



# Multi-objective generalized normal distribution optimization: a novel algorithm for multi-objective problems

Nima Khodadadi<sup>1</sup> · Ehsan Khodadadi<sup>2</sup> · Benyamin Abdollahzadeh<sup>3</sup> · El-Sayed M. El-Kenawy<sup>4</sup> · Pezhman Mardanpour<sup>5</sup> · Weiguo Zhao<sup>6</sup> · Farhad Soleimanian Gharehchopogh<sup>3</sup> · Seyedali Mirjalili<sup>7,8,9</sup>

Received: 29 November 2023 / Revised: 5 February 2024 / Accepted: 26 March 2024  
© The Author(s) 2024

## Abstract

This study introduces the Multi-objective Generalized Normal Distribution Optimization (MOGNDO) algorithm, an advancement of the Generalized Normal Distribution Optimization (GNDO) algorithm, now adapted for multi-objective optimization tasks. The GNDO algorithm, previously known for its effectiveness in single-objective optimization, has been enhanced with two key features for multi-objective optimization. The first is the addition of an archival mechanism to store non-dominated Pareto optimal solutions, ensuring a detailed record of the best outcomes. The second enhancement is a new leader selection mechanism, designed to strategically identify and select the best solutions from the archive to guide the optimization process. This enhancement positions MOGNDO as a cutting-edge solution in multi-objective optimization, setting a new benchmark for evaluating its performance against leading algorithms in the field. The algorithm's effectiveness is rigorously tested across 35 varied case studies, encompassing both mathematical and engineering challenges, and benchmarked against prominent algorithms like MOPSO, MOGWO, MOHHO, MSSA, MOALO, MOMVO, and MOAOS. Utilizing metrics such as Generational Distance (GD), Inverted Generational Distance (IGD), and Maximum Spread (MS), the study underscores MOGNDO's ability to produce Pareto fronts of high quality, marked by exceptional precision and diversity. The results affirm MOGNDO's superior performance and versatility, not only in theoretical tests but also in addressing complex real-world engineering problems, showcasing its high convergence and coverage capabilities. The source codes of the MOGNDO algorithm are publicly available at <https://nimakhodadadi.com/algorithms-%2B-codes>.

**Keywords** Generalized normal distribution optimization · Multi-objective GNDO · Engineering problems

## 1 Introduction

Computers have recently become a primary tool in various areas for tackling challenging problems. Computer-helped design is an area that stresses using computers in determining problems as well as making systems. A system's design process would have necessitated direct human involvement in the past. For example, if a developer intended to locate an optimum form for a high-rise building, s/he would need to initially develop a model and then utilize a wind tunnel to examine it.

Clearly, this design method was expensive and required significant time, with both these elements escalating sharply each year in line with human advancement.

The creation of computers accelerated the design procedure substantially several years back. This indicates that we can utilize computers to make a system without also the demand for a solitary model. Consequently, the design procedure's price and time are considerably less than before. Despite the reality that the device is a fantastic help, making a system still needs straight human participation. This causes a collection of experimentation where the developer attempts to develop an effective strategy. It is indisputable that a developer is prone to errors, which makes the design procedure undependable. The primary duty of a designer entails establishing the framework and utilizing computer software to discover optimal designs.

Optimization methods are widely regarded as practical approaches for determining the best computer designs. Most of the approximate techniques are global search

---

Extended author information available on the last page of the article

methods, which are called metaheuristics. These methods are developed to alleviate the weaknesses of classical approaches. Metaheuristic algorithms can discover nearly global or global solutions through informed decision-making. Over the past several decades, a wide array of metaheuristic algorithms inspired by natural phenomena, collective behavior, and scientific principles have been proposed, such as Differential Evolution (DE) [1], Ant Colony Optimization (ACO) [2], Particle Swarm Optimization (PSO) [3], Hippopotamus Optimization (HO) [4] algorithm, Mountain Gazelle Optimizer (MGO) [5], Al-Biruni Earth Radius (BER) [6], Puma Optimization (PO) [7] and Stochastic Paint Optimizer (SPO) [8]. Algorithms such as Advanced Charged System Search (ACSS) [9], Chaos Game Optimization (CGO) [10], and Dynamic Arithmetic Optimization Algorithm (DAOA) [11] are called physics- or mathematics-based algorithms, which obey the rules in physics or mathematics. Many researchers utilized these algorithms to solve structural optimization problems such as trusses [12–15], frames [16–18], real structural engineering applications [19–24] and applications of medicine [25–27]. The research indicates that when applied to complex optimization issues, metaheuristic algorithms have the capability to provide highly accurate solutions within a practical timeframe. Ease of implementation, simple framework, good accuracy, and reasonable execution time are advantages of metaheuristic algorithms compared with the analytical techniques.

Addressing real-world problems presents several challenges that necessitate the use of specific tools. Multi-objectivity is one of the essential characteristics of real-world challenges that makes them difficult to solve. When multiple objectives need to be improved, a challenge is referred to as a multi-objective problem. Obviously, numerous objective optimizers need to be utilized to resolve such issues.

David Schaffer suggested an advanced concept in 1985 [28]. He explained how to use stochastic optimization techniques to solve multi-objective problems. Ever since, remarkably, a considerable variety of investigations have been committed to establishing multi-objective evolutionary/heuristic formulas. The application of stochastic optimization techniques to real-world scenarios has been made more accessible through the utilization of gradient-free methods and strategies that prevent getting trapped in local optima. Multi-objective optimization approaches are being used in various fields these days. Strength–Pareto Evolutionary Algorithm (SPEA) [29], Multi-objective Particle Swarm Optimization (MOPSO) [30], Multi-objective

Artificial Vultures Optimization Algorithm (MOAVOA) [31], Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [32], Multi-objective Flower Algorithm (MOFA), Multi-objective Thermal Exchange Optimization (MOTEO) [33], Multi-objective Seagull Optimization Algorithm (MOSOA) [34], Multi-objective Stochastic Paint Optimizer (MOSPO) [35], Pareto–frontier Differential Evolution (PDE) [36], Multi-objective Moth-Flame Optimization (MMFA) [37], Multi-objective Salp Swarm Algorithm (MSSA) [38] and Multi-objective Artificial Hummingbird Algorithm (MOAHA) [39] are multi-objective optimization methods. Many of the single-objective methods that have been developed may also be used to address multi-objective optimization problems. Several of the most current ones are the Multi-objective Ant Lion Optimizer (MOALO) [40], Multi-objective Arithmetic Optimization Algorithm (MAOA) [41], Multi-objective Grey Wolf Optimizer [42], Multi-objective Material Generation Algorithm (MOMGA) [43], Multi-objective Multi-Verse Optimization (MOMVO) [44] and Multi-objective Harris Hawks Optimization (MOHHO) [45].

The generalized normal distribution optimization (GNDO), proposed recently by Zhang et al. [46], is recognized for its proficiency in globally searching optimal solutions to single-objective optimization problems. An initial glance at pertinent research demonstrates that GNDO has successfully resolved complex optimization challenges across numerous fields. The introduction of a novel multi-objective algorithm can be driven by various factors, including:

- I. This multi-objective algorithm provides superior performance compared to existing algorithms. This can be due to the incorporation of new techniques, optimization strategies, and design methodologies. The improved performance can make the algorithm more suitable for real-world applications.
- II. Presenting a new algorithm that has not been explored before can bring new insights into multi-objective optimization. By introducing a novel approach, one can contribute to the advancement of the field and potentially lead to breakthroughs in solving challenging problems.
- III. Multi-objective optimization problems can be diverse, and some may require specialized algorithms that are tailored to their specific requirements.
- IV. By presenting a new algorithm for the first time, one can compare it with state-of-the-art algorithms and demonstrate its superiority. This can help establish

the new algorithm's effectiveness and provide a benchmark for future comparisons.

The main motivation for presenting MOGNDO is to create a multi-objective (MO) variant of GNDO, equipping it to tackle multi-objective optimization (MOO) algorithm issues. The authors have leveraged the CEC'20 test suite to gauge MOGNDO's performance. The concept proposed in the No Free Lunch (NFL) theory [47] influenced the development of a multi-objective variant of the existing single-objective GNDO method. To sum up, this paper offers the following contributions:

- The MOGNDO algorithm was formulated as a multi-objective adaptation of the GNDO algorithm, leveraging its benefits. The effectiveness of MOGNDO was measured against seven cutting-edge multi-objective optimization (MOO) algorithms using 35 mathematical benchmarks, encompassing CEC09, ZDT, DTLZ, and CEC2020.
- Five indicators were used to demonstrate MOGNDO's strength and robustness.
- In order to differentiate the proposed MOGNDO algorithm from other selected algorithms, a statistical test was conducted.
- The application of MOGNDO to optimize engineering problems further illustrated its credibility in solving real-world problems.
- Visual representations of the Pareto sets and Pareto fronts obtained by MOGNDO were also provided.
- The proposed MOGNDO algorithm was found to be superior in both qualitative and quantitative analyses.
- The MOGNDO algorithm was evaluated against other cutting-edge algorithms across multiple optimization problems, using diverse performance indicators.

According to the NFL [47] theorem, no optimization algorithm can solve all optimization issues, allowing scientists to propose new or improve existing algorithms to solve optimization challenges. This assertion holds valid for both single and multi-objective optimization methodologies. This theorem establishes that excellent results achieved by an optimizer on one set of problems do not assure equivalent outcomes on a different set of problems. This principle underlies numerous studies in the field, enabling researchers to modify existing methods for novel problem classes or develop new optimization algorithms. It also serves as both the basis and the inspiration for this study. This research introduces a multi-objective version of the recently suggested GNDO [46], capable of addressing optimization issues for diverse applications. As per the

NFL theorem, current algorithms documented can address a wide range of problems, but they are only universal solutions to some optimization challenges. This research promotes the use of multi-objective GNDO to tackle emerging complexities. The processes utilized resemble those employed by MOGWO [42], yet the exploration and exploitation stages of MOGNDO are inherited from the GNDO algorithm.

The paper's remainder is prepared as follows: Sect. 2 explains the terminologies of MOO problems and their standard interpretations. Section 3 provides the standard version of GNDO as well as suggests the MOGNDO algorithm. Section 4 describes the outcomes, discussions, and evaluation of the test and the engineering problems utilized. As the last point, Sect. 5 gives the final thoughts on the work and also the future.

The methodologies employed bear similarity to those used by MOGWO [42], but the discovery and utilization phases of MOGNDO are taken from the GNDO algorithm. Here are some of the advantages that come with adopting this approach:

- The GNDO algorithm has been modified to include an archive to ensure that non-dominated solutions are recovered.
- It has been decided to include a grid mechanism in the GNDO to improve the non-dominated solutions currently in the archive.
- It has really been advised that a leader selection method be implemented based on the current best position of the population.

## 2 A study of the literature

Addressing single-objective optimization problems typically presents a more straightforward endeavor as compared to multi-objective optimization problems, owing primarily to the existence of a singular, unique solution governed by one objective function. This singularity in objective paves the way for a more facile process of comparing solutions and ascertaining the absolute optimal solution in single-objective contexts. Conversely, multi-objective optimization problems are characterized by a plurality of solutions, adding layers of complexity to the solution evaluation process [48, 49]. The following is an illustration of a MOO that can be formulated as a problem of minimization [50]:

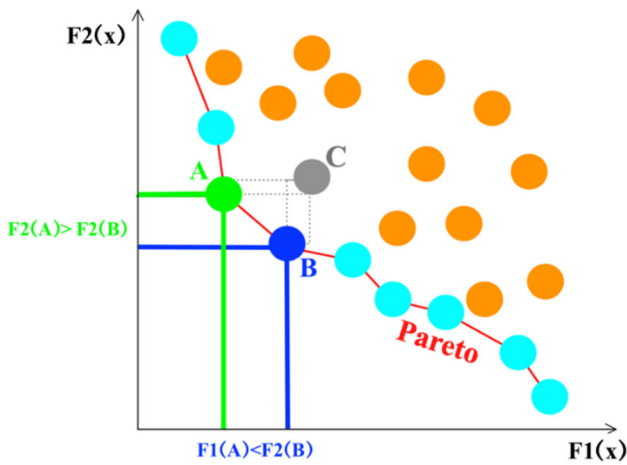


Fig. 1 Pareto dominance

$$\text{Maximization : } F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_z(\vec{x})\} \quad (1)$$

$$\text{Subject to : } g_i(\vec{x}) \geq 0 \quad i = 1, 2, \dots, Q \quad (2)$$

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, P \quad (3)$$

$$L_i \leq x_i \leq U_i \quad i = 1, 2, \dots, D \quad (4)$$

Here, the number of inequality constraints, equality constraints, variables, and objective functions are denoted by  $Q, P, D,$  and  $Z,$  respectively. The lower and upper limits of the  $i$ th variable are represented by  $L_i$  and  $U_i.$

**First Definition:** Pareto dominance.  $\vec{x}$  and  $\vec{y}$  are two solutions with cost functions as:

$$\vec{x} = (x_1, x_2, \dots, x_k) \text{ and } \vec{y} = (y_1, y_2, \dots, y_k).$$

Referring to Fig. 1 and considering a minimization problem, it is stated that solution  $\vec{x}$  dominates solution  $\vec{y}$  (noted as  $\vec{x} \prec \vec{y}$ ) only if none of  $\vec{y}$ 's cost components are less than the equivalent cost components of  $\vec{x},$  and at least one component of  $\vec{x}$  must be smaller than that of  $\vec{y}.$  This can be formally represented as follows:

$$\forall i \in \{1, 2, \dots, k\} : f_i(\vec{x}) \leq f_i(\vec{y}) \wedge \exists i \in \{1, 2, \dots, k\} : f_i(\vec{x}) < f_i(\vec{y}) \quad (5)$$

The concept of Pareto optimality is based on the definition of Pareto dominance [51]:

**Second Definition:** Pareto optimality. A solution  $\vec{x} \in X$  is called Pareto-optimal if and only if:

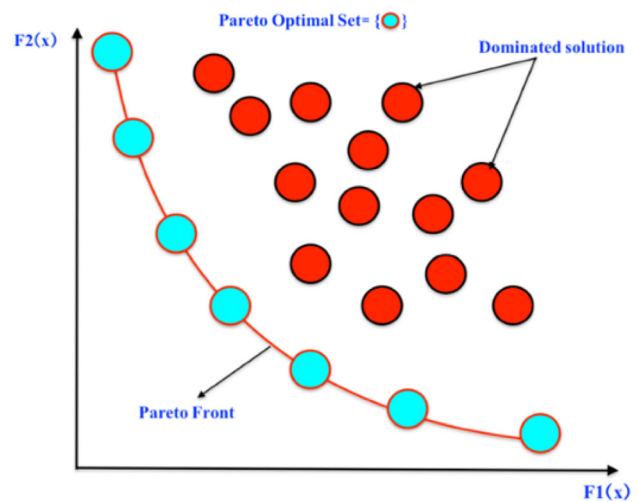


Fig. 2 Pareto optimal solutions

$$\exists \vec{y} \in X | \vec{y} \prec \vec{x} \quad (6)$$

The following description is provided for the Pareto optimal set, which comprises all non-dominated solutions to a specified problem [52]:

**Third Definition:** Pareto optimal set. The computation of the Pareto optimal set ( $P_s$ ) for a specific MOO is outlined as per Eq. (7). No feasible solution within this set can be dominated by any other feasible solution in the same set. The ensemble of Pareto optimal solutions is illustrated in Fig. 2.

$$P_s := \{\vec{x}, \vec{y} \in X | \nexists \vec{y} \prec \vec{x}\} \quad (7)$$

Here is the formulation for the Pareto optimal front:

**Fourth Definition:** Pareto optimal front. As depicted in Fig. 2, a Pareto front ( $P_f$ ) represents the Pareto optimal set in the objective space. Based on the preceding definitions, the equation can be articulated as Eq. (8).

$$P_f = \{F(\vec{x}), \vec{x} \in P_s\} \quad (8)$$

The comprehensive process of multi-objective optimization is illustrated in Fig. 3. This represents an intermediate or current front of non-dominated solutions found by the optimization process. It is an approximation of the True Pareto Front.

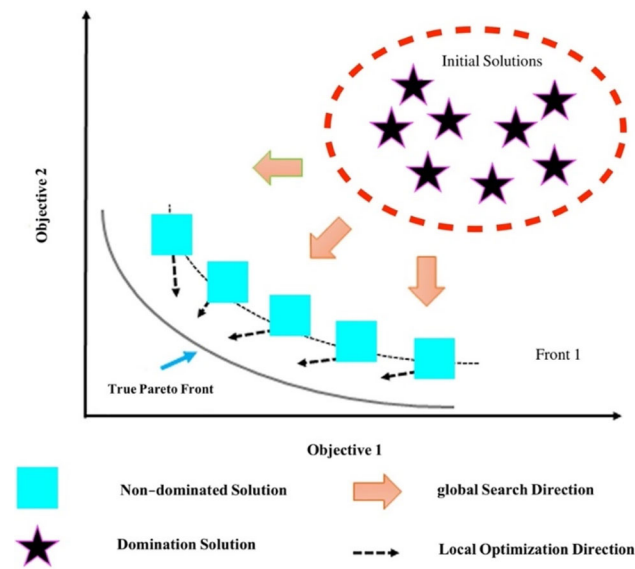


Fig. 3 Multi-objective optimization process

One of the most often used multi-objective GAs is NSGA II [53]. In this version, Pareto sets are labeled starting with the first non-dominated Front by a non-dominated sorting mechanism. A crowded-comparison operator assigns a crowding distance metric to each solution, subsequently steering the selection process based on this metric. In alignment with the concept of elitism, the algorithm opts for solutions with lower domination ranks for survival, and prefers solutions positioned in less congested locations, thereby maintaining solution diversity. In order to achieve a population that is the same size as the initial population, the technique for selecting non-dominated individuals must be performed several times. Finally, these steps are taken before an end condition is achieved.

Particle swarm optimization (PSO) [3] is another known technique that draws inspiration from the collective behavior of various species such as birds and fish. In every iteration, the particles (solutions) are inclined to align with the best personal and global solutions experienced by all particles until the iteration concludes. Coello and Lechuga [54] establishes a repository with a specific capacity to gather non-dominated solutions, which can be deployed in further process steps. In order to pinpoint areas of the objective function's search space that have been less explored, the search space is divided into equal-sized hypercubes. This feature allows the algorithm to preserve solution diversity and distribute them across the entire Pareto space, offering designers a broad range of options instead of concentrating on specific zones. MOPSO

incorporates a mutation method known as disturbance to enhance solution randomness and variability [55, 56].

### 3 Multi-objective generalized normal distribution optimization

In the subsequent section, the GNDO algorithm is initially introduced. Following this, a multi-objective GNDO, aimed at resolving multi-objective optimization issues, is formulated and put forth in the scholarly work. Lastly, the computational complexity of MOGNDO is suggested.

#### 3.1 Generalized normal distribution optimization (GNDO)

The GNDO algorithm was introduced by Zhang et al. [46]. Both exploration and exploitation are balanced via several strategies in the proposed method. Typically, the procedure of searching in methods that use a group of solutions involves three key phases. Initially, every starting solution is spread out. Next, these solutions begin moving towards the best solution, guided by specific searching and refining strategies. Eventually, they all converge near the best-found solution. This process can be explained using multiple normal distributions, where the location of each solution is considered as a variable that follows a normal distribution. At the beginning, the average location and the best location are far apart, and the variation in all solutions' locations is quite large. As the search progresses, the gap between the average location and the best location narrows, and the variation in locations decreases. In the final phase, both the distance from the average to the best location and the variation in solutions' locations are minimized.

GNDO features a straightforward framework, with its information sharing mechanisms comprising local exploitation and global exploration. Exploitation leverages a generalized normal distribution model, steered by the present average and optimal positions. Meanwhile, exploration involves the selection of three individuals at random. As outlined below, the procedure started with the creation of a random population:

$$x_{i,j}^t = l_j + (u_j - l_j) \times \lambda_5 \quad (9)$$

where:  $i = 1, 2, 3, \dots, N$ ,  $j = 1, 2, 3, \dots, D$ .

where the number of design variables is defined by  $D$ , the number of population size is defined by  $N$  the lower and upper boundary of the  $j$ th design variable is described by  $l_j$



and  $u_j$ , respectively.  $\lambda_5$  is a random number in the interval of  $[0, 1]$ .

