



Many-Objective Grasshopper Optimization Algorithm (MaOGOA): A New Many-Objective Optimization Technique for Solving Engineering Design Problems

Kanak Kalita^{1,2} · Pradeep Jangir^{3,4,5} · Robert Čep⁶ · Sundaram B. Pandya⁷ · Laith Abualigah^{8,9,10,11}

Received: 27 May 2024 / Accepted: 31 July 2024
© The Author(s) 2024

Abstract

In metaheuristic multi-objective optimization, the term effectiveness is used to describe the performance of a metaheuristic algorithm in achieving two main goals—converging its solutions towards the Pareto front and ensuring these solutions are well-spread across the front. Achieving these objectives is particularly challenging in optimization problems with more than three objectives, known as many-objective optimization problems. Multi-objective algorithms often fall short in exerting adequate selection pressure towards the Pareto front in these scenarios and difficult to keep solutions evenly distributed, especially in cases with irregular Pareto fronts. In this study, the focus is on overcoming these challenges by developing an innovative and efficient a novel Many-Objective Grasshopper Optimisation Algorithm (MaOGOA). MaOGOA incorporates reference point, niche preserve and information feedback mechanism (IFM) for superior convergence and diversity. A comprehensive array of quality metrics is utilized to characterize the preferred attributes of Pareto Front approximations, focusing on convergence, uniformity and expansiveness diversity in terms of IGD, HV and RT metrics. It acknowledged that MaOGOA algorithm is efficient for many-objective optimization challenges. These findings confirm the approach effectiveness and competitive performance. The MaOGOA efficiency is thoroughly examined on WFG1-WFG9 benchmark problem with 5, 7 and 9 objectives and five real-world (RWMaOP1- RWMaOP5) problem, contrasting it with MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA algorithms. The findings demonstrate MaOGOA superior performance against these algorithms.

Keywords Many-objective optimization · Grasshopper optimization algorithm · Reference point strategies · Information feedback mechanism · Diversity maintenance

1 Introduction

Numerous practical scenarios are categorized as many-objective optimization problems (MaOPs), encompassing areas like engineering design, drone routing and large-scale issues [1–3]. This study investigates a specific MaOP,

$$\min f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \subseteq \mathbb{R}^m \text{ s.t. } \mathbf{x} \in \Omega \subseteq \mathbb{R}^n \quad (1)$$

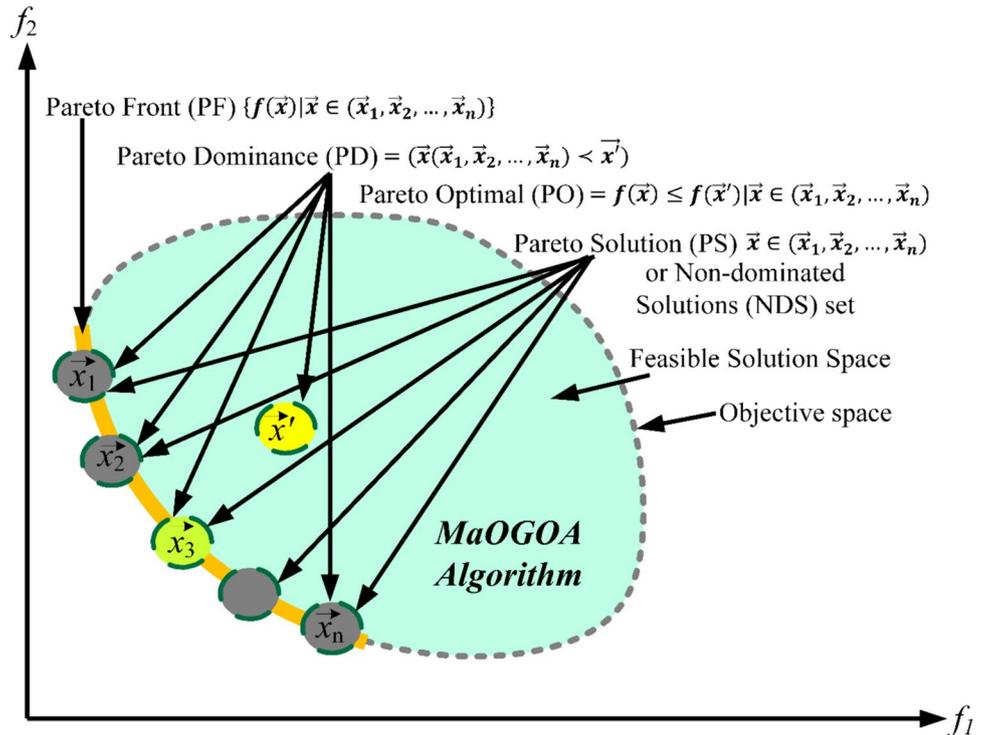
where $\Omega \subseteq \mathbb{R}^n$ identifying the decision space as the realm of $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ potential solutions. The objective space $f : \Omega \rightarrow \mathbb{R}^m$ is defined by multiple $m \geq 4$ objectives, each requiring minimization. The Pareto Optimal Set (PS) is characterized as the decision-making domain formed by

the aggregation of Pareto optimal solutions. In the realm of objective space, this set representation is known as the Pareto Optimal Front (PF). The primary objective in addressing a Multi-Objective Problem (MaOP) lies in pinpointing the PS shown in Fig. 1.

Approaches such as many-objective evolutionary algorithms (MaOEAs) [4] have been widely acknowledged in literature [5]. These population-driven methodologies strive to derive a suite of optimal solutions, which effectively converge and exhibit diversity [6] on the true Pareto front (PF) associated with MaOPs. Recent decades have seen the emergence of numerous MaOEAs, tailored for MaOPs [7] and demonstrating their effectiveness [8]. Despite their successes, the performance of MaOEAs, particularly those based on Pareto principles [9], in high-dimensional scenarios warrants further examination. A key challenge [10] in such contexts is the comparability of solutions; in

Extended author information available on the last page of the article

Fig. 1 Many-objective all definitions in search space of MaO-Problem



high-dimensional spaces, the proportion of nondominated solutions escalates. This issue is exacerbated as the number of objectives increases, leading to diminished selection pressure [11]. A secondary challenge is the preservation of population diversity [12]. Traditional methods for diversity maintenance [13], like the k th nearest distance [14], crowding distance [15] etc. are not suitable for vast objective spaces. In order to overcome these challenges, a number of algorithms have been proposed and these can be classified into three main categories:

1) **Pareto dominance-based strategies:** Pareto dominance-based strategies redefine classical Pareto dominance as a way to better compare solutions. Methods such as fuzzy dominance [16], ϵ -dominance [17] and θ -dominance [18] utilize these advanced techniques for picking the solutions. For instance, Tian et al. [19] proposed a new dominance relation that combines convergence and diversity based on adaptive approaches. Also, Qiu et al. [20] introduced fractional dominance which makes the convergence faster by comparing the objective values of two solutions. Furthermore, a new method integrating classical dominance with convergence-based approach has been proposed and Zhang et al. [21] proposed the KnEA to improve convergence speed.

2) **Indicator-based MaOEAs:** Indicator-based MaOEAs employ certain metrics to assess the nondomination of solutions and steer the search procedure. Some of the commonly used metrics in this category are R2 indicator [22], S-metric [23], I_ϵ indicator [24] and the hypervolume (HV) indicator [25]. HV-based MaOEAs [26–28] estimate the solutions by their contributions to the HV value and this flexibility allows the use of indicators within the IBEA framework [24].

3) **Decomposition-based methods:** Decomposition-based methods simplify complex multi-objective optimization problems (MaOPs) into subproblems [29] or a series of single-objective problems (SOPs) [30]. Algorithms such as MOEA/D [30], RVEA [31], Many-Objective Particle Swarm Optimizer (MaOPSO) [32], Many-Objective Sine Cosine Algorithm (MaOSCA) [33] and Non-Dominated Sorting Genetic Algorithm-III (NSGA-III) [13] are prominent in this area. MOEA/D optimizes each subproblem with data from adjacent subproblems to minimize computational complexity. NSGA-III uses a set of reference points to partition the objective space, enhancing diversity. From these principles, various environmental selection strategies have emerged, such as integrating dominance with decomposition methods (MOEA/DD) by Li et al. [34] and utilizing perpendicular distances

from solutions to reference vectors by Yuan et al. [35]. Innovations such as adaptive weight adjustments during the evolutionary process [36], learning the distribution of reference vectors via a growing neural gas (GNG) network [37] and generating reference vectors through a modified k-means clustering technique [38].

Additionally, many MaOEAAs do not align strictly with the above categories, such as the determinantal point process-based algorithm [39], adaptive clustering-based algorithm [40] and the two-archive algorithm [41, 42], which maintains separate archives for convergence and diversity. Other notable methods are the dimension reduction and knowledge-guided solving algorithm [43], the voting-based algorithm [44], the two-stage algorithm [45] and the decision variable classification-based algorithm [46]. Ding et al. [44] introduced a multi-stage evolutionary strategy that uses knowledge fusion and a statistical guidance vector to identify concentrations of elite solutions through a voting method. Li et al. [45] structured objective optimization into two phases: a convergence stage where individuals compete and a diversity stage that focuses on selecting well-spaced solutions. Liu et al. [46] developed a novel decision variable classification method, increasing the likelihood of generating offspring with superior convergence and diversity.

Although there are numerous studies on MaOEAAs, more work is required to address the issue of balance in terms of convergence [47] and diversity [48], particularly in high-dimensional problems [49]. There are other methods that have been proposed to address the issue of diversity in many-objective optimization such as the Grid-based Evolutionary Algorithm (GrEA) [50]. GrEA, therefore, effectively controls population diversity without using explicit density estimation through dividing the search space into a grid structure. Bi-goal Evolution (BiGE) [51] is another approach that focuses on two goals only, thus making the optimization process easier but without losing the ability to explore and exploit the search space effectively. Shift-based Density Estimation (SDE) [52] is another interesting approach to density estimation that shifts according to the distribution of solutions to guarantee the maintenance of solution diversity in high-dimensional objective space. These algorithms must not only retain variation in the population but also must have high selection pressure towards actual PFs.

This research paper is motivated by the need for effective approaches that can handle the intricacies of many-objective problems particularly, the challenges related to convergence to the Pareto front and maintaining solution diversity across it. Current strategies either fail to adequately converge or cannot ensure a well-spread set of solutions, especially when dealing with irregular Pareto fronts. This leads to suboptimal decision-making in practical applications. To address these

gaps, the MaOGO is proposed, an innovative approach that integrates Grasshopper Optimisation Algorithm (GOA) [53], reference point strategies, niche preservation and an IFM. It is specifically designed to improve both the convergence and diversity of solutions in many-objective problems.

This paper primary contributions are

- The selection of GOA algorithm due to its distinct characteristics that are particularly suited to addressing the challenges of MaOPs. The key reasons for this choice are
- GOA simulates the swarming behaviors of grasshoppers, which is a natural phenomenon noted for its efficiency in exploring and exploiting spatial resources.
- The algorithm's ability to dynamically balance between the different phases of search driven by its inherent components of social interaction, gravitational forces and wind advection.
- It is shown that GOA is highly scalable and flexible, which are essential when addressing the fact that the objective space can be quite diverse in many-objective problems. Some of these are the degree of attraction or repulsion amongst the individuals which can be adjusted depending on the problem at hand to ensure that there is a reasonable spread of the population and yet getting close to the Pareto front.
- Many-objective optimization problems have Pareto fronts that are complex and can be non-contiguous. GOA model, which is based on the differential movement of agents (grasshoppers), is best suited to tackle such scenarios. It can search for a larger set of regions in the search space and thus, identify more potential Pareto-optimal solutions.
- This paper offers an IFM approach to overcome previous shortcomings that are related to the non-consideration of valuable information. In the IFM, the decision-making process is made using the weighted sum method to promote the historical information of the whole population to the next generation in order to increase the convergence.
- The selection process is based on the reference point approach, which makes it possible to reach not just the true Pareto frontier and its closest solutions (convergence), but also to distribute solutions across the entire Pareto diagram (diversity). This is done by assigning each solution to the nearest reference point with the help of perpendicular distance measurement which helps in identifying the well explored region of the objective space. An NSGA-II algorithm uses non-dominated sorting which sorts solutions closer to the Pareto-optimal front and thus improves convergence.
- A niche preservation strategy specifically targets boundary individuals to enhance diversity and mitigate over-

crowding in certain areas of the objective space, thus improving the overall convergence rate of the algorithm. Additionally, a density estimation strategy is detailed to maintain diversity, ensuring both uniformity and broad coverage in the population distribution.

- The efficacy of the newly developed MaOGOA is demonstrated through comparative analysis against algorithms such as MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA across the WFG1-WFG9 benchmark sets with objectives ranging from 5 to 9, as well as five real-world problems (RWMaOP1-RWMaOP5). These experiments highlight MaOGOA ability to adeptly handle a variety of problem types and its robust performance.

The structure of this paper is organized as follows: Sect. 2 offers an overview of GOA algorithm. Section 3 provides a detailed description of the proposed MaOGOA. Experimental results and discussions are presented in Sect. 4. Finally, Sect. 5 concludes the paper and outlines future research directions.

2 Grasshopper Optimisation Algorithm

The GOA, initially introduced by Saremi et al. [53], is inspired by the swarming patterns of grasshoppers in nature and is utilized for addressing optimization challenges. Center of Gravity is based on the cohesive social behavior of grasshoppers. This swarming characteristic makes

them fit naturally for explorative and exploitative tasks in a many-dimensional search domain. The GOA mathematical model successfully provides an overall representation of this balance. The social interaction, gravitational pull and wind advection in the model enable the specific elements of the algorithm to change their search behavior in an appropriate manner. This algorithm represents potential solutions to optimization problems through the positions of grasshoppers within a swarm. Grasshoppers exhibit a distinctive flight pattern, which is mathematically modeled in GOA shown in Fig. 2. This model suggests that grasshopper movement is predominantly influenced by three components—social interaction, gravitational pull and wind advection.

