin{aligned} \text{Minimize : } & F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_K(\mathbf{x})], \\ \text{Subject to : } & \\ & g_l(\mathbf{x}) \leq 0, \quad l = 1, 2, \dots, m; \\ & h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, p; \\ & \mathbf{x} = [x_1, x_2, \dots, x_n]; \quad x_i^L \leq x_i \leq x_i^U. \end{aligned} \quad (1)$$

where  $K$ ,  $m$ ,  $p$ , and  $n$  represent the number of objective functions, inequality constraints, equality constraints, and variables, respectively.  $f_k$  is the  $k$ th objective function in the vector  $F$ ,  $g_l$  denotes the  $l$ th inequality constraints,  $h_j$  defines the  $j$ th equality constraints, and  $[x_i^L, x_i^U]$  denotes the interval of lower and upper boundaries of  $i$ th variable.

It is worth mentioning that for single-objective function, the optimal solution can be obtained easily due to the unary

single criterion or objective. For the minimization problem, the solution  $\mathbf{u}$  is better than  $\mathbf{v}$  if and only if  $f(\mathbf{u}) < f(\mathbf{v})$ . However, the solutions in multiple objectives cannot be compared due to the presence of multiple criteria that judge the solutions. In this case, the concepts of Vilfredo Pareto [1] are employed that state that a solution dominates (is better than) another solution if and only if it provides better or equal on all of the objectives and shows a better value in at least one of the objective functions. Without loss of generality, the definitions regarding Pareto dominance for a minimization problem are as follows [1]:

**Definition 1 Pareto dominance** Assume that there are two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , where vector  $\mathbf{x}_1$  dominates vector  $\mathbf{x}_2$  (denote as  $\mathbf{x}_1 \succ \mathbf{x}_2$ ) iff :

$$\forall i \in 1, 2, \dots, K : f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \wedge \exists j \in 1, 2, \dots, K : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2). \tag{2}$$

Figure 1 provides an overview of Definition 1, where three are solutions  $\mathbf{a}$ ,  $\mathbf{b}$ , and  $\mathbf{c}$ . It can be seen that the solution  $\mathbf{c}$  has the biggest values for both  $f_1$  and  $f_2$ . This explains that the solution  $\mathbf{c}$  is dominated by solutions  $\mathbf{a}$  and  $\mathbf{b}$ . In contrast, solutions  $\mathbf{a}$  and  $\mathbf{b}$  are non-dominated solutions, as neither of them dominates each other.

**Definition 2 Pareto optimality** The solution  $\mathbf{x}_1 \in X$  is defined as Pareto-optimal iff:

$$\exists \mathbf{x}_2 \in X : F(\mathbf{x}_2) \succ F(\mathbf{x}_1) \tag{3}$$

where  $X$  is feasible region that is defined as

$$X = \{ \mathbf{x} \in \mathbb{R}^n : g_l(\mathbf{x}) \leq 0 \forall l; h_j(\mathbf{x}) = 0 \forall j; x_i^l \leq x_i \leq x_i^U \forall i \} \tag{4}$$

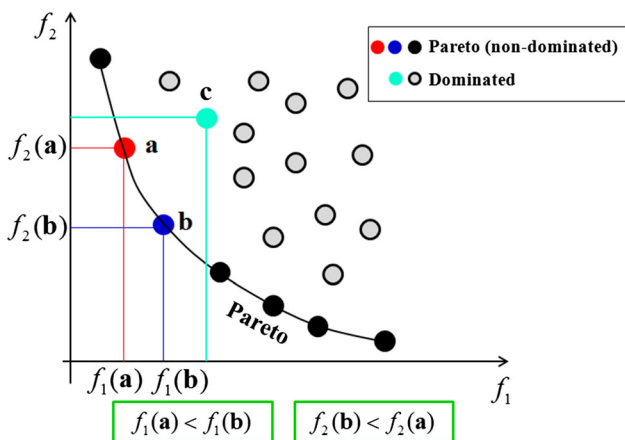


Fig. 1 Pareto solutions  $\mathbf{a}$ ,  $\mathbf{b}$  for the 2-D domain

It is worth mentioning that the set which includes all non-dominated solutions of a problem is named Pareto-optimal set and it is shown as follows:

**Definition 3 Pareto-optimal set** A set all Pareto-optimal solutions is called Pareto set as follows:

$$PS = \{ \mathbf{x}_i : \forall \mathbf{x}_i, \mathbf{x}_j \in X \wedge F(\mathbf{x}_i) \succ F(\mathbf{x}_j) \} \tag{5}$$

**Definition 4 Pareto -optimal front** The set containing the value of objective functions for Pareto solutions set is called Pareto-optimal front:

$$PF = \{ F(\mathbf{x}_i) | \mathbf{x}_i \in PS \} \tag{6}$$

The Pareto-optimal set and the Pareto-optimal front are illustrated for a minimization problem in Fig. 2.

### 2.2 Performance indices

To perform fair quantitative judgments and evaluations among different types of metaheuristic algorithms, some performance indices that are widely employed are investigated to evaluate the performances of metaheuristic algorithms. These performance indices are defined in detail in the following subsections.

#### 2.2.1 Generational distance index

The generational distance (GD) index was proposed by Veldhuizen and Lamont [36] with the aim to clarify the capability of the candidate algorithms for obtaining a set of non-dominated solutions at a lowest distance to the true Pareto-optimal front (PF<sub>O</sub>). Meanwhile, the algorithm that has a minimum value of GD possesses the best convergence to PF<sub>O</sub>. The GD index can be defined mathematically as follows [37]:

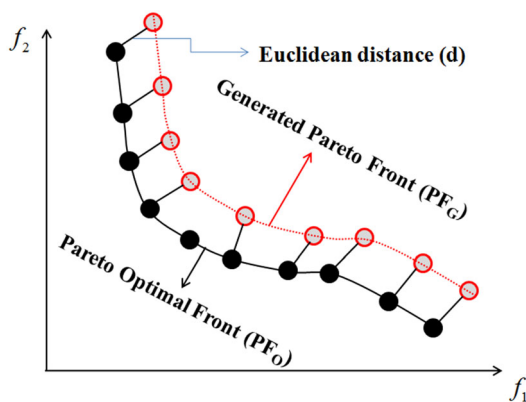
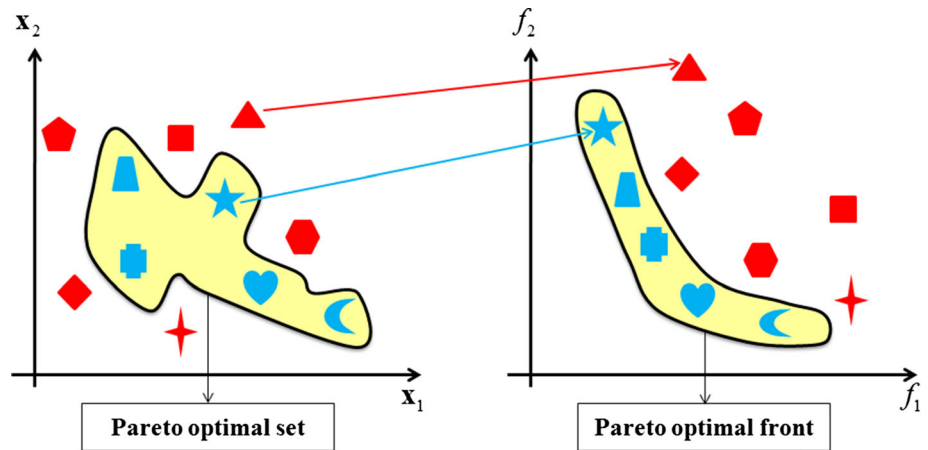
$$GD = \frac{1}{n_{PF}} \left( \sum_{i=1}^{n_{PF}} d_i^2 \right)^{1/2} \tag{7}$$

where  $n_{PF}$  indicates the number of solutions in generated Pareto front (PF<sub>G</sub>) and  $d_i$  denotes the Euclidean distance between  $i$ th member of the PF<sub>G</sub> and the nearest one in PF<sub>O</sub>. Meanwhile, the Euclidean distance ( $d$ ) is determined as follows:

$$d(r, l) = d(l, r) = \left[ \sum_{i=1}^n (f_{il} - f_{ir})^2 \right]^{1/2} \tag{8}$$

where  $l = (f_{1l}, f_{2l}, \dots, f_{nl})$  is a point on PF<sub>G</sub> and  $r = (f_{1r}, f_{2r}, \dots, f_{nr})$  is the nearest member to  $l$  in PF<sub>O</sub>. Figure 3 provides a schematic diagram of this performance index in

**Fig. 2** Decision variable space and objective space of MOP for minimization case



**Fig. 3** Schematic diagram of the generational distance (GD) index for MOPs

the 2D space. The best value for the GD index is equal to zero which occurs when the  $PF_G$  exactly coincides with the  $PF_0$ . On the other hand, the inverted generational distance (IGD) can be defined mathematically as follows [37]:

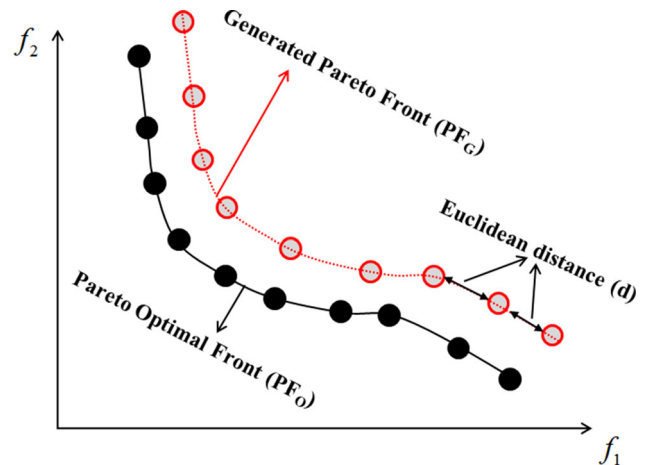
$$IGD = \frac{1}{nt} \left( \sum_{i=1}^m (d'_i)^2 \right)^{1/2} \tag{9}$$

where  $nt$  denotes the number of solutions in the true Pareto-optimal front ( $PF_0$ ) and  $d'_i$  defines the Euclidean distance between the  $i$ th true Pareto-optimal solution and the closest Pareto solution obtained by a certain methodology.

**2.2.2 Metric of spacing**

The metric of spacing ( $S$ ) was developed by Scott [38] to show the distribution among non-dominated solutions which are obtained by a particular algorithm. This criterion can be defined mathematically as follows [6]:

$$S = \sqrt{\frac{1}{n_{PF} - 1} \sum_{i=1}^{n_{PF}} (d_i - \bar{d})^2} \tag{10}$$



**Fig. 4** Schematic diagram of spacing metric ( $S$ ) for MOPs

where  $n_{PF}$  represents the number of solutions in  $PF_G$ , while  $d_i$  denotes the Euclidean distance between  $i$ th member and its nearest one in  $PF_G$ , and  $\bar{d}$  is the average (mean) of all distances. The small value of  $S$  provides a better uniform distribution in  $PF_G$ . If all non-dominated solutions are uniformly distributed in the  $PF_G$ , this implies that  $d_i$  is equal to  $\bar{d}$  and therefore, the value of  $S$  metric equals zero. Figure 4 presents a schematic diagram for the spacing index.

**2.2.3 Spread metric**

The spread metric presents the third performance index which was proposed by Deb [6]. It aims to determine the extent of spread achieved by the non-dominated solutions found from a certain algorithm. To be more precise, this criterion shows how the obtained solutions are extended across  $PF_0$ . This criterion is stated as follows [6]:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n_{pf}} |d_i - \bar{d}|}{d_f + d_l + (n_{pf} - 1) \bar{d}} \tag{11}$$

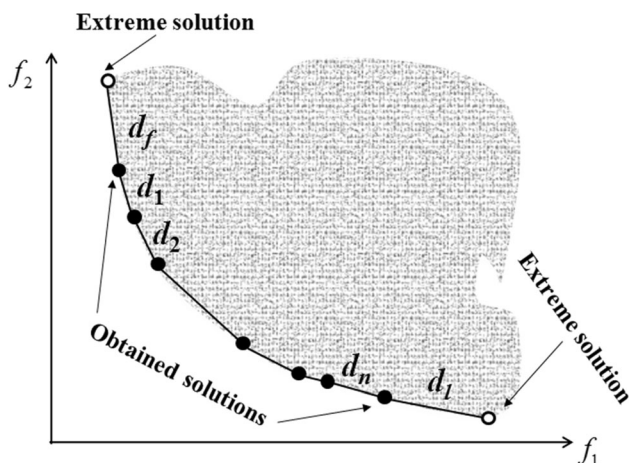


Fig. 5 Schematic diagram of the spread metric ( $\Delta$ ) for MOPs

where  $d_f$  and  $d_i$  represent the Euclidean distances between the extreme solutions in the  $PF_O$  and  $PF_G$ , respectively.  $n_{pf}$  and  $\bar{d}$  are the total number of members in  $PF_G$  and the mean (average) of all distances, respectively. Further,  $d_i$  represents the Euclidean distance between each member in  $PF_G$  and the closest one in  $PF_O$ . Equation (11) shows that the value of  $\Delta$  index is always greater than zero, while the small value of  $\Delta$  means spread of the solutions and well distribution. In this regard, when the value of  $\Delta$  is equal to zero this indicates that the extreme solutions of  $PF_O$  have been found and  $d_i = \bar{d}$  for all non-dominated points. Figure 5 shows a schematic structure of the index  $\Delta$  for a given Pareto front.

### 3 Multi-objective crow search algorithm

This section focuses on extending the classical CSA to develop a multi-objective approach variant. In this context, two subsections are discussed. The first one presents the concepts of classical CSA. The second subsection incorporates multi-objective strategies based on orthogonal opposition improvements to the CSA leading to the multi-objective orthogonal opposition-based crow search algorithm (M2O-CSA).