### 3.1.1 Exploration

All created populations are evaluated via objective function. The following steps will be continued until satisfying the end criterion. Exploration and exploitation are switched with a generated random number for every solution in the population. Global exploration involves scanning the entire search space to identify areas with potential. In GNDO, this exploration is conducted using three individuals chosen at random. The proposed method used an equation for exploration as follows:

$$v_i^t = x_i^t + \underbrace{\beta \times (|\lambda_3| \times v_1)}_{\text{Local learning term}} + \underbrace{(1 - \beta) \times (|\lambda_4| \times v_2)}_{\text{Global learning term}} \quad (10)$$

where two random distributions were created  $\lambda_3$  and  $\lambda_4$  from the normal distribution. The parameter  $\beta$  is a random number between 0 and 1. The following are the two trail vectors  $v_1$  and  $v_2$ :

$$v_1 = \begin{cases} x_i^t - x_{p1}^t, & \text{if } f(x_i^t) < f(x_{p1}^t) \\ x_{p1}^t - x_i^t, & \text{otherwise} \end{cases} \quad (11)$$

$$v_2 = \begin{cases} x_{p2}^t - x_{p3}^t, & \text{if } f(x_{p2}^t) < f(x_{p3}^t) \\ x_{p3}^t - x_{p2}^t, & \text{otherwise} \end{cases} \quad (12)$$

where three random integers number ( $p1$ ,  $p2$ , and  $p3$ ) must satisfy  $p1 \neq p2 \neq p3 \neq i$  are chosen from 1 to  $N$ . Based on Eqs. (11) and (12), the second term on the right side of Eq. (10) is referred to as the local learning term. This denotes that solution  $p1$  exchanges information with solution  $i$ . The third term on the right of Eq. (10) is termed global information sharing, signifying that individual  $i$  receives information from individuals  $p2$  and  $p3$ . The adjustment parameter  $\beta$  is utilized to strike a balance between these two strategies for sharing information. Additionally,  $\lambda_3$  and  $\lambda_4$  are random variables following a standard normal distribution, enhancing GNDO's capability to explore a broader search space during global search activities. The inclusion of the absolute value symbol in Eq. (10) ensures alignment with the screening mechanism outlined in Eqs. (11) and (12).

### 3.1.2 Exploitation

Local exploitation involves the pursuit of improved solutions within the vicinity of the existing positions of all individuals in the search space. This approach is grounded in the correlation between the population's distribution of

individuals and the normal distribution, enabling the construction of a generalized normal distribution model for optimization.

The proposed method used an equation for exploitation as follows:

$$v_i^t = \mu_i + \delta_i \times \eta, \quad i = 1, 2, 3, \dots, N \quad (13)$$

$$M = \sum_{i=1}^N x_i^t \quad (14)$$

$$\mu = \frac{1}{3} (x_i^t + x_{\text{Best}}^t + M) \quad (15)$$

$$\delta = \sqrt{\frac{1}{3} [(x_i^t - \mu)^2 + (x_{\text{Best}}^t - \mu)^2 + (M - \mu)^2]} \quad (16)$$

$$\eta = \begin{cases} \sqrt{-\log(\lambda_1)} \times \cos(2\pi\lambda_2), & \text{if } a \leq b \\ \sqrt{-\log(\lambda_1)} \times \cos(2\pi\lambda_2 + \pi), & \text{otherwise} \end{cases} \quad (17)$$

where  $v_i^t$  represents the trial vector for the  $i$ th individual at time  $t$ ,  $\mu_i$  denotes the generalized mean position of the  $i$ th individual,  $\delta_i$  signifies the generalized standard variance, and  $\eta$  is the penalty factor. In addition, three random numbers ( $a, b, \lambda_1$  and  $\lambda_2$ ) are generated between 0 and 1,  $x_{\text{Best}}^t$  is the best position and the mean position of the current population is defined as  $x_{\text{Best}}^t$  and  $M$ , respectively.

It's important to recognize that the  $i$ th individual might not always identify a superior solution through either local exploitation or global exploration strategies. To ensure that improved solutions are carried forward into the subsequent generation's population, a selection mechanism has been devised, which can be described as follows:

$$x_i^{t+1} = \begin{cases} v_i^t, & \text{iff } (v_i^t) < f(x_i^t) \\ x_i^t, & \text{otherwise} \end{cases} \quad (18)$$

The subsequent section will introduce the multi-objective variant of GNDO.

## 3.2 Multi-objective generalized normal distribution optimization (MOGND)

To carry out multi-objective optimization, two new components have been incorporated into the GNDO. These components resemble those used in MOPSO [30]. The initial element is an archive, which serves the purpose of preserving non-dominated Pareto optimal solutions that have been attained thus far. The subsequent element comprises a leader selection strategy that facilitates the identification of the most appropriate existing position solutions from the archive to serve as leaders for the search operation.

### 3.2.1 Archive mechanism (AM)

The designed external archive aims to preserve the solutions that are not dominated, as obtained up until now. It consists of two key elements: an archive controller and a grid. The archive controller determines whether or not a solution is to be included in the archive. Solutions that are dominated by those already in the archive are immediately excluded. Conversely, solutions not dominated are added to the archive. Should a current archive member be dominated by a new solution, the new solution replaces the older one. It is important to note that the archive has a limit on the number of its members.

When the archive reaches its capacity, an adaptive grid mechanism is activated. The grid's role is to maintain the diversity of solutions within the archive as much as possible. The objective space is segmented into multiple areas. If a new solution falls outside the existing grid boundaries, the grid is adjusted to encompass this new solution. If the solution is already within the grid, it is allocated to the area with the fewest solutions. There are four distinct scenarios that could occur:

- I. If a new member is outclassed by any existing archive solution, it is denied entry.
- II. A new solution that outclasses one or more archived solutions leads to the removal of those solutions, allowing the new solution to be included.
- III. New solutions that neither dominate nor are dominated by existing solutions are added to the archive.
- IV. When the archive is full, the grid mechanism reorganizes the objective space, removing a solution from the most crowded segment and adding the new solution to the least crowded segment to enhance diversity.

The likelihood of removing a solution rises in relation to the quantity of solutions within a given hypercube (segment). To free up space for new entries when the archive is at capacity, solutions are first targeted for removal from the most densely populated segments, with a solution being randomly selected for elimination. A unique situation arises when a solution is added outside the existing hypercubes; in such instances, all segments are expanded to encompass the new solution, potentially altering the segmentation of other solutions as well.

### 3.2.2 Leader selection mechanism (LSM)

The second element involves the mechanism for selecting a leader. The best solutions obtained so far are used as the

current best position. This leading position then steers other search agents towards the most promising areas within the search space, with the objective of uncovering a solution that approximates the global optimum as closely as possible. However, in a search space with multiple objectives, comparing solutions directly becomes complex due to the principles of Pareto optimality outlined in the previous section. To navigate this complexity, a leader selection strategy is implemented. This strategy utilizes an archive that records the best non-dominated solutions identified up to the present. It selects leader from the less dense areas of the search space, offering one of these non-dominated solutions as the new optimal position. This selection process employs a roulette-wheel method, where the likelihood of each hypercube proposing a new leader is detailed in Eq. (19), highlighting that sparsely populated hypercubes are more likely to influence the selection of a new leader.

$$P_i = \frac{C}{K_i} \quad (19)$$

In the equation,  $C$  is a constant number greater than one, and  $K$  represents the number of acquired Pareto optimal solutions in the  $i$ th section.

The chance of a hypercube being chosen for leader selection is boosted as the count of solutions within it diminishes, acknowledging that certain exceptional scenarios may necessitate specific leader selections. As a result, the search consistently gravitates towards areas of the search space that have not been thoroughly explored or exposed, since the leader selection framework prioritizes hypercubes with minimal crowding and suggests leaders from various segments.

To improve the performance of MOGND0 for multi-objective problems, Eq. (14) needs to be adjusted as follows:

$$M = \frac{\sum_{i=1}^N x_i^{leader}}{N} \quad (20)$$

In Eq. (20),  $M$  stands for the average position of the leader population. With regards to computational complexity, where  $n$  stands for the total population count and  $m$  indicates the overall number of objectives, the computational complexity of MOGND0 is expressed as  $O(mn^2)$ . This computational complexity is more efficient than methods such as NSGAI [53], which have a complexity of  $O(mn^3)$ . The pseudo-code of MOGND0 is ultimately provided as follows:

---

**Pseudo-code of MOGNDO**


---

**Set** initial parameters for MOGNDO (alpha, nGrid, Beta, gamma)  
**Begin** with an initial GNDO population  
**Evaluate** objectives for each member of the population  
**Find** non-dominated solutions and create the archive  
**Select** a leader and determine  $M$   
**While** (Iteration < Max-Iteration)  
    **For** each population  
        Create a random alpha between zero and one  
        **If** alpha > 0.5  
            Update the position of current population with Eq (3-3) for each objective  
        **Else**  
            Update the position of current population with Eq (3-4) for each objective  
    **End for**  
    **Evaluate** objectives for the entire population  
    **Find** non-dominated solutions  
    **Update** the archive with these non-dominated solutions  
        **If** the archive is full.  
            Apply grid mechanism to remove older archives  
            Insert the new solution into the archive  
        **End if**  
        **If** a new solution in the archive falls outside the current hypercubes.  
            Update the grids  
        **End if**  
    **End while**  
**Return** the compiled Archive of solutions

---

To grasp the theoretical efficiency of the proposed MOGNDO algorithm for multi-objective problems, a few points can be highlighted as follows:

- The external archive efficiently stores the top non-dominated solutions discovered up to now.
- The GNDO mechanism ensures both exploration and exploitation within the search for MOGNDO.
- The grid mechanism, alongside the leader selection component, preserves the archive's diversity throughout the optimization process.
- The roulette-wheel method used in selecting leader assigns a lower probability for choosing leader from the most populated hypercubes.
- The MOGNDO algorithm retains all the features of GNDO, indicating that the search agents explore and exploit the search space in an identical way.

The convergence of the MOGNDO algorithm undoubtedly derives from the GNDO algorithm since it employs the same mathematical framework to seek out optimal solutions. Search particles adjust their positions rapidly at the beginning of the optimization process and more slowly towards the end. This pattern ensures the algorithm's

convergence within the search space. Selecting a single solution from the archive allows the MOGNDO algorithm to enhance its effectiveness. However, identifying a set of Pareto optimal solutions that also exhibits significant diversity poses a considerable challenge. To address this issue, we have drawn inspiration from the MOPSO algorithm, adopting its leader selection mechanism. We employed a roulette wheel approach and Eq. (19) to select a non-dominated solution from the archive and archive management strategies. It is evident that the archive must have a capacity limit, and the selection of solutions from the archive must aim to enhance overall distribution. In addition, if one solution from the archive is selected, the quality can be improved by the GNDO algorithm. However, it remains challenging to discover the set of Pareto optimal solutions with a wide range. This challenge has been surmounted through the integration of leader feature selection and archive maintenance.

In the MOGNDO algorithm, the initial step involves estimating the true Pareto optimal front for a given multi-objective optimization problem, starting with a randomly chosen collection of solutions. Each solution is evaluated based on multiple objectives. In single-objective optimization, comparing solutions is straightforward because



**Table 1** Parameters setting of all algorithms

Parameters	MOPSO	MOGWO	MOHHO	MSSA	MOALO	MOMVO	MOAOS	MOGNDO
Mutation probability ( $P_w$ , or pro)	0.5	–	0.5	–	–	–	–	–
Population size ( $N_{pop}$ )	100	100	100	100	100	100	100	100
Archive size ( $N_{rep}$ , or TM)	100	100	100	100	100	100	100	100
Number of adaptive grid ( $N_{grid}$ )	30	30	30	30	30	30	30	30
Personal learning coefficient ( $C_1$ )	1	–	1	–	–	–	–	–
Global learning coefficient ( $C_2$ )	2	–	2	–	–	–	–	–
Inertia weight ( $w$ )	0.4	–	0.4	–	–	–	–	–
Beta	4	4	4	4	4	4	4	4
Gamma	2	2	2	2	2	2	2	2

there is only one objective function. For problems where the goal is to maximize, solution  $X$  is considered superior to solution  $Y$  if  $X > Y$ . However, in the context of multi-objective optimization, solutions cannot be directly compared using simple relational operators due to the presence of multiple criteria for comparison. Here, one solution is deemed better than (or dominates) another if it has equal or superior performance across all objectives and excels in at least one objective function. The algorithm identifies and stores all non-dominated solutions in an archive. After the first iteration, it continually adjusts the solutions' positions using Eq. (9). This equation allows for the exchange of variables with either an archived solution or a non-dominated solution in the current set. The first method focuses on utilizing the best Pareto optimal solutions obtained so far, while the second method aids in exploring the search space more thoroughly. This process of refining the solutions continues until a predetermined stopping criterion is fulfilled. The algorithm also improves the distribution of solutions across all objectives by selecting solutions from less crowded areas of the archive.

All features of the MOGNDO are inherited by the GNDO algorithm, which suggests that search agents explore and exploit the search space in a similar fashion. The key difference is that MOGNDO employs an external archive to store the non-dominated services and conducts a search surrounding a set of archive members. Moreover, the proposed MOGNDO comes with a few constraints, which are as follows:

- The algorithm can only be used to optimize multi-objective problems with no more than three or four objectives. As the number of objectives increases, the effectiveness of MOGNDO decreases, which is a

common issue with algorithms based on the Pareto principle. This is because the archive fills up quickly with non-dominated solutions, making it challenging to find optimal solutions for problems with more than four objectives.

- The algorithm is designed to solve optimization problems involving continuous variables, limiting its applicability to other types of problems, such as discrete or mixed-integer optimization problems.

## 4 Results and discussion

The effectiveness of the proposed approach is assessed in this section using performance metrics and 35 distinct case studies. These encompass unconstrained and constrained bi-objective and tri-objective mathematical problems, as well as practical engineering design problems. These tests and mathematical functions are used to ascertain the capability of multi-objective optimizers in addressing non-convex and non-linear challenges. The algorithm has been implemented in MATLAB 2022a, the details of which are described below. The computer specifications used to carry out this project are as follows: a Macintosh computer with OS X, powered by an Intel Core i9 platform, equipped with 16GB 2400MHz DDR4 RAM and a 2.3 GHz CPU (macOS Ventura).

### 4.1 Performance metrics

In order to evaluate the results of the algorithms, we employed five metrics in the following manner [58, 59]:

**Table 2** Mathematic ZDT and DTLZ problems

Function	Mathematical formulation	D	Range
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	$x_i \in [0, 1]$
ZDT2	$F_1 = x_1, F_2 = g \left( 1 - \left( F_1/g \right)^2 \right), g = 1 + \frac{9}{d-1} \sum_{i=2}^d x_i$	30	$x_i \in [0, 1]$
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	$x_i \in [0, 1]$
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	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 1, \dots, D$
ZDT6	$F_1 = 1 - \exp(-4x_1) \sin^6(6\pi x_1), F_2 = g \left( 1 - \left( F_1/g \right)^2 \right), g = 1 + 9 \left( \frac{\sum_{i=2}^d x_i}{d-1} \right)^{0.25}$	10	$x_i \in [0, 1]$
DTLZ2	$F_1 = (1+g) \cos(x_1(\frac{\pi}{2})) \cos(x_2(\frac{\pi}{2})), F_2 = (1+g) \cos(x_1(\frac{\pi}{2})) \sin(x_2(\frac{\pi}{2})),$ $F_3 = (1+g) \sin(x_1(\frac{\pi}{2})), g = \sum_{i=3}^d (x_i - 0.5)^2$	12	$x_i \in [0, 1]$
DTLZ4	$F_1 = (1+g) \cos(x_1^\pi(\frac{\pi}{2})) \cos(x_2^\pi(\frac{\pi}{2})), F_2 = (1+g) \cos(x_1^\pi(\frac{\pi}{2})) \sin(x_2^\pi(\frac{\pi}{2})),$ $F_3 = (1+g) \sin(x_1^\pi(\frac{\pi}{2})), g = \sum_{i=3}^d (x_i - 0.5)^2$	12	$x_i \in [0, 1]$

**Generational distance (GD):** The aggregate distance between potential solutions generated through various methods provides a smart metric for evaluating the convergence traits of multi-objective meta-heuristic algorithms.

$$GD = \left( \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} dis_i^2 \right)^{\frac{1}{2}} \quad (21)$$

**Spacing (S):** A metric to demonstrate the degree of divergence among solution candidates when contrasted against various result sets obtained by multiple algorithms.

$$S = \left( \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} (d_i - \bar{d})^2 \right)^{\frac{1}{2}} \quad \text{where } \bar{d} = \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} d_i \quad (22)$$

**Maximum spread (MS):** This illustrates the dispersion of potential solutions amongst other achieved sets, taking into account the diverse optimal choices.

$$MS = \left[ \frac{1}{m} \sum_{i=1}^m \left[ \frac{\min(f_i^{max}, F_i^{max}) - \max(f_i^{min}, F_i^{min})}{F_i^{max} - F_i^{min}} \right]^2 \right]^{\frac{1}{2}} \quad (23)$$

**Inverted generational distance (IGD):** This measure enables precise performance assessment of Pareto front

approximations using multi-objective optimization methods [60].

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (24)$$

**Wilcoxon rank-sum Test (WRT):** The Wilcoxon rank-sum test, a non-parametric statistical technique, is employed to determine whether two or more datasets originate from an identical distribution. This test employs a significance level of 5%. This examination is utilized to conduct a comprehensive evaluation of the efficacy of the algorithm. As per the null hypothesis, if the mean metrics attained by the two algorithms under comparison are the same, there is no discernible difference in their performance. Conversely, the alternative hypothesis proposes that a difference exists in the mean metrics generated by the algorithms under comparison. The paper presents a comparative analysis of the algorithms, highlighting their differences with mathematical symbols such as subtraction, addition, and equality operators. These indicators denote whether the algorithm exhibits inferior performance, significantly superior performance, or no distinguishable difference, correspondingly.

The evaluation of the algorithms' performance in approximating Pareto optimal solutions is determined by

**Table 3** CEC-09 problems with two objectives

Function	Mathematical formulation	D	Range
UF1	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} \left[ x_j - \sin \left( (6\pi x_1 + \frac{j\pi}{n}) \right) \right]^2,$ $J_1 = \{j   j \text{ is odd and } 2 \leq j \leq n\},$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} \left[ x_j - \sin \left( 6\pi x_1 + \frac{j\pi}{n} \right) \right]^2$ $J_2 = \{j   j \text{ is even and } 2 \leq j \leq n\},$	30	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 1, \dots, D$
UF2	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2,$ $J_1 = \{j   j \text{ is odd and } 2 \leq j \leq n\},$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $J_2 = \{j   j \text{ is even and } 2 \leq j \leq n\}$ $y_j = \begin{cases} x_j - \left[ 0.3x_1^2 \cos \left( 24\pi x_1 + \frac{4j\pi}{n} \right) + 0.6x_1 \right] \cos \left( 6\pi x_1 + \frac{j\pi}{n} \right) & \text{if } j \in J_1 \\ x_j - \left[ 0.3x_1^2 \cos \left( 24\pi x_1 + \frac{4j\pi}{n} \right) + 0.6x_1 \right] \cos \left( 6\pi x_1 + \frac{j\pi}{n} \right) & \text{if } j \in J_2 \end{cases}$	30	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 1, \dots, D$
UF3	$f_1 = x_1 + \frac{2}{ J_1 } \left( 4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos \left( \frac{20y_j\pi}{\sqrt{j}} \right) + 2 \right)$ $f_2 = \sqrt{x_1} + \frac{2}{ J_2 } \left( 4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos \left( \frac{20y_j\pi}{\sqrt{j}} \right) + 2 \right)$	30	$x_i \in [0, 1]$
UF4	<p><math>J_1</math> and <math>J_2</math> are the same as those of UF1, <math>y_j = x_j - x_1^{0.5(1.0 + \frac{3j-2}{\pi-2})}</math>, <math>j = 2, 3, \dots, n</math></p> $f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ <p><math>J_1</math> and <math>J_2</math> are the same as those of UF1, <math>y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n})</math>, <math>j = 2, 3, \dots, n</math></p> $h(t) = \frac{ t }{1+e^{2 t }}$	30	$x_1 \in [0, 1]$ $x_i \in [-2, 2]$ $i = 1, \dots, D$

Table 3 (continued)

Function	Mathematical formulation	D	Range
UF5	$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right)  \sin(2N\pi x_1)  + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j)$ $f_1 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right)  \sin(2N\pi x_1)  + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ <p><math>J_1</math> and <math>J_2</math> are identical to those of UF1, <math>\varepsilon &gt; 0, y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n</math>  <math>h(t) = 2t^2 - \cos(4\pi t) + 1</math></p>	30	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 1, \dots, D$
UF6	$f_1 = x_1 + \max\left\{0, 2\left(\frac{1}{2N} + \varepsilon\right)  \sin(2N\pi x_1) \right\} + \frac{2}{ J_1 } \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1\right)$ $f_2 = 1 - x_1 + \max\left\{0, 2\left(\frac{1}{2N} + \varepsilon\right)  \sin(2N\pi x_1) \right\} \frac{2}{ J_1 } \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1\right)$ <p><math>J_1</math> and <math>J_2</math> are identical to those of UF1, <math>\varepsilon &gt; 0, y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n</math></p>	30	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 1, \dots, D$
UF7	$f_1 = \sqrt[n]{x_1} + \frac{2}{ J_2 } \sum_{j \in J_1} y_j^2, f_2 = 1 - \sqrt[n]{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ <p><math>J_1</math> and <math>J_2</math> are identical to those of UF1, <math>\varepsilon &gt; 0, y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, 3, \dots, n</math></p>	30	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 1, \dots, D$

the S and MS metrics, while the measurement of their convergence is based on the IGD and GD performance metrics. After conducting mean-based assessments to evaluate the performance of the algorithm and comparing it using the Wilcoxon rank-sum test, it was determined that the algorithm exhibited a high level of competitiveness and effectiveness.