The position of each grasshopper, labeled as X_i , is expressed as

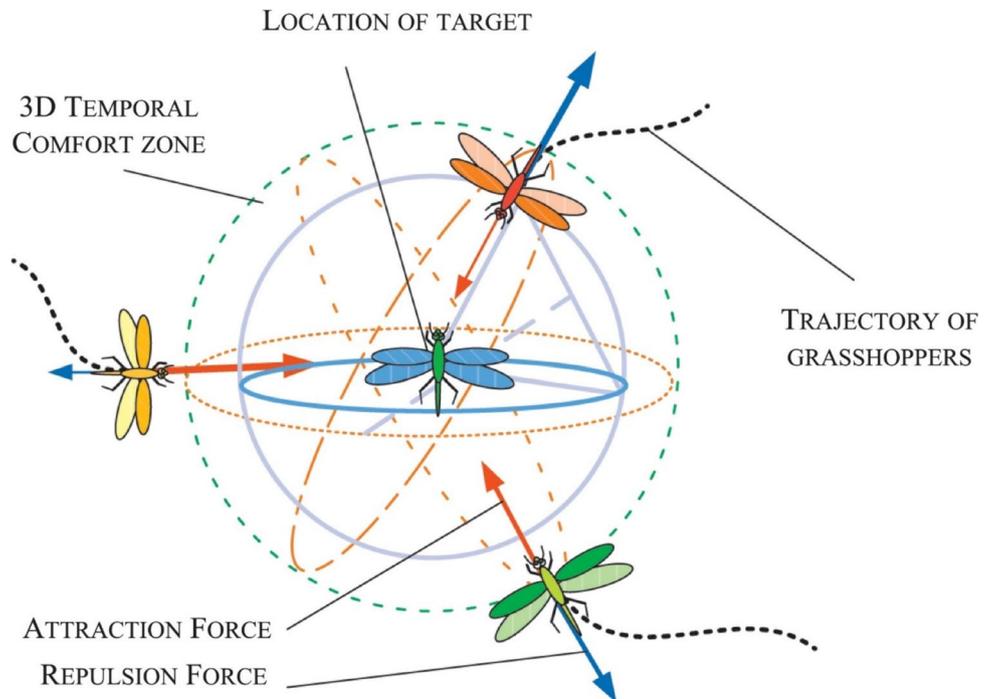
$$X_i = S_i + G_i + A_i, \tag{2}$$

where S_i signifies social interaction, G_i denotes gravitational force and A_i represents wind advection. The pivotal element in this model is social interaction which is calculated as

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(d_{ij}) \hat{d}_{ij}, \tag{3}$$

$$d_{ij} = |x_j - x_i|, \tag{4}$$

Fig. 2 Basic interaction patterns among individual grasshoppers within a group



$$\hat{d}_{ij} = (x_j - x_i) / d_{ij} \tag{5}$$

$$s(r) = fe^{-r} - e^{-r} \tag{6}$$

In these equations, d_{ij} represents the distance between the i th and j th grasshopper and \hat{d}_{ij} a unit vector directs from the i th and j th grasshopper. The social force, defined by the s function, is modifiable through parameters f and l . It important to note that the s function social forces diminish with increasing distances between grasshoppers. For effective interaction, grasshopper distances are mapped within the range of [1, 4]. Gravitational influence on a grasshopper is described as,

$$G_i = -g\hat{e}_g \tag{7}$$

In this context, g stands for the gravitational constant and \hat{e}_g a unit vector points towards Earth center. The wind advection effect on grasshoppers is given as

$$A_i = u\hat{e}_w \tag{8}$$

where u is a constant drift and \hat{e}_w a unit vector indicates wind direction. Consequently, Eq. (2) is elaborated as,

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(|x_j - x_i|) \frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \tag{9}$$

N denotes the total grasshoppers in the swarm. to enhance the optimization process, additional parameters are incorporated into the model, optimizing exploration and exploitation phases. The gravitational impact on grasshoppers is relatively minor and generally disregarded, while the wind direction (A component) is presumed to always orient towards the optimal solution \hat{T}_d . The final form of the mathematical model is as,

$$X_i^d = c \left(\sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j^d - x_i^d}{d_{ij}} \right) + \hat{T}_d \tag{10}$$

In this equation, ub_d and lb_d are the upper and lower bounds in the d th dimension, respectively. \hat{T}_d represents the d th dimension value in the optimal solution discovered so far. The parameter c evolves according to the equation below:

$$c = cmax - l \frac{cmax - cmin}{L} \tag{11}$$

where $cmax$ and $cmin$ are the maximum and minimum values, respectively, l is the current iteration and L signifies the maximum iteration limit.

3 Proposed Many-Objective Grasshopper Optimization Algorithm

3.1 Overview of MaOGOA

MaOGOA is designed to address the complexities of many-objective optimization problems by effectively balancing exploration and exploitation, ensuring diversity and converging towards the Pareto front. This algorithm leverages the natural swarming behaviors of grasshoppers, modeled through social interaction, gravitational pull and wind advection mechanisms, to navigate the search space dynamically.

3.2 Initialization

MaOGOA commences with an initialization phase where a random population of size N is established. Each individual in this population represents a potential solution within a d -dimensional decision space, where d stands for the number of decision variables. The initial population is evaluated based on M objective functions, which define the initial objective space.

3.3 Generation of Reference Points

Following initialization, a set of reference points is generated using Das and Dennis technique. This technique strategically partitions the objective space to enhance the diversity of solutions and guide the search towards covering the entire Pareto front effectively. The number of reference points is approximately equal to the population size N , calculated as $H = \binom{M+p-1}{p}$, where p is the number of partitions.

3.4 Grasshopper Optimization Mechanism

Each iteration of MaOGOA involves updating the position of each grasshopper (solution) in the search space. The position update is influenced by three main components:

- Social Interaction S_i : Each grasshopper adjusts its position relative to others based on their proximity, promoting local exploration.

- Gravitational Pull G_i : A collective movement towards better solutions, facilitating exploitation.
- Wind Advection A_i : Random perturbations that prevent premature convergence and encourage global exploration.

The position update formula for each grasshopper X_i is given in Eq. (1).

3.5 Information Feedback Mechanism (IFM)

A novel aspect of MaOGOA is the incorporation of an IFM. This mechanism integrates historical performance data to adjust the search dynamically. For each grasshopper, a weighted combination of its own past position and the best positions encountered so far is used to update its position, enhancing the algorithm ability to converge on optimal solutions while maintaining diversity.

3.6 Environmental Selection and Niche Preservation

After updating the positions of grasshoppers, the total population, including both parents and offspring, undergoes evaluation. The individuals are organized into non-dominated fronts and selection is focused on preserving a diverse and high-quality solution set. This process are

- Niche Preservation: Ensuring that solutions spread across all regions of the objective space, particularly focusing on less crowded niches.
- Density Estimation: Evaluating the density around each solution to avoid overcrowding and maintain diversity.

The selection process emphasizes maintaining a balance between convergence towards the Pareto front and diversity across it.

3.7 Algorithm Termination

The MaOGOA algorithm progresses through these stages until a specified termination criterion is met, which may be a predefined number of generations, a convergence threshold, or another problem-specific benchmark. At the end of the process, the algorithm delivers a set of diverse, high-quality solutions that approximate the Pareto front effectively. The main process of MaOGOA can be expressed as follows:

MaOGOA algorithm starts with a random population of size N , M no. of objectives, p no. of partitions and generate a set of reference points using Das and Dennis technique

$H = \binom{M+p-1}{p}$, as $H \approx N$. the current generation is t , x_i^t and x_i^{t+1} the i th individual at t and $(t + 1)$ generation. u_i^{t+1} the i th individual at the $(t + 1)$ generation generated through the GOA algorithm and parent population P_t . the fitness value of u_i^{t+1} is f_i^{t+1} and U^{t+1} is the set of u_i^{t+1} . Next, x_i^{t+1} is computed according to u_i^{t+1} generated through the GOA algorithm and IFM as per Eq. (12)

$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t; \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t}, \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t}, \partial_1 + \partial_2 = 1, \tag{12}$$

where x_k^t is the k th individual we chose from the t th generation, the fitness value of x_k^t is f_k^t , ∂_1 and ∂_2 are weight coefficients. Generate offspring population Q_t . Q_t is the set of x_i^{t+1} . The combined population $R_t = P_t \cup Q_t$ is sorted into different w -non-dominant levels $(F_1, F_2, \dots, F_l, \dots, F_w)$. Begin from F_1 , all individuals in level 1 to l are added to S_t and remaining members of R_t are rejected. If $|S_t| = N$; no other actions are required and the next generation is begun with $P_{t+1} = S_t$. Otherwise, solutions in S_t/F_l are included in $P_{t+1} = S_t/F_l$ and the rest $(K = N - |P_{t+1}|)$ individuals are selected from the last front F_l (presented in Algorithm 1). For selecting individuals from F_l , we use a niche-preserving operator First, each population member of P_{t+1} and F_l is normalized (presented in Algorithm 2) using the current population spread so that all objective vectors and reference points have commensurate values. Thereafter, each member of P_{t+1} and F_l is associated (presented in Algorithm 3) with a specific reference point using the shortest perpendicular distance ($d()$) of each population member with a reference line created by joining the origin with a supplied reference point. Then, a careful niching strategy (described in Algorithm 4) that improve the diversity of MaOGOA algorithm is employed to choose those F_l members that are associated with the least represented reference points niche count ρ_i in P_{t+1} and check termination condition is met. If the termination condition is not satisfied, $t = t + 1$ than repeat and if it is satisfied, P_{t+1} is generated, it is then applied to generate a new population Q_{t+1} by GOA algorithm. Such a careful selection strategy is found to computational complexity of M -Objectives $O(N^2 \log^{M-2} N)$ or $O(N^2 M)$, whichever is larger. The MaOGOA algorithm, integrating the IFM, effectively directs the search towards improved convergence, coverage and diversity, essential for many-objective optimization. Importantly, the algorithm operates without requiring additional parameters beyond standard GOA settings such as population size and termination criteria.

Algorithm 1 Generation t of MaOGOA Algorithm with IFM Procedure

Input: N (Population Size), M (No. of Objectives), GOA algorithm parameters, and Initial population $P_t(t=0)$,

Output: $Q_{t+1} = \text{GOA}(P_{t+1})$

- 1: H Calculated using Das and Dennis's technique, structured reference points Z^s , supplied aspiration points Z^a , $S_t = \phi$, $i = 1$
- 2: **Proposed Information Feedback Mechanism (IFM)**
GOA algorithm apply on the initial population P_t to generate u_i^{t+1} , calculate x_i^{t+1} according to u_i^{t+1} can be expressed as follows:

$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t; \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t}, \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t}, \partial_1 + \partial_2 = 1$$

$$Q_t = Q_t; (Q_t \text{ is the set of } x_i^{t+1})$$
- 3: $R_t = P_t \cup Q_t$
- 4: Different non-domination levels $(F_1, F_2, \dots, F_l) = \text{Non-dominated-sort}(R_t)$
- 5: **repeat**
- 6: $S_t = S_t \cup F_i$ and $i = i+1$
- 7: **until** $|S_t| \geq N$
- 8: Last front to be included: $F_l = \bigcup_{i=1}^l F_i$
- 9: **if** $|S_t| = N$ **then**
- 10: $P_{t+1} = S_t$
- 11: **else**
- 12: $P_{t+1} = S_t / F_l$
- 13: Point to chosen from last Front (F_l): $K = N - |P_{t+1}|$
- 14: Normalize objectives and create reference set Z^r :
Normalize (μ, S_t, Z^r, Z^s, Z^a); Brief Explanation in **Algorithm-2**
- 15: Associate each member s of S_t with a reference point:
 $[\pi(s), d(s)] = \text{Associate}(S_t, Z^r)$; Brief Explanation in **Algorithm-3**
 $\% \pi(s)$: closest reference point, d : distance between s and $\pi(s)$
- 16: Compute niche count of reference point $j \in Z^r$:
 $\rho_j = \sum_{s \in S_t / F_l} (\pi(s) = j), \mathbf{1} : \mathbf{0}$;
- 17: Choose K members one at a time F_l to construct
 $P_{t+1} : \text{Niching}(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$; **Represent in Algorithm-4**
- 18: **end if**

Algorithm 2 Normalize (f^n , S_t , Z^r , Z^s/Z^a) procedure

Input: S_t, Z^s (structured points) or Z^a (supplied points)
Output: f^n, Z^r (reference points on normalized hyper-plane)

- 1: **for** $j=1$ **to** M **do**
- 2: Compute ideal point: $Z_j^{min} = \min_{s \in S_t} f_j(s)$
- 3: Translate objectives: $f_j^i(s) = f_j(s) - Z_j^{min} \forall s \in S_t$
- 4: Compute extreme points: $Z^{j,max} = s$:
 $\text{argmin}_{s \in S_t} ASF(s, w^j) = \text{where } w^j = (\epsilon_1, \dots, \epsilon_j)^T$,
 $\epsilon = 10^{-6}$, and $w_j^j = 1$
- 5: **end for**
- 6: Compute intercepts a_j for $j= 1, \dots, M$
- 7: Normalize objectives $f_i^n(X)$ using

$$f_i^n(X) = \frac{f_i(X)}{a_i - Z_i^{min}}, \text{ for } i = 1, 2, \dots, M$$
- 8: **if** Z^a is given **then**
- 9: Map each (aspiration) point on normalized hyper-plane
 $f_i^n(X)$ and save the points in the set Z^r
- 10: **else**
- 11: $Z^r = Z^s$
- 12: **end if**

Algorithm 3 Associate (S_t, Z^r) procedure

Input: S_t, Z^r
Output: $\pi(s \in S_t), d(s \in S_t)$

- 1: **for** each reference point $Z \in Z^r$ **do**
- 2: Compute reference line $w=Z$
- 3: **end for**
- 4: **for** each $(s \in S_t)$ **do**
- 5: **for** each $w \in Z^r$ **do**
- 6: Compute $d^\perp(s, w) = s - w^T s / \|w\|$
- 7: **end for**
- 8: Assign $\pi(s) = w: \text{argmin}_{w \in Z^r} d^\perp(s, w)$
- 9: Assign $d(s) = d^\perp(s, \pi(s))$
- 10: **end for**

Algorithm 4 Niching ($K, \rho_j, \pi, d, Z^r, F_t, P_{t+1}$) procedure

Input: $K, \rho_j, \pi(s \in S_t), d(s \in S_t), Z^r, F_t$,
Output: P_{t+1}

- 1: $k=1$
- 2: **while** $k \leq K$ **do**
- 3: $J_{min} = \{j : \text{argmin}_{j \in Z^r} \rho_j\}$
- 4: $\bar{j} = \text{random}(J_{min})$
- 5: $I_{\bar{j}} = \{s : \pi(s) = \bar{j}, s \in F_t\}$
- 6: **if** $I_{\bar{j}} \neq \phi$ **then**
- 7: **if** $\rho_{\bar{j}} = 0$ **then**
- 8: $P_{t+1} = P_{t+1} \cup (s : \text{argmin}_{s \in I_{\bar{j}}} d_s)$
- 9: **else**
- 10: $P_{t+1} = P_{t+1} \cup \text{random}(I_{\bar{j}})$
- 11: **end if**
- 12: $\rho_{\bar{j}} = \rho_{\bar{j}} + 1, F_t = F_t/s$
- 13: $k = k+1$
- 14: **else**
- 15: $Z^r = Z^r / \{\bar{j}\}$
- 16: **end if**
- 17: **end while**

Fig. 3 Flowchart of MaOGOA algorithm

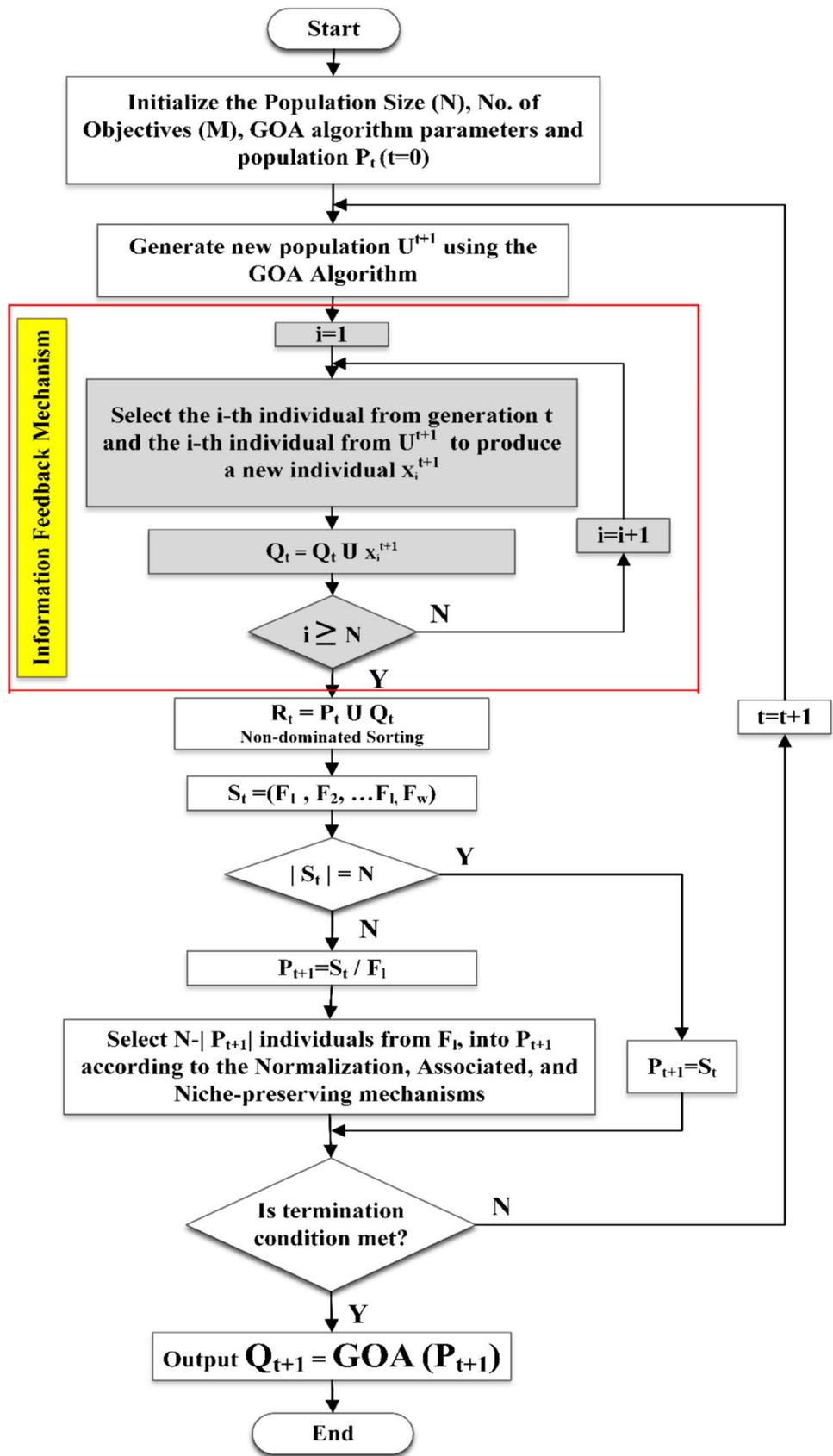


Table 1 Properties of the quality indicators

Quality indicator	Convergence [60, 61]	Diversity	Uniformity	Cardinality	Computational Burden
RT					✓
IGD	✓	✓	✓		
HV	✓	✓	✓	✓	