#### 3.1 Crow search algorithm

Crow search algorithm is introduced by Askarzadeh [35] which mimics the intelligent behaviors of crows. Crows exhibit awesome behaviors during collecting and storing foods. In this sense, crows memorize the positions where they hide exceeded food and during this process two cases may be occur. The first one, a specific crow pilfers the hidden food of another crow when it does not realize that it

is followed by another one. The second one, the crow protects hidden food if it realizes that followed by another crow and therefore it can fool this crow.

From a searching point of view, CSA is a population-based search algorithm with population  $\delta^t = \{\delta_1^t, \delta_2^t, \dots, \delta_N^t\}$  of  $N$  crows (individuals) is evolved during each iteration  $t$ . The  $i$ th crow  $\delta_i$  is represented by a  $n$ -dimensional vector as  $\delta_i = \delta_{i,1}, \delta_{i,2}, \dots, \delta_{i,n}$ , where each of its element (dimension) corresponds to a decision variable of the candidate problem to be solved. According to the two scenarios of crow behavior, the new population  $\delta^{t+1}$  is updated by considering two options whether it is aware (realizes) or not aware (not realizes) being followed by another crow. Thus each new element can be determined as follows:

$$\delta_i^{t+1} = \begin{cases} \delta_i^t + r_i \times fl(\mathbf{mc}_j^t - \delta_i^t) & r_j \geq AP \\ \text{a random position} & \text{otherwise} \end{cases} \quad (12)$$

where the memory of each crow ( $\mathbf{mc}$ ) is updated as follows:

$$\mathbf{mc}_i^{t+1} = \begin{cases} \delta_i^{t+1} & f(\delta_i^{t+1}) > f(\mathbf{mc}_i^t) \\ \mathbf{mc}_i^t & \text{otherwise} \end{cases} \quad (13)$$

where  $r_i$  and  $r_j$  are random numbers between 0 and 1 which are drawn from a uniform distribution,  $fl$  defines the flight length parameter that controls the searching extent,  $\mathbf{mc}_j^t$  denotes the memory of the crow  $j$  at iteration  $t$ ; it saves the best location obtained so far by the crow  $j$ , and  $AP$  is awareness probability of the following. The working steps of the CSA are listed in Algorithm 1.

#### 3.2 The proposed M2O-CSA algorithm

In this section, the CSA is extended to deal with multiple objectives which is named multi-objective orthogonal opposition CSA (M2O-CSA), where some modifications are developed. The first is that an external archive is equipped to store the obtained Pareto-optimal solutions during runs of the proposed algorithm. The second, all crows employ an external archive as the memory of the hoard positions. Finally, a parallel-orthogonal-opposition (P2O) strategy is introduced to enhance the convergence to the true Pareto front as well as acquire its well-spread solutions over the Pareto front.

##### 3.2.1 Updating mechanism

The algorithm starts with the initialization process, and the archive is filled by non-dominated solutions by this process. Afterward, the updating strategy is presented to enhance the convergence of solutions, improve the distribution of the whole Pareto front, and complete the finite archive until it is filled. Also, the redundancy or the

existence of similar solutions is prevented according to the dominance concepts. To perform the updating strategy, one of the memorized solutions in the archive (**Ac**) is selecting by a roulette-wheel mechanism with the following probability for each segment:

$$P_i = \frac{t}{T} \tag{14}$$

where  $t$  and  $T$  define the current iteration and maximum number of iterations, respectively. In this regard, the

removed from the archive. If the archive is full, some solutions are removed using the rule of Coello Coello et al. [39].

### 3.2.2 Orthogonal opposition strategy

The orthogonal opposition strategy (OOS) is a novel phase that combines both the merits of orthogonal arrays and oppositional points. This strategy is incorporated with CSA

**Algorithm 1: Principles of the CSA**

```

1: Input:  $T$  : Maximum iterations;  $N$  : Crows' flock ;  $AP$  :Awareness of crow;  $fl$  :flight length.
2: Initialize : initialize the places (positions) of  $N$  crows randomly within the search space
3: Evaluate the places (positions) of  $N$  crows
4: Engage the memory with the initial places of crows
5    $t = 0$ 
6: While  $t \leq T$  do
7:   for  $i = 1$  to  $N$  do
8:     Extract randomly one crow from the flock to follow it (for example  $k$  )
9:     Generate  $r \in [0, 1]$ 
10:    if  $r < AP$  then
11:       $\delta_i^{t+1} = \delta_i^t + r_i \times fl (\mathbf{mc}_k^t - \delta_i^t)$ 
12:    else
13:       $\delta_i^{t+1} = \text{randomplace}$ 
14:    end
15:  end
16:  Examine the feasibility of new places
17:  Evaluate the new places (positions) of the crows
18:  Update the memory of each crow:  $\mathbf{mc}_i^{t+1} = \begin{cases} \delta_i^{t+1} & f(\delta_i^{t+1}) > f(\mathbf{mc}_i^t) \\ \mathbf{mc}_i^t & \text{otherwise} \end{cases}$ 
19:     $t = t + 1$ 
20: end
21: Output: best position

```

solution is determined as follows:

$$\delta_i^{t+1} = \begin{cases} \delta_i^t + r_i \times fl (\mathbf{Ac}_j^t - \delta_i^t) & r_j \geq P_i^t \\ \text{a random position} & \text{otherwise} \end{cases} \tag{15}$$

Therefore, the crows will be encouraged to update their positions by such archive' solutions. On the other hand, the archive is updated regularly in each iteration and may be fully filled during optimization. In this sense, the archive is managed by dominance criteria such that the solution is prevented from entering if it is dominated by at least one of the archive residences. Also, it should be allowed to enter the archive if it is non-dominated with respect to all of the solutions in the archive, and if the solution dominates some of the Pareto solutions of the archive, they all should be

to solve large-scale MOPS. Therefore, a wider space can be explored during evolution. The OOS is parallelized to perverse the spread and convergence of solutions. This strategy is working by three stages, namely forming the orthogonal arrays-based *crossover*, generating the opposition points, and parallelization. The strategy of the OOS is stated by the pseudo-code as in Algorithm 2.

**3.2.2.1 Orthogonal crossover** In this stage, the orthogonal array of  $N$  factors (variables) through  $G$  levels and  $C$  combinations is often denoted as  $L_C(G^N)$ . For example, if an experiment has four factors (variables) and three levels, then there are 81 combinations for this experiment, but when the orthogonal array is employed, only nine

combinations can be attained ( $L_9(3^4)$ ). The term orthogonal means that each level per factor occurs at the same time for any column.

The orthogonal crossover strategy is introduced based on generating two individuals ( $\mathbf{v}_1, \mathbf{v}_2$ ) randomly, and these individuals are quantified to generate the dynamic third level that involves  $(G^2 - 2)$  combinations indexed from 2 to  $G^2 - 1$  arrays in which the difference among any two successive arrays is the same. The first array (level) is the minimum among the individuals  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , while the last array or level represents the maximum among them. By this strategy,  $C = G \times G$  individuals or combinations are obtained that are incorporated for survival. In this sense, the general idea behind the orthogonal crossover strategy is to provide a set of solutions that are controlled by the dynamic  $G$  levels to maintain a better diversification in the early stages and enhance the intensification of convergence behavior toward the promising regions of the Pareto front at the later stages. Besides, the proposed dynamic orthogonal crossover strategy provides a general intelligent framework to overcome the lack of conventional crossover that generates only two individuals which leads to insufficient population diversity and falling into non-efficient solutions. In this context, the different combinations are obtained as follows:

$$l_{ij} = \begin{cases} \min(\mathbf{v}_{1j}, \mathbf{v}_{2j}), & \text{for } i = 1 \text{ and } 1 \leq j \leq N \\ \min(\mathbf{v}_{1j}, \mathbf{v}_{2j}) + (i - 1) * \left( \frac{|\mathbf{v}_{1j} - \mathbf{v}_{2j}|}{G - 1} \right), & \text{for } 2 \leq i \leq G^2 - 1 \text{ and } 1 \leq j \leq N \\ \max(\mathbf{v}_{1j}, \mathbf{v}_{2j}), & \text{for } i = G \text{ and } 1 \leq j \leq N \end{cases} \tag{16}$$

For large-scale dimensions, the  $d$  (i.e., the number of dimensions) is usually much larger than  $N$ . In such a situation, the different variables are group into  $N$  sub-vectors. Finally,  $C$  arrays are obtained according to the orthogonal

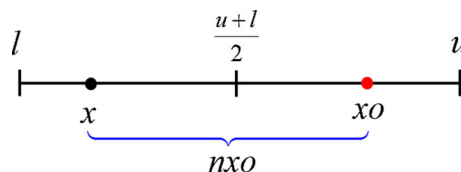


Fig. 6 The point  $(x)$ , its opposite  $(xo)$ , and new opposite  $nxo$

array  $L_C(G^N)$ . In this work,  $L_9(3^4)$  is used to define an orthogonal-based crossover and it is denoted as ObC9.

**3.2.2.2 Oppositional learning** Oppositional learning is proposed by Tizhoosh et al. [40] which aims to accelerate the convergence speeds of optimization algorithms. It operates to consider the opposite of each individual. The concepts of opposition learning are as follows:

**Definition 5** Assume that  $x$  is a real number with bounds  $l$  and  $u$  such that  $x \in [l, u]$ , then its opposite value  $xo$  is obtained as (Fig. 6).

$$xo = l + u - x \tag{17}$$

**Definition 6** Suppose that  $x$  is a real number such that  $x \in [l, u]$ , then its new opposite value  $xo$  is determined as (Fig. 6).

$$nxo = rand(x, xo) \tag{18}$$

where  $xo$  is the opposite of  $x$ .

**Algorithm 2: Orthogonal opposition strategy (OOS)**

1. **Input:** Elicit vectors  $(\mathbf{v}_1, \mathbf{v}_2)$  randomly
2. Split the large-scale dimensions into four subgroups
3. Apply the crossover stage to constitute three levels by Equation (16)
4. Generate the orthogonal arrays according to  $L_9(3^4)$
5. Execute the new oppositional strategy to generate the opposite arrays  $OL_9(3^4)$  of  $L_9(3^4)$  by Eq. (18)
6. **Output:** Recorded the  $L_9(3^4)$  and  $OL_9(3^4)$



### 3.3 Eliciting the satisfactory solution

The multiple objective optimization algorithms yield a set of Pareto-optimal solutions due to the conflict natures among the objectives. The practical situations are concerned with one design point or one operating point for these purposes. Such a design point is called the best (satisfactory) compromise solution. MOORA (multi-objective optimization on the basis of ratio analysis) method has the ability to extract the one option from a set of available options. The idea of MOORA can be stated in a series of steps.

1. Exhibit the performance for  $N$  alternatives over  $K$  criteria.

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & \cdots & f_{1K} \\ f_{21} & f_{22} & \cdots & \cdots & f_{2K} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f_{N1} & f_{N2} & \cdots & \cdots & f_{NK} \end{bmatrix}$$

where  $f_{ij}$  presents the performance measure of the  $i$ th option on  $j$ th attribute or criterion which also  $f_{ij}$  represents the value of the  $i$ th solution at the  $j$ th criterion. The row of the matrix  $F$  represents the values of all  $K$  objective functions for  $i$ th solution.

2. Obtain the normalized form of the decision matrix so that it becomes dimensionless. This normalization or the system ratio is expressed as follows:

$$f_{ij}^* = f_{ij} / \left( \sum_{i=1}^N f_{ij}^2 \right)^{1/2}, \quad j = 1, 2, \dots, K$$

where  $f_{ij}^*$  is a dimensionless value that belongs to the interval  $[0, 1]$  and represents the normalized performance of the  $i$ th option on  $j$ th attribute or criterion.

- (3) Add the obtained normalized performances for maximization cases (for beneficial attributes) and subtract these performances for minimization cases (for non-beneficial attributes). This step is expressed as follows:

$$Y_i = \sum_{j=1}^l f_{ij}^* - \sum_{j=l+1}^K f_{ij}^*$$

where  $Y_i$  denotes the normalized assessment for  $i$ th option (alternative) with considering all the attributes.

The  $Y_i$  may be positive or negative according to the totals of its maxima and minima, respectively.

4. Rank the values of  $Y_i$  to show the final preference.
5. Recommend the highest value of  $Y_i$  as the best alternative.

All the parameters of the M2O-CSA are typical of the original work of the CSA by introducing a new parameter for denoting archive size. After all, the working of the M2O-CSA is exhibited by the pseudo-code as in Algorithm 3 and the flowchart of M2O-CSA is provided in Fig. 7.

## 4 Simulation results

In this section, a set of multi-objective benchmark problems are employed to clarify and validate the performance of the proposed M2O-CSA. These multi-objective problems involve two categories of experiments. The first category is to test our algorithm on a well-known ZDT test suites that have been selected from some of the credible research studies and these suites are listed in Table 1, where the ZDT5 isn't considered in Table 1 because this problem involves a discrete Pareto front and thus it needs a different methodology to generate discrete Pareto front. Also, in this category, we focus on investigating the performance of the proposed M2O-CSA regarding two scales of dimensions, namely,  $n = 30$ , and  $n = 100$ . In the second category, a set of real engineering multi-objective design problems are employed for investigating the performance of the proposed M2O-CSA. Moreover, the M2O-CSA as a modified optimizer is compared with its classical version (i.e., MOCSA [41]) and chaotic version (MOCCSA [41]).

To achieve unbiased results, all algorithms are performed under the same conditions. In this regard, all algorithms are uniformly randomly initialized within the candidate range and the maximum number of iterations is set to 200 and the number of individuals in the population is set to 100 for all algorithms. Also, the number of parallel branches for M2O-CSA is set to 4. Furthermore, in order to mitigate the impact of randomness, all algorithms were run 20 times independently of all categories. The selected parameters of the candidate algorithms are set as in their respective works, where they are: population size, archive size, flight length, awareness probability, and the number of iterations which are 100, 100, 2, 0.1, and 200, respectively. For a fair comparison among MOCSA, MOCCSA and M2O-CSA, the first randomly generated population is used for the first run of MOCSA, MOCCSA and M2O-CSA, the second randomly generated population is used for the second run of MOCSA, MOCCSA and M2O-CSA, and so on.