#### 4.1.1 Numerical setup

This section presents a comparison among MOPSO, MOGWO, MOHHO, MSSA, MOALO, MOMVO, MOAOS, and MOGND. The best figure from a set of Pareto optimal figures is emphasized. To ensure a fair comparison, all experiments are performed on the same device. Table 1 displays all initial parameters for the aforementioned algorithms. The parameters for the algorithm are set to their default values, minimizing the chance of superior parametrization bias, as advocated by Arcuri and Fraser [61]. Furthermore, the control parameters for the algorithms under comparison were derived from their

respective references. The optimal parameter for each algorithm, as outlined in their references, is utilized in this context. It is important to note that all experiments incorporated 100 populations and a maximum of 1000 iterations. The efficiency of the proposed algorithm is evaluated in this section on 35 cases, including 7 unconstrained and constrained functions, ten traditional multi-objective CEC-09, eight engineering design problems, and 10 CEC-2020 problems.

The death penalty was used in multi-objective problems. However, the death penalty's role is to be used to discard solutions that are not feasible, and the knowledge of those solutions that are useful in solving the problem of controlled inviolable regions is not utilized [54]. Due to its simplicity and low computational cost, the MOGND algorithm has been equipped with a death penalty feature to deal with multiple constraints. The benchmark problems are offered in Tables 2, 3 and 4. Engineering design problems are considered one of the most challenging examination problems in literary works that give various multi-objective search spaces with different Pareto optimal

**Table 4** CEC-09 problems with three objectives

Function	Mathematical formulation	D	Range
UF8	$f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right)^2 \right)$ $f_2 = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right)^2 \right)$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right)^2 \right)$ $J_1 = \{j 3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j 3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}$	30	$x_1 \in [0, 1]$ $x_2 \in [0, 1]$ $x_i \in [-2, 2]$ $i = 1, \dots, D$
UF9	$f_1 = 0.5[\max\{0, (1 + \varepsilon)(1 - 4(2x_1 - 1)^2)\}] + \frac{2}{ J_1 } \sum_{j \in J_1} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right)^2 \right)$ $f_2 = 0.5[\max\{0, (1 + \varepsilon)(1 - 4(2x_1 - 1)^2)\}] + 2x_1x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right)^2 \right)$ $f_3 = 1 - x_2 + \frac{2}{ J_3 } \sum_{j \in J_3} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right)^2 \right)$ $J_1 = \{j 3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j 3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}, \varepsilon = 0.1$	30	$x_1 \in [0, 1]$ $x_2 \in [0, 1]$ $x_i \in [-2, 2]$ $i = 1, \dots, D$



Table 4 (continued)

Function	Mathematical formulation	D	Range
UF10	$f_2 = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} [4y_j^2 - \cos(8\pi y_j) + 1]$ $f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} [4y_j^2 - \cos(8\pi y_j) + 1]$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} [4y_j^2 - \cos(8\pi y_j) + 1]$ $J_1 = \{j   3 \leq j \leq n, \text{ and } j - 1 \text{ is a multiplication of } 3\}$ $J_2 = \{j   3 \leq j \leq n, \text{ and } j - 2 \text{ is a multiplication of } 3\}$ $J_3 = \{j   3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}$	30	$x_1 \in [0, 1]$ $x_2 \in [0, 1]$ $x_i \in [-2, 2]$ $i = 1, \dots, D$

fronts: convex, non-convex, discontinuous, and also multimodal. It might be observed that examination functions with varied features are picked to evaluate the efficiency of MOGNDO from various points of view. Although examination functions can help analyze an algorithm, solving real problems is constantly much more difficult. Tables 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 provide the results of 30 independent runs on the 100,000 function evaluations for all algorithms in this study.

## 4.2 Numerical discussion of the ZDT and DTLZ test function

The initial set of problems consist of mathematical problems. The mean and standard deviation results for all performance metrics for ZDT and DTLZ problems are displayed in Tables 5, 6, 7 and 8. The results for the GD performance metric are indicated in Table 5. As per this table, the Pareto optimal solutions garnered by MOGNDO

exhibit greater convergence than MOPSO, MOGWO, and MOHHO on ZDT1, ZDT2, and ZDT3.

The results when using the IGD metric are provided in Table 6. It can be seen that MOGNDO is slightly better than MOPSO in this metric. These outcomes reveal the efficiency of the MOGNDO algorithm is much steadier than MOPSO. Considering that IGD is an excellent measurement to benchmark an algorithm's convergence, these outcomes show that the proposed algorithm has a much better quantitative efficiency, both considering convergence and distribution convergence on this benchmark function.

The value of the outcomes for the IGD metric is depicted in Fig. 4. The worst outcomes come from MOHHO handling these benchmark problems. The box-plots depicted in Fig. 4 reveal that this algorithm yields highly unfavorable results, while MOGNDO's outcomes are more competitive in comparison. In addition, the box-plot of the MOGNDO is narrower than MOPSO, MOGWO and also MOHHO in most problems, revealing the strength

**Table 5** Mathematic problem statistics results for the GD metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGNDO</i>
ZDT1				
Ave	0.00179	0.00206	0.00214	<b>0.00161</b>
SD	0.00505	0.00124	0.00106	<b>0.00022</b>
WRT	+	+	+	
ZDT2				
Ave	0.14749	0.00201	0.00453	<b>0.00158</b>
SD	0.06521	0.00062	0.00077	<b>0.00032</b>
WRT	+	+	+	
ZDT3				
Ave	0.00020	0.00026	0.00022	<b>0.00017</b>
SD	<b>0.00003</b>	0.00009	0.00017	0.00015
WRT	+	+	+	
ZDT4				
Ave	4.12985	6.87465	<b>0.00072</b>	0.94821
SD	3.95190	7.46401	<b>0.00132</b>	1.71557
WRT	+	+	-	
ZDT6				
Ave	<b>0.02580</b>	0.02583	0.03702	0.10548
SD	0.05801	0.05801	<b>0.03179</b>	0.06141
WRT	-	+	+	
DTLZ2				
Ave	0.00686	<b>0.00126</b>	0.00499	0.00688
SD	0.00091	0.00052	0.00258	<b>0.00040</b>
WRT	+	-	+	
DTLZ4				
Ave	0.01010	<b>0.00187</b>	0.02706	0.01406
SD	0.00246	<b>0.00039</b>	0.00755	0.01922
WRT	+	-	+	
$W^+/W^-$	12/2	10/4	12/2	
$+/-/=$	6/1/0	5/2/0	6/1/0	

**Table 6** Mathematic problem statistics results for the IGD metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGNDO</i>
ZDT1				
Ave	0.00081	0.00122	0.00126	<b>0.00050</b>
SD	0.00148	0.00035	0.00013	<b>0.00006</b>
WRT	+	+	+	
ZDT2				
Ave	0.05202	0.01819	0.02311	<b>0.00051</b>
SD	0.00950	0.00974	0.00300	<b>0.00009</b>
WRT	+	+	+	
ZDT3				
Ave	<b>0.00026</b>	0.00049	0.00064	0.00035
SD	0.00004	0.00007	0.00018	<b>0.00003</b>
WRT	-	+	+	
ZDT4				
Ave	0.70747	0.36983	<b>0.00695</b>	0.05723
SD	0.31784	0.33799	<b>0.00997</b>	0.07820
WRT	+	+	-	
ZDT6				
Ave	0.00823	0.00359	<b>0.00088</b>	0.00633
SD	0.02486	0.00856	<b>0.00024</b>	0.00826
WRT	+	+	-	
DTLZ2				
Ave	<b>0.00047</b>	0.00301	0.00058	0.00098
SD	0.00003	0.00040	0.00004	<b>0.00002</b>
WRT	-	+	+	
DTLZ6				
Ave	<b>0.00168</b>	0.00740	0.00228	0.00434
SD	<b>0.00009</b>	0.00037	0.00024	0.00267
WRT	-	+	+	
$W^+/W^-$	8/6	14/0	12/2	
$+/-/=$	4/3/0	7/0/0	6/1/0	

of the MOGNDO algorithm is converging towards the true Pareto optimal front.

Table 7 shows the MS performance metric of all algorithms. The coverage of Pareto optimal solutions for algorithms is calculated by the maximum spread metric. It is evident that the results of MOGNDO are better than MOPSO, MOHHO and MOGWO for all test problems. Thanks to this, the proposed method has competitive coverage in comparison with the other mentioned methods. MOGNDO obtains the best results in ZDT1, ZDT2, ZDT6, DTLZ2 and DTLZ4 problems.

Based on the analysis of the true Pareto optimal front and the best-obtained Pareto optimal fronts for the ZDT and DTLZ problems, it is evident from Fig. 5 that MOGNDO’s Pareto optimal front consistently outperforms those of MOPSO, MOGWO, and MOHHO in the majority of scenarios. This is further substantiated by the Pareto optimal fronts for ZDT1, ZDT2, and ZDT3 shown in Fig. 5, where all Pareto optimal solutions estimated by MOGNDO align with the true Pareto Front. This confirms that the suggested MOGNDO algorithm has the potential to offer remarkable outcomes on multi-objective problems.

**Table 7** Mathematic problem statistics results for the MS metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGNDO</i>
ZDT1				
Ave	0.99892	0.82895	0.97339	<b>0.99961</b>
SD	0.02378	0.07455	0.01906	<b>0.00318</b>
WRT	+	+	+	
ZDT2				
Ave	0.07431	0.22038	0.00002	<b>0.99501</b>
SD	0.23499	0.40821	0.00532	<b>0.00489</b>
WRT	+	+	+	
ZDT3				
Ave	<b>0.99865</b>	0.97302	0.99105	0.99116
SD	0.00179	0.02115	0.00914	<b>0.00128</b>
WRT	-	+	+	
ZDT4				
Ave	0.46863	<b>6.21295</b>	0.75264	1.86344
SD	1.48194	6.92916	<b>0.37945</b>	3.27514
WRT	+	-	+	
ZDT6				
Ave	1.11704	2.16386	1.74734	<b>3.31874</b>
SD	0.53983	1.60103	<b>0.45030</b>	1.30800
WRT	+	+	+	
DTLZ2				
Ave	1.09765	0.08909	1.05236	<b>2.21711</b>
SD	0.03087	<b>0.03033</b>	0.06055	0.40584
WRT	+	+	+	
DTLZ4				
Ave	1.24373	0.21199	1.57037	<b>2.32417</b>
SD	0.14187	<b>0.06520</b>	0.15206	0.37802
WRT	+	+	+	
$W^+/W^-$	12/2	12/2	14/0	
$+/-/=$	6/1/0	6/1/0	7/0/0	

**Table 8** Mathematic problem statistics results for the S metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGNDO</i>
ZDT1				
Ave	0.01132	0.01489	0.00811	<b>0.00909</b>
SD	0.00108	0.00619	0.00278	<b>0.00094</b>
WRT	+	+	+	
ZDT2				
Ave	0.00073	0.00500	<b>0.00001</b>	0.01043
SD	0.00232	0.01031	0.00210	<b>0.00151</b>
WRT	+	+	-	
ZDT3				
Ave	0.01327	0.01512	0.01180	<b>0.01028</b>
SD	0.00236	0.00652	0.00278	<b>0.00209</b>
WRT	+	+	+	
ZDT4				
Ave	<b>0.00438</b>	1.34460	0.00833	0.10730
SD	0.01385	1.67668	<b>0.00458</b>	0.17431
WRT	-	+	+	
ZDT6				
Ave	0.01811	0.05767	<b>0.01105</b>	0.09873
SD	0.03007	0.09388	<b>0.00399</b>	0.07624
WRT	+	+	-	
DTLZ2				
Ave	0.05774	<b>0.00471</b>	0.04096	0.18045
SD	0.00495	<b>0.00128</b>	0.01724	0.04137
WRT	+	-	+	
DTLZ4				
Ave	0.06132	<b>0.01174</b>	0.05573	0.16177
SD	0.00533	<b>0.00332</b>	0.01273	0.03750
WRT	+	-	+	
$W^+/W^-$	12/2	10/4	10/4	
$+/-/=$	6/1/0	7/2/0	7/2/0	

The statistical results for tri-objective benchmarks (DTLZ2 and DTLZ4) are more challenging benchmarks, showing that this method is competitive for solving more than two objective problems. Figure 6 presents the true and the achieved Pareto Front for tri-objective issues from two viewing angles.

To compare the diversity of MOGNDO with MOPSO, MOGWO and MOHHO algorithms concerning the spacing metric, the statistical results are obtained from 30 individual runs for ZDT and DTLZ benchmarks. Table 8 shows

the results for the spacing performance metric. MOGNDO, MOHHO and MOGWO obtained the best results in terms of this metric simultaneously. It is clear that the MOGNDO presented here will provide a reasonably reliable estimate of true Pareto optimal solutions. Evidence of the MOGNDO's superiority over alternative approaches is presented in Tables 5, 6, 7, and 8 via the Wilcoxon's rank sum test.

**Table 9** CEC problem statistics results for the IGD metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
UF1				
Ave	0.00533	0.00627	0.00751	<b>0.00389</b>
SD	0.00296	0.00236	0.00073	<b>0.00049</b>
WRT	+	+	+	
UF2				
Ave	0.00416	0.00312	0.00386	<b>0.00243</b>
SD	0.00048	0.00014	0.00031	<b>0.00011</b>
WRT	+	+	+	
UF3				
Ave	0.01723	<b>0.01176</b>	0.01282	0.01470
SD	0.00108	0.00109	<b>0.00032</b>	0.00529
WRT	+	-	+	
UF4				
Ave	0.00283	0.00277	0.00222	<b>0.00218</b>
SD	0.00026	0.00043	<b>0.00009</b>	0.00028
WRT	+	+	+	
UF5				
Ave	0.34235	<b>0.31605</b>	0.50037	0.61373
SD	0.15317	0.10235	0.08916	<b>0.08465</b>
WRT	+	-	+	
UF6				
Ave	0.02925	<b>0.01638</b>	0.04112	0.04917
SD	0.00914	<b>0.00145</b>	0.00233	0.01154
WRT	+	-	+	
UF7				
Ave	0.00434	<b>0.00427</b>	0.00773	0.00763
SD	0.00242	0.00205	0.00560	<b>0.00533</b>
WRT	+	-	+	
UF8				
Ave	0.00928	0.00450	<b>0.00344</b>	0.00491
SD	0.00161	0.00100	0.00065	<b>0.00063</b>
WRT	+	+	-	
UF9				
Ave	0.01230	0.00411	0.00471	<b>0.00387</b>
SD	0.00295	0.00095	0.00073	<b>0.00054</b>
WRT	+	+	+	
UF10				
Ave	0.06330	0.03439	0.04096	<b>0.01288</b>
SD	0.01376	0.00908	<b>0.00001</b>	0.00497
WRT	+	+	+	
$W^+/W^-$	20/0	12/8	20/0	
$+/-/=$	10/0/0	6/4/0	10/0/0	

**Table 10** CEC problem statistics results for the GD metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
UF1				
Ave	0.02743	0.03497	0.02261	<b>0.01148</b>
SD	0.02539	0.03199	<b>0.00320</b>	0.00905
WRT	+	+	+	
UF2				
Ave	0.02066	0.01122	0.02675	<b>0.00386</b>
SD	0.00468	0.00212	0.00807	<b>0.00075</b>
WRT	+	+	+	
UF3				
Ave	0.09601	0.07206	0.04656	<b>0.01776</b>
SD	0.02348	0.01027	<b>0.00213</b>	0.01237
WRT	+	+	+	
UF4				
Ave	0.00930	0.01017	0.00960	<b>0.00906</b>
SD	0.00093	0.00348	0.00024	<b>0.00091</b>
WRT	+	+	+	
UF5				
Ave	0.42733	0.28272	<b>0.23459</b>	0.84808
SD	0.29837	0.15805	<b>0.03298</b>	0.16888
WRT	+	+	-	
UF6				
Ave	0.20278	<b>0.11532</b>	0.12362	0.52201
SD	0.16569	0.05047	<b>0.00917</b>	0.16704
WRT	+	-	+	
UF7				
Ave	0.01794	<b>0.00913</b>	0.02365	0.04279
SD	0.01381	0.01232	<b>0.00297</b>	0.04476
WRT	+	-	+	
UF8				
Ave	0.26587	0.03391	0.05737	<b>0.00142</b>
SD	0.08220	0.01331	0.03652	<b>0.00074</b>
WRT				
UF9				
Ave	0.40311	<b>0.05196</b>	0.11025	0.07831
SD	0.07992	<b>0.02176</b>	0.07425	0.08931
WRT	+	-	+	
UF10				
Ave	1.38278	0.69388	0.30025	<b>0.17344</b>
SD	0.21017	0.20777	0.18031	<b>0.17751</b>
WRT	+	+	+	
$W^+/W^-$	20/0	14/6	18/2	
$+/-/=$	10/0/0	7/3/0	9/1/0	

**Table 11** CEC problem statistics results for the MS metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
UF1				
Ave	0.97019	0.96206	0.97885	<b>0.99976</b>
SD	0.38257	0.11721	<b>0.06003</b>	0.24170
WRT	+	+	+	
UF2				
Ave	1.20372	1.20071	1.15449	<b>1.21871</b>
SD	0.15132	0.05715	0.15595	<b>0.03751</b>
WRT	+	+	+	
UF3				
Ave	1.30353	1.09227	<b>1.00000</b>	0.70004
SD	0.71625	0.09263	<b>0.00320</b>	0.48298
WRT	+	+	–	
UF4				
Ave	1.02946	1.00491	0.97920	<b>1.03597</b>
SD	0.01980	<b>0.00867</b>	0.01456	0.02154
WRT	+	+	+	
UF5				
Ave	0.63155	1.20349	0.99002	<b>3.42353</b>
SD	0.72675	0.86410	<b>0.43264</b>	1.12945
WRT	+	+	+	
UF6				
Ave	0.87827	1.02984	0.68589	<b>2.64456</b>
SD	1.20854	<b>0.03734</b>	0.51371	0.70124
WRT	+	+	+	
UF7				
Ave	1.08922	1.02022	0.99098	<b>1.29359</b>
SD	0.25804	<b>0.04926</b>	0.13153	0.31392
WRT	+	+	+	
UF8				
Ave	<b>5.50407</b>	0.93689	1.43347	0.85455
SD	1.46549	0.21879	0.30954	<b>0.03824</b>
WRT	–	+	+	
UF9				
Ave	<b>7.61837</b>	1.42589	1.79948	2.31543
SD	1.60882	<b>0.48994</b>	1.13938	1.70518
WRT	–	+	+	
UF10				
Ave	<b>15.76728</b>	6.34414	0.00301	1.27560
SD	2.35969	1.47038	<b>0.03404</b>	1.70290
WRT	–	+	+	
$W^+/W^-$	14/6	20/0	18/2	
$+/-/=$	7/3/0	10/0/0	9/1/0	

**Table 12** CEC problem statistics results for the S metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
UF1				
Ave	0.03565	0.05871	0.03584	<b>0.02627</b>
SD	0.00999	0.04308	0.02451	<b>0.00742</b>
WRT	+	+	+	
UF2				
Ave	0.01857	0.01443	0.01283	<b>0.01101</b>
SD	0.00314	0.00409	<b>0.00281</b>	0.00510
WRT	+	+	+	
UF3				
Ave	0.02696	0.07400	0.03743	<b>0.01001</b>
SD	0.01571	0.02562	0.01381	<b>0.00707</b>
WRT	+	+	+	
UF4				
Ave	0.01018	0.02014	<b>0.00620</b>	0.01025
SD	0.00176	0.01079	0.00257	<b>0.00139</b>
WRT	+	+	–	
UF5				
Ave	0.02576	0.08949	<b>0.01185</b>	0.22659
SD	0.03309	0.07070	<b>0.00377</b>	0.10846
WRT	+	+	–	
UF6				
Ave	0.04377	0.13105	<b>0.00569</b>	0.19548
SD	0.07466	0.18372	<b>0.00525</b>	0.07518
WRT	+	+	–	
UF7				
Ave	<b>0.01688</b>	0.02665	0.04893	0.04361
SD	<b>0.00830</b>	0.04571	0.05361	0.02058
WRT	–	+	+	
UF8				
Ave	0.27757	0.04431	0.03625	<b>0.03090</b>
SD	0.07907	<b>0.01164</b>	0.02026	0.01788
WRT	+	+	+	
UF9				
Ave	0.37998	0.06695	0.23247	<b>0.05123</b>
SD	0.08938	<b>0.02289</b>	0.02539	0.09431
WRT	+	+	+	
UF10				
Ave	1.09983	0.41590	0.50043	<b>0.16842</b>
SD	0.19502	<b>0.13872</b>	0.33063	0.21597
WRT	+	+	+	
$W^+/W^-$	18/2	20/0	14/6	
$+/-/=$	9/1/0	10/0/0	7/3/0	