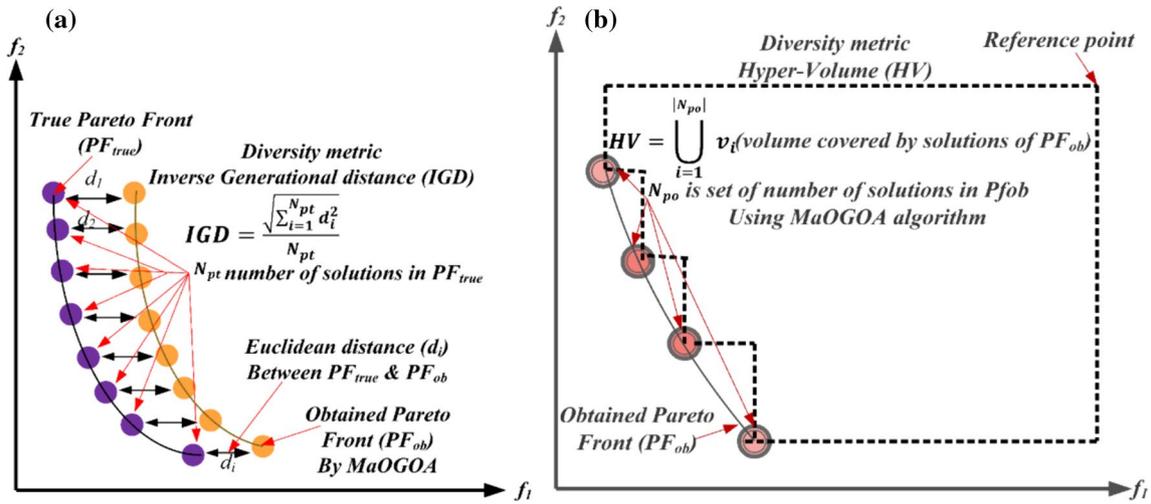


Fig. 4 Mathematical and schematic view of **a** IGD and **b** HV metrics

The flowchart of MaOGOA algorithm can be shown in Fig. 3.

4 Results and Discussion

4.1 Experimental Settings

4.1.1 Benchmarks

To evaluate the MaOGOA effectiveness, this research utilizes the WFG1-WFG9 [54] benchmark (Appendix A) along with five real-world engineering design challenges (Appendix B). Car cab design (RWMaOP1) [55], 10-bar truss structure (RWMaOP2) [56], water and oil repellent fabric development (RWMaOP3) [57], ultra-wideband antenna design (RWMaOP4) [58] and liquid-rocket single element injector design (RWMaOP5) [59] are the five considered real-world engineering design problems. The WFG problems define the number of decision variables as $k + l$, with k equating to $M - 1$ and l being 10.

4.1.2 Comparison Algorithms and Parameter Settings

The performance of MaOGOA is assessed by conducting empirical comparisons against leading multi-objective algorithms (MOAs) like MaOPSO [32], MaOSCA [33], MOEA/DD [34], NSGA-III [13], KnEA [21], RvEA [31] and GrEA [50]. The testing environment consists of Matlab R2020a on an Intel Core i7-9700 CPU. Each algorithm is executed 30 times with varying population sizes: N is set to 210, 156 and 276, corresponding to objective problems with $M = 5, 7$ and 9, respectively. The maximum number of function evaluations ($MaxFEs$) for each test is capped at 10,000.

4.1.3 Performance Measures

To ensure a robust assessment of the proposed algorithm, this paper adopts quality evaluation metrics Hypervolume (HV) and Run Time (RT), each chosen for their proven reliability and relevance in current many-objective optimization where the true Pareto fronts are unknown [60]. Additionally, Inverse Generational distance (IGD) metric is also used where the true Pareto fronts are known. HV and IGD metric help in providing a more accurate representation of an

Table 2 Results of IGD metric for WFG problems

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA
WFG1	5	14	1.089 ± 0.198	1.23 ± 0.225	1.151 ± 0.122	1.348 ± 0.165	1.259 ± 0.121	1.346 ± 0.121	1.021 ± 0.09	0.963 ± 0.033
	7	16	1.483 ± 0.126	1.748 ± 0.144	2.153 ± 0.1	1.791 ± 0.195	1.525 ± 0.148	1.809 ± 0.219	1.369 ± 0.04	1.328 ± 0.101
	9	18	1.972 ± 0.167	1.953 ± 0.305	2.677 ± 0.135	2.176 ± 0.154	2.104 ± 0.131	1.946 ± 0.24	1.619 ± 0.15	1.846 ± 0.295
WFG2	5	14	0.475 ± 0.006	0.537 ± 0.012	0.862 ± 0.027	0.521 ± 0.014	0.503 ± 0.005	0.509 ± 0.013	0.566 ± 0.001	0.528 ± 0.023
	7	16	0.898 ± 0.017	0.85 ± 0.077	1.934 ± 0.347	0.923 ± 0.072	0.883 ± 0.006	0.837 ± 0.021	0.919 ± 0.01	0.904 ± 0.029
	9	18	1.02 ± 0.045	1.01 ± 0.009	2.299 ± 0.235	1.167 ± 0.093	0.988 ± 0.007	1.274 ± 0.258	1.152 ± 0.036	1.147 ± 0.029
WFG3	5	14	0.68 ± 0.023	0.684 ± 0.1	0.73 ± 0.124	0.742 ± 0.078	0.799 ± 0.057	0.909 ± 0.047	0.588 ± 0.128	0.555 ± 0.032
	7	16	1.678 ± 0.689	1.152 ± 0.078	1.635 ± 0.286	0.727 ± 0.296	1.601 ± 0.058	0.83 ± 0.038	1.487 ± 0.567	1.11 ± 0.217
	9	18	2.726 ± 0.751	2.494 ± 1.07	2.879 ± 0.331	0.942 ± 0.085	2.458 ± 0.254	1.256 ± 0.099	1.577 ± 0.068	1.813 ± 0.032
WFG4	5	14	1.23 ± 0.011	1.252 ± 0.006	1.213 ± 0.007	1.269 ± 0.01	1.22 ± 0.001	1.23 ± 0.004	1.296 ± 0.008	1.28 ± 0.013
	7	16	2.612 ± 0.021	2.562 ± 0.027	2.554 ± 0.022	2.749 ± 0.023	2.65 ± 0.018	2.63 ± 0.01	2.731 ± 0.026	2.531 ± 0.013
	9	18	4.316 ± 0.263	4.224 ± 0.138	4.206 ± 0.012	4.486 ± 0.122	4.274 ± 0.036	4.345 ± 0.145	4.641 ± 0.096	4.507 ± 0.047
WFG5	5	14	1.218 ± 0.008	1.265 ± 0.013	1.196 ± 0.009	1.269 ± 0.03	1.209 ± 0.001	1.208 ± 0.01	1.27 ± 0.018	1.262 ± 0.019
	7	16	2.618 ± 0.019	2.696 ± 0.055	2.547 ± 0.006	2.656 ± 0.004	2.635 ± 0.008	2.57 ± 0.016	2.693 ± 0.015	2.523 ± 0.016
	9	18	4.461 ± 0.151	4.474 ± 0.018	4.193 ± 0.027	4.287 ± 0.054	4.217 ± 0.025	4.262 ± 0.077	4.581 ± 0.066	4.422 ± 0.061
WFG6	5	14	1.257 ± 0.01	1.302 ± 0.017	1.232 ± 0.002	1.316 ± 0.028	1.235 ± 0.008	1.239 ± 0.007	1.371 ± 0.038	1.323 ± 0.026
	7	16	3.08 ± 0.403	2.769 ± 0.079	2.583 ± 0.019	2.912 ± 0.118	2.662 ± 0.017	2.626 ± 0.012	2.85 ± 0.128	2.583 ± 0.021
	9	18	4.796 ± 0.376	4.369 ± 0.157	4.24 ± 0.035	4.327 ± 0.079	4.276 ± 0.046	4.399 ± 0.051	4.874 ± 0.025	4.48 ± 0.158
WFG7	5	14	1.245 ± 0.005	1.265 ± 0.013	1.268 ± 0.017	1.291 ± 0.008	1.24 ± 0.008	1.256 ± 0.005	1.327 ± 0.036	1.292 ± 0.016
	7	16	2.723 ± 0.087	2.615 ± 0.038	2.628 ± 0.04	2.68 ± 0.023	2.648 ± 0.014	2.633 ± 0.005	2.757 ± 0.03	2.551 ± 0.02
	9	18	4.238 ± 0.029	4.04 ± 0.029	4.278 ± 0.035	4.501 ± 0.311	4.26 ± 0.029	4.268 ± 0.022	4.559 ± 0.084	4.459 ± 0.02
WFG8	5	14	1.298 ± 0.012	1.331 ± 0.024	1.314 ± 0.038	1.355 ± 0.008	1.271 ± 0.01	1.296 ± 0.003	1.36 ± 0.048	1.308 ± 0.008
	7	16	2.756 ± 0.069	2.791 ± 0.014	2.7 ± 0.013	2.812 ± 0.218	2.695 ± 0.01	2.657 ± 0.029	2.874 ± 0.045	2.679 ± 0.012
	9	18	4.641 ± 0.096	4.484 ± 0.191	4.424 ± 0.032	4.635 ± 0.19	4.396 ± 0.017	4.723 ± 0.192	4.85 ± 0.418	5.256 ± 0.099
WFG9	5	14	1.227 ± 0.043	1.263 ± 0.02	1.249 ± 0.023	1.272 ± 0.007	1.219 ± 0.013	1.192 ± 0.004	1.216 ± 0.013	1.22 ± 0.008
	7	16	2.688 ± 0.044	2.599 ± 0.035	2.752 ± 0.084	2.587 ± 0.038	2.57 ± 0.012	2.52 ± 0.027	2.485 ± 0.016	2.497 ± 0.007
	9	18	4.229 ± 0.046	4.28 ± 0.068	4.356 ± 0.082	4.325 ± 0.095	4.239 ± 0.053	4.23 ± 0.019	4.292 ± 0.064	4.365 ± 0.156

algorithm performance across both convergence and diversity, shown in Table 1 and Fig. 4. A higher value of *HV*, lower value of *IGD* and *RT* refers to better performance. As per the guidelines outlined in [61], using *IGD* without a known Pareto front can indeed result in biased evaluations. This insight is vital, especially for the real-world application scenarios, where the true Pareto fronts are not predetermined. The Wilcoxon rank sum test (WRST), conducted at a 0.05 significance level, is used to analyze performance variations—indicating superior (“+”), inferior (“−”), or equivalent (“=”) outcomes when compared to MaOGOA. Additional statistical evaluations, such as the Friedman rank test, are employed to comprehensively assess the differences in performance between MaOGOA and other algorithms.

4.2 Experimental Results on WFG Problems

Table 2 illustrates MaOGOA effectiveness relative to established algorithms like MaOSCA, MaOPSO, MOEA/DD,

NSGA-III, KnEA, RvEA and GrEA within the WFG benchmark suite. This effectiveness is measured using the Inverted Generational Distance (*IGD*) metric, which evaluates the solution sets’ convergence and diversity towards the true Pareto front. Lower *IGD* values are preferred as they signify solution sets that are not only closer to, but also more diverse within, the Pareto front. In the WFG1 problem with 5 objectives and 14 decision variables, MaOGOA outperforms all the other algorithms with an *IGD* value of 0.963 ± 0.033 , suggesting its superior ability to PF closely while maintaining diversity in the solutions. As the complexity increases to 7 objectives and 16 decision variables in WFG1, MaOGOA maintains a commendable *IGD* value of 1.328 ± 0.101 , standing out as the algorithm with the closest proximity to the PF like MaOPSO and MOEA/DD, which exhibit higher *IGD* values. A trend of MaOGOA efficiency is further seen in WFG2, where for 5 objectives and 14 decision variables,

Table 3 Friedman test and p value based on IGD metric for WFG problems

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA	p -value
WFG1	5	14	3.67	5.33	4	6	6	6.67	2.67	1.67	0.134
	7	16	3.33	5.67	8	6	3.67	6	1.67	1.67	0.012
	9	18	3.33	4	8	6	6	3.33	1.33	4	0.036
WFG2	5	14	1	5.67	8	3.67	2.67	3.33	7	4.67	0.009
	7	16	4.67	3.33	8	4.67	3.33	1.33	5.67	5	0.064
	9	18	2.33	2.33	8	5.67	1.33	6	5.33	5	0.012
WFG3	5	14	3.33	3.33	5.33	5.67	6.33	8	2	2	0.021
	7	16	6	4.33	6.67	1.67	7	1.67	5.33	3.33	0.028
	9	18	6.33	6	7.33	1	6.33	2	3	4	0.008
WFG4	5	14	3.33	5	1	6.33	2	3.67	7.67	7	0.005
	7	16	4.33	2.67	2.33	8	6	4.67	7	1	0.004
	9	18	3.67	2.33	2	5.67	4	4.67	7.67	6	0.078
WFG5	5	14	4	6.33	1	6.67	2.67	2.33	7	6	0.010
	7	16	4	7.33	2	6.33	5	3	7.33	1	0.005
	9	18	6.33	6.33	1.33	3.33	2.33	3	7.67	5.67	0.011
WFG6	5	14	4	5.33	1.67	6.67	2.33	2	7.33	6.67	0.009
	7	16	7	5.67	1.33	7	4	3	6.33	1.67	0.009
	9	18	6.67	3.67	2.33	4	1.67	4.67	7.33	5.67	0.053
WFG7	5	14	1.67	4.33	4.67	7	1.33	3.33	7.67	6	0.008
	7	16	6.33	3	3.67	6	5	3.33	7.67	1	0.023
	9	18	2	1	4.67	6.67	3.33	4	7.67	6.67	0.006
WFG8	5	14	3	6.33	4.33	7	1	3.33	6.67	4.33	0.035
	7	16	5.33	6.33	4	5	3.33	1.67	7.67	2.67	0.057
	9	18	4.67	3.33	2	5	1.67	6	5.67	7.67	0.039
WFG9	5	14	4	6.67	6.33	7	4	1	3.67	3.33	0.042
	7	16	7.67	5	7.33	5	4.67	3	1	2.33	0.009
	9	18	2.67	3.67	7	5.33	4.33	3.33	4	5.67	0.429
$\pm/=$			2/25/0	1/26/0	6/21/0	1/26/0	4/23/0	2/25/0	4/23/0	7/20/0	
Rank			4	5	2	5	3	4	3	1	

it achieves an IGD of 0.528 ± 0.023 , again surpassing other algorithms in finding a diverse and accurate representation of the Pareto front. Notably, in WFG3 for 9 objectives and 18 decision variables, MaOGOA demonstrates a significantly better performance with an IGD of 1.813 ± 0.032 , highlighting its scalability and robustness even in higher dimensional objective spaces. In WFG6 with 7 objectives, MaOGOA performance remains strong, recording an IGD of 2.583 ± 0.021 , which is considerably lower than MaOPSO and MOEA/DD. This reflects MaOGOA persistent ability to generate high-quality solutions across various problem configurations. WFG8 and WFG9 follow the same pattern, with MaOGOA consistently showing lower IGD values such as 2.679 ± 0.012 and 2.497 ± 0.007 , respectively, for problems with 7 objectives and 16 decision variables. This is

indicative of MaOGOA efficient exploration and exploitation capabilities, resulting in better coverage of the Pareto front.