### 4.1 Results of ZDT test functions

In this subsection, the proposed M2O-CSA is evaluated and compared with the MOCSA and MOCCSA algorithms on the ZDT test functions. The validation is investigated through the different assessment metrics such as (1)

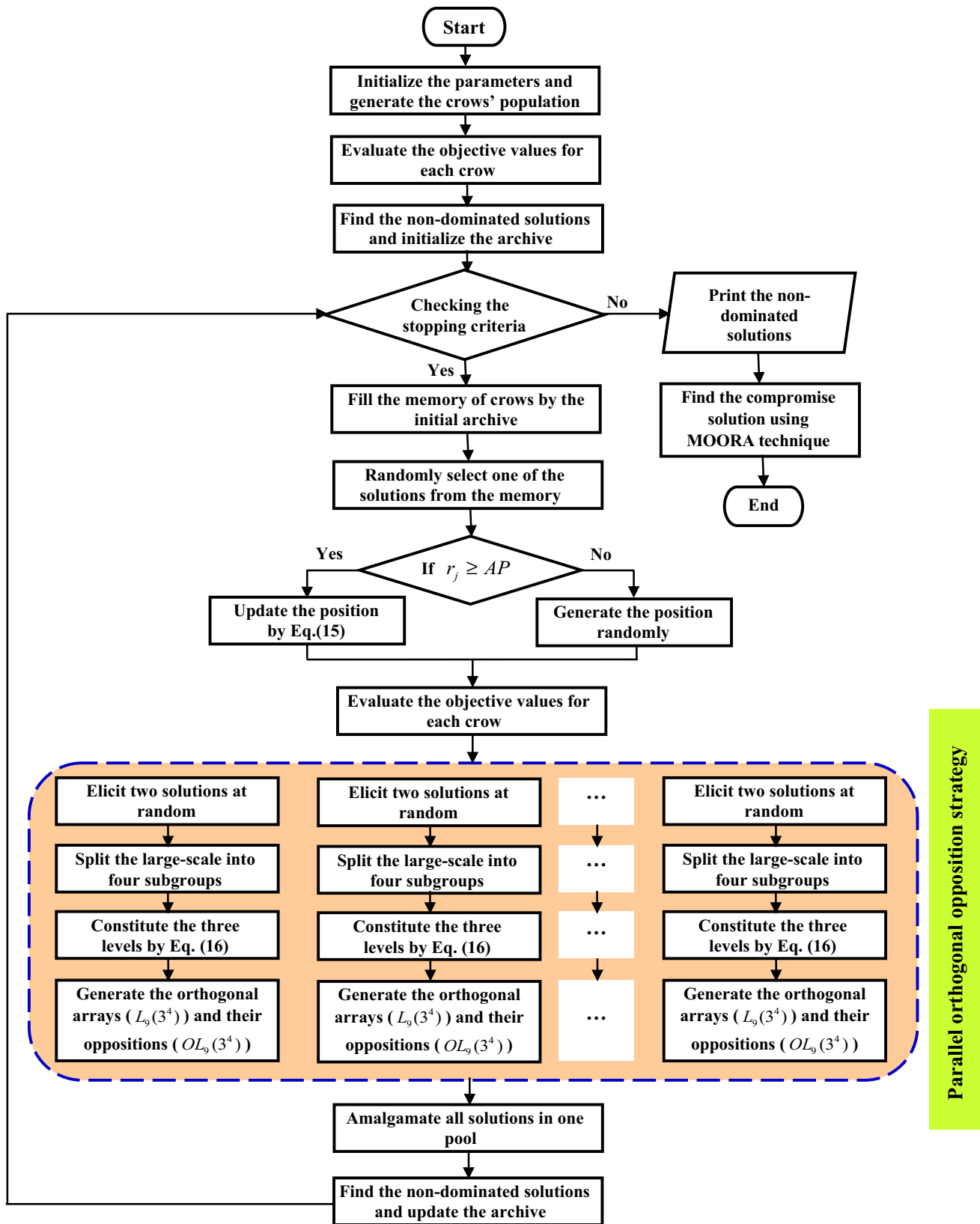


Fig. 7 Flowchart of the proposed M2O-CSA

**Table 1** ZDT test suits (*D*: Dimension)

Problem	Functions' descriptions	Range	<i>D</i>	Optimal front
ZDT1	Minimize: $f_1(\mathbf{x}) = x_1$ Minimize: $f_2(\mathbf{x}) = g(x) * (1 - \sqrt{f_1/g(x)})$ , where : $g(x) = 1 + \frac{9}{n-1} * \sum_{i=2}^n x_i$	[0, 1]	30, 100	$x_1 \in [0, 1]$ , $x_i = 0, i = 2 : n$
ZDT2	Minimize: $f_1(\mathbf{x}) = x_1$ Minimize: $f_2(\mathbf{x}) = g(x) * (1 - (f_1/g(x))^2)$ , where : $g(x) = 1 + \frac{9}{n-1} * \sum_{i=2}^n x_i$	[0, 1]	30, 100	$x_1 \in [0, 1]$ , $x_i = 0, i = 2 : n$
ZDT3	Minimize: $f_1(\mathbf{x}) = x_1$ Minimize: $f_2(\mathbf{x}) = g(x) * (1 - \sqrt{f_1/g(x)} - (f_1/g(x)) \sin(10\pi x_1))$ , where : $g(x) = 1 + \frac{9}{n-1} * \sum_{i=2}^n x_i$	[0, 1]	30, 100	$x_1 \in [0, 1]$ , $x_i = 0, i = 2 : n$
ZDT4	Minimize: $f_1(\mathbf{x}) = x_1$ Minimize: $f_2(\mathbf{x}) = g(x) * (1 - \sqrt{f_1/g(x)})$ , where : $g(x) = 1 + \left(10(n-1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)]\right)$	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ , $i = 2 : n$	30, 100	$x_1 \in [0, 1]$ , $x_i = 0, i = 2 : n$
ZDT6	Minimize: $f_1(\mathbf{x}) = 1 - \exp(-4x_1) * \sin^6(6\pi x_1)$ Minimize: $f_2(\mathbf{x}) = g(x) * (1 - (f_1/g(x))^2)$ , where : $g(x) = 1 + 9 \left(\frac{1}{n-1} * \sum_{i=2}^n x_i\right)^{0.25}$	[0, 1]	30, 100	$x_1 \in [0, 1]$ , $x_i = 0, i = 2 : n$

generational distance (GD), (2) inverted generational distance (IGD), (3) spacing (SP), and (4) the maximum spread (MS). The proposed M2O-CSA is investigated through two experiments. Experiment 1 employs thirty variables ( $n = 30$ ), while, experiment 2 employs a hundred variables ( $n = 100$ ).

In experiment 1, the assessment metrics (i.e., GD, IGD, SP, and MS) are obtained and reported in Tables 2, 3, 4, and 5. The obtained results regarding these metrics show that the M2O-CSA outperforms the compared versions, namely MOCSA and MOCCSA for all ZDT test suits with  $n = 30$ . The superiority is demonstrated by the mean values, where the reported values show a higher accuracy of M2O-CSA compared to the other algorithms. Also, the

shape of the obtained Pareto-optimal front by the proposed M2O-CSA and the compared ones on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 are portrayed in Fig. 8. Inspecting this figure, it is noted that MOCCSA exhibits the poorest convergence, while the proposed M2O-CSA provides a good convergence and useful diversity toward the true Pareto fronts on all ZDT test suits. On the other hand, the MOCSA provides good convergence for ZDT2 and ZDT6. The most interesting features behind the superior performance of the proposed M2O-CSA are contained in parallel-orthogonal-opposition (P2O) strategy that preserves the diversity of solutions and enhances the convergence toward the true Pareto fronts.

**Algorithm 3: M2O-CSA pseudo code**

1. **Input:**  $T$  =Maximum number of iterations;  $N$  = flock size;  $fl$  =flight length.
2. Initialize the flock of crows randomly
3. Evaluate the multiple objectives of each crow
4. Apply the dominance concepts to obtain the initial non-dominated solutions
5. Fill the archive with the initial non-dominated solutions
6.  $t = 0$
7. **Repeat**
8.     **for** each crow
7.         **if**  $rand < P_i$
8.             Elicit one solution from the archive according to roulette wheel aspect
9.             Move the crow to new location using Eq. (15)
10.         **else**
11.             Generate a new solution randomly
12.         **end**
13.     **end**
14. Evaluate the multiple objectives of each crow
15.     **for** each parallel branch
16.         Elicit vectors  $(\mathbf{v}_1, \mathbf{v}_2)$  randomly
17.         Split the large-scale dimensions into four subgroups
18.         Apply the crossover stage to constitute three levels by Equation (16)
19.         Generate the orthogonal arrays according to  $L_9(3^4)$
20.         Generate  $OL_9(3^4)$  of  $L_9(3^4)$
21.         Evaluate the multiple objectives the  $OL_9(3^4)$  of  $L_9(3^4)$
22.         Store the non-dominated among  $L_9(3^4)$  and  $OL_9(3^4)$
23.     **end**
24. Store the non-dominated solutions among  $\delta$ ,  $L_9(3^4)$  and  $OL_9(3^4)$
25. Update the archive (**Ac**) with the obtained non-dominated solutions
26.  $t = t + 1$
27. **Until**  $t \geq T$
28. **Output:** archive

**Table 2** Statistical results for GD on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 30 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.440022	0.328264	0.248344	0.157755	0.895094	0.46616	0.57907	0.33286	0.004793	0.87973
MOCCSA	2.255486	2.241715	0.138	2.047034	2.527346	3.31984	3.26807	0.121017	3.209934	3.54827
M2O-CSA	0.000308	0.000239	0.000211	8.55E−05	0.00064	4.72E−05	4.72E−05	4.38E−06	4.41E−05	5.03E−05
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.443457	0.425613	0.220698	0.08474	0.739996	14.8949	15.43874	12.9013	0.464534	37.68461
MOCCSA	2.153157	2.149536	0.164283	1.837869	2.314501	80.6278	74.95245	18.28148	63.13257	125.334
M2O-CSA	3.08E−03	3.00E−03	4.60E−04	2.58E−03	3.88E−03	3.72E−04	4.80E−04	2.30E−04	1.08E−04	5.27E−04
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	1.250934	0.098801	2.127796	0.01975	5.2473					
MOCCSA	6.006829	6.050078	0.206774	5.55698	6.275276					
M2O-CSA	8.08E−02	3.57E−04	1.70E−01	2.72E−04	4.37E−01					

**Table 3** Statistical results for IGD on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 30 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.425131	0.320903	0.230445	0.156945	0.841729	0.506639	0.629995	0.362325	0.006922	0.959283
MOCCSA	1.953679	1.918242	0.095759	1.848102	2.115394	3.236088	3.204521	0.114286	3.046126	3.405229
M2O-CSA	0.000101	9.96E−05	1.30E−05	8.50E−05	0.000123	9.33E−05	9.33E−05	2.04E−06	9.18E−05	9.47E−05
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.245108	0.243769	0.043168	0.191951	0.302796	4.069727	0.783467	6.939034	0.284181	22.37024
MOCCSA	1.304741	1.369996	0.192513	0.842974	1.488893	19.60672	21.07413	5.505553	11.54718	27.01433
M2O-CSA	9.56E−03	9.57E−03	4.03E−05	9.50E−03	9.61E−03	9.91E−05	9.40E−05	1.04E−05	9.22E−05	1.11E−04
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	1.185574	0.003905	2.49198	0.002299	5.913776					
MOCCSA	6.442483	6.474644	0.194266	6.055729	6.740418					
M2O-CSA	1.08E−04	9.60E−05	3.25E−05	7.14E−05	1.66E−04					

In experiment 2, the behavior of the proposed M2O-CSA is further investigated and compared with the MOCSA and MOCCSA on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 test suits with  $n = 100$  to analyze their

scalability. The statistical measures of the assessment metrics include the best, mean, worst values, and standard deviation (SD) are obtained and listed in Tables 6, 7, 8, and 9. The reported results provide that the proposed M2O-

**Table 4** Statistical results for SP on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 30 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.08995	0.080937	0.024638	0.065584	0.144665	0.11547	0.138093	0.032189	0.076346	0.140155
MOCCSA	0.400523	0.418763	0.09661	0.230749	0.541592	0.191213	0.205418	0.069586	0.056327	0.274667
M2O-CSA	0.077091	0.076677	0.008803	0.062098	0.091813	0.079145	0.079145	0.005304	0.075395	0.082896
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.295456	0.285195	0.031873	0.262202	0.355798	7.207835	8.830306	5.126228	0.08122	13.72118
MOCCSA	0.435521	0.441755	0.088688	0.261037	0.579891	33.59741	33.75935	10.59575	15.79048	51.62622
M2O-CSA	0.246764	0.241037	0.028862	0.202053	0.295544	0.072254	0.071163	0.006025	0.066848	0.07875
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	0.254635	0.143016	0.266595	0.070052	0.77021					
MOCCSA	0.169937	0.17807	0.059717	0.064676	0.241464					
M2O-CSA	0.154685	0.073659	0.175822	0.061163	0.53179					

**Table 5** Statistical results for MS on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 30 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	1.639651	1.600426	0.129396	1.485744	1.847334	1.218761	1.169757	0.114077	1.105779	1.401842
MOCCSA	3.333086	3.310255	0.194899	3.02852	3.640172	1.826214	1.822699	0.545682	0.910689	2.610752
M2O-CSA	0.983795	0.987311	0.012179	0.956188	0.996147	0.997497	0.997497	0.002747	0.995555	0.999439
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	2.265764	2.280908	0.099991	2.077011	2.376677	26.32447	29.27306	19.18476	0.766169	54.10424
MOCCSA	3.820768	3.746044	0.487154	3.276426	4.660992	139.3872	140.308	24.10723	106.3494	177.5755
M2O-CSA	1.379455	1.379417	0.009072	1.361235	1.390662	0.992575	0.994707	0.006172	0.98562	0.997399
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	2.853867	1.168652	2.94905	0.741725	8.597349					
MOCCSA	1.650576	1.665099	0.686172	0.81896	2.937119					
M2O-CSA	1.366481	0.826164	1.148256	0.819532	3.774333					