**Table 13** Engineering problem statistics results for the IGD metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
<b>BNH</b>				
Ave	0.00107	0.00338	0.00369	<b>0.00095</b>
SD	0.00057	0.00115	<b>0.00030</b>	0.00333
WRT	+	+	+	
<b>CONSTR</b>				
Ave	<b>0.00052</b>	0.00074	0.00117	0.00075
SD	0.00036	0.00028	0.00023	<b>0.00020</b>
WRT	–	+	+	
<b>DISK BREAKER</b>				
Ave	<b>0.00059</b>	0.00072	0.00167	0.00326
SD	<b>0.00006</b>	0.00011	0.00024	0.00036
WRT	–	+	+	
<b>4-BAR TRUSS</b>				
Ave	0.02011	0.02141	0.02020	<b>0.02003</b>
SD	0.00004	0.00015	0.00008	<b>0.00002</b>
WRT	+	+	+	
<b>WELDED BEAM</b>				
Ave	<b>0.00060</b>	0.00134	0.00167	0.00142
SD	<b>0.00005</b>	0.00039	0.00047	0.00054
WRT	–	+	+	
<b>OSY</b>				
Ave	0.01466	0.00762	0.01336	<b>0.00687</b>
SD	0.00869	<b>0.00064</b>	0.00556	0.00258
WRT	+	+	+	
<b>SPEED REDUCER</b>				
Ave	0.06653	0.01424	0.03930	<b>0.00985</b>
SD	0.03131	0.00320	0.01257	<b>0.00231</b>
WRT	+	+	+	
<b>SRN</b>				
Ave	0.00045	0.00248	0.00140	<b>0.00024</b>
SD	0.00012	0.00106	0.00024	<b>0.00002</b>
WRT	+	+	+	
$W^+/W^-$	10/6	16/0	16/0	
$+/-/=$	5/3/0	8/0/0	8/0/0	

**Table 14** Engineering problem statistics results for the GD metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
<b>BNH</b>				
Ave	0.03389	0.06247	0.03159	<b>0.03024</b>
SD	0.00214	0.01565	0.00195	<b>0.00147</b>
WRT	+	+	+	
<b>CONSTR</b>				
Ave	0.00083	0.00089	0.00085	<b>0.00061</b>
SD	0.00003	0.00015	0.00015	<b>0.00012</b>
WRT	+	+	+	
<b>DISK BREAKER</b>				
Ave	<b>0.00230</b>	0.00433	0.00919	0.04542
SD	<b>0.00056</b>	0.00181	0.01395	0.04185
WRT	–	+	+	
<b>4-BAR TRUSS</b>				
Ave	14.09479	<b>11.69562</b>	13.45197	15.03955
SD	<b>0.51580</b>	2.33396	0.73413	1.37877
WRT	+	–	+	
<b>WELDED BEAM</b>				
Ave	0.01085	0.03921	0.02821	<b>0.00539</b>
SD	0.00200	0.03810	0.02939	0.00096
WRT	+	+	+	
<b>OSY</b>				
Ave	3.57651	<b>1.27359</b>	3.34079	6.07962
SD	2.52498	<b>0.40179</b>	2.55480	4.46173
WRT	+	–	+	
<b>SPEED REDUCER</b>				
Ave	18.39390	8.28890	15.34775	<b>5.92256</b>
SD	5.31010	1.61293	4.10279	<b>1.10406</b>
WRT	+	+	+	
<b>SRN</b>				
Ave	0.03162	<b>0.01674</b>	0.02839	0.07264
SD	0.01069	<b>0.00399</b>	0.01362	0.01051
WRT	+	–	+	
$W^+/W^-$	14/2	10/6	16/0	
$+/-/=$	7/1/0	5/3/0	8/0/0	

### 4.3 Discussion of the CEC-09 test function

This section focuses on the results obtained from the evaluations of the CEC-09 test functions. Tables 9 and 10 present the statistical outcomes for the IGD and GD metrics, respectively. Table 9 highlights explicitly that the

recommended MOGND0 algorithm demonstrates superior performance in terms of IGD metrics for UF1, UF2, UF4, UF9, and UF10. The statistical analysis of the IGD metric for the CEC-09 test functions is visualized in boxplot Fig. 7. It is worth discussing here that MOGND0 is really efficient for solving more challenging than two-objective

**Table 15** Engineering problem statistics results for the MS metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
BNH				
Ave	<b>1.00000</b>	0.87156	0.98678	<b>1.00000</b>
S D	0.09000	0.06696	<b>0.01173</b>	0.08778
WRT	=	+	+	
CONSTR				
Ave	0.99384	0.97790	0.95021	<b>0.99922</b>
SD	<b>0.00686</b>	0.02011	0.02024	0.03270
WRT	+	+	+	
DISK BREAKER				
Ave	0.99928	0.99582	1.00021	<b>1.06589</b>
SD	<b>0.00114</b>	0.01281	0.01393	0.20205
WRT	+	+	+	
4-BAR TRUSS				
Ave	1.48748	1.37874	1.45302	<b>1.48753</b>
SD	0.00055	0.04612	0.02948	<b>0.00001</b>
WRT	+	+	+	
WELDED BEAM				
Ave	1.00725	<b>1.11069</b>	1.10316	0.93094
SD	0.06105	0.10194	0.10980	<b>0.06025</b>
WRT	+	-	+	
OSY				
Ave	0.32390	0.66490	0.59830	<b>0.71947</b>
SD	0.34284	<b>0.02512</b>	0.41630	0.14521
WRT	+	+	+	
SPEED REDUCER				
Ave	0.00001	0.76707	0.38107	<b>0.77661</b>
SD	<b>0.00020</b>	0.02418	0.21697	0.03764
WRT	+	+	+	
SRN				
Ave	0.90900	0.70014	0.87800	<b>0.97434</b>
SD	0.04546	0.08018	0.03172	<b>0.01792</b>
WRT	+	+	+	
$W^+/W^-$	14/0	14/2	16/0	
$+/-/=$	7/0/1	7/1/0	8/0/0	

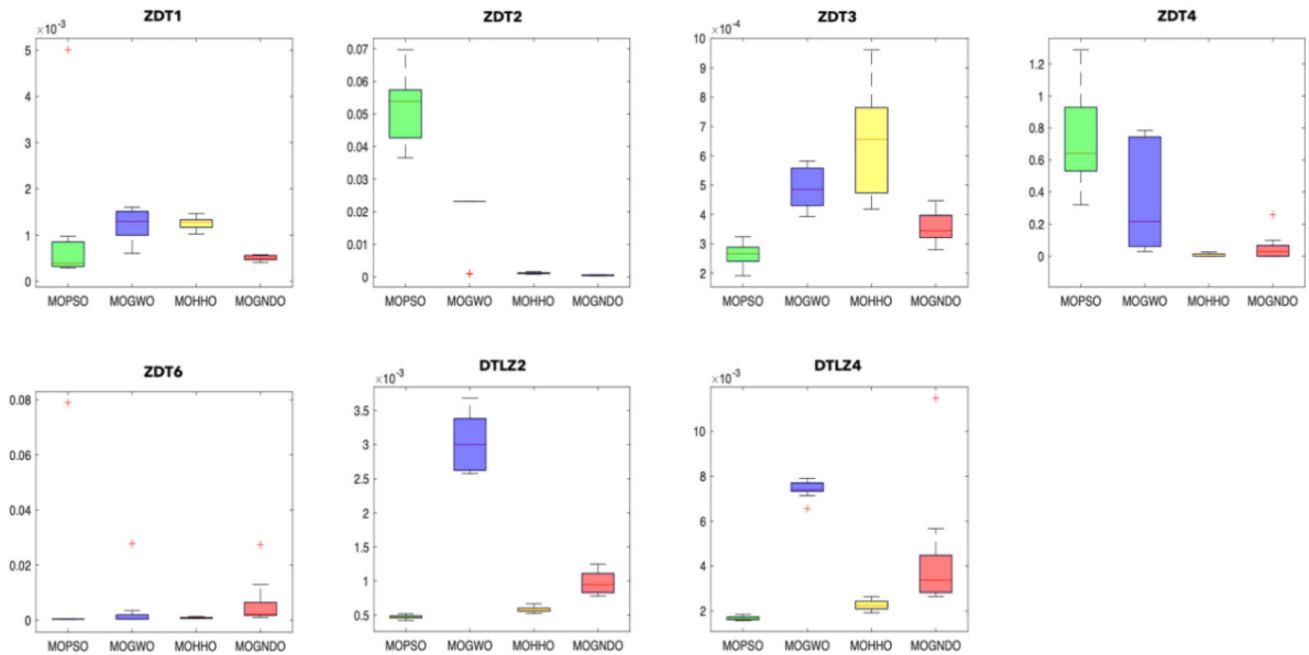
**Table 16** Engineering problem statistics results for the S metric

Functions	Algorithm			
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MOGND0</i>
BNH				
Ave	1.09006	1.74769	0.79799	<b>0.94528</b>
SD	<b>0.13631</b>	0.47352	0.40572	0.22767
WRT	+	+	+	
CONSTR				
Ave	0.05874	0.05489	<b>0.04275</b>	0.10070
SD	<b>0.00589</b>	0.00831	0.02570	0.03944
WRT	+	+	-	
DISK BREAKER				
Ave	0.11452	0.13331	<b>0.10633</b>	0.58348
SD	<b>0.01302</b>	0.01932	0.02281	0.57396
WRT	+	+	-	
4-BAR TRUSS				
Ave	5.36052	5.90453	5.99245	<b>5.26845</b>
SD	<b>0.26169</b>	1.74617	1.78906	0.66605
WRT	+	+	+	
WELDED BEAM				
Ave	0.23432	0.44716	<b>0.23361</b>	0.29580
SD	<b>0.02570</b>	0.21585	0.07086	0.06496
WRT	+	+	-	
OSY				
Ave	1.12775	1.62745	0.91658	<b>0.90137</b>
SD	1.46753	0.47312	0.66900	<b>0.05630</b>
WRT	+	+	+	
SPEED REDUCER				
Ave	31.0450	20.62414	9.15388	<b>7.01506</b>
SD	<b>3.00438</b>	3.18605	3.54455	5.74217
WRT	+	+	+	
SRN				
Ave	0.00142	2.64908	1.56020	<b>2.98903</b>
SD	<b>0.00074</b>	1.25584	0.55377	0.38569
WRT	+	+	+	
$W^+/W^-$	16/0	16/0	10/6	
$+/-/=$	8/0/0	8/0/0	5/3/0	

problems since UF9 and UF10 are tri-objective function benchmarks. The outcomes and analyses prove that the MOGND0 algorithm can offer really more competitive and appealing outcomes on the multi-objective examination functions concerning the obtained Pareto Front, as seen in Fig. 8. The Pareto optimal front for the UF5 test problem differs from the first four tests in that it has discontinuities, making it a challenge to find the Pareto optimal set. This is evident in Fig. 8, where the numerous discontinuous areas pose a problem for all algorithms. The Pareto front for the UF6 test problem has three distinct

regions and is quite similar to UF1. As shown in Fig. 8, MOGND0 has results that are closest to the Pareto front.

True and obtained Pareto fronts of UF8, UF9, and UF10 are seen in Fig. 9 from different perspectives. According to Fig. 9, MOGND0 has generated one of the best Pareto sets for the three objective test problems. The figure demonstrates that MOGND0 has a good coverage and convergence, whereas the other algorithms have low convergence rates. In addition, the Pareto optimal set shown in Fig. 9 confirms that MOGND0 has better coverage than the other algorithms. As depicted in Table 10, 60% (UF1, UF2, UF3,



**Fig. 4** Boxplot of the statistical results (IGD) for ZDT and DTLZ problems

UF4, UF8, and UF10) of all CEC-09 test functions for all algorithms in MOGNDO obtained the best results, which show that the distance between true and obtained Pareto Front is low in these examples.

By running 30 times on CEC-09 optimization problems independently, the MS metric for MOPSO, MOGWO, MOHHO, and the proposed method is listed in Table 11. It is evident that MOGNDO shows good results compared with the other mentioned methods. These outcomes indicate the MOGNDO gains the best results of 5 CEC-09 benchmarks out of 10. The average and standard deviation of the S performance metric are listed in Table 12. UF1, UF2, UF3, UF8, UF9, and UF10 have the best results in terms of average. Inspecting the results on UF8, UF9, and UF10, which have three objective functions, the MOGNDO algorithm obtained the first rank. It proves the proposed method will also be a good choice for more than two objective function problems. The results of the Wilcoxon rank sum test, presented in Tables 9, 10, 11 and 12, provide strong evidence that MOGNDO surpasses the other evaluated methods significantly.

#### 4.4 Discussion of the engineering problem results.

The last collection of examination functions is one of the most difficult ones and also consists of eight real engineering design problems. Equations (25) to (75) reveal that these problems have varied features. As a result, the very fit benchmarking the efficiency of the recommended MOGNDO algorithm. The different behavior of the best Pareto optimal fronts and the obtained Pareto fronts are shown compared to the true ones in Fig. 10. MOGNDO has

superior coverage and convergence among all the algorithms. Based on the four superior metrics, as well as its acceptable coverage and convergence factors, MOGNDO is better equipped to provide better solutions for the BNH, CONSTR, WELDED BEAM, and SRN design problems than the other algorithms.

The convergence of the MOGNDO algorithm in the mentioned design problems is nearly 100% similar to the true Pareto Front, as shown in this figure. Convergence is reasonable due to this, and coverage is exceptionally high and nearly uniform.

Constraint multi-objective test problem:

##### 4.3.1 CONSTR

There are two constraints and two design variables in this problem, which has a convex Pareto front.

$$\text{Minimize : } f_1(x) = x_1 \quad (25)$$

$$\text{Minimize : } f_2(x) = (1 + x_2)/x_1 \quad (26)$$

$$\text{where : } g_1(x) = 6 - (x_2 + 9x_1) \quad (27)$$

$$g_2(x) = 1 + (x_2 - 9x_1) \quad (28)$$

$$0.1 \leq x_1 \leq 1, 0 \leq x_2 \leq 5$$

##### 4.3.2 SRN

Srinivas and Deb [63] suggested a continuous Pareto optimal front for the next problem as follows:

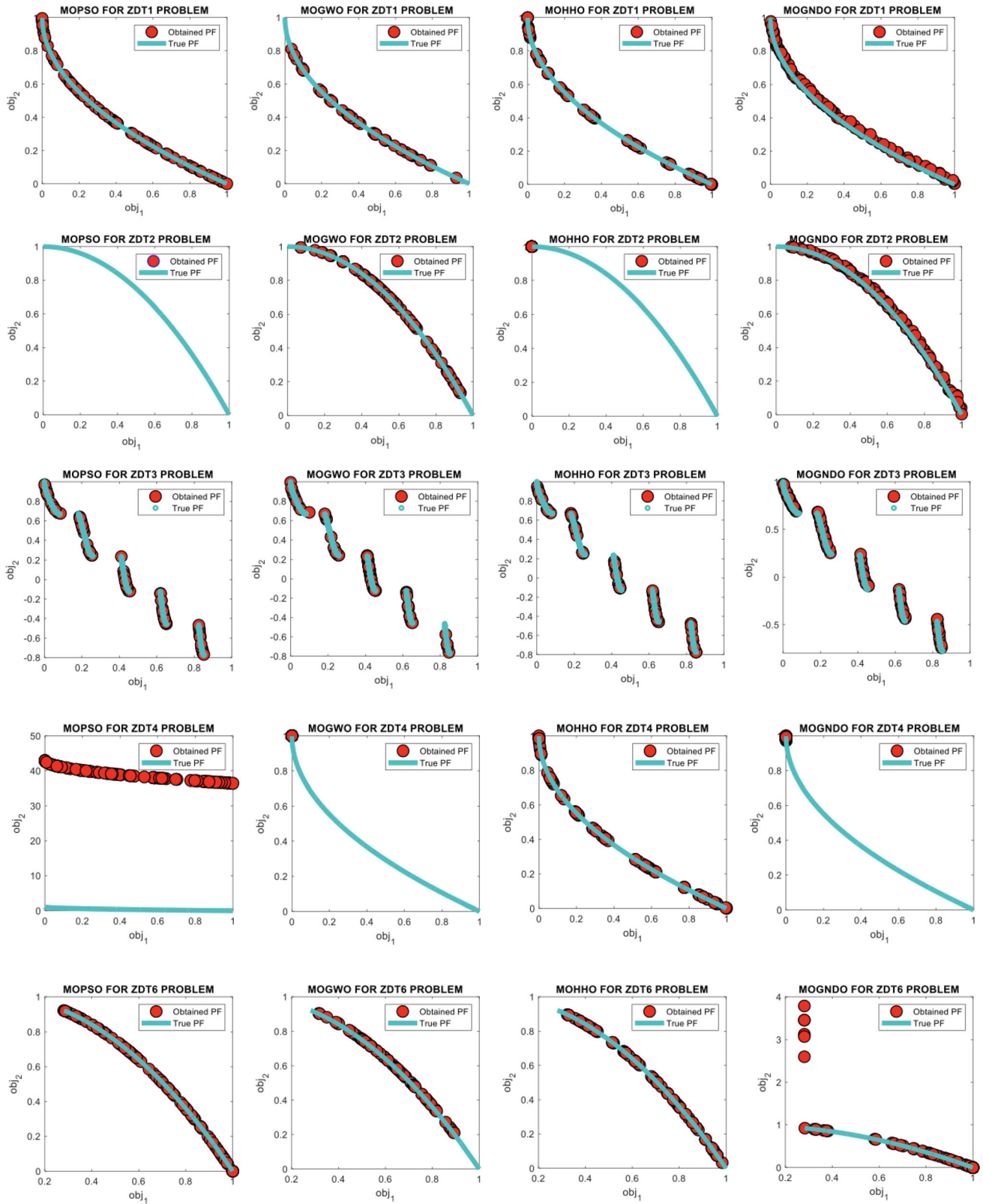


Fig. 5 True and obtained Pareto front for ZDT benchmarks

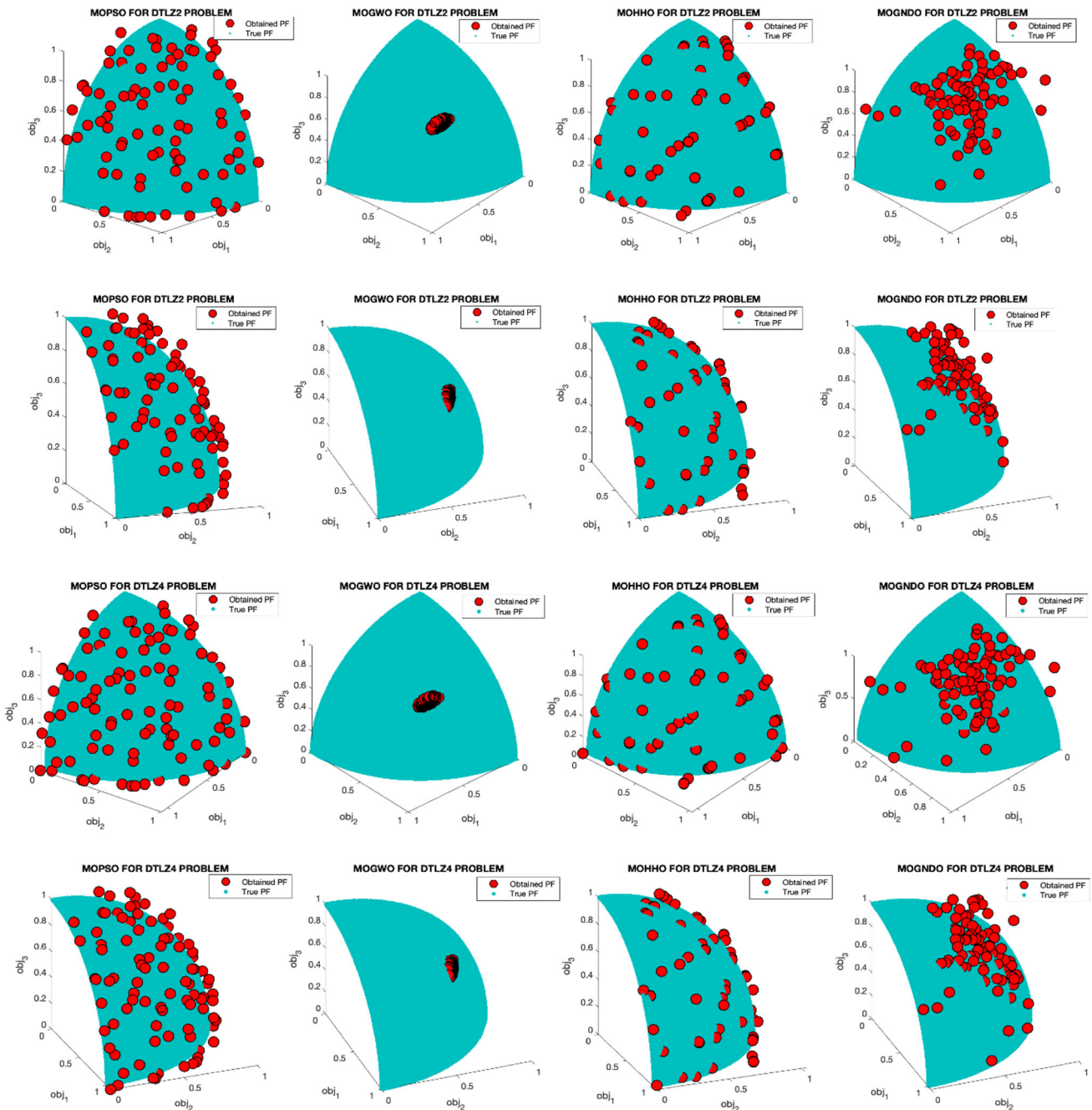


Fig. 6 True and obtained Pareto front for DTLZ2 and DTLZ4

$$\text{Minimize : } f_1(x) = 2 + (x_1 - -2)^2 + (x_2 - -1)^2 \quad (29)$$

$$\text{Minimize : } f_2(x) = 9x_1 - -(x_2 - -1)^2 \quad (30)$$

$$\text{where : } g_1(x) = x_1^2 + x_2^2 - 255 \quad (31)$$

$$g_2(x) = x_1 - 3x_2 + 10 \quad (32)$$

$$- 20 \leq x_1 \leq 20, -20 \leq x_2 \leq 20$$

### 4.3.3 BNH

Binh and Korn [64] were the first to propose this problem as follows:

$$\text{Minimize : } f_1(x) = 4x_1^2 + 4x_2^2 \quad (33)$$

$$\text{Minimize : } f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 \quad (34)$$

$$\text{where : } g_1(x) = (x_1 - 5)^2 + x_2^2 - 25 \quad (35)$$



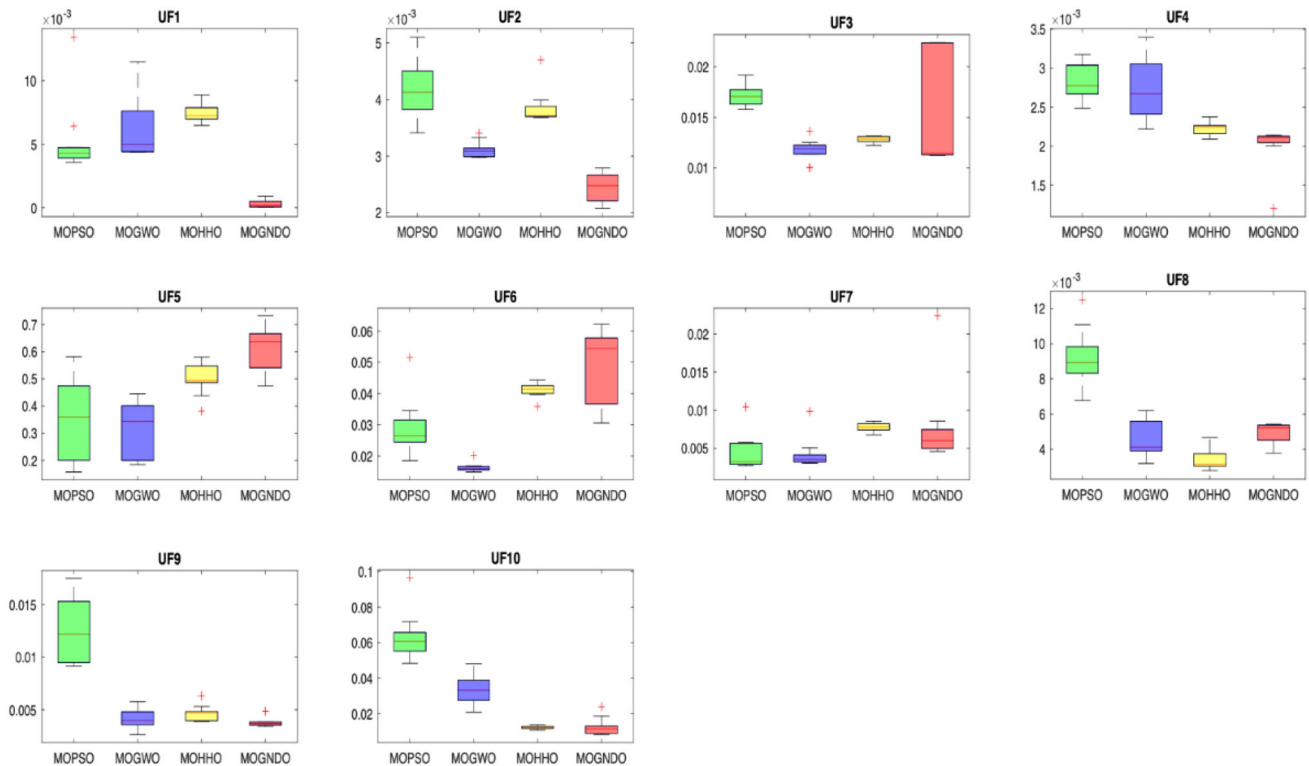


Fig. 7 Boxplot of the statistical results (IGD) for CEC-09 problems

$$g_2(x) = 7.7 - (x_1 - 8)^2 - (x_2 + 3)^2 \quad (36)$$

$$0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3$$

#### 4.3.4 OSY

Oszycza and Kundu [65] proposed five distinct regions for the OSY test issue. There are also six constraints and six design variables to consider as bellow:

$$\text{Minimize : } f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \quad (37)$$

Minimize :

$$f_2(x) = [25(x_1 - 2)^2 + (x_2 - 1)^2 + (x_3 - 1)^2(x_4 - 4)^2 + (x_5 - 1)^2] \quad (38)$$

$$\text{Where : } g_1(x) = 2 - x_1 - x_2 \quad (39)$$

$$g_3(x) = -2 - x_1 + x_2 \quad (40)$$

$$g_4(x) = -2 + x_1 - 3x_2 \quad (41)$$

$$g_5(x) = -4 + x_4 + (x_3 - 3)^2 \quad (42)$$

$$g_6(x) = 4 - x_6 - (x_5 - 3)^3 \quad (43)$$

$$0 \leq x_1 \leq 10, 0 \leq x_2 \leq 10, 1 \leq x_3 \leq 5$$

$$0 \leq x_4 \leq 6, 1 \leq x_5 \leq 5, 0 \leq x_6 \leq 10$$

Constraint multi-objective engineering problems:

#### 4.3.5 Four-bar truss design problem

The 4-bar truss design problem [66], in which the structural volume ( $f_1$ ) and displacement ( $f_2$ ) of a 4-bar truss should be minimized, is a well-known problem in the structural optimization field. There are four design variables ( $x_1$ - $x_4$ ) connected to the cross-sectional area of members 1, 2, 3, and 4, as shown in the equations below:

Minimize :

$$f_1(x) = 200 \times (2 \times x(1) + \text{sqrt}(2 \times x(2)) + \text{sqrt}(x(3)) + x(4)) \quad (44)$$

Minimize :

$$f_2(x) = 0.01 \times \left( \frac{2}{x(1)} \right) + \left( \frac{2 \times \text{sqrt}(2)}{x(2)} \right) - ((2 \times \text{sqrt}(2))/x(3)) + (2/x(1)) \quad (45)$$

$$\text{Where : } 1 \leq x_1 \leq 3, 1.4142 \leq x_2 \leq 3$$

$$1.4142 \leq x_3 \leq 3, 1 \leq x_4 \leq 3$$

#### 4.3.6 Welded beam design problem

Ray and Liew [67] suggested four constraints for the welded beam design issue. In this issue, the fabrication cost ( $f_1$ ) and beam deflection ( $f_2$ ) of a welded beam should be

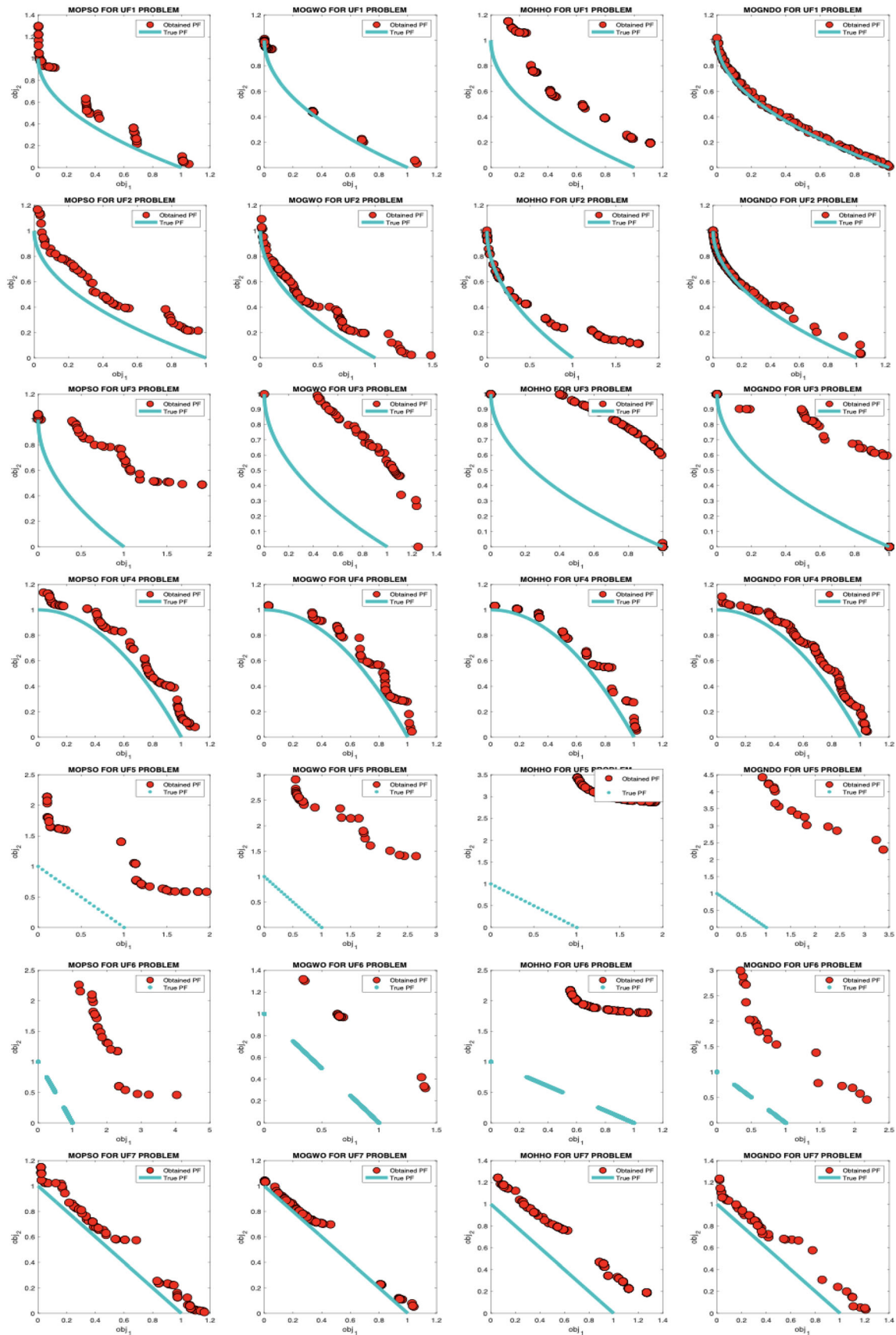
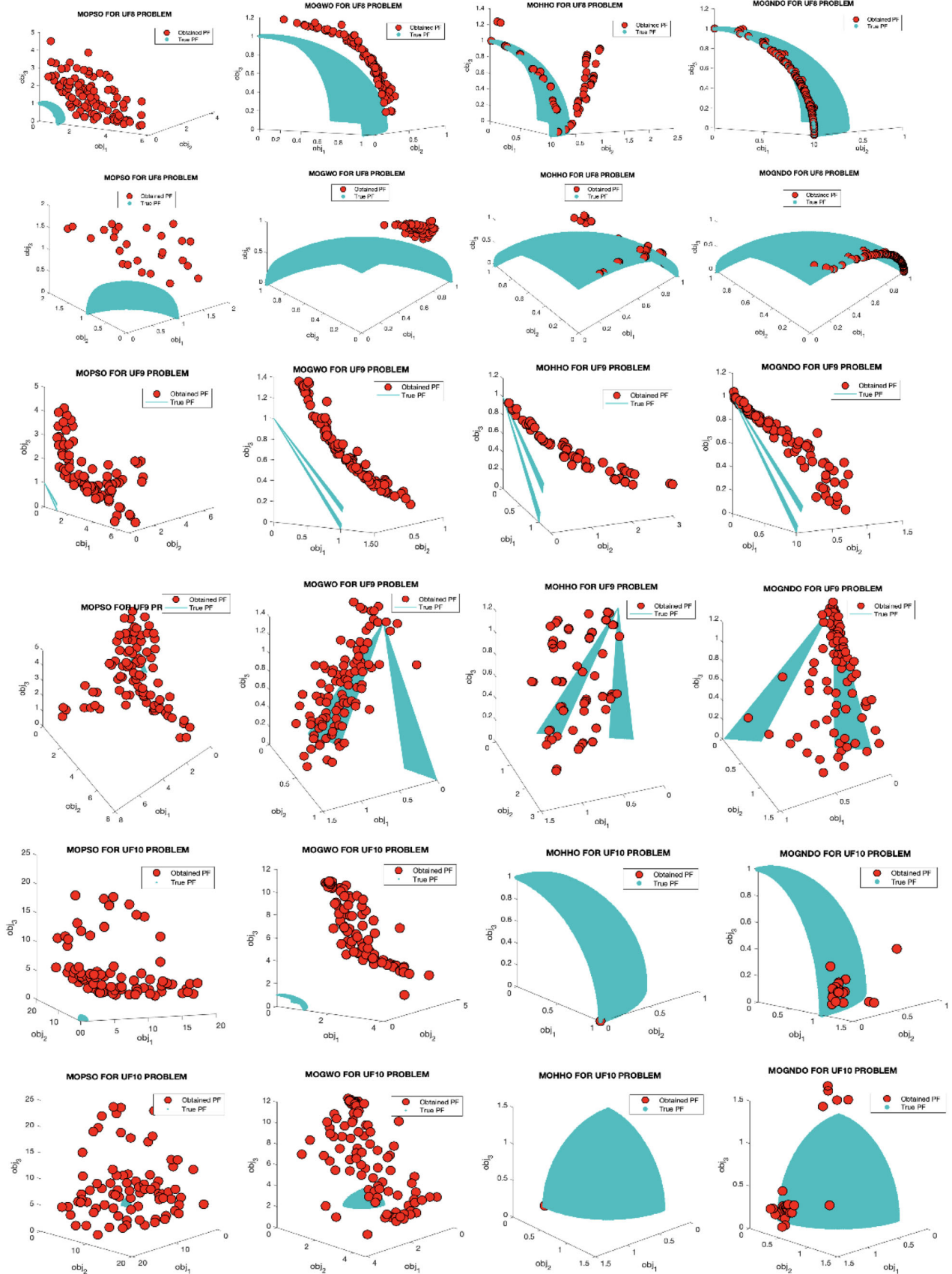


Fig. 8 True and obtained Pareto front for CEC-09 benchmarks (UF1-7)



◀**Fig. 9** True and obtained Pareto front for CEC-09 benchmarks (UF8-10)

minimized. The thickness of the weld ( $x_1$ ), the length of the clamped bar ( $x_2$ ), the height of the bar ( $x_3$ ) and the thickness of the bar ( $x_4$ ) are the four design variables.

Minimize :

$$f_1(x) = 1.10471 \times x(1)^2 \times x(2) + 0.04811 \times x(3) \times x(4) \times (14 + x(2)) \tag{46}$$

Minimize :

$$f_2(x) = 65856000 / (30 \times 10^6 \times x(4) \times x(3)^3) \tag{47}$$

Where :  $g_1(x) = \tau - 13600$  (48)

$g_2(x) = \sigma - 30000$  (49)

$g_3(x) = x(1) - x(4)$  (50)

$g_4(x) = 6000 - P$   
 $0.125 \leq x_1 \leq 5, 0.1 \leq x_2 \leq 10$  (51)

$0.1 \leq x_3 \leq 10, 0.125 \leq x_4 \leq 5$

$$q = 6000 * \left(14 + \frac{x(2)}{2}\right); D$$

$$= \text{sqrt} \left( \frac{x(2)^2}{4} + \frac{x(1) + x(3))^2}{4} \right) \tag{52}$$

$J = 2$   
 $* \left( x(1) * x(2) * \text{sqrt}(2) * \left( \frac{x(2)^2}{12} + \frac{(x(1) + x(3))^2}{4} \right) \right)$  (53)

$$\alpha = \frac{6000}{\text{sqrt}(2) * x(1) * x(2)} \tag{54}$$

$$\beta = Q * \frac{D}{J} \tag{55}$$

**4.3.7 Disk brake design problem [67]**

Ray and Liew [67] proposed the disc brake design issue, which has multiple constraints. Stopping time ( $f_1$ ) and brake mass ( $f_2$ ) for a disc brake are the two objectives to be minimized. The inner radius of the disc ( $x_1$ ), the outer radius of the disc ( $x_2$ ), the engaging force ( $x_3$ ), and the number of friction surfaces ( $x_4$ ) as well as four constraints, are shown in the following equations.

Minimize :

$$f_1(x) = 4.9 \times (10)^{(-5)} \times (x(2))^{(2)} - x(1)^{(2)} \times (x(4) - 1) \tag{56}$$

Minimize :

$$f_2(x) = (9.82 \times (10)^{(6)}) \times (x(2))^{(2)} - x(1)^{(2)}) / ((x(2))^{(3)} - x(1)^{(3)}) \times x(4) \times x(3)) \tag{57}$$

$g_1(x) = 20 + x(1) - x(2)$  (58)

$g_2(x) = 2.5 + (x(4) + 1) - 30$  (59)

$g_3(x) = (x(3)) / (3.14 \times (x(2)^2 - x(1)^2)^2) - 0.4$  (60)

$g_4(x) = (2.22 \times (10)^{(-3)} \times x(3) \times (x(2)^3 - x(1)^3)) / ((x(2)^2 - x(1)^2)^2) - 1$  (61)

$g_5(x) = 900 - (2.66 \times (10)^{(-2)} \times x(3) \times x(4) \times (x(2)^3 - x(1)^3)) / ((x(2)^2 - x(1)^2)^2)$  (62)

Where :  $55 \leq x_1 \leq 80, 75 \leq x_2 \leq 110$   
 $1000 \leq x_3 \leq 3000, 2 \leq x_4 \leq 20$

**4.3.8 Speed reducer design problem**

The weight ( $f_1$ ) and stress ( $f_2$ ) of a speed reducer should be minimized in the speed reducer design issue, which is well-known in the field of mechanical engineering [66, 68]. There are seven design variables: gear face width ( $x_1$ ), teeth module ( $x_2$ ), number of teeth of pinion ( $x_3$  integer variable), distance between bearings 1 ( $x_4$ ), distance between bearings 2 ( $x_5$ ), diameter of shaft 1 ( $x_6$ ), and diameter of shaft 2 ( $x_7$ ) as well as eleven constraints.