The p values from the Wilcoxon signed-rank test are detailed in Table 3, indicating significant superiority of MaOGOA over other conventional algorithms. Specifically, MaOGOA outperformed MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA in 5, 6, 1, 6, 3, 5 and 3 instances, respectively, with p values below 0.05, showcasing notable effects. Moreover, the Friedman test revealed fewer instances where MaOGOA performed worse than the mentioned algorithms: 2, 1, 6, 1, 4, 2 and 4 cases, respectively.

To visually demonstrate MaOGOA's efficiency, the average Inverted Generational Distance (IGD) convergence

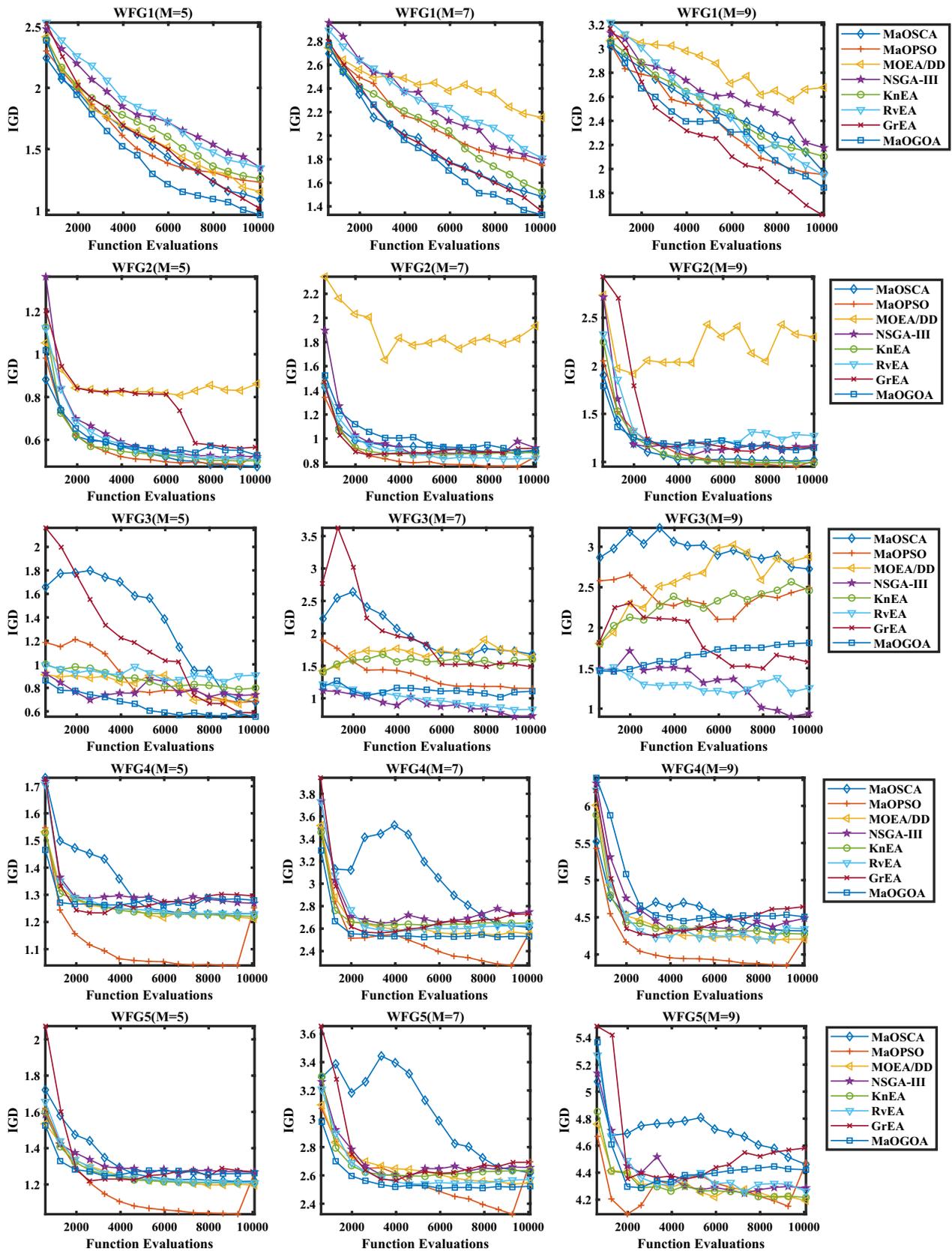


Fig. 5 Convergence curve based on the IGD metric for WFG problems

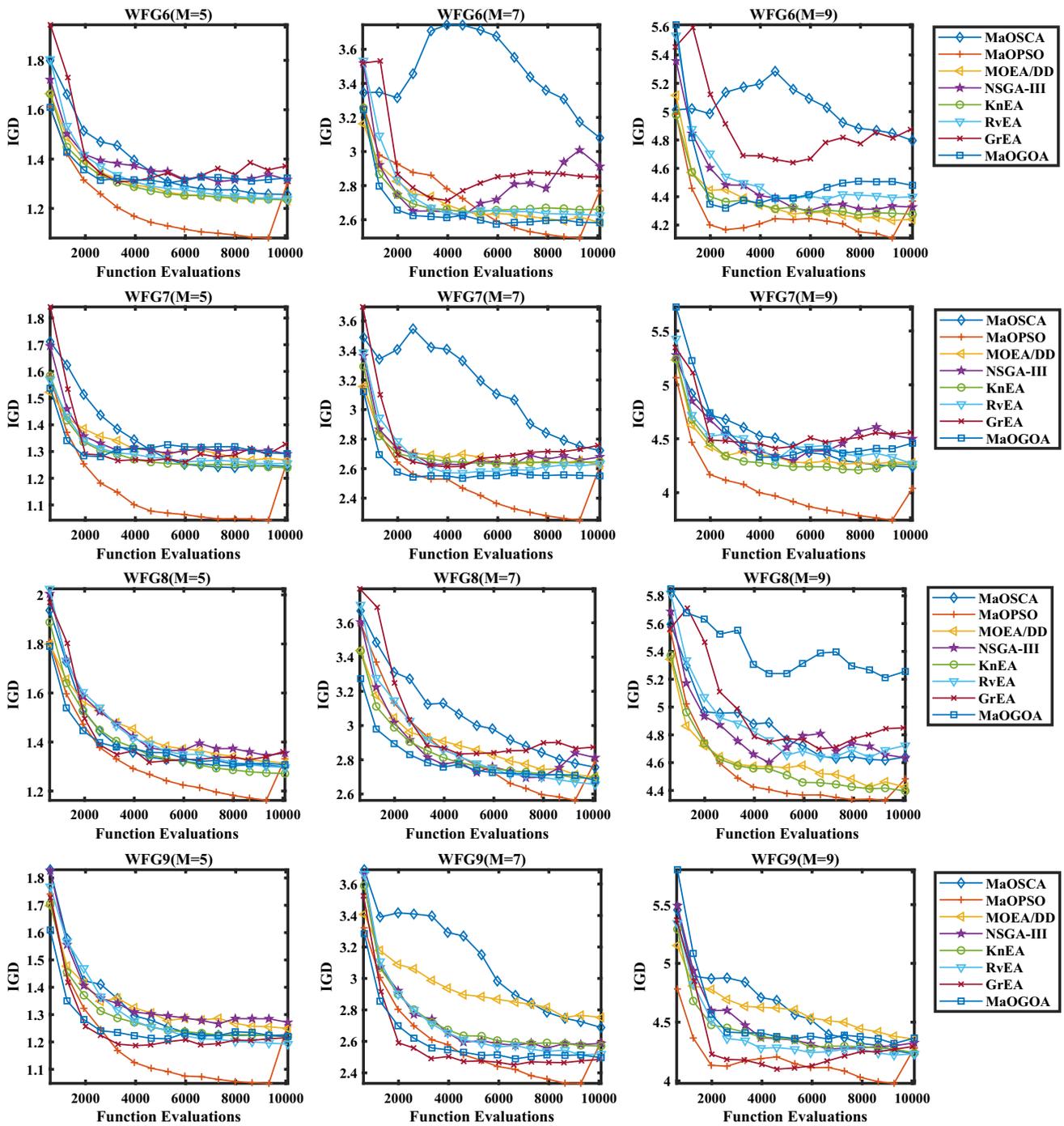


Fig. 5 (continued)

curves and box plots over 30 runs of all algorithms on the WFG benchmark problems are displayed in Figs. 5 and 6, respectively. MaOGOA is significantly superior to MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA

and GrEA algorithms, which shows that the MaOGOA has the ability to jump out from the local optimal in the iterative process and reaches a near-optimal solution more quickly. From these results, it is evident that MaOGOA is not only

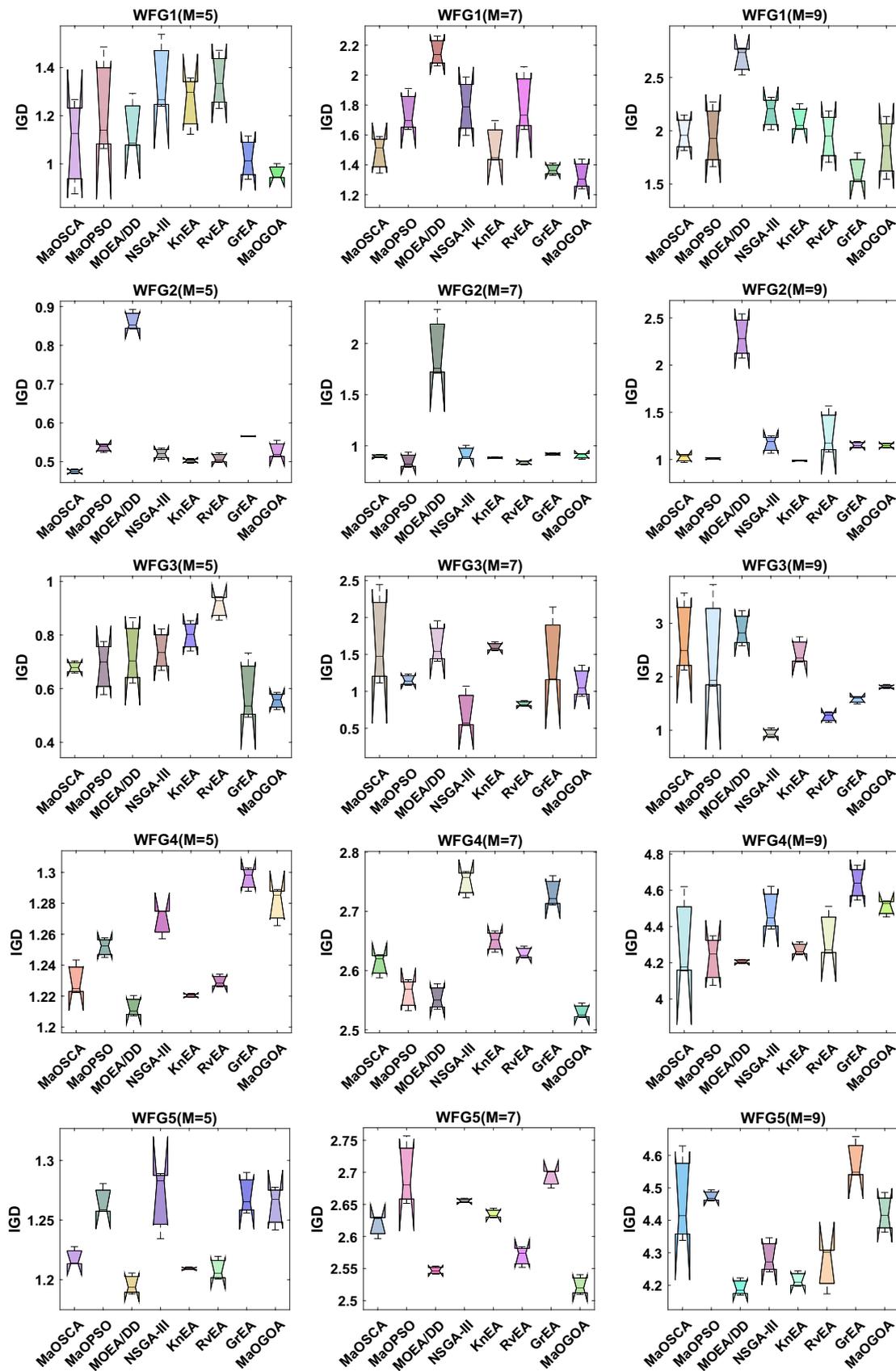


Fig. 6 Box plot based on the IGD metric for WFG problems

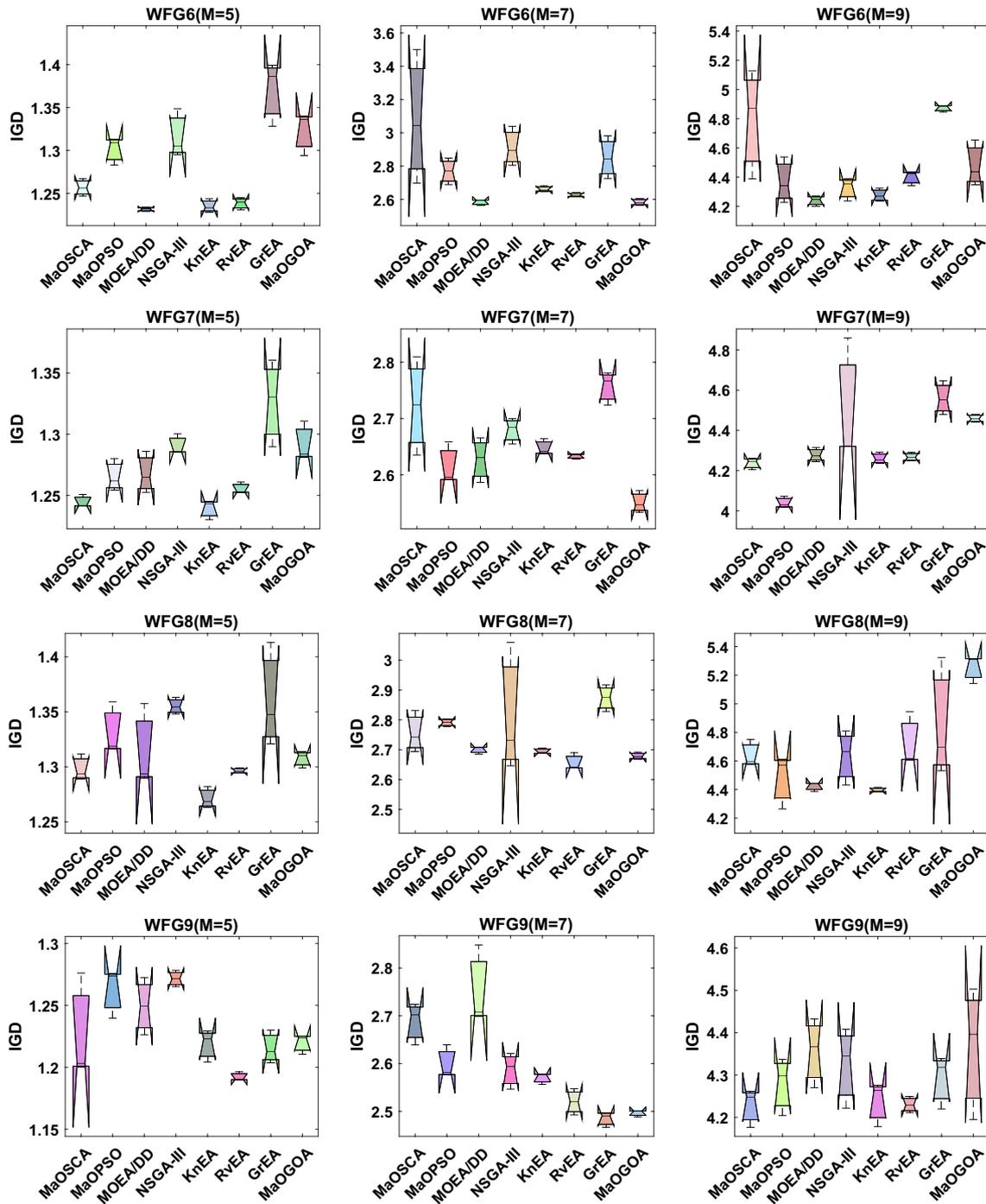


Fig. 6 (continued)

competitive but often surpasses the performance of other state-of-the-art algorithms in terms of the IGD metric across the WFG test suite. The consistently lower IGD values achieved by MaOGOA shown its capacity for generating

well-distributed and accurate approximations of the Pareto front shown in Fig. 7, making it a highly reliable option for complex many-objective optimization problems.