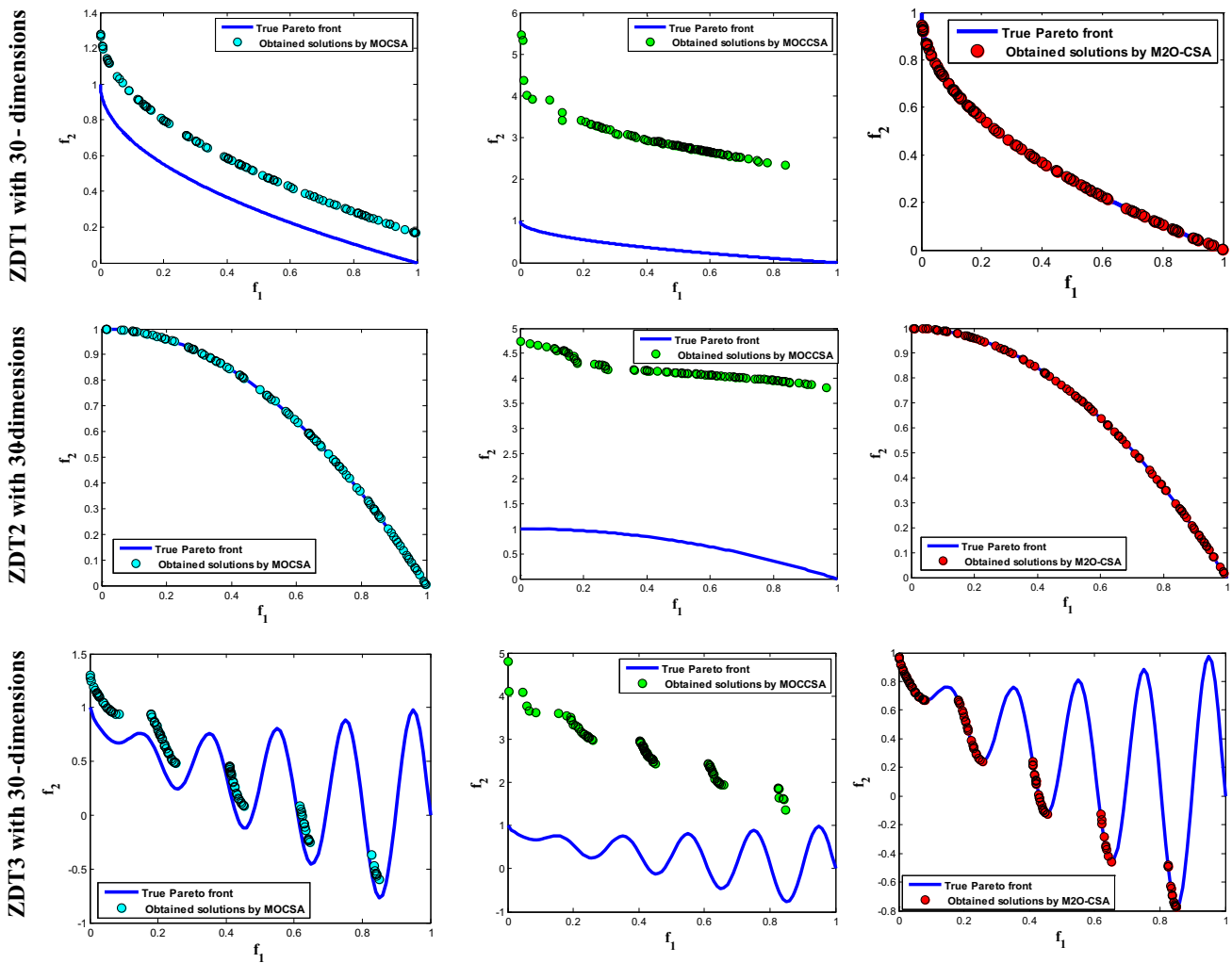


Fig. 8 Obtained Pareto front by MOCSA, MOCCSA, and M2O-CSA on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 30 dimensions

CSA achieves better results of GD, IGD, SP, and MS in terms of mean values under increasing the dimensionality. Thus, we can conclude that the proposed M2O-CSA is insensitive to grow the dimensionality. Compared to the MOCSA and MOCCSA algorithms, the proposed M2O-CSA gives better results for all ZDT test suits. Furthermore, the Pareto front shapes obtained by the proposed M2O-CSA and the other algorithms are depicted in Fig. 9. Based on Fig. 9, it is noted that M2O-CSA provides the boost convergence in finding Pareto-optimal fronts. The compared algorithms suffer from achieving the Pareto-optimal fronts. The superior performance regarding the proposed M2O-CSA is due to the parallel-orthogonal-opposition (P2O) strategy that helps in exploring more regions of the search space and focuses on approaching the true Pareto fronts.

### 4.2 Comparison with state-of-the-art optimization algorithms

In this subsection, the proposed M2O-CSA is further verified and compared against some recent state-of-the-art algorithms: MOGOA [33], MOALO [33], MOPSO [33], NSGA-II [33], MOLAPO [32], MOGWO [32], HEIA [31], MOEA/D-DRA [31], FRRMAB [31], EAG [31], BCE [31], and EF-PD [31]. The performance evaluations of these algorithms are performed with four assessment criteria including GD, IGD, SP, and MS, where lower GD, IGD, and SP values indicate better performance, while the high MS value defines the better performance of the candidate method. For the mentioned algorithms, the results are reported in terms of mean and standard deviation (St. dev) values. Table 10 provides the comparisons among the

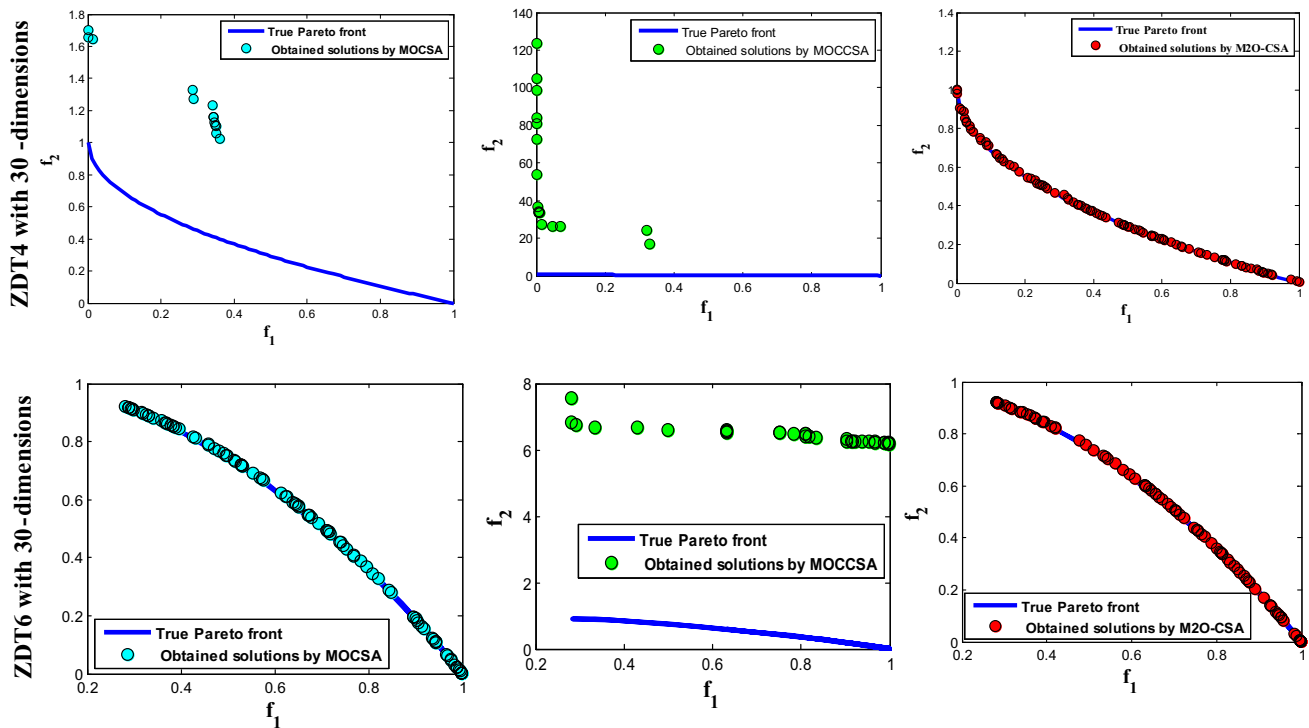


Fig. 8 continued

proposed M2O-CSA and the comparative ones, M2O-CSA MOGOA, MOALO, MOPSO, NSGA-II, MOLAPO, and MOGWO, where the comparisons show that the M2O-CSA outperforms the other comparative algorithms for GD and MS metrics for most ZDT problems and it can provide competitive results in terms of SP metric. In terms of the IGD, Table 10 shows that the proposed M2O-CSA affirms its superiority over the HEIA, MOEA/D-DRA, FRRMAB, EAG, BCE, and EF-PD algorithms for all ZDT problems. Also, the rank for each algorithm at each ZDT problem is provided and the average rank (AR) regarding all ZDT problems is provided. Based on the results of average rank, the presented algorithms follow this order based on GD metric:

M2O-CSA > MOLAPO > MOGOA > MOALO > MOGWO > MOPSO > NSGA-II, M2O-CSA > NSGA-II > MOLAPO > MOGWO > MOPSO > MOALO > MOGOA in terms of MS metric, and follow this order for SP metric: MOLAPO > M2O-CSA  $\approx$  MOGWO > MOGOA > MOALO > MOPSO > NSGA-II. Also in terms of IGD metric, the following order is conducted: M2O-CSA > EF-PD > HEIA > BCE > MOEA/D-DRA  $\approx$  EAG > FRRMAB. According to the assessment criteria, the M2O-CSA has a superior performance in terms of GD and MS metrics, and a strong

competitive edge based on SP metric. Here, the symbol > is to indicate that a certain algorithm better than another, while  $\approx$  is to indicate similar results for both competitors in terms of average rank.

### 4.3 Results of multi-objective constrained problems (MOCPs)

In this section, a set of well-known multi-objective constrained optimization problems are employed to evaluate and clarify the performance of the proposed M2O-CSA. The constrained test suits problems are named as follows: SRN, CONSTR, BNH, KIT, and TNK. These problems have diverse features of Pareto fronts such as continuous convex and discrete natures. The details of the MOCPs are shown in Table 11.

To compare the performance of the proposed M2O-CSA against the other ones, the assessment metrics including GD and IGD are utilized to quantify and compare the performances in terms of convergence and coverage. These metrics are obtained and reported in Tables 12 and 13 for the proposed M2O-CSA and the compared ones. Based on the obtained results, we can note that the M2O-CSA outperforms the other algorithms in terms of means for GD



**Table 6** Statistical results for GD on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 100 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	1.101661	1.131581	1.85E−01	0.808119	1.321551	2.299403	2.318409	2.99E−01	1.860917	2.682559
MOCCSA	2.668672	2.690509	9.58E−02	2.500291	2.823633	3.873993	3.905588	1.25E−01	3.661894	4.041318
M2O-CSA	0.000268	0.000172	0.000171	0.000133	0.000529	0.001295	0.001472	0.000619	0.000607	0.001805
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	1.268118	1.201477	3.23E−01	0.828948	1.652988	1.649261	1.452268	4.73E−01	1.306326	2.189188
MOCCSA	2.616507	2.555031	1.57E−01	2.430174	2.875724	2.355743	1.899442	1.10E+00	1.625925	3.998164
M2O-CSA	0.002851	0.002865	0.000259	0.00257	0.003237	0.000172	0.000172	5.44E−05	0.000134	0.000211
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	6.52502	6.48634	1.83E−01	6.354613	6.759463					
MOCCSA	6.655139	6.638391	1.84E−01	6.408671	6.856911					
M2O-CSA	0.081744	0.000328	0.182067	0.000291	0.407435					

**Table 7** Statistical results for IGD on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 100 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	1.02E−02	1.06E−02	1.82E−03	7.22E−03	1.21E−02	2.47E−02	2.49E−02	3.13E−03	2.01E−02	2.87E−02
MOCCSA	2.46E−02	2.47E−02	5.06E−04	2.38E−02	2.53E−02	3.91E−02	3.91E−02	1.42E−03	3.60E−02	4.12E−02
M2O-CSA	9.97E−05	0.0001	9.72E−06	8.57E−05	0.000111	9.33E−05	9.01E−05	7.66E−06	8.78E−05	1.02E−04
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	2.18E−02	1.92E−02	4.77E−03	1.73E−02	2.78E−02	1.50E−02	1.55E−02	1.31E−03	1.35E−02	1.60E−02
MOCCSA	5.74E−02	5.74E−02	3.65E−03	5.01E−02	6.25E−02	1.74E−02	1.70E−02	2.95E−03	1.44E−02	2.13E−02
M2O-CSA	9.61E−03	0.009593	4.41E−05	9.57E−03	0.009675	9.46E−05	9.46E−05	2.12E−06	9.31E−05	9.61E−05
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	7.23E−02	7.19E−02	1.90E−03	7.06E−02	7.48E−02					
MOCCSA	7.35E−02	7.33E−02	1.82E−03	7.12E−02	7.58E−02					
M2O-CSA	9.93E−05	0.000103	3.50E−05	5.70E−05	0.000137					

and IGD metrics. In addition, the Pareto-optimal fronts for all algorithms are depicted and shown in Fig. 10. It is noted that the proposed M2O-CSA provides high coverage toward the true Pareto fronts for all multi-objective

constrained test suits. The results show that the proposed M2O-CSA managed to achieve the Pareto fronts successfully for all multi-objective constrained problems.