Minimize :  $f_1(x) = 0.7854 \times x(1) \times x(2)^2 \times (3.3333 \times x(3)^2 + 14.9334 \times x(3)) \dots - 43.0934 - 1.508 \times x(1) \times (x(6)^2 + x(7)^2)$  (63)

Minimize :

$$f_2(x) = ((\text{sqrt}(((745 * x(4))/x(2) * x(3))))^2 + 19.9e6)/(0.1 * x(6)^3)) \tag{64}$$

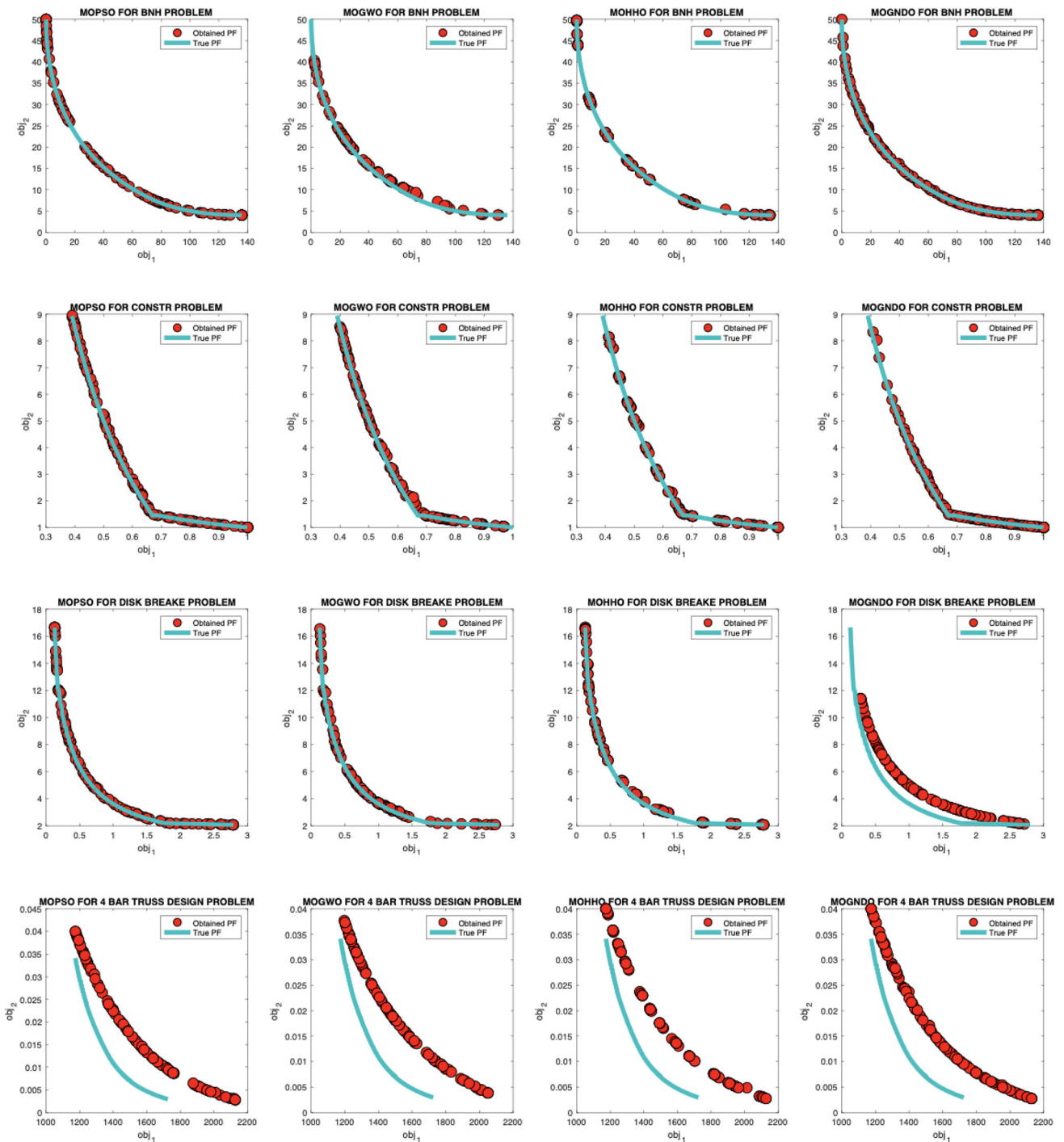


Fig. 10 True and obtained Pareto front for the engineering design problem

$$where : \quad g_1(x) = 27/(x(1) \times x(2)^2 \times x(3)) - 1 \quad (65)$$

$$g_2(x) = 397.5/(x(1) \times x(2)^2 \times x(3)^2) - 1 \quad (66)$$

$$g_3(x) = (1.93 \times (x(4)^3)/(x(2) \times x(3) \times (x(6)^4) - 1 \quad (67)$$

$$g_4(x) = (1.93 \times (x(5)^3)/(x(2) \times x(3) \times (x(7)^4) - 1 \quad (68)$$



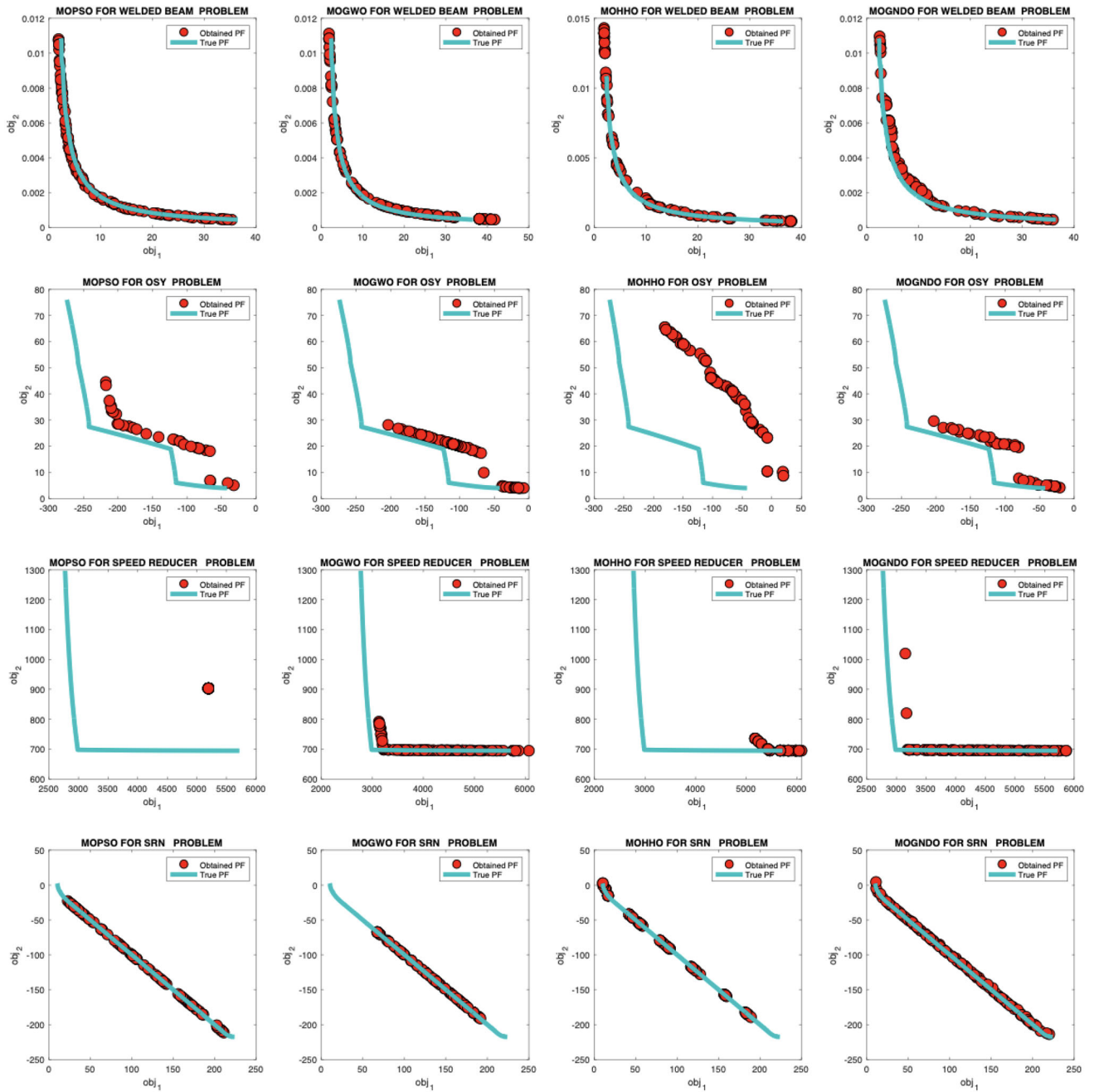
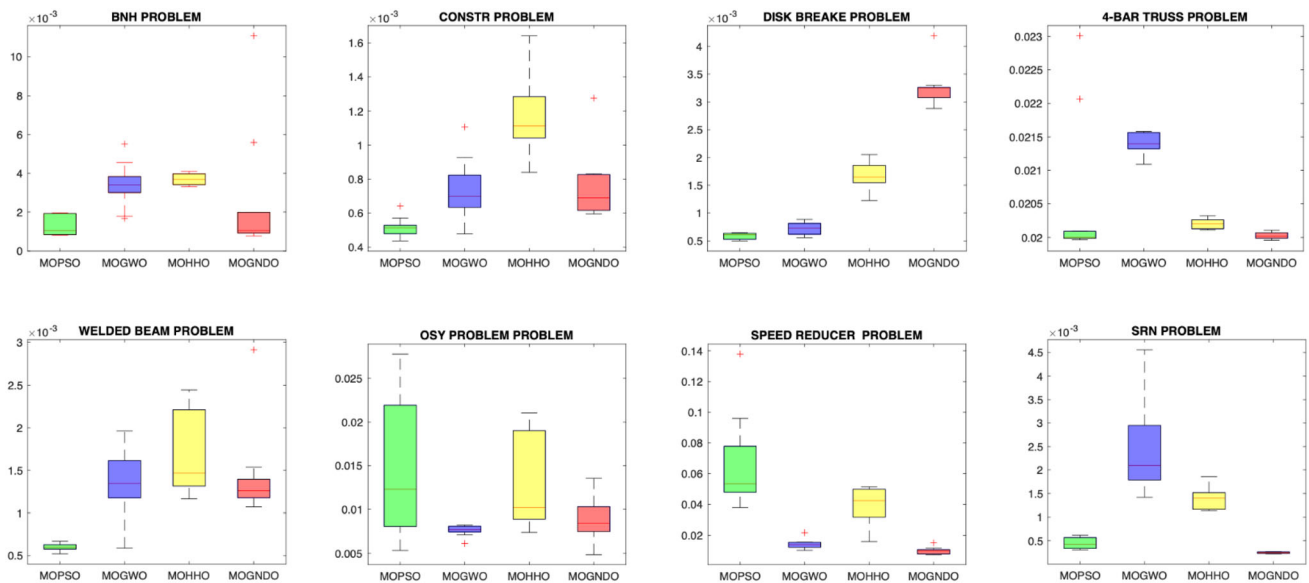


Fig. 10 continued

$$g_5(x) = ((\text{sqrt}((745 \times x(4))/x(2) \times x(3)))^2 + 16.9e6)/(110 \times x(6)^3)) - 1 \tag{69}$$

$$g_6(x) = ((\text{sqrt}((745 \times x(4))/x(2) \times x(3)))^2 + 157.5e6)/(85 \times x(7)^3)) - 1 \tag{70}$$



**Fig. 11** Boxplot of the statistical results (IGD) for engineering problems

$$g_7(x) = ((x(2) \times x(3))/40) - 1 \quad (71)$$

$$\tau = \text{sqrt}\left(\alpha^2 + 2 \times \alpha \times \beta \times \frac{x(2)}{2 \times D} + \beta^2\right) \quad (72)$$

$$\sigma = \frac{504000}{x(4) \times x(3)^2} \quad (73)$$

$$\text{tmpf} = 4.013 \times \frac{30 \times 10^6}{196} \quad (74)$$

$$P = \text{tmpf} \times \text{sqrt}\left(x(3)^2 \times \frac{x(4)^6}{36}\right) \times \left(1 - x(3) \times \frac{\text{sqrt}\left(\frac{30}{48}\right)}{28}\right) \quad (75)$$

The performance of MOGNDO, in comparison with MOPSO, MOGWO, and MOHHO, shows a high degree of competitiveness, particularly in the best Pareto front-of-rest problems. The outcomes of these algorithms are quantitatively compared using a similar set of efficiency metrics, and the corresponding results are displayed in Tables 13, 14, 15 and 16.

Most of the constrained test function results, as depicted in Tables 13 and 14, highlight the superior performance of MOGNDO over the other three algorithms. This dominance is further evident in the Generational Distance (GD) and Inverted Generational Distance (IGD) metrics, suggesting outstanding convergence. Table 14 GD performance measurement results distinctly indicate the superiority of the newly proposed algorithm over MOPSO,

MOGWO, and MOHHO. Additionally, the boxplot of the IGD metric for engineering problems, displayed in Fig. 11, verifies that MOGNDO outperformed the others in 75% of the engineering problems based on the IGD metric.

According to Tables 15 and 16, the statistical results for the MS and S metrics show that the proposed MOGNDO algorithm competes well with other algorithms. These tables list the average and standard deviation results for the MS and S performance metrics. The MOGNDO algorithm excels in terms of the MS performance metric. Furthermore, in some problems, the coverage of the proposed method on engineering test functions surpasses that of other algorithms. As evidenced by the results in these tables, MOGNDO can surpass other methods in identifying Pareto optimal fronts with distinct regions.

The results of examinations show that the MOGNDO algorithm has high convergence and coverage. The high convergence of this algorithm is inherited by its standard version and leader selection mechanism. The results showed that this method's merit is high coverage and is guaranteed by archive maintenance and leader feature selection function. Tables 13, 14, 15, and 16 present the results of a Wilcoxon rank sum test, demonstrating without a shadow of a doubt that the MOGNDO is the most effective of the other tested methods.

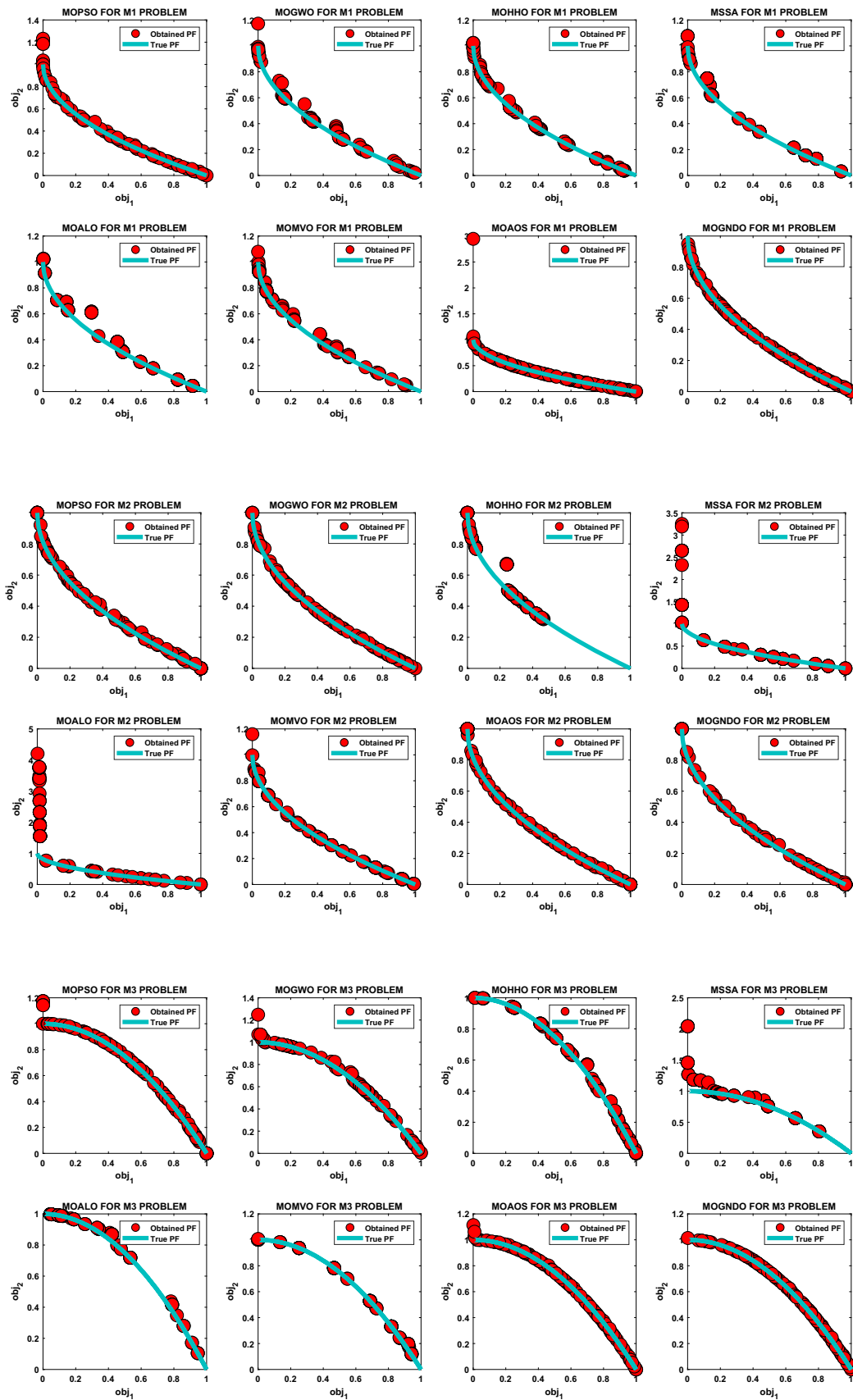
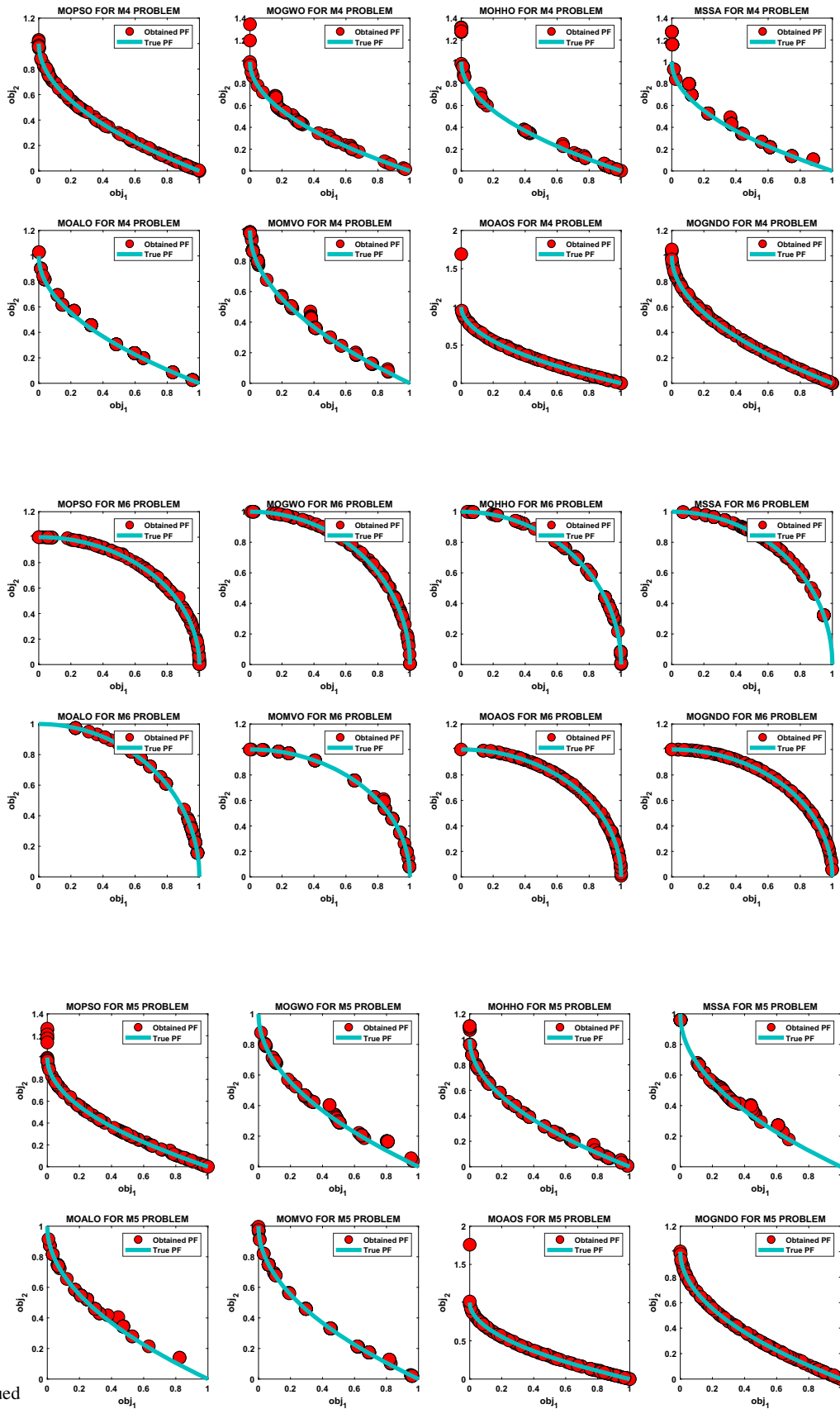
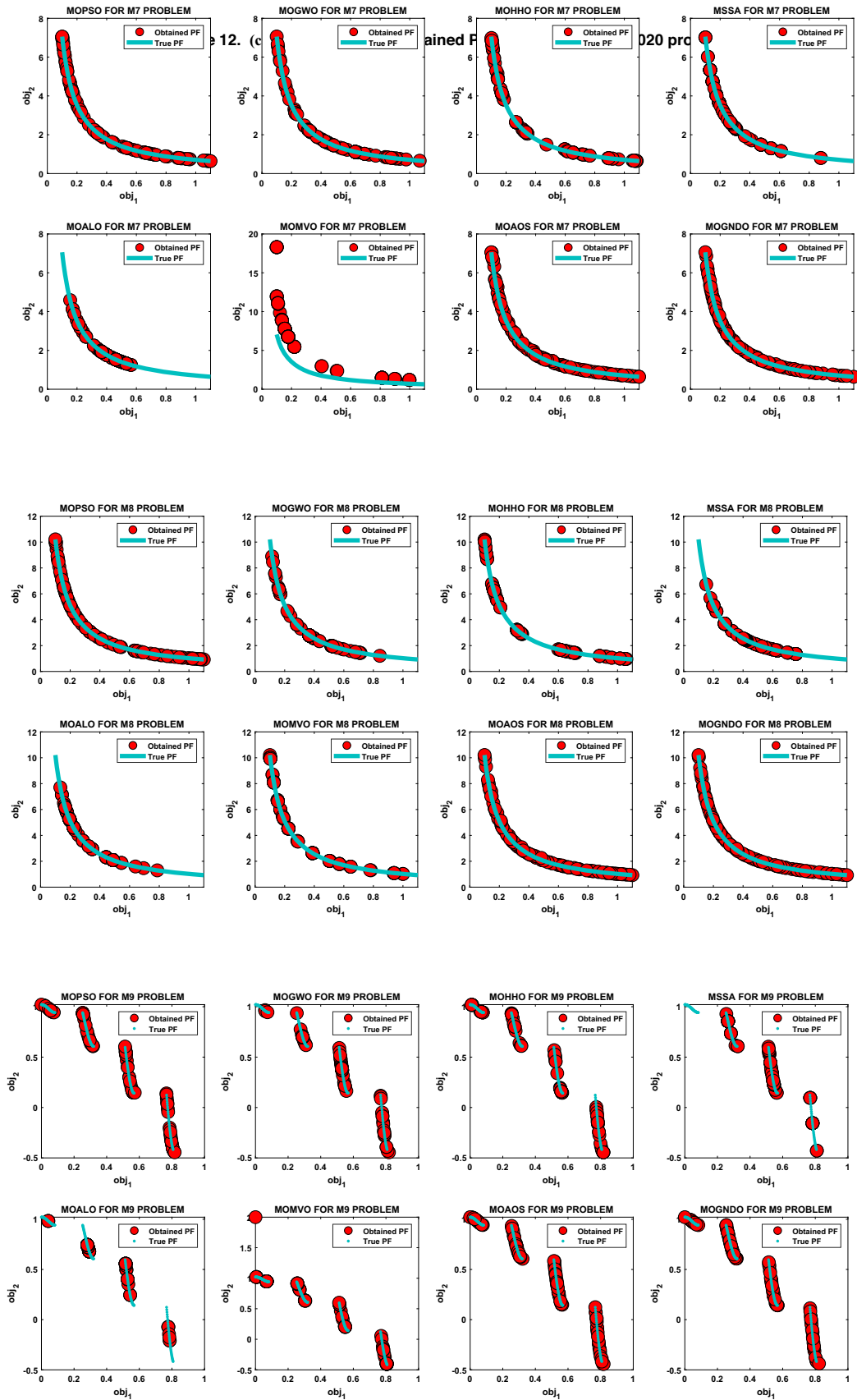