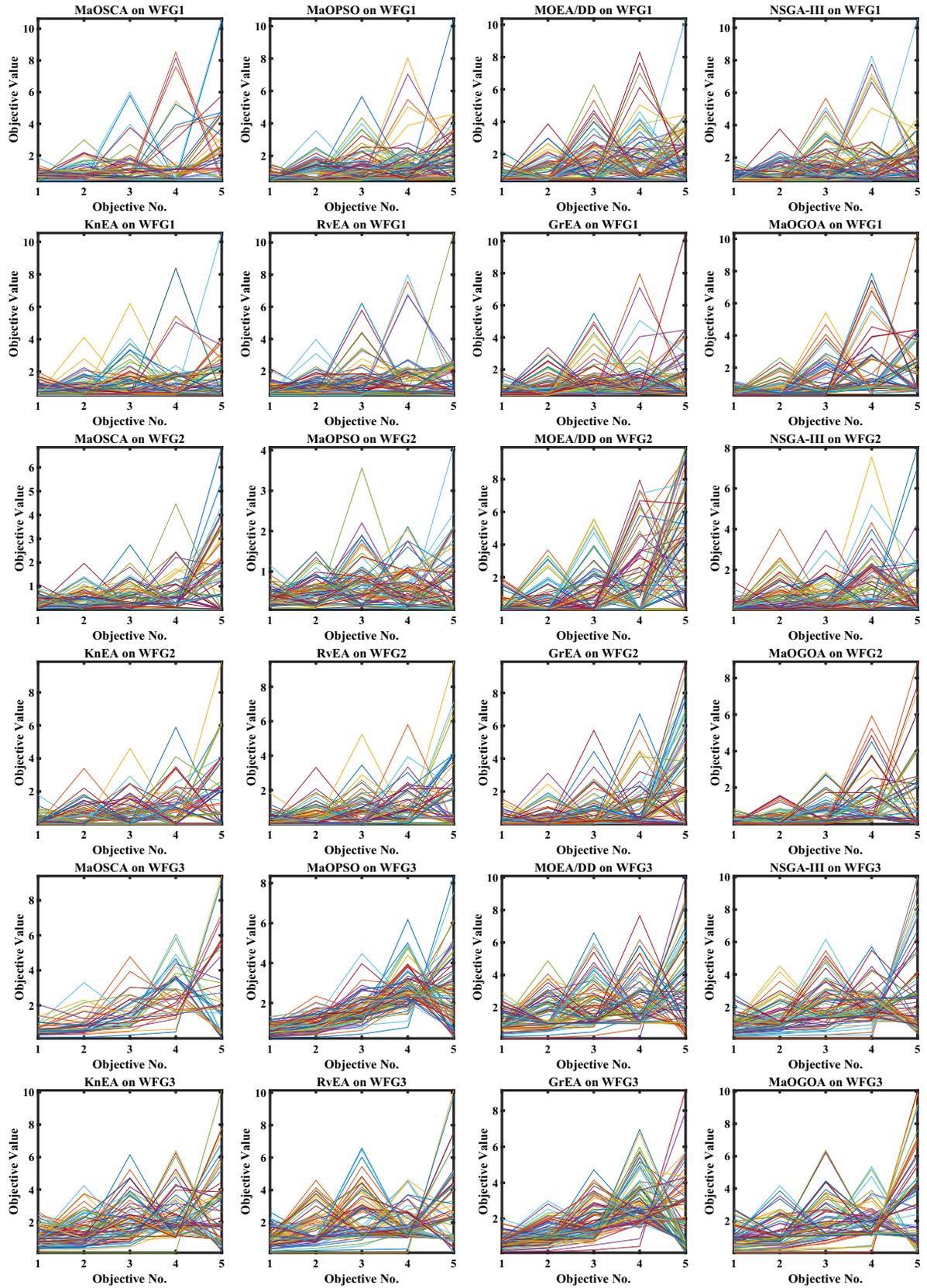


Fig. 7 Best Pareto optimal fronts for WFG problems

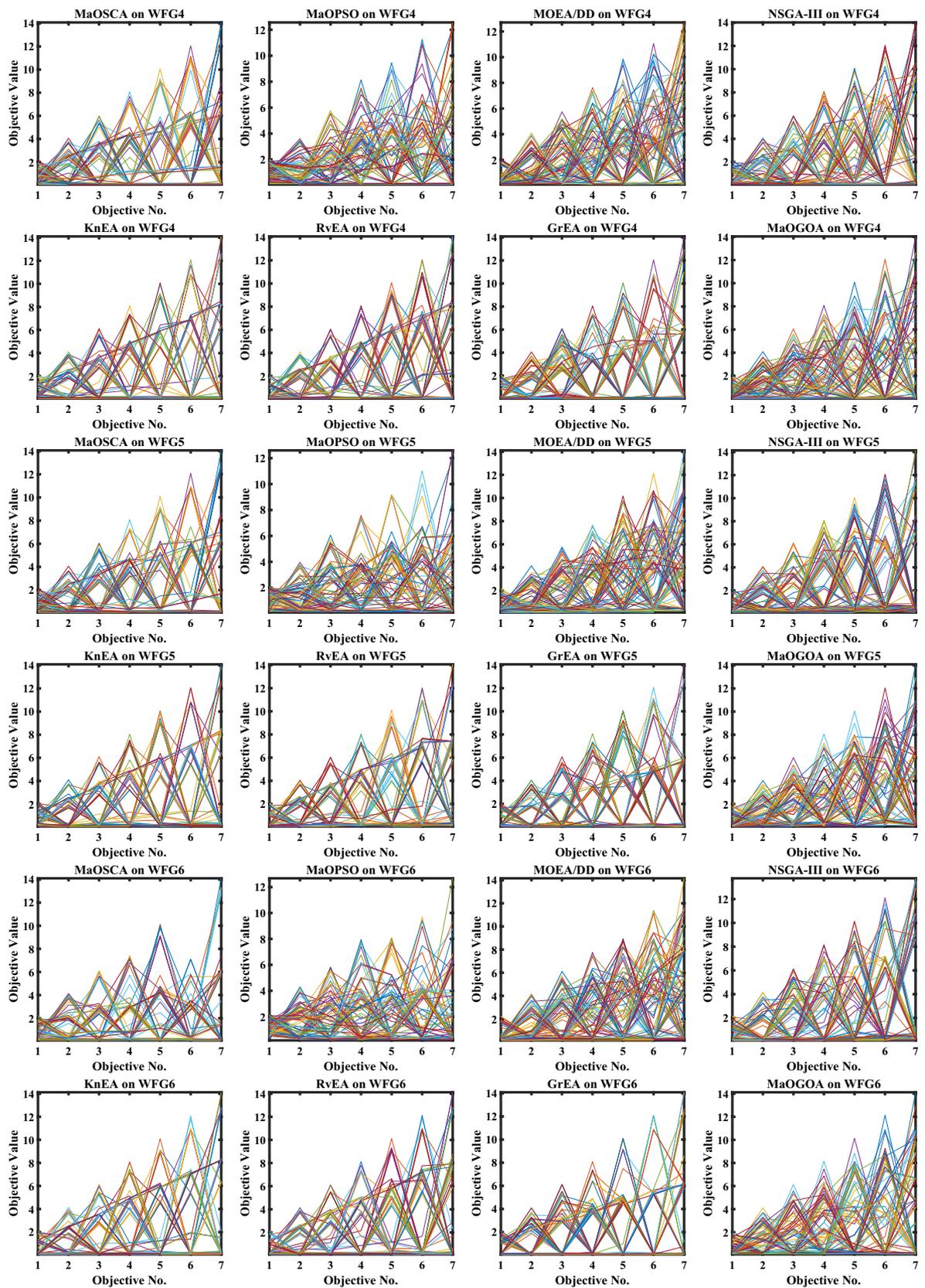


Fig. 7 (continued)

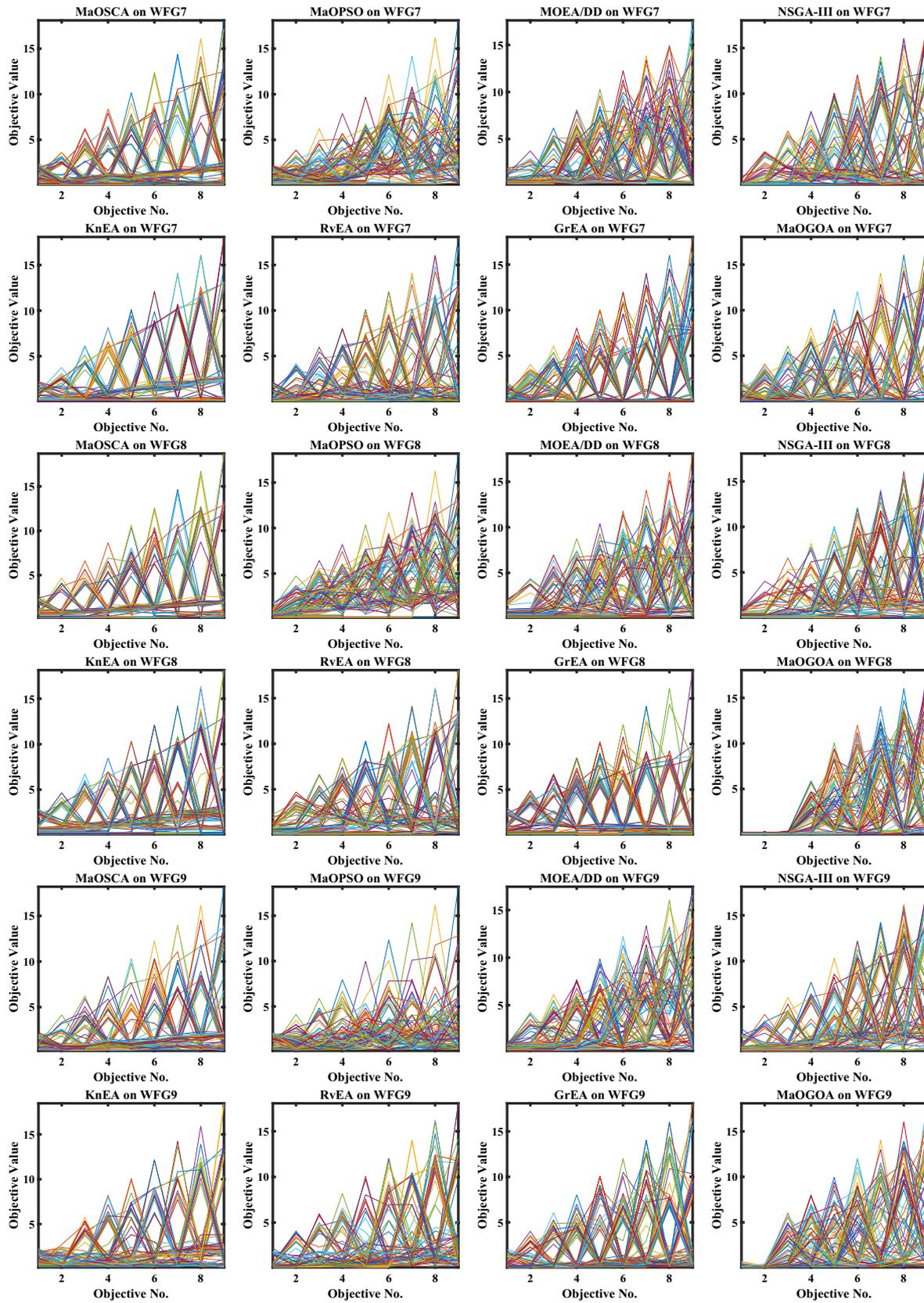


Fig. 7 (continued)

Table 4 Results of HV metric for WFG problems

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA
WFG1	5	14	0.602±0.066	0.535±0.065	0.631±0.056	0.516±0.041	0.511±0.047	0.51±0.049	0.618±0.018	0.671±0.039
	7	16	0.58±0.065	0.452±0.095	0.433±0.014	0.492±0.06	0.518±0.038	0.453±0.057	0.601±0.059	0.664±0.039
	9	18	0.415±0.023	0.426±0.081	0.381±0.061	0.411±0.027	0.377±0.037	0.511±0.076	0.562±0.055	0.493±0.102
WFG2	5	14	0.927±0.027	0.904±0.018	0.959±0.012	0.938±0.006	0.969±0.006	0.955±0.011	0.969±0.004	0.943±0.014
	7	16	0.913±0.015	0.913±0.025	0.903±0.098	0.943±0.012	0.955±0.003	0.959±0.008	0.969±0.004	0.947±0.011
	9	18	0.91±0.03	0.87±0.039	0.93±0.024	0.966±0.009	0.938±0.015	0.963±0.008	0.954±0.015	0.947±0.018
WFG3	5	14	0.062±0.027	0.012±0.009	0.081±0.026	0.057±0.013	0.044±0.032	0.047±0.006	0.052±0.032	0.103±0.025
	7	16	0±0	0±0	0±0	0.028±0.022	0±0	0±0	0±0	0.242±0.004
	9	18	0±0	0±0	0±0	0±0	0±0	0±0	0±0	0±0
WFG4	5	14	0.7±0.003	0.641±0.005	0.644±0.012	0.674±0.007	0.692±0.004	0.703±0.007	0.715±0.01	0.718±0.002
	7	16	0.757±0.024	0.672±0.014	0.659±0.031	0.757±0.012	0.772±0.014	0.772±0.007	0.808±0.004	0.78±0.009
	9	18	0.729±0.053	0.611±0.008	0.633±0.03	0.761±0.015	0.771±0.028	0.731±0.024	0.845±0.037	0.845±0.015
WFG5	5	14	0.691±0.006	0.61±0.005	0.624±0.005	0.649±0.008	0.678±0.002	0.68±0.008	0.691±0.004	0.695±0.005
	7	16	0.74±0.009	0.623±0.01	0.603±0.005	0.716±0.012	0.748±0.016	0.757±0.011	0.773±0.007	0.747±0.016
	9	18	0.709±0.022	0.521±0.026	0.535±0.035	0.709±0.002	0.703±0.017	0.603±0.021	0.821±0.012	0.8±0.011
WFG6	5	14	0.637±0.036	0.565±0.012	0.611±0.025	0.612±0.027	0.649±0.012	0.653±0.02	0.654±0.021	0.68±0.012
	7	16	0.595±0.174	0.519±0.054	0.605±0.019	0.705±0.011	0.73±0.025	0.728±0.002	0.724±0.024	0.723±0.018
	9	18	0.601±0.063	0.453±0.033	0.595±0.03	0.657±0.017	0.704±0.038	0.616±0.007	0.798±0.005	0.776±0.017
WFG7	5	14	0.709±0.01	0.623±0.008	0.583±0.008	0.676±0.025	0.681±0.006	0.69±0.016	0.74±0.007	0.75±0.008
	7	16	0.717±0.023	0.616±0.063	0.551±0.048	0.765±0.01	0.749±0.047	0.782±0.013	0.815±0.008	0.808±0.014
	9	18	0.683±0.014	0.493±0.065	0.523±0.029	0.777±0.012	0.728±0.037	0.696±0.06	0.857±0.012	0.854±0.004
WFG8	5	14	0.563±0.01	0.516±0.017	0.528±0.017	0.534±0.018	0.587±0.006	0.56±0.014	0.577±0.011	0.602±0.005
	7	16	0.508±0.027	0.475±0.077	0.568±0.022	0.58±0.036	0.653±0.009	0.581±0.02	0.599±0.016	0.595±0.006
	9	18	0.46±0.045	0.427±0.069	0.557±0.029	0.666±0.025	0.564±0.017	0.573±0.034	0.624±0.045	0.742±0.014
WFG9	5	14	0.596±0.042	0.57±0.033	0.551±0.008	0.57±0.054	0.594±0.02	0.64±0.017	0.659±0.066	0.692±0.013
	7	16	0.59±0.138	0.546±0.011	0.485±0.079	0.679±0.016	0.628±0.032	0.692±0.032	0.749±0.017	0.69±0.018
	9	18	0.539±0.132	0.473±0.021	0.488±0.018	0.624±0.088	0.556±0.029	0.626±0.032	0.743±0.072	0.759±0.022