**Table 8** Statistical results for SP on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 100 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.164975	0.159783	0.022454	0.140636	0.197018	0.183083	0.19615	0.030556	0.142103	0.213283
MOCCSA	0.31113	0.310176	0.083744	0.199198	0.423325	0.134581	0.12706	0.07726	0.020284	0.258232
M2O-CSA	0.082504	0.083094	0.002923	0.078227	0.085732	0.068091	0.064761	0.015171	0.054861	0.08465
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.400808	0.366387	0.067837	0.332685	0.489705	0.270997	0.252905	0.049712	0.232864	0.327221
MOCCSA	0.417804	0.427419	0.085503	0.309344	0.558356	0.315281	0.255026	0.137204	0.230704	0.520367
M2O-CSA	0.230272	0.232538	0.028177	0.190255	0.258216	0.06442	0.06442	0.014459	0.054195	0.074644
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	0.216646	0.217595	0.020525	0.188602	0.245112					
MOCCSA	0.173067	0.162902	0.039285	0.121247	0.244444					
M2O-CSA	0.171633	0.069821	0.230163	0.053281	0.582852					

**Table 9** Statistical results for MS on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 100 dimensions

Method	ZTD1					ZTD2				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	1.374971	1.379563	0.063469	1.28346	1.462082	0.731866	0.731015	0.012403	0.714519	0.747711
MOCCSA	2.027612	2.086932	0.202502	1.548546	2.206083	0.791625	0.790086	0.202621	0.546052	1.201445
M2O-CSA	0.99444	0.993853	0.004918	0.986985	0.999471	0.990139	0.990019	0.000773	0.989433	0.990964
Method	ZTD3					ZTD4				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	1.787694	1.777542	0.12239	1.660037	1.913739	1.586838	1.624982	0.1768	1.394079	1.741453
MOCCSA	2.460545	2.494443	0.172447	2.205154	2.672856	1.660605	1.559032	0.244915	1.499667	2.02469
M2O-CSA	1.368546	1.374133	0.012954	1.354012	1.381514	0.987574	0.987574	0.005226	0.983879	0.99127
Method	ZTD6									
	Mean	Median	SD	Best	Worst					
MOCSA	0.986023	1.095015	0.275789	0.51511	1.178402					
MOCCSA	0.658219	0.58485	0.148562	0.546531	0.905849					
M2O-CSA	1.374337	0.826126	1.226659	0.824412	3.568651					

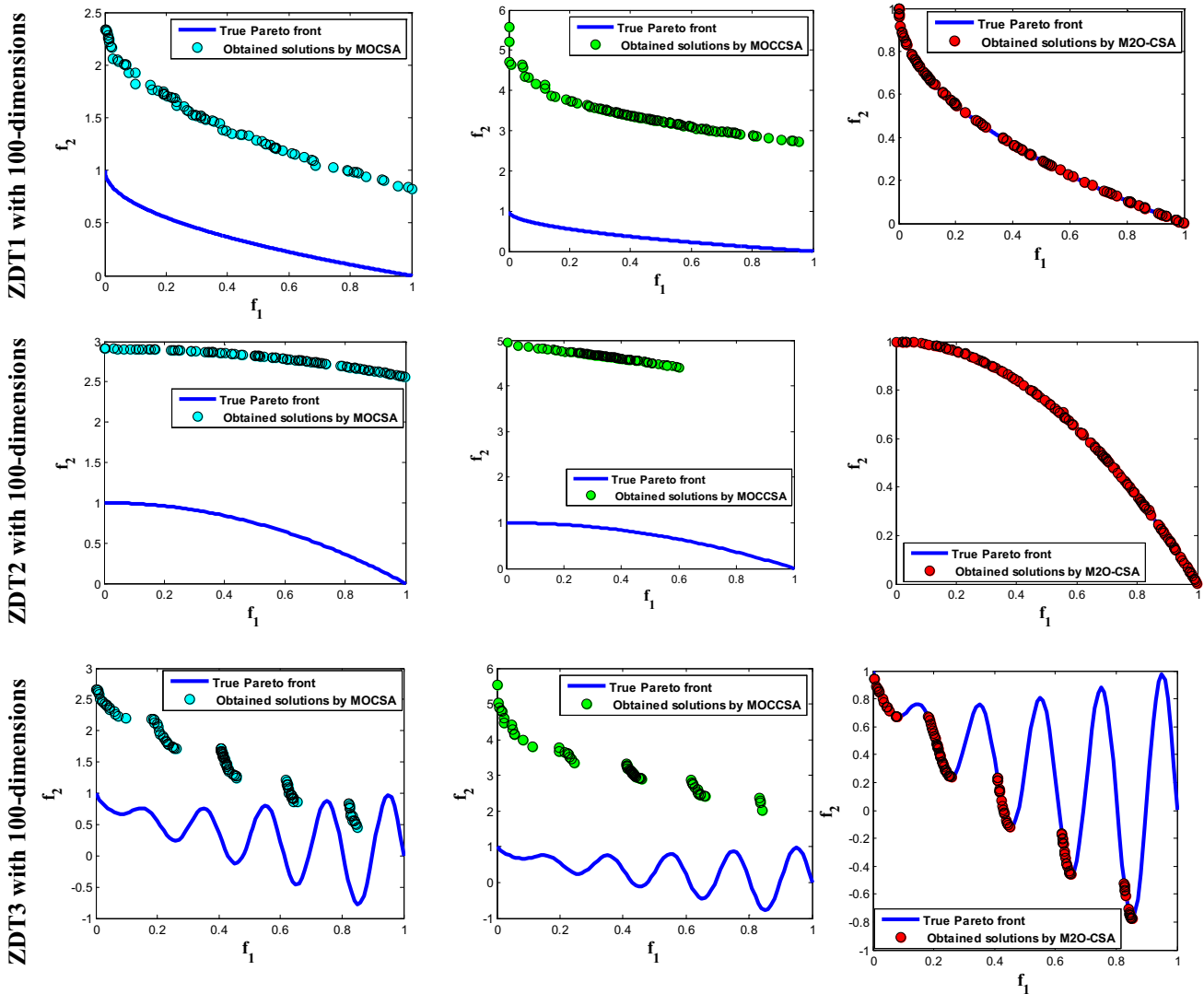


Fig. 9 Obtained Pareto front by MOCSA, MOCCSA, and M2O-CSA on ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 for 100 dimensions

#### 4.4 Results of multi-objective constrained engineering designs (MOCEDs)

In this subsection, M2O-CSA has experimented on four multi-objective constrained engineering designs (MOCEDs) which are popular in the engineering design field. The engineering design suits are as follows: four bar truss (FBT) design, welded beam (WB) design, disk brake (DB) design and speed reduced (SR) design, where they involve different characteristics. The details for all engineering designs are provided in Table 14, where their figures are appended in Fig. 11. The proposed M2O-CSA is compared with MOCSA and MOCCSA algorithms for all engineering design suits. All engineering design suits have

experimented through a population size of 60, 200 for iterations, archive size of 100, flight length of 2, and 0.1 for awareness probability.

For design suits, 100 Pareto-optimal points are generated. In this context, a new methodology-based MOORA technique is employed to help the designer for extracting the operating point as the best compromise or satisfactory solution to execute the candidate engineering design.

The Pareto fronts for all engineering designs are presented in Fig. 12, where the proposed M2O-CSA provides spread Pareto fronts for all designs and also can dominate the most points of the compared algorithms. On the other hand, MOCSA and MOCCSA provide the poorest convergence of Pareto fronts. As the designers or engineers

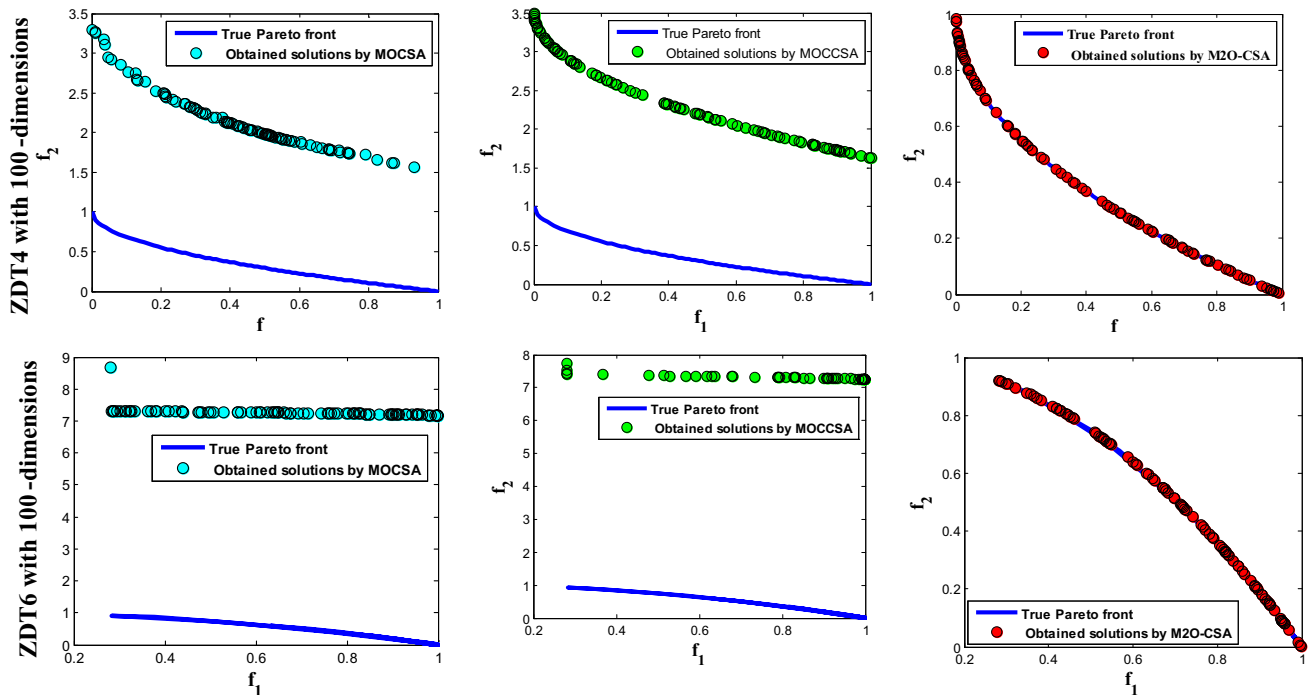


Fig. 9 continued

concern and realize the carrying out of the design tasks, MOORA technique is introduced as a decision making tool to acquire the operating point of the candidate design. This operating point is denoted as the compromise solution that is elicited from the obtained Pareto front based on MOORA technique, where the results of the compromise solutions for all designs are reported in Table 15. Based on the obtained results, we can see that the proposed M2O-CSA gives superior solutions in terms of the constraints violations compared with the MOCSA and MOCCSA.

#### 4.5 Discussion

The convergence and coverage of the Pareto fronts obtained by the proposed M2O-CSA are shown based on its benefits of qualitative and quantitative results. The convergence of M2O-CSA is guaranteed by implanting the orthogonal-based crossover strategy of solutions, while the coverage is emphasized through adopting the opposition strategy. In this context, the proposed method is applied on different tests such as ZDT test suits, some constrained

problems such as SRN, CONSTR, BNH, KIT, TNK problems and some of the real applications such as FBT, WB, DB, and SR designs. The comparisons among the different algorithms prove that the M2O-CSA algorithm performs the test suits well and does not suffer from attaining the Pareto fronts, while the compared algorithms suffer from achieving the Pareto solutions. Furthermore, visualizing the Pareto fronts for engineering design problems can hesitate inexperienced designer to perform them, so the MOORA technique is employed to extract the compromise solution of the candidate design. Therefore, it helps and facilitates the mission of designs for engineering fields. Finally, the feasibility of employing the proposed M2O-CSA methodology for solving the multi-objective design problems has been practically approved, where it was employed to study four real-world optimization problems related to computational design fields that involve multiple objectives optimization. These designs include four bar truss design, welded beam design, disk brake design, and speed reduced design. The recorded simulations show that the M2O-CSA provides a powerful assistance for designing

**Table 10** Statistical results of by M2O-CSA and state-of-the-art algorithms

Algorithm	Statistics	GD			AR			SP			AR						
		ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	AR
MOGOA	Mean	0.04	0.06	0.09	0.34	0.19	0.14	0.17	0.21	0.51	0.25						
	SD	0.01	0.02	0.03	0.13	0.06	0.03	0.03	0.11	0.32	0.19						
	Rank	4	4	3	3	2	3.2	4	5	2	2	3.4					3.4
MOALO	Mean	0.06	0.09	0.08	0.32	0.21	0.17	0.23	0.24	0.53	0.34						
	SD	0.02	0.04	0.03	0.11	0.05	0.04	0.09	0.14	0.38	0.13						
	Rank	5	5	2	2	3	3.4	5	6	3	3	4.4					4.4
MOPSO	Mean	0.09	0.16	0.09	0.42	0.25	0.26	0.31	0.20	0.62	0.46						
	SD	0.05	0.08	0.04	0.17	0.12	0.15	0.11	0.13	0.34	0.23						
	Rank	6	6	3	5	4	4.8	6	4	5	4	5					5
NSGA-II	Mean	0.21	0.23	0.31	0.37	0.32	0.45	0.56	0.34	0.60	0.57						
	SD	0.13	0.12	0.09	0.16	0.19	0.21	0.20	0.15	0.32	0.23						
	Rank	7	7	6	3	5	5.6	7	7	4	5	6					6
MOLAPO	Mean	<b>2.49E-4</b>	2.80E-4	1.24E-1	N/A	N/A	<b>4.71E-3</b>	5.06E-3	<b>8.67E-2</b>	N/A	N/A						
	SD	8.80E-6	3.70E-5	1.55E-2	N/A	N/A	1.80E-4	4.30E-4	3.96E-2	N/A	N/A						
	Rank	<b>1</b>	2	4	N/A	N/A	<b>1</b>	2	1	N/A	N/A	<b>1.3</b>					
MOGWO	Mean	1.58E-2	2.45E-2	1.42E-1	N/A	N/A	0.01472	<b>2.55E-3</b>	3.15E-1	N/A	N/A						
	SD	2.52E-3	5.45E-3	9.36E-3	N/A	N/A	0.00751	7.56E-3	8.74E-2	N/A	N/A						
	Rank	3	3	5	N/A	N/A	2	<b>1</b>	3	N/A	N/A	<b>2</b>					2
<b>M2O-CSA</b>	Mean	3.08E-4	<b>4.72E-5</b>	<b>3.08E-3</b>	<b>3.72E-4</b>	<b>8.08E-2</b>	7.71E-2	7.91E-2	2.46E-1	<b>7.22E-2</b>	<b>1.54E-1</b>						
	SD	2.11E-4	4.38E-6	4.60E-4	2.30E-4	1.70E-1	8.80E-3	5.30E-3	2.88E-2	6.03E-3	1.75E-1						
	Rank	2	<b>1</b>	1	1	1	<b>1.2</b>	3	2	1	1	<b>2</b>					2
MS																	
AR																	
MOGOA	Mean	0.15	0.31	0.30	0.51	0.49											
	SD	0.09	0.15	0.21	0.25	0.21	0.21	0.25	0.21	0.25	0.21						
	Rank	7	6	7	5	5	5	5	6	5	6						
MOALO	Mean	0.23	0.26	0.34	0.67	0.82											
	SD	0.12	0.13	0.16	0.37	0.34	0.34	0.37	0.34	0.34	0.34						
	Rank	6	7	6	3	3	5	3	5	3	5						
MOPSO	Mean	0.34	0.42	0.36	0.61	0.67											
	SD	0.16	0.16	0.20	0.23	0.25	0.25	0.23	0.25	0.25	0.25						
	Rank	5	5	5	4	4	4.6	4	4	4	4.6						
NSGA-II	Mean	<b>1.1</b>	0.85	0.89	1.2	1.2											
	SD	0.54	0.41	0.35	0.36	0.46	0.46	0.36	0.46	0.46	0.46						
	Rank	<b>1</b>	2	2	2	2	1.8	2	2	2	1.8						