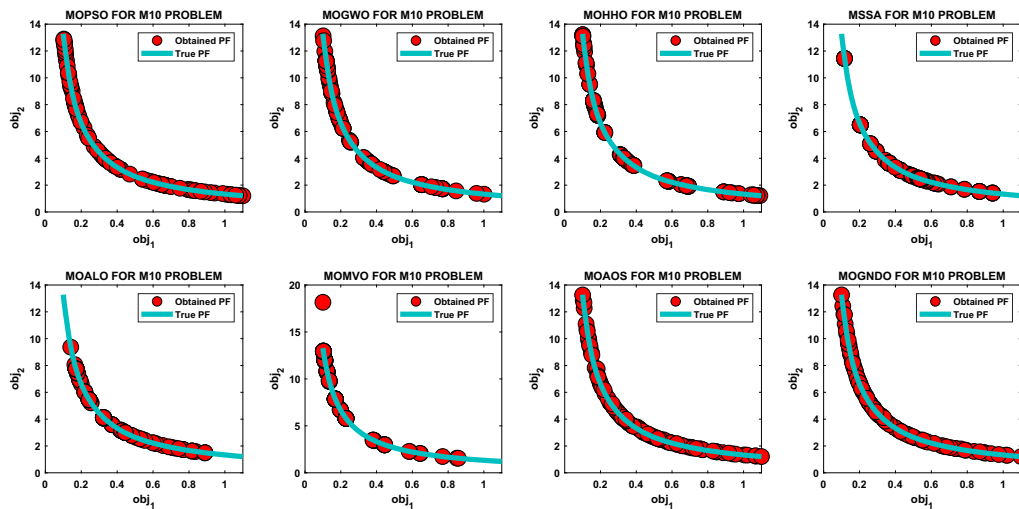
Fig. 12 True and obtained Pareto front for CEC-2020 problems



◀ Fig. 12 continued



◀ Fig. 12 continued



◀ Fig. 12 continued

#### 4.4 Discussion of the CEC20 problem results

This section assesses MOGNDO, the proposed multi-objective approach, and its ability to address various issues at CEC-2020. A comparison is made between MOPSO, MOGWO, MOHHO, MSSA, MOALO, MOMVO, and MOAOS. Information about the MMO evaluation functions is provided in [62]. The MMO evaluation process encompasses linear, non-linear, convex, and concave problems.

Figure 12 compares the ideal Pareto fronts against those achieved for the CEC-2020 benchmark problems labeled M1 through M10. The graphical representation clearly shows the disparity between the theoretical optimal Pareto fronts and those derived through our algorithm is remarkably small. For each of the ten benchmark problems presented, the results obtained from our algorithm align closely with the true Pareto fronts, showcasing a consistency not observed in the results from alternative algorithms, which exhibit notable variations for each problem scenario.

Table 17 elaborates on the statistical evaluation of the performance based on the GD metric. Here, the MOGNDO method stands out by achieving the best average GD values for a majority of the problems, precisely seven out of ten. The MOAOS follows as the second-best method, leading in mean values for three problems, while other evaluated methods did not produce notable results in this comparison. Furthermore, when examining the standard deviation of the GD performance, our method demonstrates a superior capability in pinpointing eight out of ten optimal solutions, outperforming MOAOS and MOHHO. The latter methods only manage to secure a commendable position in one instance for MOAOS, based on SD, underlining the

robustness of our current approach in identifying solutions that closely adhere to the ideal outcomes.

Table 18 provides an overview of the IGD metric's statistical results, indicating the MOGNDO method's dominance in obtaining the best average IGD values across eight of the ten problems. When considering the standard deviation of these results, MOGNDO again excels by securing the best outcomes in six instances. MOAOS is recognized for its performance according to the standard deviation of IGD, securing the second rank, whereas other methods lag behind in average and standard deviation measures.

The assessment continues with Table 19, which details the statistics for the S metric, highlighting that our method identifies the optimal average values in eight instances and excels in standard deviation measures for six instances. This demonstrates the method's effectiveness in ensuring diverse yet accurate solution sets. Other methods like MOPSO and MOHHO are noted for finding a couple and one optimal standard deviation value, respectively, but they do not match the overall efficacy of our approach.

Finally, Table 20 summarizes the performance evaluation based on the MS metric, where MOGNDO again leads by achieving the highest number of optimal average and standard deviation values across the benchmarks. This superiority is affirmed by results from the Wilcoxon rank sum test across Tables 17, 18, 19 and 20, solidifying MOGNDO's position as significantly more efficient in navigating and solving the CEC-2020 benchmark problems compared to the competing methods.

Following is a summary of the experiment's findings:

- As shown in Tables 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 and 20, the experimental results show that the MOGNDO algorithm has excellent coverage and convergence capabilities. Both the archive and non-

**Table 17** Mathematic problem statistics results for the GD metric

Functions	Algorithm							
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MSSA</i>	<i>MOALO</i>	<i>MOMVO</i>	<i>MOAOS</i>	<i>MOGND</i>
<b>M1:MMF1</b>								
Ave	0.00432	0.00277	0.01197	0.00471	0.00341	0.01026	0.00670	<b>0.00249</b>
SD	0.00341	0.00296	0.00971	0.00391	0.00220	0.01943	0.00662	<b>0.00106</b>
WRT	+	+	+	+	+	+	+	
<b>M2:MMF2</b>								
Ave	0.00092	0.00042	0.00120	0.04935	0.04536	0.01476	<b>0.00031</b>	0.00078
SD	0.00040	0.00006	0.00135	0.06612	0.05900	0.03383	0.00007	<b>0.00002</b>
WRT	+	+	+	+	+	+	–	
<b>M3:MMF4</b>								
Ave	0.00505	0.00549	0.00589	0.01193	0.01232	0.00253	0.00712	<b>0.00225</b>
SD	0.00357	0.00565	<b>0.00332</b>	0.01345	0.01954	0.00567	0.00819	0.01002
WRT	+	+	+	+	+	+	+	
<b>M4:MMF5</b>								
Ave	0.00358	0.00702	0.02826	0.02656	0.00713	0.00273	0.00816	<b>0.00259</b>
SD	0.00707	0.00684	0.01704	0.02810	0.01194	0.00314	0.00721	<b>0.00276</b>
WRT	+	+	+	+	+	+	+	
<b>M5:MMF7</b>								
Ave	0.00217	0.00182	0.00988	0.00184	0.00153	0.00126	0.00284	<b>0.00115</b>
SD	0.00279	0.00048	0.00507	0.00065	0.00077	0.00069	<b>0.00024</b>	0.00469
WRT	+	+	+	+	+	+	+	
<b>M6:MMF8</b>								
Ave	0.00061	0.02742	0.01084	0.03746	0.08258	0.01600	0.00055	<b>0.00043</b>
SD	0.00006	0.11960	0.02304	0.08765	0.17631	0.04433	0.00007	<b>0.00005</b>
WRT	+	+	+	+	+	+	+	
<b>M7:MMF10</b>								
Ave	0.00377	0.01595	0.04521	0.00571	0.01912	0.19716	0.00260	<b>0.00253</b>
SD	0.00031	0.05592	0.09149	0.01403	0.04955	0.09511	0.00037	<b>0.00027</b>
WRT	+	+	+	+	+	+	+	
<b>M8:MMF11</b>								
Ave	0.00540	0.00363	0.00568	0.00269	0.00275	0.02926	<b>0.00126</b>	0.00338
SD	0.00064	0.00093	0.00075	0.00185	0.00121	0.05908	0.00070	<b>0.00044</b>
WRT	+	+	+	+	+	+	–	
<b>M9:MMF12</b>								
Ave	0.00053	0.00047	0.00051	0.00070	0.00118	0.00389	0.00054	<b>0.00043</b>
SD	0.00007	0.00008	0.00006	0.00099	0.00329	0.00416	0.00005	<b>0.00004</b>
WRT	+	+	+	+	+	+	+	
<b>M10:MMF13</b>								
Ave	0.00234	0.00175	0.00237	0.00138	0.00143	0.03714	<b>0.00107</b>	0.00183
SD	0.00025	0.00066	0.00019	0.00091	0.00053	0.04818	0.00025	<b>0.00015</b>
WRT	+	+	+	+	+	+	–	
$W^+/W^-$	20/0	20/0	20/0	20/0	20/0	20/0	14/6	
$+/-/=$	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	7/3/0	



**Table 18** Mathematic problem statistics results for the IGD metric

Functions	Algorithm							
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MSSA</i>	<i>MOALO</i>	<i>MOMVO</i>	<i>MOAOS</i>	<i>MOGND0</i>
<b>M1:MMF1</b>								
Ave	0.00095	0.00260	0.00276	0.00448	0.00355	0.00304	0.00061	<b>0.00057</b>
SD	0.00016	0.00030	0.00026	0.00177	0.00064	0.00109	0.00006	<b>0.00004</b>
WRT	+	+	+	+	+	+	+	
<b>M2:MMF2</b>								
Ave	0.00171	<b>0.00094</b>	0.00269	0.00379	0.00222	0.00188	0.00102	0.00227
SD	0.00053	0.00026	0.00049	0.00105	0.00061	0.00021	<b>0.00023</b>	0.00098
WRT	+	–	+	+	+	+	+	
<b>M3:MMF4</b>								
Ave	0.00088	0.00203	0.00259	0.00465	0.00431	0.00415	0.00069	<b>0.00061</b>
SD	0.00013	0.00086	0.00081	0.00240	0.00231	0.00149	0.00015	<b>0.00010</b>
WRT	+	+	+	+	+	+	+	
<b>M4:MMF5</b>								
Ave	0.00063	0.00158	0.00227	0.00283	0.00223	0.00198	0.00042	<b>0.00040</b>
SD	0.00014	0.00030	0.00063	0.00061	0.00037	0.00029	0.00003	<b>0.00002</b>
WRT	+	+	+	+	+	+	+	
<b>M5:MMF7</b>								
Ave	0.00060	0.00185	0.00173	0.00477	0.00416	0.00229	0.00042	<b>0.00041</b>
SD	0.00009	0.00025	0.00040	0.00218	0.00228	0.00042	<b>0.00003</b>	0.00004
WRT	+	+	+	+	+	+	+	
<b>M6:MMF8</b>								
Ave	0.00049	0.00076	0.00146	0.00229	0.00279	0.00231	0.00048	<b>0.00042</b>
SD	0.00005	0.00021	0.00014	0.00064	0.00143	0.00032	0.00010	<b>0.00004</b>
WRT	+	+	+	+	+	+	+	
<b>M7:MMF10</b>								
Ave	0.00087	0.00180	0.00308	0.00663	0.01050	0.00985	0.00084	<b>0.00071</b>
SD	0.00012	0.00087	0.00040	0.00287	0.00467	0.00233	<b>0.00006</b>	0.00015
WRT	+	+	+	+	+	+	+	
<b>M8:MMF11</b>								
Ave	0.00084	0.00380	0.00269	0.00892	0.01328	0.00414	0.00074	<b>0.00068</b>
SD	0.00013	0.00202	0.00065	0.00428	0.00717	0.00130	0.00019	<b>0.00013</b>
WRT	+	+	+	+	+	+	+	
<b>M9:MMF12</b>								
Ave	0.00057	0.00170	0.00105	0.00960	0.00798	0.00320	0.00053	<b>0.00047</b>
SD	0.00007	0.00074	0.00023	0.00747	0.00474	0.00084	<b>0.00003</b>	0.00005
WRT	+	+	+	+	+	+	+	
<b>M10:MMF13</b>								
Ave	0.00048	0.00160	0.00188	0.00439	0.00851	0.00277	<b>0.00034</b>	0.00044
SD	0.00007	0.00077	0.00043	0.00319	0.00239	0.00086	0.00009	<b>0.00008</b>
WRT	+	+	+	+	+	+	–	
$W^+ / W^-$	20/0	18/2	20/0	20/0	20/0	20/0	18/2	
+/-/=	10/0/0	9/1/0	10/0/0	10/0/0	10/0/0	10/0/0	9/1/0	

**Table 19** Mathematic problem statistics results for the S metric

Functions	Algorithm							
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MSSA</i>	<i>MOALO</i>	<i>MOMVO</i>	<i>MOAOS</i>	<i>MOGND0</i>
<b>M1:MMF1</b>								
Ave	0.01283	0.02015	0.01202	0.01742	0.01135	0.02695	0.05310	<b>0.01027</b>
SD	<b>0.00192</b>	0.01828	0.00724	0.01203	0.00630	0.03799	0.05246	0.03360
WRT	+	+	+	+	+	+	+	
<b>M2:MMF2</b>								
Ave	0.01704	0.01138	0.01163	0.01693	0.06958	0.09397	0.01618	<b>0.01051</b>
SD	0.00331	0.00214	0.00401	0.01170	0.10175	0.28142	0.00123	<b>0.00110</b>
WRT	+	+	+	+	+	+	+	
<b>M3:MMF4</b>								
Ave	0.01408	0.02913	<b>0.00964</b>	0.01607	0.01888	0.01111	0.04611	0.05561
SD	0.00665	0.02644	<b>0.00319</b>	0.01021	0.01161	0.01997	0.05149	0.05422
WRT	+	+	-	+	+	+	+	
<b>M4:MMF5</b>								
Ave	0.01627	0.03014	0.01259	0.02066	0.01751	0.02427	0.06485	<b>0.01120</b>
SD	0.01788	0.01702	0.00250	0.02796	0.01549	0.03187	0.05158	<b>0.00196</b>
WRT	+	+	+	+	+	+	+	
<b>M5:MMF7</b>								
Ave	0.01170	0.01906	0.01273	0.01136	0.01228	0.03100	0.02713	<b>0.01016</b>
SD	0.00205	0.00587	0.00649	0.00698	0.00627	0.00478	0.02212	<b>0.00191</b>
WRT	+	+	+	+	+	+	+	
<b>M6:MMF8</b>								
Ave	0.01018	0.01622	0.00849	0.01218	0.03591	0.16707	0.00951	<b>0.00757</b>
SD	<b>0.00088</b>	0.01993	0.00336	0.00673	0.07048	0.44251	0.00208	0.00187
WRT	+	+	+	+	+	+	+	
<b>M7:MMF10</b>								
Ave	0.06569	0.08182	0.04964	0.05304	0.07248	0.13529	0.05963	<b>0.04584</b>
SD	0.00587	0.01858	0.01902	0.02913	0.04273	0.17111	0.01051	<b>0.00093</b>
WRT	+	+	+	+	+	+	+	
<b>M8:MMF11</b>								
Ave	0.06776	0.10962	0.08428	0.07500	0.06044	0.16776	0.07602	<b>0.07445</b>
SD	0.05818	0.03815	0.01711	0.06970	0.03524	0.29591	0.01087	<b>0.01003</b>
WRT	+	+	+	+	+	+	+	
<b>M9:MMF12</b>								
Ave	0.01104	0.01359	0.01097	0.01055	0.00720	0.02963	0.00869	<b>0.00829</b>
SD	0.00141	0.00507	0.00121	0.00762	0.01027	0.03453	0.00095	<b>0.00091</b>
WRT	+	+	+	+	+	+	+	
<b>M10:MMF13</b>								
Ave	0.08452	0.14062	<b>0.05877</b>	0.08827	0.08249	0.27118	0.09202	0.11004
SD	0.01080	0.03907	0.02099	0.07310	0.03804	0.28914	<b>0.01045</b>	0.03168
WRT	+	+	-	+	+	+	+	
$W^+ / W^-$	20/0	20/0	16/4	20/0	20/0	20/0	20/0	
+/-/=	10/0/0	10/0/0	8/2/0	10/0/0	10/0/0	10/0/0	10/0/0	

**Table 20** Mathematic problem statistics results for the MS metric

Functions	Algorithm							
	<i>MOPSO</i>	<i>MOGWO</i>	<i>MOHHO</i>	<i>MSSA</i>	<i>MOALO</i>	<i>MOMVO</i>	<i>MOAOS</i>	<i>MOGND0</i>
<b>M1:MMF1</b>								
Ave	1.10102	1.08055	1.18416	0.99793	0.98744	1.16757	1.36164	<b>1.38216</b>
SD	0.07762	0.15887	0.20901	0.08508	0.05765	0.37172	0.39062	<b>0.06629</b>
WRT	+	+	+	+	+	+	+	
<b>M2:MMF2</b>								
Ave	1.00000	1.00000	1.00000	1.76484	<b>2.31815</b>	1.68734	1.00000	0.99479
SD	<b>0.00000</b>	<b>0.00000</b>	<b>0.00000</b>	1.03604	1.62026	1.95090	<b>0.00000</b>	<b>0.00000</b>
WRT	=	=	=	+	-	+	=	
<b>M3:MMF4</b>								
Ave	1.13708	1.17601	1.09484	1.09506	1.16260	1.01655	1.36134	<b>1.51770</b>
SD	0.12218	0.21583	<b>0.04344</b>	0.25945	0.49183	0.14625	0.42309	0.48666
WRT	+	+	+	+	+	+	+	
<b>M4:MMF5</b>								
Ave	1.11008	1.24573	1.53080	1.37191	1.10965	1.04177	1.45189	<b>1.59036</b>
SD	0.26860	0.24312	0.35816	0.43900	0.29093	0.22198	0.42076	<b>0.19200</b>
WRT	+	+	+	+	+	+	+	
<b>M5:MMF7</b>								
Ave	1.04370	0.94314	1.17927	0.77612	0.77803	0.95502	1.14042	<b>1.33231</b>
SD	0.06681	0.03254	0.09733	0.15152	0.12102	<b>0.03177</b>	0.15302	0.23171
WRT	+	+	+	+	+	+	+	
<b>M6:MMF8</b>								
Ave	0.99484	1.60437	1.18288	1.79952	<b>3.12249</b>	2.01156	0.99522	0.99902
SD	0.00379	2.81011	0.42595	2.17966	4.79795	3.03677	0.00535	<b>0.00196</b>
WRT	+	+	+	+	-	+	+	
<b>M7:MMF10</b>								
Ave	0.99962	1.00572	1.09775	0.82084	0.77244	1.45234	0.99786	<b>1.50009</b>
SD	0.00143	0.14770	0.27115	0.12721	0.21374	0.26215	0.00579	<b>0.00121</b>
WRT	+	+	+	+	+	+	+	
<b>M8:MMF11</b>								
Ave	0.99978	0.88170	0.97312	0.70722	0.55842	1.06549	0.99804	<b>1.08277</b>
SD	<b>0.00055</b>	0.06674	0.00856	0.10053	0.13715	0.21975	0.00366	0.03497
WRT	+	+	+	+	+	+	+	
<b>M9:MMF12</b>								
Ave	1.01428	0.96975	0.99646	0.80791	0.81452	1.08214	1.01521	<b>1.11335</b>
SD	0.00274	0.02686	0.01460	0.11526	0.11459	0.17556	0.00136	<b>0.00119</b>
WRT	+	+	+	+	+	+	+	
<b>M10:MMF13</b>								
Ave	0.99879	0.91838	0.97968	0.70201	0.54678	0.99125	<b>0.99652</b>	0.99547
SD	0.00162	0.04679	0.01101	0.12979	0.09708	0.14647	0.00754	<b>0.00154</b>
WRT	+	+	+	+	+	+	-	
$W^+ / W^-$	18/0	18/0	18/0	20/0	16/4	20/0	16/2	
+/-/=	9/0/1	9/0/1	9/0/1	10/0/0	8/2/0	10/0/0	8/1/1	

dominated sorting maintenance methods contribute to MOGNDO's high coverage, but high convergence is a privilege from the GNDO algorithm.