Table 4 presents the Hypervolume (HV) performance metrics for the optimization algorithms MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA, GrEA and MaOGOA across WFG test problems involving 5, 7 and 9 objectives. The Hypervolume metric is crucial in many-objective optimization because it quantitatively assesses an algorithm's ability to cover the Pareto front. A higher HV value is preferable, as it signifies a more extensive coverage of the objective space by the algorithm's solutions, indicating superior performance in terms of both convergence to and diversity within the Pareto front. In WFG1, the MaOGOA algorithm exhibits commendable results with HV values of 0.671, 0.664 and 0.493 for the 5, 7 and 9-objective cases, respectively. These values consistently outperform those achieved by other algorithms like MaOSCA, MaOPSO and NSGA-III, highlighting MaOGOA dominance in various many-dimensional objective spaces.

In WFG2, MaOGOA maintains a leading stance, with HV values consistently above 0.943, 0.947 and 0.947 for the 5, 7 and 9-objective instances, respectively. The algorithm demonstrates superior performance, although closely followed by GrEA and RvEA. WFG3 presents a unique challenge where for the 5-objective case, MaOGOA achieves the highest HV value reported as 0.103, significantly surpassing all compared algorithms. However, for the 7-objective case, the HV metric drops to 0.242, respectively, indicating a diminishing performance as the number of objectives increases, a trend observed across all algorithms for this problem. In the WFG4 scenario, MaOGOA consistently displays robust HV values of 0.718, 0.78 and 0.845 for 5, 7 and 9-objective problems, respectively. It demonstrates not only the ability of MaOGOA to handle an increasing number of objectives but also its potential to outperform other algorithms such as KnEA and MOEA/DD, especially as the complexity of

Table 5 WRST and p value based on HV metric for WFG problems

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA	p value
WFG1	5	14	5	3.33	6.67	2.33	2.67	2.67	6	7.33	0.051
	7	16	6.67	3	2.33	3.33	4.33	2.33	6.33	7.67	0.030
	9	18	4	3.67	2.33	4	2.33	6.67	7.67	5.33	0.072
WFG2	5	14	2.33	1	6.67	3.67	6.67	4.67	7.33	3.67	0.013
	7	16	2	1.67	4	4.33	5.33	6	8	4.67	0.038
	9	18	2.67	1	3.33	7.33	4.33	7	6	4.33	0.019
WFG3	5	14	5	1.33	6.33	5	3.33	3.33	4.33	7.33	0.090
	7	16	3.83	3.83	3.83	8	3.83	3.83	3.83	5	0.011
	9	18	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	1.000
WFG4	5	14	5.67	1.33	1.67	3	4	5.67	7	7.67	0.006
	7	16	4.33	1.67	1.33	4	5.33	5	8	6.33	0.015
	9	18	3.67	1.33	1.67	4.67	5.67	4	7.67	7.33	0.008
WFG5	5	14	6.33	1	2	3	4.33	5	6.67	7.67	0.007
	7	16	5	2	1	3	5	7	7.67	5.33	0.008
	9	18	4.67	1.33	1.67	5.33	5	3	8	7	0.006
WFG6	5	14	4.67	1	2.67	2.33	5.33	6	6	8	0.009
	7	16	4	1.33	2.33	4.33	5.67	6.33	6.33	5.67	0.094
	9	18	3.33	1	2.67	4.67	6	3.33	8	7	0.007
WFG7	5	14	5.67	2	1	3.67	3.67	5	7.33	7.67	0.006
	7	16	3.33	2	1	4.33	4.33	6	7.67	7.33	0.005
	9	18	3.33	1.67	1.33	6	4.67	4	7.67	7.33	0.006
WFG8	5	14	4.33	1.67	1.67	3	6.67	4.67	6	8	0.009
	7	16	1.67	2	3.33	5	8	4.33	6	5.67	0.026
	9	18	1.67	1.33	3.67	6.67	3.67	5	6	8	0.007
WFG9	5	14	4	3	1.67	3.33	4	6.33	6	7.67	0.053
	7	16	3.33	2	1.67	5.33	4	6	8	5.67	0.023
	9	18	2.67	1.33	2.67	5	4	5.33	7.67	7.33	0.012
$\pm/=$			0/26/1	0/26/1	0/26/1	2/24/1	2/24/1	0/26/1	11/15/1	11/15/1	
Rank			3	3	3	2	2	3	1	1	

the problem space increases. For WFG5, the HV metric for MaOGOA is impressive, indicating better Pareto front coverage than most other algorithms across all objective instances, although it sees some competition from GrEA and RvEA in the 5-objective case. In WFG6, the HV values for MaOGOA are consistently high, although it is closely followed by other algorithms, especially in the 9-objective case. This suggests that while MaOGOA is effective, there is strong competition and it does not always hold a clear lead. WFG7 results show that MaOGOA has HV values indicating superior Pareto front coverage across the 5, 7 and 9-objective cases, with performance generally superior to that of MOEA/DD and NSGA-III, although it does not always outperform RvEA. For WFG8, MaOGOA performance is robust in the 5-objective case but sees a slight decrease in the 7 and 9-objective

scenarios, where it remains competitive but does not always secure the top position, particularly against the strong performance of GrEA in the 9-objective case. Lastly, in WFG9, MaOGOA shows a high HV in the 5-objective case and remains competitive in the 7 and 9-objective instances, although it is closely matched by RvEA and surpassed by GrEA in the 9-objective scenario.

In addition to the initial experiments, Table 5 outlines the p values from the Wilcoxon signed-rank test, highlighting MaOGOA's significant advantage over other foundational algorithms. Specifically, MaOGOA outperformed MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA in 11, 11, 11, 9, 9, 11 and 1 instances, respectively, with all corresponding p values falling below 0.05, indicating a significant effect. Furthermore, the Friedman test shows

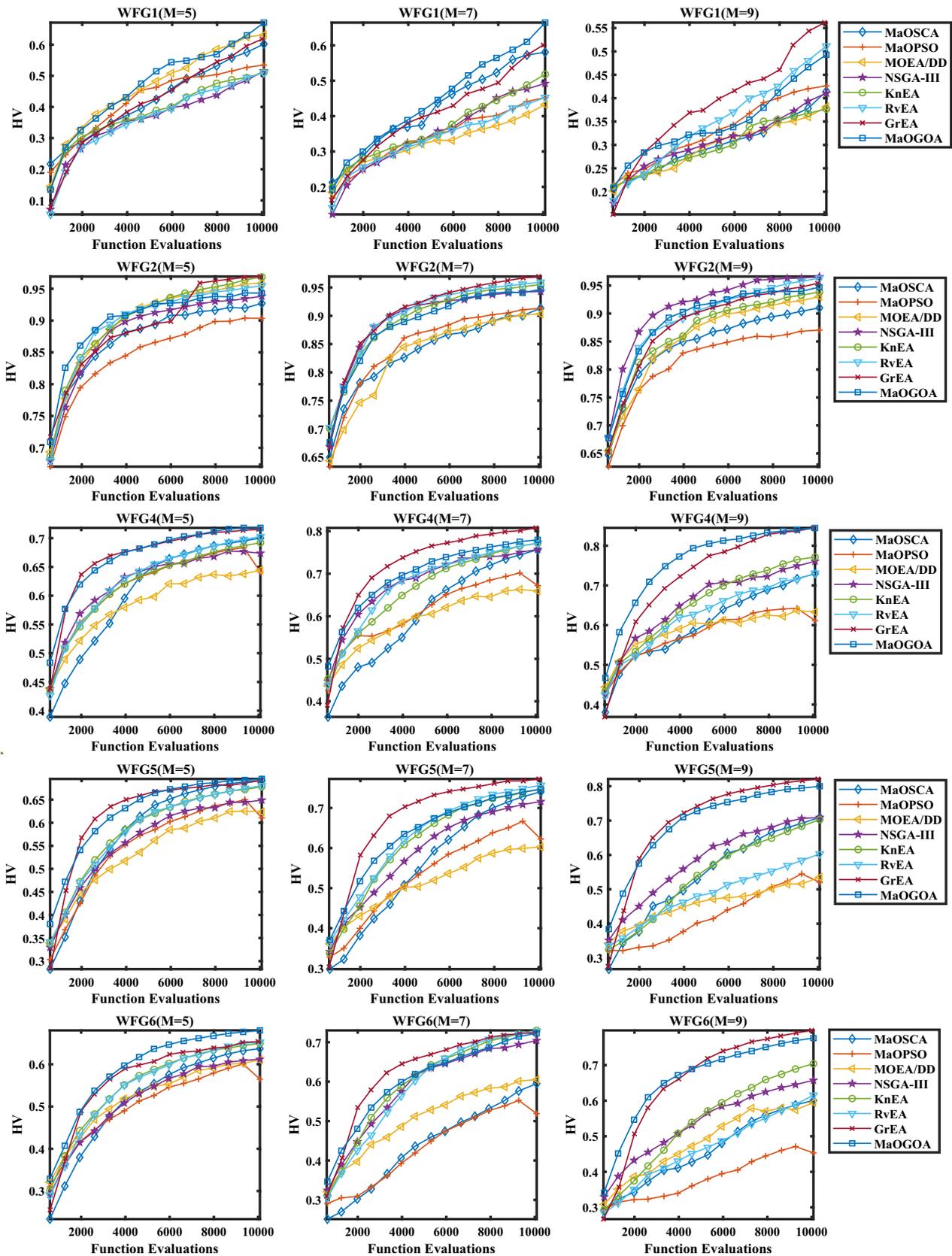


Fig. 8 Convergence curve based on the HV metric for WFG problems

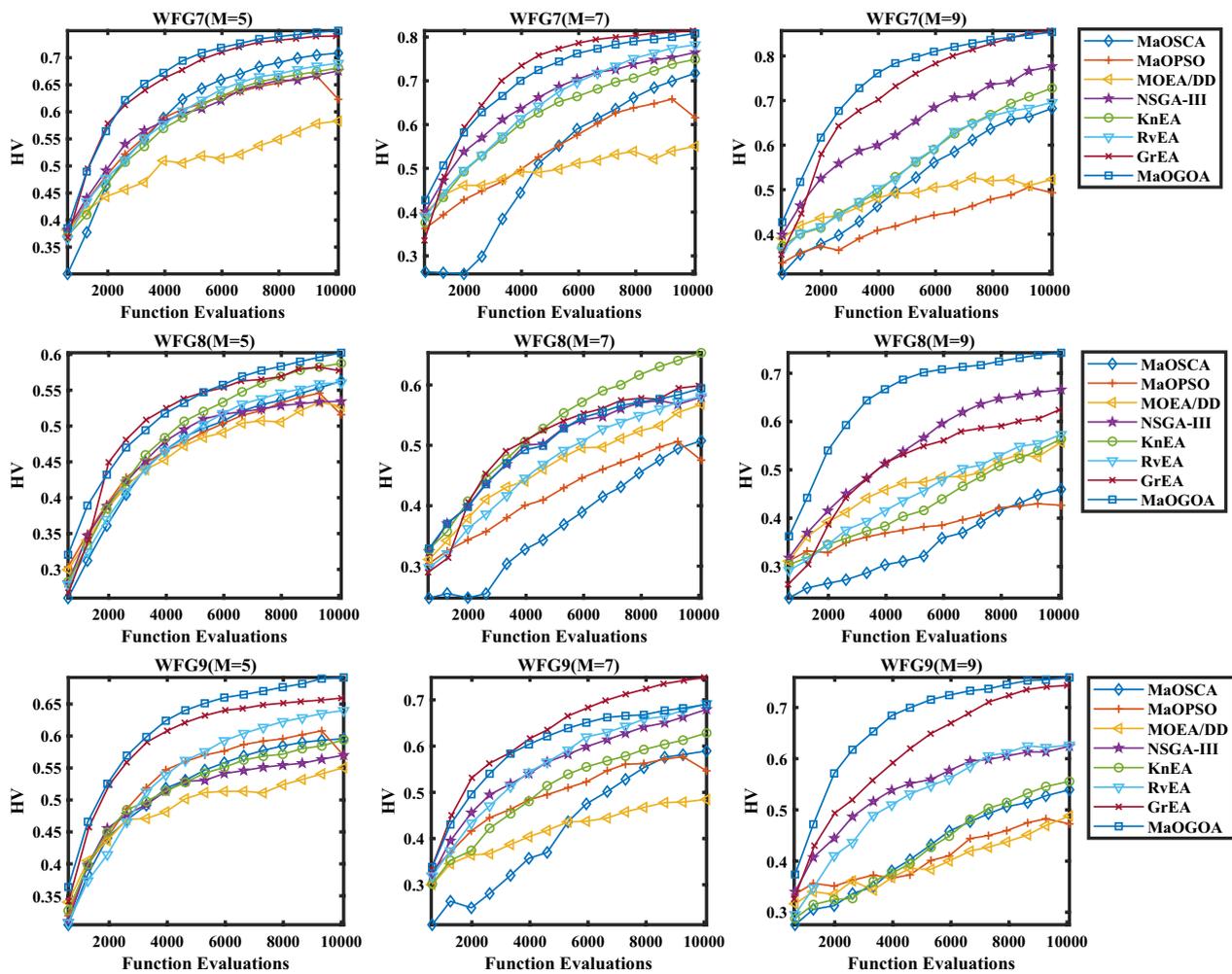


Fig. 8 (continued)

fewer instances where MaOGOA performed worse than these algorithms: 0, 0, 0, 2, 2, 0 and 11 cases, respectively. Collectively, these results highlight MaOGOA’s statistical significance in comparison to these established algorithms. To show the efficiency of MaOGOA clearly, the average HV convergence curves and box plot over 30 times run of all algorithms on WFG benchmark problems are plotted in Figs. 8 and 9, respectively. MaOGOA is significantly superior to MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA algorithms, which shows that the MaOGOA has the ability to jump out from the local optimal in the iterative process and reaches a near-optimal solution more quickly. The overall performance of the MaOGOA algorithm, characterized by high HV values, elucidates its ability to cover a larger volume of the Pareto front

effectively. It exhibits robustness in problems with a lower number of objectives and maintains competitive efficacy as the complexity increases, although it faces stiff competition from other algorithms, notably RvEA and GrEA, in higher dimensional objective spaces.