Table 10 (continued)

Algorithm	Statistics	MS						AR					
		ZDT1		ZDT2		ZDT3		ZDT4		ZDT6		AR	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
MOLAPO	Mean	0.74	0.70	0.87	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	SD	0.02	0.08	0.03	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	Rank	3	3	3	3	3	3	3	3	3	3	3	
MOGWO	Mean	0.54	0.45	0.68	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	SD	0.17	0.01	0.09	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	Rank	4	4	4	4	4	4	4	4	4	4	4	
M2O-CSA	Mean	0.98	<b>0.99</b>	<b>1.37</b>	<b>0.99</b>	<b>1.36</b>	<b>1.36</b>	<b>1.36</b>	<b>1.36</b>	<b>1.36</b>	<b>1.36</b>	<b>1.36</b>	
	SD	1.21E-2	2.74E-3	9.07E-3	6.17E-3	1.14	1.14	1.14	1.14	1.14	1.14	1.14	
	Rank	2	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1.2</b>	
HEIA	Mean	3.9E-3	4.0E-3	4.4E-3	4.0E-3	4.4E-3	4.4E-3	3.9E-3	3.9E-3	2.9E-3	2.9E-3	2.9E-3	
	SD	0.084E-3	0.07E-3	0.068E-3	0.07E-3	0.068E-3	0.068E-3	0.19E-3	0.19E-3	0.19E-3	0.19E-3	0.19E-3	
	Rank	3	3	3	3	3	3	3	3	4	4	3.2	
MOEA/D-DRA	Mean	7.4E-3	1.3E-2	1.6E-2	1.3E-2	1.6E-2	1.6E-2	4.9E-2	4.9E-2	2.4E-3	2.4E-3	2.4E-3	
	SD	3.0E-3	0.32E-2	0.46E-2	0.32E-2	0.46E-2	0.46E-2	1.1E-2	1.1E-2	0.029E-3	0.029E-3	0.029E-3	
	Rank	6	6	6	6	6	6	6	6	2	2	5.2	
FRRMAB	Mean	3.7E-2	3.1E-2	7.7E-2	3.1E-2	7.7E-2	7.7E-2	3.1E-1	3.1E-1	2.4E-3	2.4E-3	2.4E-3	
	SD	1.0E-2	0.91E-2	3.7E-2	0.91E-2	3.7E-2	3.7E-2	4.0E-1	4.0E-1	0.034E-3	0.034E-3	0.034E-3	
	Rank	7	7	7	7	7	7	7	7	3	3	6.2	
EAG	Mean	5.5E-3	4.9E-3	5.2E-3	4.9E-3	5.2E-3	5.2E-3	5.3E-3	5.3E-3	3.9E-3	3.9E-3	3.9E-3	
	SD	0.59E-3	0.24E-3	0.26E-3	0.24E-3	0.26E-3	0.26E-3	0.87E-3	0.87E-3	0.78E-3	0.78E-3	0.78E-3	
	Rank	5	5	5	5	5	5	5	5	6	6	5.2	
BCE	Mean	4.3E-3	4.2E-3	4.7E-3	4.2E-3	4.7E-3	4.7E-3	7.5E-3	7.5E-3	1.2E-2	1.2E-2	1.2E-2	
	SD	0.12E-3	0.24E-3	0.074E-3	0.24E-3	0.074E-3	0.074E-3	3.0E-3	3.0E-3	0.2E-2	0.2E-2	0.2E-2	
	Rank	4	4	4	4	4	4	4	4	7	7	4.6	
EF-PD	Mean	3.8E-3	3.8E-3	4.4E-3	3.8E-3	4.4E-3	4.4E-3	3.8E-3	3.8E-3	2.9E-3	2.9E-3	2.9E-3	
	SD	0.056E-3	0.053E-3	0.054E-3	0.053E-3	0.054E-3	0.054E-3	0.39E-3	0.39E-3	0.15E-3	0.15E-3	0.15E-3	
	Rank	2	2	2	2	2	2	2	2	5	5	2.6	
MOGWO	Mean	<b>9.96E-05</b>	<b>9.33E-5</b>	<b>9.57E-03</b>	<b>9.33E-5</b>	<b>9.57E-03</b>	<b>9.57E-03</b>	<b>9.40E-05</b>	<b>9.40E-05</b>	<b>9.60E-05</b>	<b>9.60E-05</b>	<b>9.60E-05</b>	
	SD	1.30E-5	2.04E-6	4.03E-5	2.04E-6	4.03E-5	4.03E-5	1.04E-5	1.04E-5	3.25E-5	3.25E-5	3.25E-5	
	Rank	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	

**Table 11** Multi-objective constrained test suits

Problem name	Functions and constraints	Range	Pareto front* feature
SRN	Minimize $\begin{cases} f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2 \\ f_2(x) = 9x_1 - (x_2 - 1)^2 \end{cases}$ Subject to : $g_1(x) = x_1^2 + x_2^2 \leq 225$ $g_2(x) = x_1 - 3x_2 + 10 \leq 0$	$x_1 \in [-20, 20]$ $x_2 \in [-20, 20]$	Continuous convex
CONSTR	Minimize $\begin{cases} f_1(x) = x_1 \\ f_2(x) = \frac{1 + x_2}{x_1} \end{cases}$ Subject to : $g_1(x) = 6 - (x_2 + 9x_1) \leq 0$ $g_2(x) = 1 + x_2 - 9x_1 \leq 0$	$x_1 \in [0.1, 1]$ $x_2 \in [0, 5]$	Continuous concave
BNH	Minimize $\begin{cases} f_1(x) = 4x_1^2 + 4x_2^2 \\ f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 \end{cases}$ Subject to : $g_1(x) = (x_1 - 5)^2 + x_2^2 - 25 \leq 0$ $g_2(x) = 7.7 - (x_1 - 8)^2 - (x_2 + 3)^2 \leq 0$	$x_1 \in [0, 5]$ $x_2 \in [0, 3]$	Continuous convex
KIT	Maximize $\begin{cases} f_1(x) = -x_1^2 + x_2 \\ f_2(x) = 0.5x_1 + x_2 + 1 \end{cases}$ Subject to $g_1(x) = \frac{1}{6}x_1 + x_2 - \frac{13}{2} \leq 0$ $g_2(x) = \frac{1}{2}x_1 + x_2 - \frac{15}{2} \leq 0$ $g_3(x) = 5x_1 + x_2 - 30 \leq 0$	$x_1 \in [0, 7]$ $x_2 \in [0, 7]$	Continuous convex
TNK	Minimize $\begin{cases} f_1(x) = x_1 \\ f_2(x) = x_2 \end{cases}$ Subject to $g_1(x) = -x_1^2 - x_2^2 + 1 + 0.1 \cos(16\arctan \frac{x_1}{x_2}) \leq 0$ $g_2(x) = 0.5 - (x_1 - 0.5)^2 - (x_2 - 0.5)^2 \leq 0$	$x_1 \in [0, \pi]$ $x_2 \in [0, \pi]$	Discrete

**Table 12** Statistical results of GD for constrained problems

Method	SRN					CONSTR				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	133.4477	130.6098	11.35244	114.1482	153.2962	0.00926	0.008488	0.002887	0.005898	0.012938
MOCCSA	53.37386	53.17986	6.48591	46.33395	65.90961	0.016889	0.018935	0.006813	0.006077	0.025527
M2O-CSA	0.310903	0.295142	0.060893	0.243306	0.398874	0.003865	0.003535	0.000992	0.003184	0.006248
Method	BNH					KIT				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.353228	0.369699	0.045825	0.288872	0.395399	0.378178	0.355244	0.288454	0.028031	0.958209
MOCCSA	0.376366	0.407966	0.082848	0.253671	0.435861	0.139575	0.081274	0.191616	0.007304	0.65307
M2O-CSA	0.314963	0.315786	0.017739	0.292164	0.344243	0.047906	0.007839	0.081177	0.006164	0.261082
Method	TNK									
	Mean	Median	SD	Best	Worst					
MOCSA	0.027717	0.025003	0.010958	0.010598	0.042447					
MOCCSA	0.015541	0.015386	0.005859	0.009796	0.021597					
M2O-CSA	0.003827	0.003802	0.000681	0.002665	0.005231					

**Table 13** Statistical results of IGD for constrained problems

Method	SRN					CONSTR				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.055305	0.054208	0.006128	0.045937	0.068767	0.001433	0.001412	0.000205	0.001154	0.001712
MOCCSA	0.061372	0.051048	0.026218	0.041867	0.128152	0.003419	0.003069	0.001424	0.002077	0.007092
M2O-CSA	0.01476	0.014748	0.001241	0.013048	0.017043	0.000876	0.000867	0.000155	0.000705	0.001139
Method	BNH					KIT				
	Mean	Median	SD	Best	Worst	Mean	Median	SD	Best	Worst
MOCSA	0.174789	0.17816	0.069794	0.091334	0.272204	0.005587	0.005864	0.00328	0.001351	0.011421
MOCCSA	0.223935	0.190016	0.097121	0.150988	0.364718	0.010865	0.007598	0.00766	0.004728	0.02491
M2O-CSA	0.046038	0.045814	0.003606	0.039822	0.052371	0.001051	0.00103	6.74E−05	0.000961	0.001152
Method	TNK									
	Mean	Median	SD	Best	Worst					
MOCSA	0.002858	0.002857	8.92E−04	0.00145	0.004386					
MOCCSA	0.002446	0.0027	1.13E−03	0.000926	0.003458					
M2O-CSA	0.000187	0.000185	1.24E−05	0.00017	0.000208					



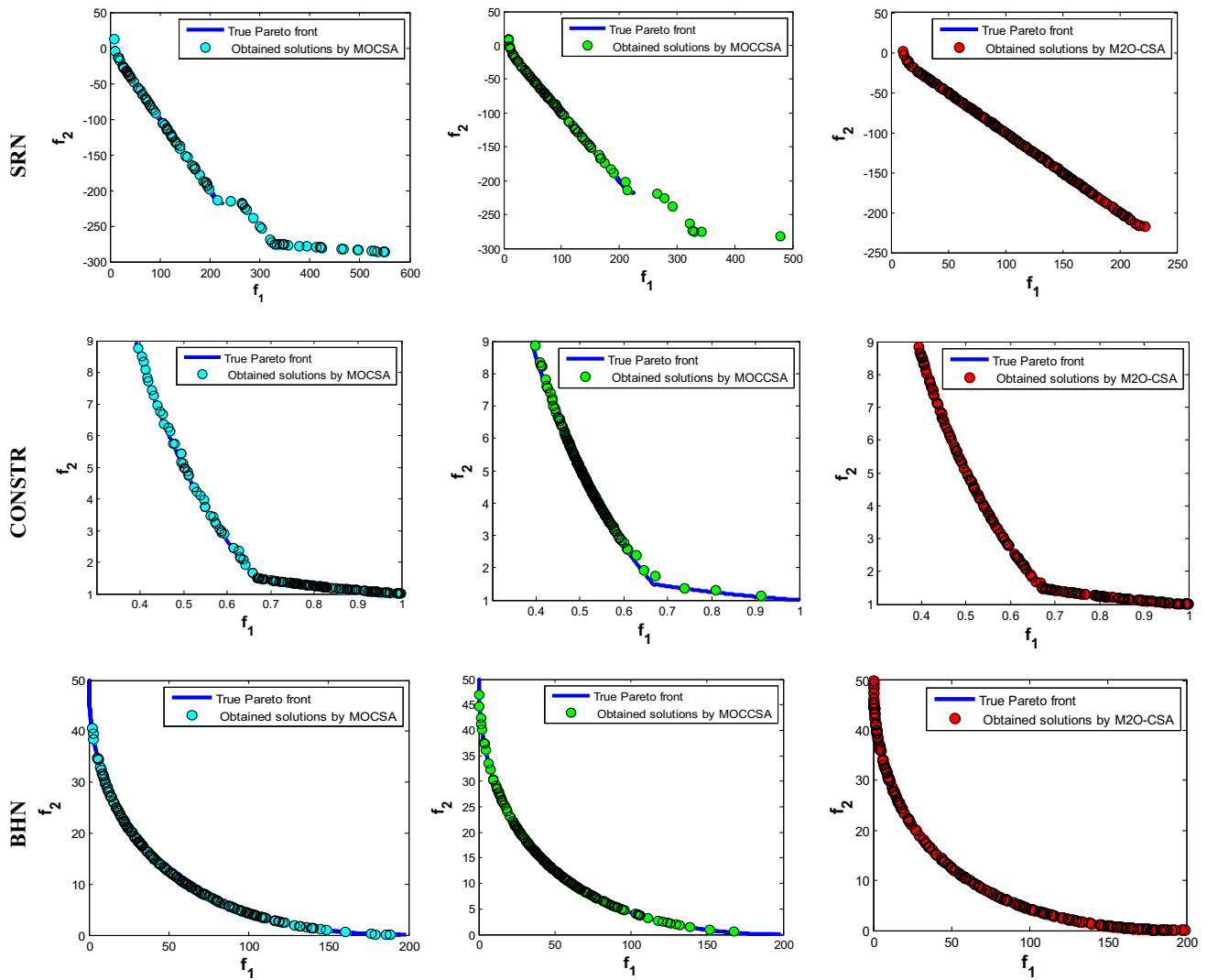


Fig. 10 Obtained Pareto front by MOCSA, MOCCSA, and M2O-CSA for SRN, CONSTR, BNH, KIT, and TNK

tasks not only for inexpert designers by invoking the MOORA tool to attain satisfactory or preferred solution from the Pareto front to execute the candidate engineering design but also for experts through envisioning various scenarios of designing conditions that are denoted by Pareto front and they can choose the most preferred

scenario according to their preferences. Also, it is useful when no a priori knowledge about the candidate application is available. Prudently, this methodology can be a very promising candidate tool in handling practical situations with complicated operating conditions as well. Furthermore, we hope that the released methodology will