- In the same way that other archive-based MO algorithms do, the MOGNDO makes use of the archive memory in order to store the non-dominated solutions discovered through the optimization process.
- Compared to other MO-like algorithms, the MOGNDO's superior performance can be attributed to its use of the original GNDO update mechanism, allowing it to investigate a greater number of non-dominated solutions while maintaining a high convergence and coverage rate.

## 5 Conclusion and future works

This research developed an early multi-objective variant of MOGNDO, which incorporated an archive to store non-dominated solutions and a selection mechanism to identify the “best” solutions for the GNDO algorithm. The archive had a crucial role in preserving and updating the solutions, while the recommended leader selection feature allowed the MOGNDO algorithm to demonstrate excellent coverage and convergence simultaneously. The algorithm was evaluated using 25 diverse mathematical and engineering design problems, highlighting its effectiveness in achieving a balance between the exploration and exploitation stages. Metrics such as GD, IGD, S, and MS were employed to assess the dominance and performance of the proposed method.

Qualitative outcomes documenting the most effective Pareto optimal front identified in 30 runs were also documented to validate these results. The proposed algorithm was compared with highly esteemed algorithms such as MOPSO, MOGWO, and MOHHO to validate these results. This comparison demonstrated that our algorithm offered a highly competitive performance relative to the mentioned algorithms. The results and findings thus suggested that advantages are held by the proposed algorithm over existing multi-objective algorithms, positioning it as an appealing solution for multi-objective optimization problems.

Future research will involve the development of a binary variant of the MOGNDO algorithm to address a broader spectrum of complex, real-world problems. Another prospective contribution lies in the creation of a version of the proposed algorithm that considers multiple objectives simultaneously. Moreover, future work is encouraged to utilize MOGNDO for a range of other engineering design problems, such as truss structures, real-time applications, and the development of structural health assessment.

**Acknowledgements** The authors gratefully acknowledge the financial support from the National Science Foundation I/U-CRC Center for Integration of Composites into Infrastructure (CICI) under grant #1916342. The authors express their sincere gratitude to the Inter-American Cement Federation (FICEM) for their invaluable support and assistance during the internship period.

**Author contributions** N.K Methodology, Writing – original draft, Software, E.K Methodology, Writing – original draft, B. A, Investigation, Validation, S.K, Investigation, Visualization, P. M, Formal analysis, Validation, W.Z, Data curation, F.S.G, Investigation, S.M, Supervision

**Funding** The authors have not disclosed any funding.

**Data availability** The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**, 398–417 (2008)
2. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006)
3. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948. IEEE (1995)
4. Amiri, M.H., et al.: Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm. *Sci. Rep.* **14**, 5032 (2024)
5. Abdollahzadeh, B., Gharehchopogh, F.S., Khodadadi, N., Mirjalili, S.: Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Adv. Eng. Softw.* **174**, 103282 (2022)
6. El-Kenawy, E.-S.M., et al.: Al-Biruni Earth Radius (BER) metaheuristic search optimization algorithm. *Comput. Syst. Sci. Eng.* **45**, 1917–1934 (2023)
7. Abdollahzadeh, B., et al. Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. *Cluster Comput.* (2024). <https://doi.org/10.1007/s10586-023-04221-5>

8. Kaveh, A., Talatahari, S., Khodadadi, N.: Stochastic paint optimizer: theory and application in civil engineering. *Eng. Comput.* **38**, 1921–1952 (2020)
9. Kaveh, A., Khodadadi, N., Azar, B.F., Talatahari, S.: Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections. *Eng. Comput.* **37**, 2521–2541 (2020)
10. Talatahari, S., Azizi, M.: Chaos Game Optimization: a novel metaheuristic algorithm. *Artif. Intell. Rev.* **54**, 917–1004 (2021)
11. Khodadadi, N., Vaclav, S., Mirjalili, S.: Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints. *IEEE Access* **10**, 16188–16208 (2022)
12. Degertekin, S.O., Lamberti, L., Ugur, I.B.: Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. *Appl. Soft Comput.* **70**, 903–928 (2018)
13. Kaveh, A., Talatahari, S., Khodadadi, N.: Hybrid invasive weed optimization-shuffled frog-leaping algorithm for optimal design of truss structures. *Iran. J. Sci. Technol. Trans. Civ. Eng.* **44**, 405–420 (2019)
14. Gandomi, A.H., Talatahari, S., Yang, X., Deb, S.: Design optimization of truss structures using cuckoo search algorithm. *Struct. Des. Tall Spec. Build.* **22**, 1330–1349 (2013)
15. Khodadadi, N., Mirjalili, S.: Truss optimization with natural frequency constraints using generalized normal distribution optimization. *Appl. Intell.* **52**, 10384–10397 (2022)
16. Degertekin, S.O., Tutar, H., Lamberti, L.: School-based optimization for performance-based optimum seismic design of steel frames. *Eng. Comput.* **37**, 3283–3297 (2020)
17. Gholizadeh, S., Danesh, M., Gheytratmand, C.: A new Newton metaheuristic algorithm for discrete performance-based design optimization of steel moment frames. *Comput. Struct.* **234**, 106250 (2020)
18. Kaveh, A., Talatahari, S., Khodadadi, N.: The hybrid invasive weed optimization-shuffled frog-leaping algorithm applied to optimal design of frame structures. *Period. Polytech. Civ. Eng.* **63**, 882–897 (2019)
19. Al-Tashi, Q., et al.: Moth-Flame optimization algorithm for feature selection: a review and future trends. In: *Handbook of Moth-Flame Optimization Algorithm*, pp. 11–34 (2022)
20. Mirjalili, S.M., Davar, S., Khodadadi, N., Mirjalili, S.: Design optimization of photonic crystal filter using Moth-Flame Optimization Algorithm. In: *Handbook of Moth-Flame Optimization Algorithm*, pp. 313–322. CRC Press (2022)
21. Kaveh, A., Khodadadi, N., Talatahari, S.: A comparative study for the optimal design of steel structures using Csa and Accs algorithms. *Iran Univ. Sci. Technol.* **11**, 31–54 (2021)
22. Tayfur, B., Yilmaz, H., Daloglu, A.T.: Hybrid tabu search algorithm for weight optimization of planar steel frames. *Eng. Optim.* **53**, 1369–1383 (2020)
23. Moshtaghzadeh, M., Bakhtiari, A., Izadpanahi, E., Mardanpour, P.: Artificial Neural Network for the prediction of fatigue life of a flexible foldable origami antenna with Kresling pattern. *Thin-Walled Struct.* **174**, 109160 (2022)
24. Moshtaghzadeh, M., Bakhtiari, A., Mardanpour, P.: Artificial Neural Network-based Finite Element method for assessing fatigue and stability of an origami-inspired structure. *Eng. Struct.* **272**, 114965 (2022)
25. Abdelhamid, A.A., et al.: Classification of Monkeypox images based on transfer learning and the Al-Biruni Earth Radius Optimization Algorithm. *Mathematics* **10**, 3614 (2022)
26. Eid, M.M., et al.: Meta-heuristic optimization of LSTM-based deep network for boosting the prediction of Monkeypox cases. *Mathematics* **10**, 3845 (2022)
27. El-Kenawy, E.-S.M., et al.: Advanced dipper-throated metaheuristic optimization algorithm for digital image watermarking. *Appl. Sci.* **12**, 10642 (2022)
28. Schaffer, J.D., Grefenstette, J.J.: Multi-objective learning via genetic algorithms. In: *Ijcai*, vol. 85, pp. 593–595. Citeseer (1985)
29. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative case study. In: *International Conference on Parallel Problem Solving from Nature*, pp. 292–301. Springer (1998)
30. Coello, C.A.C., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **8**, 256–279 (2004)
31. Khodadadi, N., Soleimani Gharehchopogh, F., Mirjalili, S.: MOAVOA: a new multi-objective artificial vultures optimization algorithm. *Neural Comput. Appl.* **34**, 20791–20829 (2022)
32. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**, 712–731 (2007)
33. Khodadadi, N., Talatahari, S., Dadras Eslamlou, A.: MOTEO: a novel multi-objective thermal exchange optimization algorithm for engineering problems. *Soft Comput.* **26**, 6659–6684 (2022)
34. Aljebreen, M., Alohal, M.A., Mahgoub, H., Aljameel, S.S., Alsumayt, A., Sayed A.: Multi-objective seagull optimization algorithm with deep learning-enabled vulnerability detection for secure cloud environments. *Sensors* **23**, 9383 (2023)
35. Khodadadi, N., Abualigah, L., Mirjalili, S.: Multi-objective Stochastic Paint Optimizer (MOSPO). *Neural Comput. Appl.* (2022). <https://doi.org/10.1007/s00521-022-07405-z>
36. Abbass, H.A., Sarker, R., Newton, C.: PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 2, pp. 971–978 (2001)
37. Khodadadi, N., Mirjalili, S.M., Mirjalili, S.: Multi-objective moth-flame optimization algorithm for engineering problems. In: *Handbook of Moth-Flame Optimization Algorithm*, pp. 79–96. CRC Press (2022)
38. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191 (2017)
39. Zhao, W., et al.: An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Comput. Methods Appl. Mech. Eng.* **398**, 115223 (2022)
40. Mirjalili, S., Jangir, P., Saremi, S.: Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. *Appl. Intell.* **46**, 79–95 (2017)
41. Khodadadi, N., Abualigh, L., El-Kenawy, E.S.M., Snasel, V., Mirjalili, S.: An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems. *IEEE Access* **10**, 106673–106698 (2022)
42. Mirjalili, S., Saremi, S., Mirjalili, S.M., dos S. Coelho, L.S.: Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert Syst. Appl.* **47**, 106–119 (2016)
43. Nouhi, B., Khodadadi, N., Azizi, M., Talatahari, S., Gandomi, A.H.: Multi-Objective Material Generation Algorithm (MOMGA) for optimization purposes. *IEEE Access* **10**, 107095–107115 (2022)
44. Mirjalili, S., Jangir, P., Mirjalili, S.Z., Saremi, S., Trivedi, I.N.: Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowl. Based Syst.* **134**, 50–71 (2017)
45. Yüzgeç, U., Kusoglu, M.: Multi-objective Harris Hawks optimizer for multiobjective optimization problems. *BSEU J. Engi. Res. Technol.* **1**, 31–41 (2020)
46. Zhang, Y., Jin, Z., Mirjalili, S.: Generalized normal distribution optimization and its applications in parameter extraction of

- photovoltaic models. *Energy Convers. Manag.* **224**, 113301 (2020)
47. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997)
  48. Edgeworth, F.Y.: *Mathematical Psychics*. McMaster University Archive for the History of Economic Thought (1881)
  49. Pareto, V.: *Cours D'économie Politique*, vol. 1. Librairie Droz (1964)
  50. Coello, C.A.C.: Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Front. Comput. Sci. China* **3**, 18–30 (2009)
  51. Ngatchou, P., Zarei, A., El-Sharkawi, A.: Pareto multi objective optimization. In: *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*, pp. 84–91. IEEE (2005)
  52. Zhou, A., et al.: Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evol. Comput.* **1**, 32–49 (2011)
  53. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002)
  54. Coello, C.A.C., Lechuga, M.S.: MOPSO: a proposal for multiple objective particle swarm optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 2, pp. 1051–1056 (2002)
  55. Coello, C.A.C., Lamont, G.B.: *Applications of Multi-objective Evolutionary Algorithms*, vol. 1. World Scientific (2004)
  56. Nebro, A.J., Durillo, J.J., Coello, C.A.C.: Analysis of leader selection strategies in a multi-objective particle swarm optimizer. In: *2013 IEEE Congress on Evolutionary Computation*, pp. 3153–3160. IEEE (2013)
  57. Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the Pareto archived evolution strategy. *Evol. Comput.* **8**, 149–172 (2000)
  58. Deb, K.: An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **186**, 311–338 (2000)
  59. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput.* **8**, 173–195 (2000)
  60. Sierra, M.R. & Coello, C.A.C.: Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance. In: *International Conference on Evolutionary Multi-criterion Optimization*, pp. 505–519. Springer (2005)
  61. Arcuri, A., Fraser, G.: Parameter tuning or default values? An empirical investigation in search-based software engineering. *Empir. Softw. Eng.* **18**, 594–623 (2013)
  62. Liang, J.J., Qu, B.Y., Gong, D.W., Yue, C.T.: Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session on Multimodal Multiobjective Optimization. *Computational Intelligence Laboratory*, Zhengzhou University (2019)
  63. Srinivasan, N., Deb, K.: Multi-objective function optimisation using non-dominated sorting genetic algorithm. *Evol. Comput.* **2**, 221–248 (1994)
  64. Binh, T.T., Korn, U.: MOBES: A multiobjective evolution strategy for constrained optimization problems. In: *The Third International Conference on Genetic Algorithms (Mendel 97)*, vol. 25, p. 27. Citeseer (1997)
  65. Osyczka, A., Kundu, S.: A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Struct. Optim.* **10**, 94–99 (1995)
  66. Coello, C.A.C., Pulido, G.T.: Multiobjective structural optimization using a microgenetic algorithm. *Struct. Multidiscip. Optim.* **30**, 388–403 (2005)
  67. Ray, T., Liew, K.M.: A swarm metaphor for multiobjective design optimization. *Eng. Optim.* **34**, 141–153 (2002)
  68. Kurpati, A., Azarm, S., Wu, J.: Constraint handling improvements for multiobjective genetic algorithms. *Struct. Multidiscip. Optim.* **23**, 204–213 (2002)

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



**Nima Khodadadi** is currently a research assistant, and he is studying for his 2nd Ph.D. at the University of Miami (UM), Miami, USA. He has published over 45 publications with over 400 citations. Nima graduated with a bachelor's degree in Civil Engineering and obtained a master's degree in Structural Civil Engineering from the University of Tabriz (One of the ten top universities in Iran). Previously, he got his 1st Ph.D. at the Iran University of Science

and Technology (IUST). His main research interests include the field of wind engineering and steel structures, particularly in the experimental and numerical investigation of steel-braced frames. In addition, he has been actively involved in engineering optimization, especially in evolutionary algorithms. Using comprehensive finite element analyses, he has also investigated the use of differently shaped sections in real steel frames. Recently, he has been engaged in research in single and multi-objective engineering optimization, especially in solving large-scale and practical structural design problems.



**Ehsan Khodadadi** received the B.Sc. and M.Sc. degrees in plant breeding with a specialization in molecular genetics and genetic engineering from the Islamic Azad University of Ardabil in 2007 and 2009, respectively, and the Ph.D. degree in molecular genetics and genetic engineering from the University of Ilam, in 2017. He is currently pursuing a Ph.D. degree in material science and engineering. He joined the University of Arkansas in 2023. He has made

significant contributions to the scientific community throughout his career. He has published over 30 papers covering diverse fields, including bioinformatics, proteomics, post-translational modifications, systems biology, and optimization algorithms. These publications highlight his expertise and research interests across various domains. In addition to his research accomplishments, he was an Assistant Professor at the Islamic Azad University of Tabriz. This role allowed him to share his knowledge and passion for genetics and biotechnology by teaching science courses.





ious projects.

**Benjamin Abdollahzadeh** has a master's degree in computer engineering from Azad University of Urmia, Urmia, Iran. He is a very active and motivated researcher and he has started his research since early 2019 and now he is the senior researcher in advanced AI and machine learning methods in the Department of Computer Engineering, Azad University of Urmia. His main interest is meta-heuristic algorithms and he is currently researching various

Astronautics (AIAA), the American Society of Mechanical Engineering (ASME), and the American Physics Society (APS).



**Weiguo Zhao** received the PhD degree from School of Electrical Engineering, Hebei University of Technology in 2016. He has been a professor in Electric Engineering at Hebei University of Engineering since 2021. He is currently involved in the area of soft computing, intelligent fault diagnosis, and intelligent water conservancy. He has published over 30 research papers in international journals.



easy-to-understand manner. He is a reviewer for Computers, Materials & Continua Journal, IEEE Access, and other journals. His research interests include artificial intelligence, machine learning, optimization, deep learning, digital marketing, and data science.

**El-Sayed M. El-Kenawy** is an assistant professor at Delta Higher Institute for Engineering & Technology (DHET), Mansoura, Egypt. He published more than 35 papers with more than 1200 citations and an H-index of 22. He has launched and pioneered independent research programs. He motivates and inspires his students in different ways by providing a thorough understanding of various computer concepts. He explains complex concepts in an



For give more information, please visit his google scholar pages.

**Farhad Soleimanian Gharahchopogh** was born in 1979 in Iran. He received his B.S. in Computer Engineering from Shabestar Branch, Islamic Azad University, Iran, in 2002, his M.S. in Computer Engineering from Cukurova University, Adana/Turkey, in 2011 and his PhD degree in Computer Engineering from Hacettepe University, Ankara/Turkey in 2015. He has been an assistant professor in Computer Engineering at Urmia Branch, Islamic Azad University, Urmia, IRAN, since 2015. He is researching metaheuristic algorithms and their application issues. And he published more than 150 papers in international journals and conferences.



to serve as the assistant director of the Flutter System Research Center and to work with professor A. V. Balakrishnan on his continuum theory of aeroelasticity and its application on experimental flow-induced vibration and power harvesting. Before he joined Georgia Tech, he earned another M.S. from Sharif University of Technology in Tehran, Iran, and worked as a research scientist at the Institute of Aeroelasticity at the German Aerospace Center (DLR) in Göttingen, Germany, where he focused on the experimental aeroelasticity of high-aspect-ratio wings. A regular contributor to the Journal of Aircraft, the Journal of Fluids and Structures, the Journal of Non-linear Dynamics, and the Journal of Vibration, Sound and Control, he is a professional member of the American Institute of Aeronautics and

**Pezhman Mardanpour** is currently an associate professor at the Florida International University (FIU) and received his Ph.D. and M.S. from the School of Aerospace Engineering at the Georgia Institute of Technology. Under the supervision of professor Dewey H. Hodges, he became an expert in Hodges's non-linear composite beam theory as well as the aeroelastic design of UAVs. After graduating, he joined UCLA as a post-doctoral fellow



atic design framework to reliably benchmark, evaluate, and propose computationally cheap robust optimization algorithms. He has published over 300 publications with over 40,000 citations and an H-index of 70. As the most cited researcher in robust optimization, he has been on the list of 1% highly-cited researchers and named as one of the most influential researchers in the world by Web of Science for three consecutive years, since 2019. In 2020, he was ranked 21st across all disciplines and 4th in artificial intelligence and image processing in the Stanford University's list of World's Top Scientists. In 2021, The Australian newspaper named him as the top researcher in Australia in three fields of artificial intelligence, evolutionary

**Seyedali Mirjalili** (Senior Member, IEEE) is currently a Professor and the Founding Director of the Centre for Artificial Intelligence Research and Optimization, Torrens University Australia. He is internationally recognized for his advances in optimization and swarm intelligence, including the first set of algorithms from a synthetic intelligence standpoint—a radical departure from how natural systems are typically understood—and a systematic



computation, and fuzzy systems. His research interests include optimization, swarm intelligence, evolutionary algorithms, and machine learning. He is an Associate Editor of several AI journals including

Neurocomputing, Applied Soft Computing, Advances in Engineering Software, Computers in Biology and Medicine, Healthcare Analytics, Applied Intelligence, and IEEE ACCESS.

## Authors and Affiliations

Nima Khodadadi<sup>1</sup> · Ehsan Khodadadi<sup>2</sup> · Benyamin Abdollahzadeh<sup>3</sup> · El-Sayed M. El-Kenawy<sup>4</sup> ·  
Pezhman Mardanpour<sup>5</sup> · Weiguo Zhao<sup>6</sup> · Farhad Soleimanian Gharehchopogh<sup>3</sup> · Seyedali Mirjalili<sup>7,8,9</sup>

✉ Nima Khodadadi  
nima.khodadadi@miami.edu

Ehsan Khodadadi  
ehsank@uark.edu

Benyamin Abdollahzadeh  
benyamin.abdolahzade@gmail.com

El-Sayed M. El-Kenawy  
skenawy@ieee.org

Pezhman Mardanpour  
pmardanp@fiu.edu

Weiguo Zhao  
zgw770123@163.com

Farhad Soleimanian Gharehchopogh  
bonab.farhad@gmail.com

Seyedali Mirjalili  
ali.mirjalili@gmail.com

<sup>1</sup> Department of Civil and Architectural Engineering,  
University of Miami, Coral Gables, FL, USA

<sup>2</sup> Department of Chemistry and Biochemistry, University of  
Arkansas, Fayetteville, AR 72701, USA

<sup>3</sup> Department of Computer Engineering, Urmia Branch,  
Islamic Azad University, Urmia, Iran

<sup>4</sup> Department of Communications and Electronics, Delta  
Higher Institute of Engineering and Technology (DHIET),  
Mansoura 35111, Egypt

<sup>5</sup> Department of Mechanical & Material Engineering, Florida  
International University, Miami, FL, USA

<sup>6</sup> School of Water Conservancy and Hydropower, Hebei  
University of Engineering, Handan 056038, Hebei, China

<sup>7</sup> YFL (Yonsei Frontier Lab), Yonsei University, Seoul, Korea

<sup>8</sup> Research and Innovation Center, Obuda University,  
Budapest 1034, Hungary

<sup>9</sup> Centre for Artificial Intelligence Research and Optimisation,  
Torrens University Australia, Brisbane, Australia