Table 6 demonstrates the runtime efficiency (RT) of MaOGOA against other leading many-objective evolutionary algorithms across various WFG test problems. The RT metric, where lower values are indicative of superior computational efficiency, is utilized as the basis for this comparative analysis. For instance, in the WFG1 problem with 5 objectives (M) and 14 decision variables (D), MaOGOA showcases its computational prowess with a RT of 0.59 s. This outperforms MaOSCA, MOEA/DD and NSGA-III, which have higher RTs of 1.86 s, 2.04 s

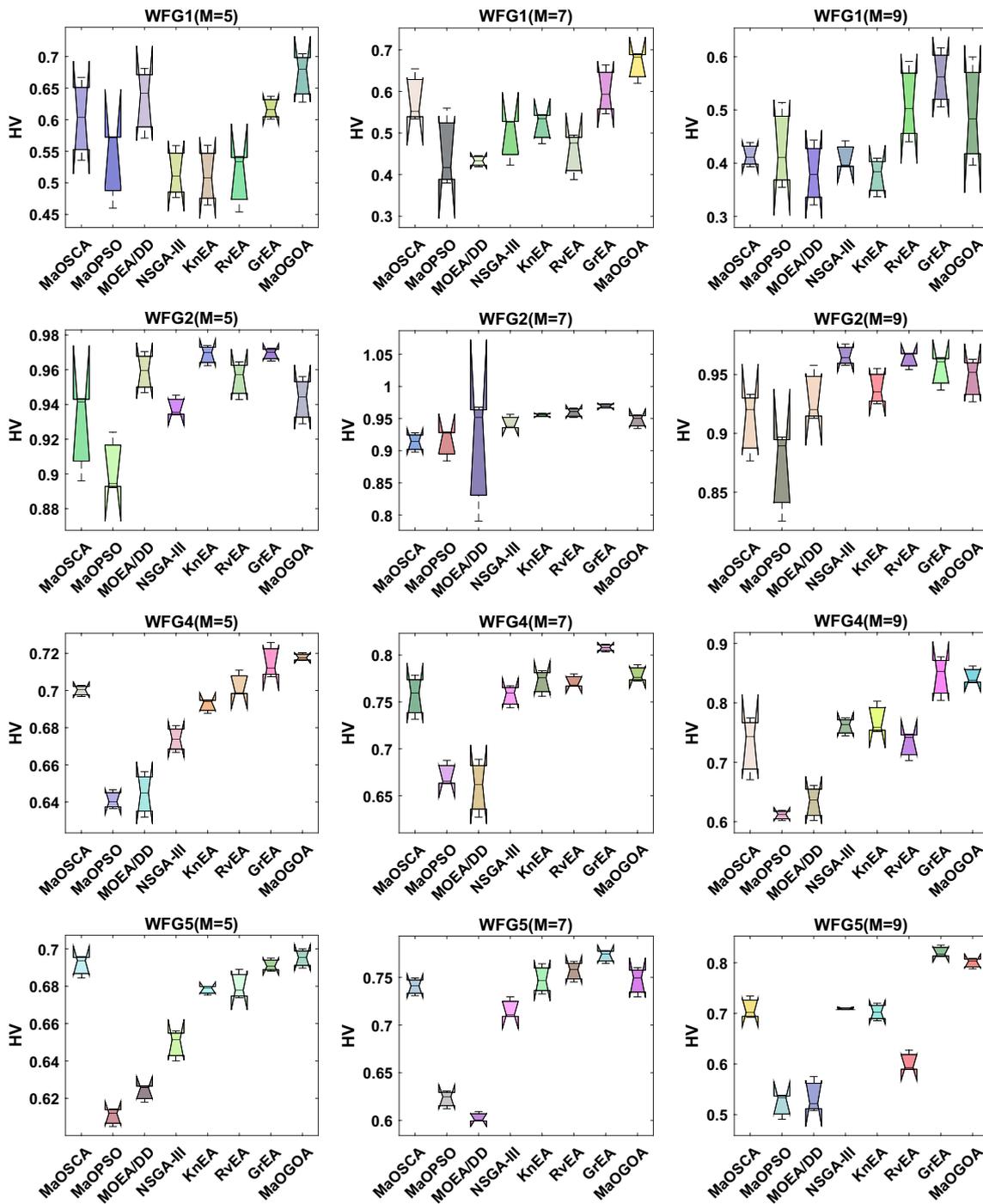


Fig. 9 Box plot based on the HV metric for WFG problems

and 1.19 s, respectively. Notably, as the complexity of the problem increases to 9 objectives and 18 decision variables in WFG1, MaOGOA maintains its efficiency with a RT of 0.52 s, whereas algorithms like MOEA/DD and KnEA show

RTs of 3.20 s and 7.26 s, demonstrating a more significant computational load. In the WFG2 scenario with 7 objectives and 16 decision variables, MaOGOA still maintains a higher RT of 0.48 s while others such as MaOPSO and RvEA have

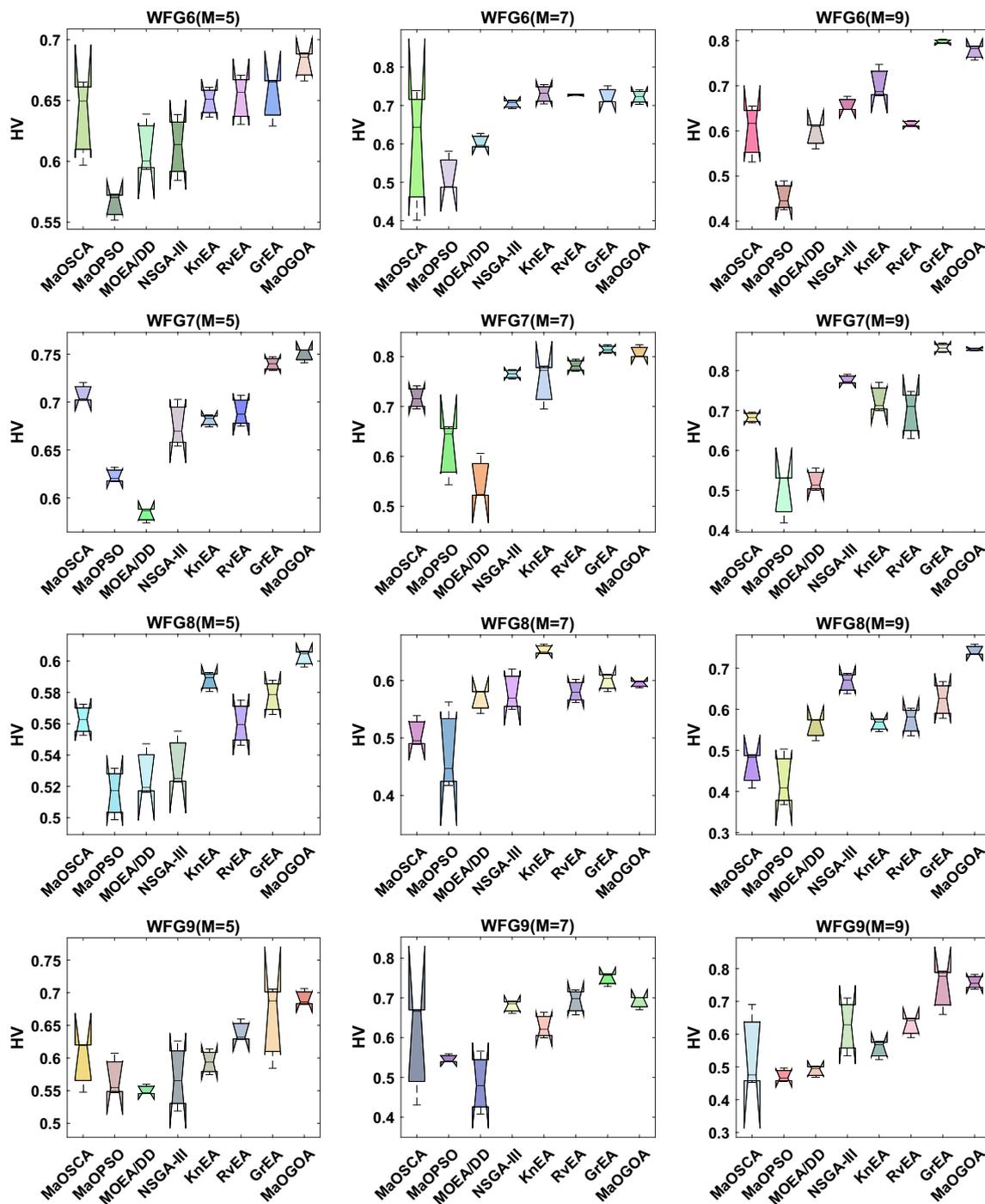


Fig. 9 (continued)

RTs of 0.8 s and 0.76. In WFG3 with 5 objectives and 14 decision variables which have shown RT of 0.44 s. This is considerably lower than the RTs claimed by MaOSCA and MOEA/DD at 1.81 s and 3.77 s, which further supports

MaOGOA’s ability to efficiently explore the search space with less computational power. For WFG7 problem with 9 objectives and 18 decision variables in the higher dimensional objective space, MaOGOA achieves an RT of 0.5 s.

Table 6 Results of RT metric for WFG problems

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA
WFG1	5	14	1.86	0.82	2.04	1.19	5.46	0.76	0.88	0.59
	7	16	1.54	0.69	2.93	1.14	6.15	0.80	0.66	0.47
	9	18	1.65	0.70	3.20	1.14	7.26	0.81	0.70	0.52
WFG2	5	14	1.84	0.85	2.36	0.98	5.70	0.71	0.80	0.53
	7	16	1.69	0.80	3.26	1.21	6.98	0.76	0.73	0.48
	9	18	1.73	0.80	3.26	1.88	7.39	1.03	0.74	0.48
WFG3	5	14	1.81	0.72	3.77	1.04	7.69	0.81	0.76	0.44
	7	16	1.88	0.71	4.97	1.95	8.88	1.75	0.79	0.44
	9	18	1.84	0.75	5.21	1.90	9.28	1.68	0.90	0.46
WFG4	5	14	1.66	0.88	3.86	0.97	7.34	0.72	0.76	0.49
	7	16	1.68	0.89	4.89	1.10	8.30	0.76	0.77	0.49
	9	18	1.71	0.89	5.09	1.66	8.83	1.19	0.81	0.49
WFG5	5	14	1.58	0.88	3.72	0.88	7.39	0.65	0.74	0.50
	7	16	2.08	1.11	4.79	1.02	8.57	0.75	0.79	0.50
	9	18	1.76	0.95	5.16	1.80	8.85	1.35	0.80	0.50
WFG6	5	14	1.65	0.86	3.09	0.89	6.54	0.63	0.72	0.51
	7	16	1.58	0.84	4.24	1.01	7.90	0.71	0.74	0.49
	9	18	1.77	0.84	4.56	1.74	8.52	1.06	0.77	0.60
WFG7	5	14	1.71	0.92	4.25	0.94	8.51	0.70	0.78	0.51
	7	16	1.73	0.92	5.38	1.08	9.26	0.74	0.81	0.51
	9	18	1.74	0.89	5.50	2.50	1.02	1.23	0.85	0.50
WFG8	5	14	1.51	0.87	2.83	0.96	6.60	0.67	0.72	0.52
	7	16	1.62	0.86	4.13	1.09	7.69	0.75	0.71	0.51
	9	18	1.74	0.86	4.51	2.13	8.98	1.27	0.74	0.49
WFG9	5	14	1.66	0.97	4.04	1.06	9.06	0.73	0.85	0.55
	7	16	1.73	0.97	5.28	1.21	1.06	0.85	0.84	0.53
	9	18	1.73	0.91	5.80	2.27	1.12	1.18	0.90	0.52

This is in stark contrast to KnEA and MOEA/DD which have RTs of 1.02 s and 5.5 s, respectively, highlighting the scalability and efficiency of MaOGOA in handling complex many-objective optimization problems. MaOGOA consistent outperformance in RT across the WFG suite, such as on an average 0.51 s in WFG8 and 0.53 s in WFG9 respectively, signifies its superiority in speed efficiency. This trend of fast solution convergence towards better solutions using MaOGOA is evident irrespective of the types of WFG problems and configurations. This enables MaOGOA to be an excellent choice for solving LSMOP problems where computational time is of the essence.

4.3 Experimental Results on RWMaOP Problems

As seen in Table 7, for RWMOPs, MaOGOA demonstrates exceptional performance as per HV metric. In RWMaOP1, with 9 objectives and 7 decision variables, the HV of MaOGOA is 0.003 ± 0.00016 , which is the highest among

the compared algorithms indicating it provides the most exhaustive coverage of the Pareto front for this problem. Similarly, for RWMaOP2 with 4 objectives and 10 decision variables, MaOGOA has an HV of 0.085 ± 0.00017 , significantly outperforming the other algorithms. In the case of RWMaOP3 with 7 objectives and 3 decision variables, MaOGOA HV achieves 0.018 ± 0.00019 . In RWMaOP4, with 5 objectives and 6 decision variables, MaOGOA exhibits an excellent HV of 0.523 ± 0.0003 , which is the highest among the tested algorithms. For RWMaOP5, which has 4 objectives and 4 decision variables, MaOGOA HV value of 0.553 ± 0.00085 is once again the highest, showing its consistent performance in capturing a wide and diverse set of optimal solutions.

Table 8 details the results of the Friedman test and the p values from the Wilcoxon signed-rank test, affirming MaOGOA significant superiority over other algorithms. Specifically, MaOGOA demonstrated better performance compared to MaOSCA, MaOPSO, MOEA/DD, NSGA-III,

Table 7 Results of the HV metric for RWMaOPs

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA
RWMaOP1	9	7	0.002 ± 0.0001	0.001 ± 0.00018	0.001 ± 0.00013	0.002 ± 0.00026	0.001 ± 0.00021	0.002 ± 0.00033	0.001 ± 0.00008	0.003 ± 0.00016
RWMaOP2	4	10	0.08 ± 0.00076	0.062 ± 0.00346	0.061 ± 0.00469	0.081 ± 0.00146	0.073 ± 0.00371	0.081 ± 0.00029	0.02 ± 0.00272	0.085 ± 0.00017
RWMaOP3	7	3	0.017 ± 0.00085	0.013 ± 0.00032	0.017 ± 0.00096	0.017 ± 0.00028	0.016 ± 0.00015	0.017 ± 0.00025	0.017 ± 0.00016	0.018 ± 0.00019
RWMaOP4	5	6	0.524 ± 0.00252	0.48 ± 0.00635	0.537 ± 0.0125	0.538 ± 0.00955	0.542 ± 0.00313	0.534 ± 0.00195	0.474 ± 0.0111	0.523 ± 0.00003
RWMaOP5	4	4	0.558 ± 0.00912	0.499 ± 0.00253	0.547 ± 0.0316	0.535 ± 0.00318	0.547 ± 0.00489	0.533 ± 0.00379	0.538 ± 0.00554	0.553 ± 0.00085

KnEA, RvEA and GrEA. Each of these comparisons yielded *p* values less than 0.05, highlighting the substantial impact and effectiveness of MaOGOA. To show the efficiency of MaOGOA clearly, the average HV convergence curves and box plot over 30 times run of all algorithms on RWMaOP problems are plotted in Figs. 10 and 11, respectively. MaOGOA is significantly superior to MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA algorithms, which shows that the MaOGOA has the ability to jump out from the local optimal in the iterative process and reaches a near-optimal solution more quickly. Overall, the HV results indicate that MaOGOA is highly capable of identifying solutions that cover a large extent of the Pareto front shown in Fig. 12, outperforming other algorithms in all the given cases. This capacity is essential for obtaining a broad representation of the optimal trade-offs between objectives in many-objective optimization problems. The consistent high performance of MaOGOA across various real-world problems establishes it as a robust and proficient algorithm for solving complex optimization problems that require a rich diversity of high-quality solutions.