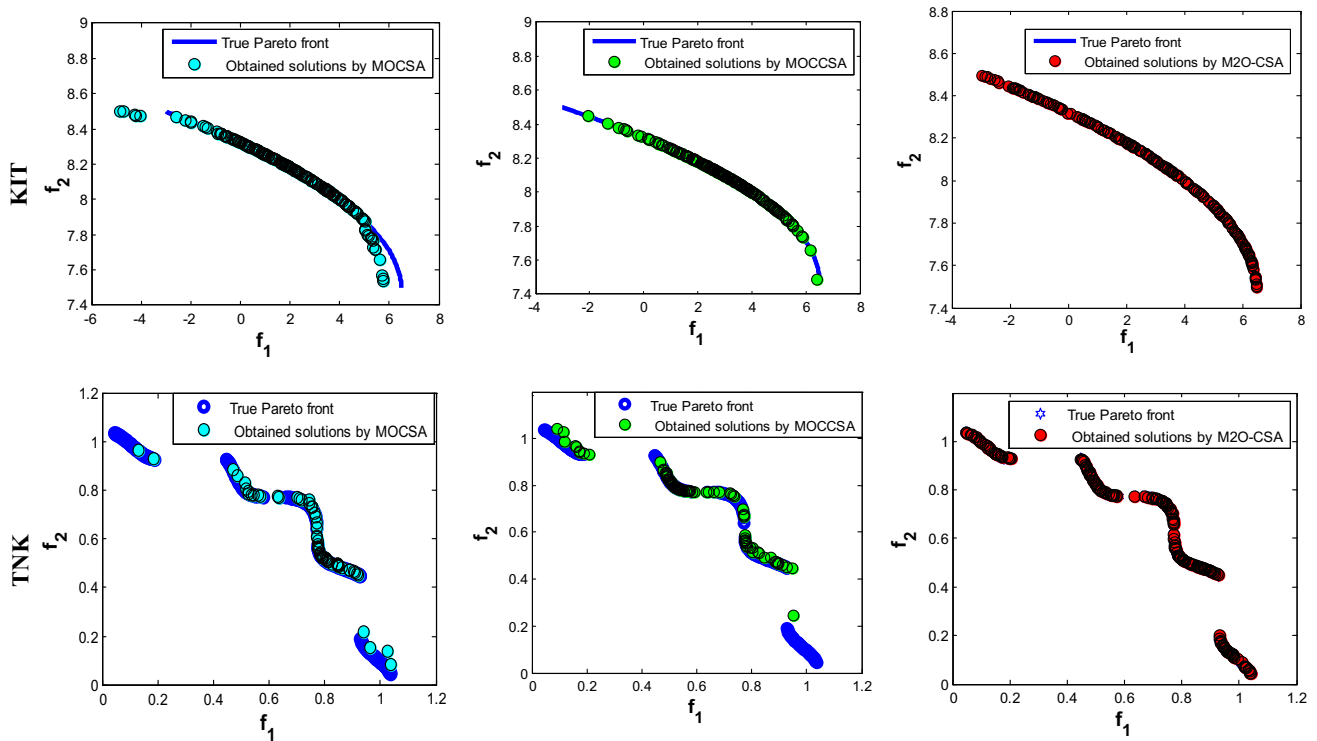


Fig. 10 continued

encourage the engineers and practitioners to apply it in multi-objective real environmental areas such as environmental-economic load dispatch problem that aims to simultaneously minimize the pollution induced by fossil fuels and the operational costs. Another further direction including the energy-based wind farm layout that aims to maximize the output power produced by this farm and minimize the operating cost is also a very environmental interesting area.

## 5 Conclusion

This paper proposed an improved multi-objective orthogonal opposition version of the CSA called M2O-CSA. With maintaining the search strategy of CSA, M2O-CSA is introduced through incorporating CSA with a multi-orthogonal opposition strategy, an archive and a new updating mechanism based on Pareto solutions as the stored food sources. The proposed M2O-CSA algorithm is

benchmarked on ten test instances including five unconstrained test suits and five constrained test suits. The results are assessed through some performance metrics as quantitative indicators such as GD, IGD, SP, and MS. The results are verified by comparisons with MOCSA and MOCSA algorithms. For further investigation, the proposed M2O-CSA is compared with some recent state-of-the-art algorithms. The obtained results prove that the proposed M2O-CSA is highly competitive or outperform the compared algorithms and it is insensitive to the high dimensional problems. Moreover, the proposed M2O-CSA algorithm was successfully investigated to tackle real engineering applications including four bar truss design, welded beam design, disk brake design, and speed reduced design. The results of M2O-CSA show the superiority over those reported by other methods, indicating that the proposed M2O-CSA algorithm has an improved convergence and coverage capabilities for these designs and can be a good alternative to effectively solve the multi-objective design tasks. In addition, qualitative results are

**Table 14** Engineering designs with their details

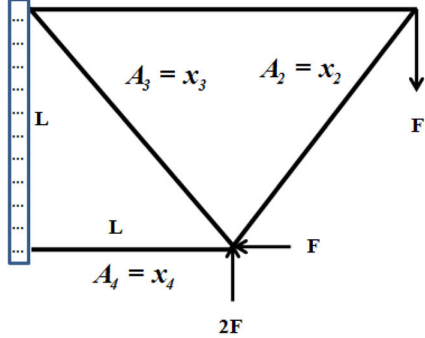
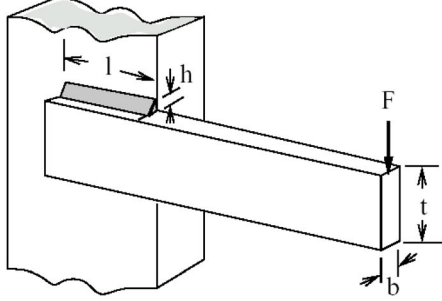
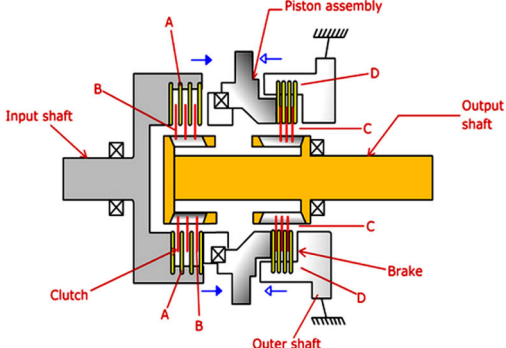
Design Name	The Design description	Mathematical model	The Shape of design
Four bar truss (FBT) design	<p>This design aims to minimize simultaneously the volume and displacement of joints. Where the area of each link represents the design variables.</p>	$\text{Min:} \begin{cases} f_1(x) = L(2x_1 + \sqrt{2}x_2 + \sqrt{x_3} + x_4) \\ f_2(x) = F \frac{L}{E} \left( \frac{2}{x_2} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4} \right) \end{cases}$ <p>Subject to :</p> $1 \leq x_1, x_4 \leq 3,$ $\sqrt{2} \leq x_2, x_3 \leq 3,$ $F = 10, E = 2e5, L = 200.$	 <p>Fig.11-(a). FBT shape</p>
Welded beam (WB) design	<p>The aim of this design is to minimize both the cost and the end deflection simultaneously. The problem involves some constraints such as shear stress, bending stress, weld length and the buckling load. Also, the design contains four different design variables which are the height, the length of the welded joint, thickness, and the width of the beam.</p>	$\text{Min:} \begin{cases} f_1(x) = 1.10471h^2l + 0.04811tb(14 + l) \\ f_2(x) = 2.1952 / t^2b \end{cases}$ <p>Subject to :</p> $g_1(x) = 13600 - \tau \geq 0,$ $g_2(x) = 30000 - \sigma \geq 0,$ $g_3(x) = b - h \geq 0,$ $g_4(x) = P - 6000 \geq 0,$ $h, b \in [0.125, 5],$ $l, t \in [0.1, 10],$ $[h, l, t, b] = [x_1, x_2, x_3, x_4],$ <p>where</p> $\tau = \sqrt{(\tau')^2 + (\tau'')^2 + l\tau'' / \sqrt{0.25(l^2 + (h+l)^2)}},$ $\tau' = 6000 / \sqrt{2hl},$ $\tau'' = \frac{6000(14 + 0.5l)\sqrt{0.25(l^2 + (h+l)^2)}}{2\sqrt{2hl}(l^2 / 12 + 0.25(h+t)^2)},$ $\sigma = 504000 / t^2b,$ $P = 64746.022(1 - .0282346t)tb^3.$	 <p>Fig.11-(b). WB shape</p>
Disk brake (DB) design	<p>The application of the multi-plate disk brake appears in airplanes, with the aim to exhibit effective braking during landing. The purpose of this design is to minimize both the mass of the brake and the stopping time simultaneously. It contains four design variables include the inner radius, outer radius, the engaging force (applied force) and the number of friction surfaces (number of friction plates). Out of these variables, number of friction surfaces takes a discrete integer value and then this can make the design mixed-integer problem. In addition, five different restrictions are considered as constraints which are the distance between the radii of the friction plates, length of the brake, pressure sustained by the plates, the maximum limitation for the temperature generated and the braking torque.</p>	$\text{Min:} \begin{cases} f_1(x) = 4.9e - 5(x_2^2 - x_1^2)(x_4 - 1) \\ f_2(x) = ((9.82e6) \frac{(x_2^2 - x_1^2)}{x_3 x_4 (x_2^2 - x_1^2)}) \end{cases}$ <p>Subject to :</p> $g_1(x) = 20 + x_1 - x_2 \leq 0$ $g_2(x) = 2.5(x_4 + 1) - 30 \leq 0$ $g_3(x) = \frac{x_3}{3.14(x_2^2 - x_1^2)} - 0.4 \leq 0$ $g_4(x) = 2.22e - 3x_3 \frac{x_2^3 - x_1^3}{(x_2^2 - x_1^2)} - 1 \leq 0$ $g_5(x) = 900 - \left( \frac{2.66e - 2x_3 x_4 (x_2^3 - x_1^3)}{(x_2^2 - x_1^2)} \right) \leq 0$ $55 \leq x_1 \leq 80, \quad 75 \leq x_2 \leq 110,$ $1000 \leq x_3 \leq 3000, \quad 2 \leq x_4 \leq 20$	 <p>Fig.11-(c). DB shape</p>

Table 14 continued

Speed reduced (SR) design	<p>The goal of this design problem is to minimize the weight of the gear assembly and the transverse deflection of the simultaneously optimize shaft. In this design, different constraints are considered on bending stress of the gear teeth, surfaces stress, transverse deflections of the shafts and stresses in the shafts while seven design variables are imposed such as the face width, module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts, respectively. This design is considered as a <b>mixed-integer</b> problem because of the third variable is an integer while the rest variables are continuous.</p>	$\text{Min: } \begin{cases} f_1(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \\ f_2(x) = \left( \sqrt{\frac{745x_4}{x_2x_3}} + 16.9e6 \right) / 110x_6^3 \end{cases}$ <p>Subject to:</p> $\begin{aligned} g_1(x) &= 27/x_1x_2^2x_3 - 1 \leq 0, \\ g_2(x) &= 397.5/x_1x_2^2x_3^2 - 1 \leq 0, \\ g_3(x) &= 1.93x_4^3/x_2x_3x_6^4 - 1 \leq 0, \\ g_4(x) &= 1.93x_5^3/x_2x_3x_7^4 - 1 \leq 0, \\ g_5(x) &= \left( \sqrt{\frac{745x_4}{x_2x_3}} + 16.9e6 \right) / 0.1x_6^3 - 1 \leq 0, \\ g_6(x) &= \left( \sqrt{\frac{745x_5}{x_2x_3}} + 157.5e6 \right) / 0.1x_7^3 - 1 \leq 0, \\ g_7(x) &= x_2x_3/40 - 1 \leq 0, \\ g_8(x) &= 5x_2/x_1 - 1 \leq 0, \\ g_9(x) &= x_1/12x_2 - 1 \leq 0, \\ g_{10}(x) &= 1.5x_6 + 1.9/x_4 - 1 \leq 0, \\ g_{11}(x) &= 1.1x_7 + 1.9/x_5 - 1 \leq 0, \\ 2.6 \leq x_1 \leq 3.6, & 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, \\ 7.3 \leq x_4, x_5 \leq 8.3, & 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5. \end{aligned}$	<p>Fig.11-(d). SR shape</p>
---------------------------	---	---	-----------------------------

demonstrated by visualizing the Pareto front for engineering design problems. Furthermore, the hesitance of the designer handled by introducing MOORA technique is to extract the compromise solution of the candidate design and therefore it provides a friendly-user system for designers.

Besides, the proposed work is giving two new directions toward practical implementation: (1) improving the crows' searchability such that multi-objective real life problems can be handled; (2) presenting a fruitful decision making model for engineers or designers to decide the best alternative regarding the operating situations when the non-dominated set is considered. Although the M2O-CSA has been demonstrated to be superior and competitive to other compared algorithms, a fact of no free lunch (NFL) theorem cannot be ignored, that is the M2O-CSA may still have

more room to be competitive enough with huge dimensions and/or many constraints that may deteriorate both of the convergence pattern and diversification abilities toward the true Pareto front. Therefore, some improvement strategies such as mutation schemes, parallel implementation, and hybridization with other algorithms are other research paths that can be implemented in the future to improve performance behavior where these strategies can be applied around the best Pareto front obtained by M2O-CSA to guide the search toward better vicinities and then the performance of the solutions is strengthened. In addition, we attempt to develop a discrete fusion of the M2O-CSA to solve combinatorial optimization applications. In this context, the discrete fusion can be implemented through a truncation strategy or mapping mechanism to transfer the continuous decision variable to a discrete decision variable.

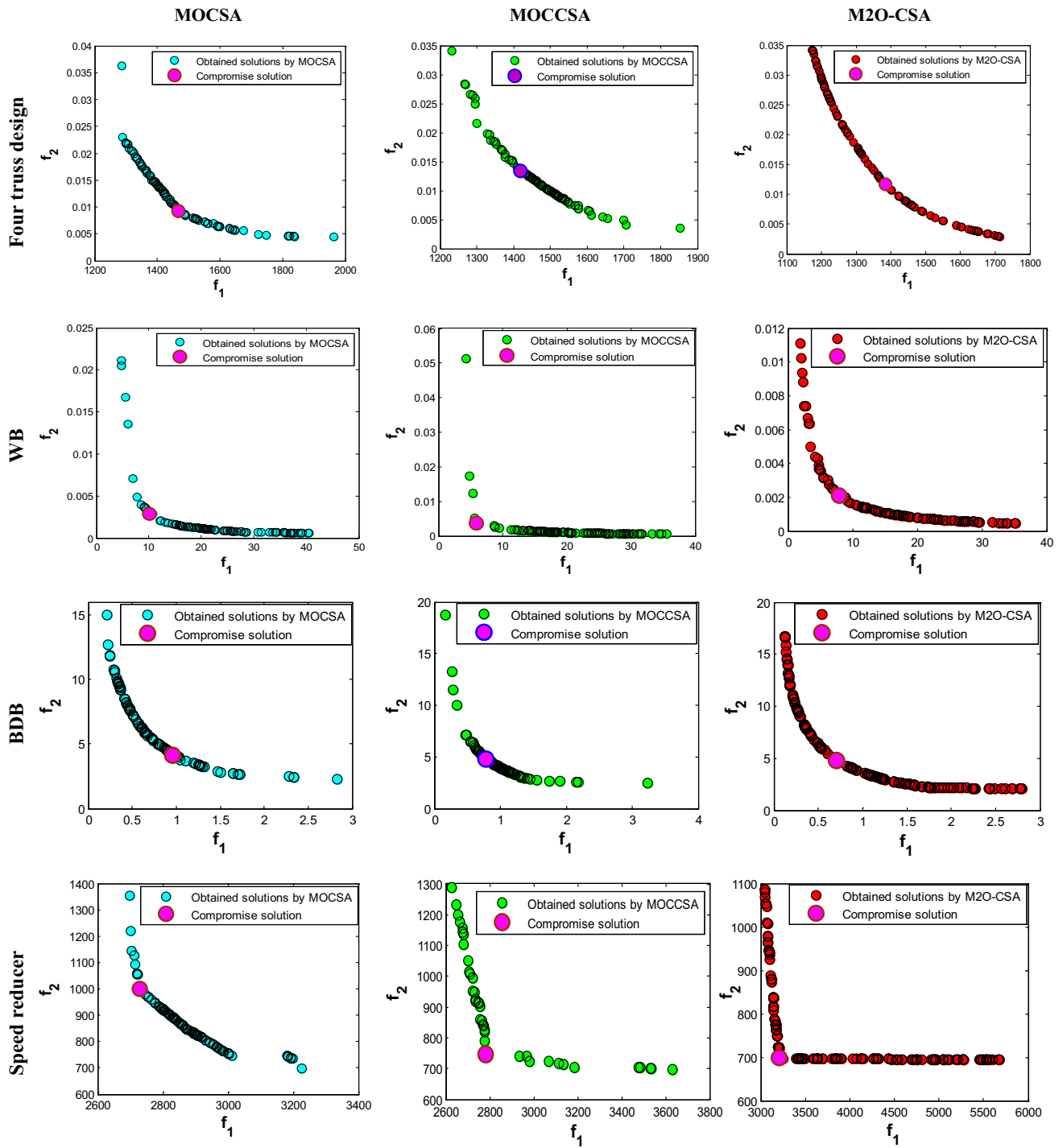


Fig. 12 Obtained Pareto front for design problems

**Table 15** The best compromise solution using MOORA mechanism for FBT, WB, DB, and SR designs

Design name	Design characteristic	MOCSA	MOCCSA	M2O-CSA
FBT	Design variables			
	$x_1$	1.059914	1.065766	1
	$x_2$	2.871824	1.916232	2.969928
	$x_3$	1.549023	1.496284	1.414214
	$x_4$	2.876502	1.099561	1.324033
	Objective functions			
	$f_1(x)$	1727.504	1282.397	1390.084
$f_2(x)$	0.005507	0.024484	0.011363	
WB	Design variables			
	$x_1$	0.737506	0.98478	0.931073
	$x_2$	0.948432	1.091289	0.57131
	$x_3$	9.433933	8.743876	10
	$x_4$	0.410887	1.96115	1.058428
	Constraints			
	$g_1(x)$	− 2589.16	− 6131.72	− 144.593
	$g_2(x)$	− 16217.7	− 26638.7	− 25238.2
	$g_3(x)$	0.326619	− 0.97637	− 0.12736
	$g_4(x)$	− 0.61251	− 0.85978	− 0.80607
	$g_5(x)$	− 43149.9	− 5078912	− 865,128
	Objective functions			
	$f_1(x)$	3.6842	13.61934	7.966978
$f_2(x)$	0.332983	0.001674	0.002074	
DB	Design variables			
	$x_1$	64.98809	63.09972	80
	$x_2$	85.69782	86.38307	100.0575
	$x_3$	2720.737	2878.739	2999.994
	$x_4$	7.115609	4.510027	7
	Constraints			
	$g_1(x)$	− 0.70973	− 3.28335	− 0.0575
	$g_2(x)$	− 9.71098	− 16.2249	− 10
	$g_3(x)$	− 0.12234	− 0.13659	− 0.13545
	$g_4(x)$	− 0.77988	− 0.79248	− 0.74994
	$g_5(x)$	− 57665.3	− 38131.3	− 74847
	Objective functions			
	$f_1(x)$	0.935153	0.598609	1.061782
$f_2(x)$	4.460186	6.692364	3.448482	
SR	Design variables			
	$x_1$	3.051515	2.693731	3.6
	$x_2$	0.732546	0.712028	0.7
	$x_3$	17	17	17
	$x_4$	7.640489	7.68032	7.785744
	$x_5$	7.972066	7.471643	7.786991
	$x_6$	3.871891	3.252395	3.9
	$x_7$	5.403132	5.162313	5.290751

**Table 15** (continued)

Design name	Design characteristic	MOCSA	MOCCSA	M2O-CSA
Constraints				
	$g_1(x)$	− 0.03009	0.162964	− 0.09964
	$g_2(x)$	− 0.16005	0.007142	− 0.22028
	$g_3(x)$	− 0.69243	− 0.35444	− 0.66913
	$g_4(x)$	− 0.90787	− 0.90636	− 0.90227
	$g_5(x)$	− 0.35219	0.093435	− 0.36556
	$g_6(x)$	− 0.06331	0.073939	− 0.00231
	$g_7(x)$	− 0.68867	− 0.69739	− 0.7025
	$g_8(x)$	0.200298	0.321639	− 0.02778
	$g_9(x)$	− 0.65286	− 0.68473	− 0.57143
	$g_{10}(x)$	0.008814	− 0.11741	− 0.00459
	$g_{11}(x)$	− 0.01613	0.014308	− 0.00863
Objective functions				
	$f_1(x)$	3193.073	2614.869	3207.198
	$f_2(x)$	712.8028	1203.452	697.8804

Furthermore, investigating the performance of M2O-CSA on dynamic many-objective optimization problems, incorporation of uncertainties in optimization models and more practical problems such as energy-based optimization, and labor-aware flexible job shop scheduling under dynamic electricity pricing optimization tasks are also worthwhile directions.

### Compliance with ethical standards

**Conflict of interest** All authors declare that they have no conflict of interest.

### References

- Miettinen K (2002) Non-linear multiobjective optimization. Kluwer Academic Publisher, Dordrecht
- Ehrgott M, Gandibleux X (2002) Multiobjective combinatorial optimization—theory, methodology, and applications. In: Ehrgott M, Gandibleux X (eds) Multiple criteria optimization: state of the art annotated bibliographic surveys. Springer, US, pp 369–444
- Berube JF, Gendreau M, Potvin JY (2009) An exact e-constraint method for bi-objective combinatorial optimization problems: application to the Traveling salesman problem with profits. *Eur J Oper Res* 194(1):39–50
- Laumanns M, Thiele L, Zitzler E (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *Eur J Oper Res* 169(3):932–942
- Branke J, Deb K (2005) Integrating user preferences into evolutionary multiobjective optimization. Knowledge incorporation in evolutionary computation. Springer, pp 461–477
- Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. Wiley, London
- Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st international conference on genetic algorithms. L. Erlbaum Associates. Inc., pp 93–100
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
- Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolut Comput* 2(3):221–248
- Corne DW, Jerram NR, Knowles JD, Oates MJ (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd annual conference on genetic and evolutionary computation. Morgan Kaufmann Publishers Inc., pp 283–290
- 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 1999), vol 1, pp 98–105
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evol Comput* 1(1):32–49
- Mousa AA, Abd El-Wahed WF, Rizk-Allah RM (2011) A hybrid ant colony optimization approach based local search scheme for multiobjective design optimizations. *J Electr Power Syst Res* 81:1014–1023
- Rizk-Allah RM, El-Sehiemy RA (2018) A novel sine cosine approach for single and multiobjective emission/economic load dispatch problem. In: International conference on innovative trends in computer engineering (ITCE 2018) Aswan University, Egypt, pp 271–277
- Rizk-Allah RM, El-Sehiemy RA, Deb S, Wang G-G (2017) A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor. *J Supercomput* 73(3):1235–1256
- Rizk-Allah RM, Abo-Sinna MA (2017) Integrating reference point, Kuhn-Tucker conditions and neural network approach for multi-objective and multi-level programming problems. *OPSEARCH* 54(4):663–683
- Rizk-Allah RM, El-Sehiemy RA, Wang G-G (2018) A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution. *Appl Soft Comput* 63:206–222
- Akbari R, Hedayatzadeh R, Ziarati K, Hassanizadeh B (2012) A multi-objective artificial bee colony algorithm. *Swarm Evol Comput* 2:39–52

19. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
20. Mirjalili S, Jangir P, Saremi S (2017) Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Appl Intell* 46:79–95
21. Mirjalili S, Saremi S, Mirjalili SM, Coelho L (2016) Multi-objective grey wolf optimizer: a novel algorithm for multicriterion optimization. *Expert Syst Appl* 47:106–119
22. Tian Y, Cheng R, Zhang X, Cheng F, Jin Y (2018) An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans Evol Comput* 22(4):609–622
23. Rong M, Gong D, Zhang Y, Jin Y, Pedrycz W (2019) Multidirectional prediction approach for dynamic multiobjective optimization problems. *IEEE Trans Cybern* 49(9):3362–3374
24. Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2018) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf Sci* 427:63–76
25. Liu Y, Gong D, Sun J, Jin Y (2017) A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans Cybern* 47(9):2689–2702
26. Liu Y, Gong D, Sun X, Zhang Y (2017) Many-objective evolutionary optimization based on reference points. *Appl Soft Comput* 50:344–355
27. Gong DW, Sun J, Miao Z (2018) A set-based genetic algorithm for interval many-objective optimization problems. *IEEE Trans Evol Comput* 22(1):47–60
28. Yue C, Qu B, Liang J (2018) A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. *IEEE Trans Evol Comput* 22(5):805–817
29. Gu F, Cheung YM (2018) Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Trans Evol Comput* 22(2):211–225
30. Adel G, Abdelouahab M, Djaafar Z (2020) A guided population archive whale optimization algorithm for solving multiobjective optimization problems. *Expert Syst Appl* 141:112972
31. Wang W, Yang S, Lin Q, Zhang Q, Wong KC, Coello CAC, Chen J (2019) An effective ensemble framework for multi-objective optimization. *IEEE Trans Evol Comput* 23(4):645–659
32. Nematollahi AF, Rahiminejad A, Vahidi B (2019) A novel multi-objective optimization algorithm based on Lightning Attachment Procedure Optimization algorithm. *Appl Soft Comput* 75:404–427
33. Tharwat A, Houssein EH, Ahmed MM, Hassanien AE, Gabel T (2018) MOGOA algorithm for constrained and unconstrained multi-objective optimization problems. *Appl Intell* 48(8):2268–2283
34. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
35. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput Struct* 169:1–12
36. Veldhuizen DAV, Lamont GB (1998) Multiobjective evolutionary algorithm research: a history and analysis, Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH
37. Coello CAC, Pulido GT (2005) Multiobjective structural optimization using a micro Genetic algorithm. *Struct Multidiscip Optim* 30(5):388–390
38. Schott JR (1995) Fault tolerant design using single and multi-criteria genetic algorithm optimization (Master's thesis), Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA
39. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8:256–279
40. Tizhoosh HR (2006) Opposition-based reinforcement learning. *J Adv Comput Intell Intell Inform* 10(3):578–585
41. Hinojosa S, Oliva D, Cuevas E, Pajares G, Avalos O, Galvez J (2018) Improving multi-criterion optimization with chaos: a novel Multi-Objective Chaotic Crow Search Algorithm. *Neural Comput Appl* 29(8):319–335

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