From Table 9, the overall running time of the MaOGOA is consistently the lowest among the compared algorithms, indicating its superior computational efficiency. For the Car Cab Design problem (RWMaOP1), MaOGOA running time is 0.47 s, which represents approximately 21%, 18%, 64% and 10% of the runtimes of NSGA-III, MaOPSO, MaOSCA and RvEA, respectively. Similarly, for the 10-Bar Truss Structure problem (RWMaOP2), MaOGOA running time of 5.68 s accounts for 94%, 78% and 75% of the runtimes of NSGA-III, MaOPSO and MaOSCA, respectively. In each of these cases, MaOGOA running time is significantly lower than that of its competitors, particularly excelling with running times of 0.31 s for RWMaOP3, 0.29 s for RWMaOP4 and 0.29 s for RWMaOP5, which are significantly lower than the next best performing algorithms. In comparison to MOEA/DD, NSGA-III, MaOPSO, MaOSCA, KnEA, GrEA and RvEA. The MaOGOA algorithm showcased enhanced performance across all five evaluated cases, featuring significantly reduced running times, which highlights its efficiency in various engineering optimization contexts. Consequently, the experimental data in Table 9 lead to the conclusion that MaOGOA not only operates at a swifter pace but also achieves greater search efficiency and effectiveness in addressing many-objective optimization challenges.

4.4 Discussion

MaOGOA incorporates a sophisticated niche preservation strategy which is crucial for maintaining diversity in the

Table 8 Freidman test and p value based on HV metric for RWMaOPs

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA	p value
RWMaOP1	9	7	4.67	1	2	6.33	4	7	3.33	7.67	0.006
RWMaOP2	4	10	5.67	2.67	2.33	6.33	4	6	1	8	0.006
RWMaOP3	7	3	4	1	4.33	4.33	2.67	5.33	6.33	8	0.024
RWMaOP4	5	6	3.67	1.67	6.67	6	7.33	6	1.33	3.33	0.009
RWMaOP5	4	4	7.33	1	4.67	3	5.67	3	4.33	7	0.023
$\pm/=$			1/4/0	0/5/0	0/5/0	0/5/0	1/4/0	0/5/0	0/5/0	3/2/0	
Rank			2	3	3	3	2	3	3	1	

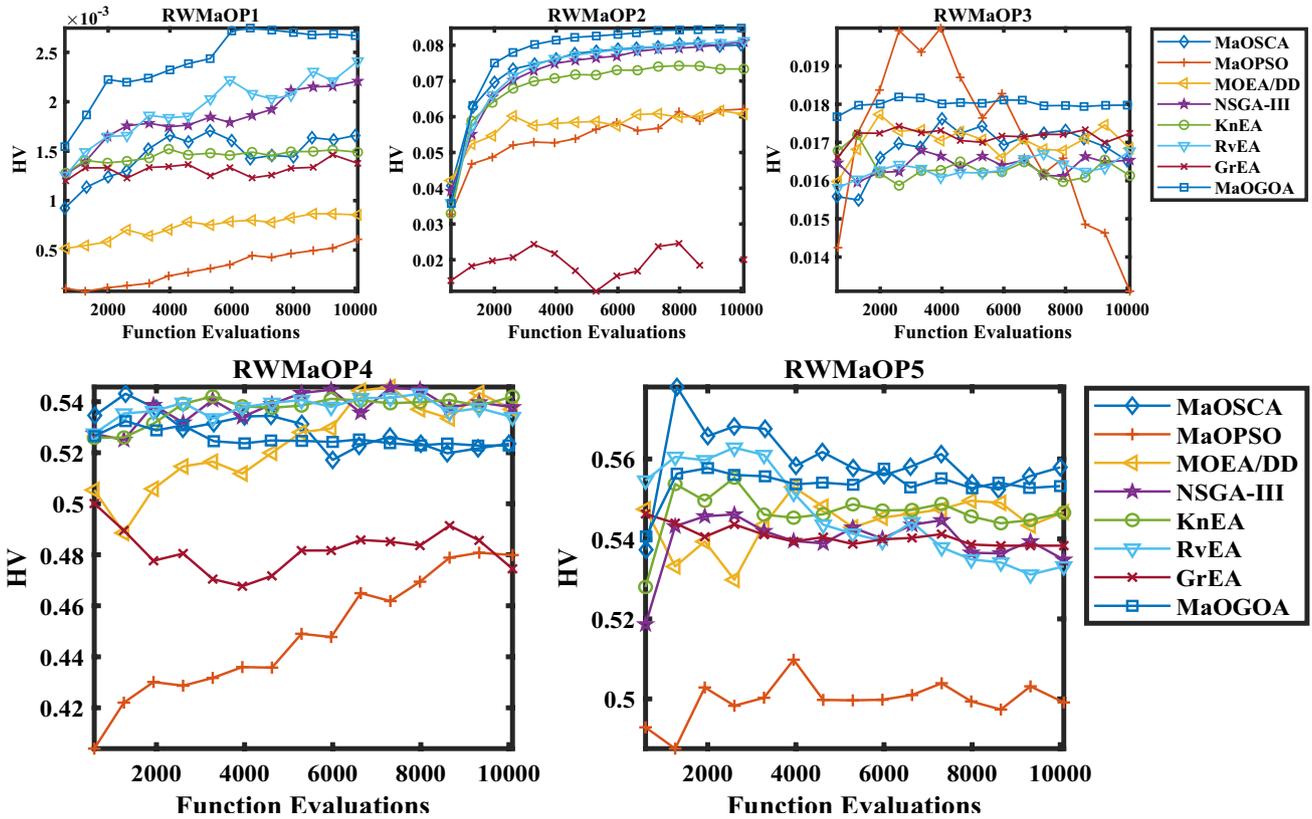


Fig. 10 Convergence curve based on HV metric for RWMaOPs

population. This method is highly useful in preventing the elimination of better solutions at the initial stages, a common occurrence in competitive environments; a weakness inherent in most algorithms. The incorporation of the IFM in MaOGOA represents a marked improvement over conventional feedback procedures in evolutionary algorithms. It uses past performance information to modify the search process on the fly, directing the computation towards promising areas of the search space. Through gathering

and applying feedback data across generations, MaOGOA provides for a more directed and efficient search, which is faster in terms of convergence, while also providing for better quality of solutions. Unlike static reference point methods that are employed by other algorithms, MaOGOA incorporates dynamic reference points that adjust with the population. This dynamic adjustment helps in achieving a better coverage and spread of the nondominated solutions since the reference points better reflect the current

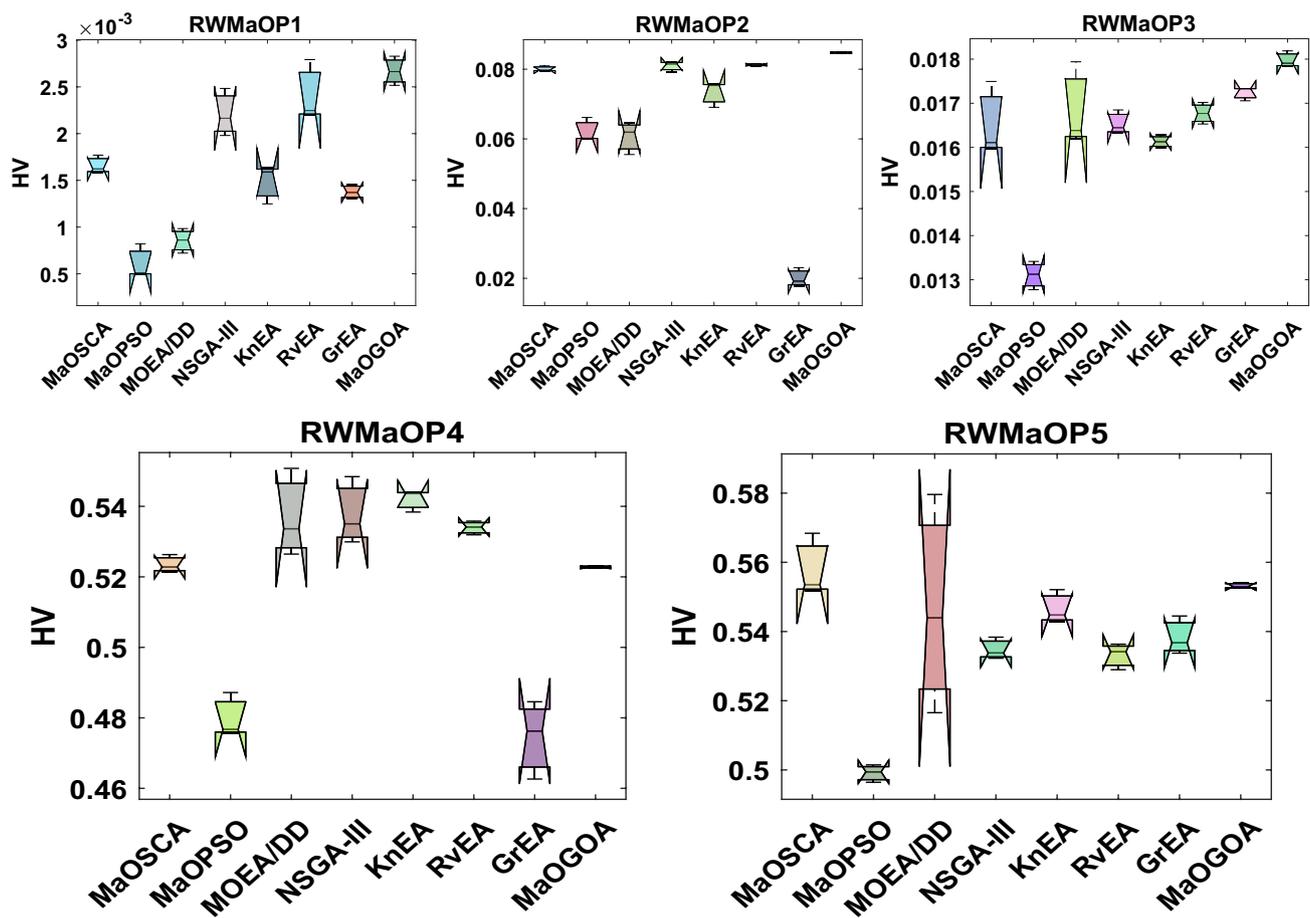


Fig. 11 Box plots based on the HV metric for RWMaOPs

Pareto front. Compared to other optimization techniques, MaOGOA has less sensitivity to changes in parameters, thereby making the algorithm more versatile to different problem contexts. This robustness is due to mechanisms that are able to learn and adjust the trade-off between exploration and exploitation without the need for fine-tuning of parameters. It was found that the performance advantage of MaOGOA is more significant in higher dimensional objective spaces. This is due to its high performance in dealing with non-linear Pareto fronts and its capacity in keeping the distribution of solutions reasonable across these fronts especially in many-objective problem. These features in combination result in slightly better performance of MaOGOA compared to other algorithms. However, it should also be emphasized that the extent of improvement depends on certain parameters of the problem and the decision space, including the number

of objectives. Additional statistical analysis and more comprehensive comparisons supported these observations, showing that MaOGOA has comparable or even better performance than the other methods and outperforms other algorithms when it is difficult to find a meaningful balance between convergence and diversity.

5 Conclusions

In this research, a new Many-Objective Grasshopper Optimization Algorithm known as MaOGOA is proposed to solve MaOPs due to difference PF shapes. Likewise, MaOGOA employs a reference point and niche preserve, which successfully partitions a favorable subset in the objective space. This process ensures both high-quality convergence and the preservation of solution diversity for subsequent selection

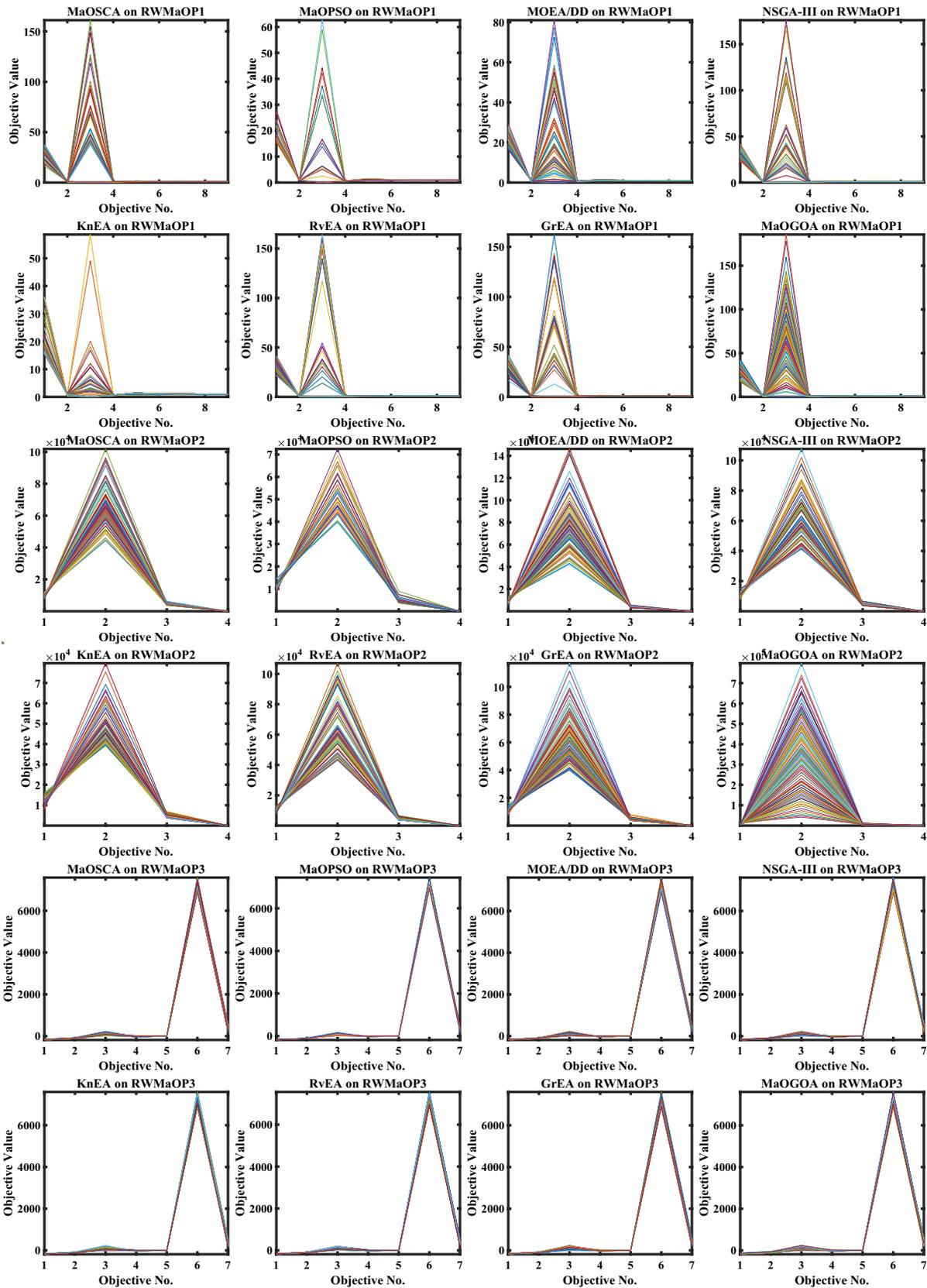


Fig. 12 Best Pareto optimal fronts for RWMaOPs

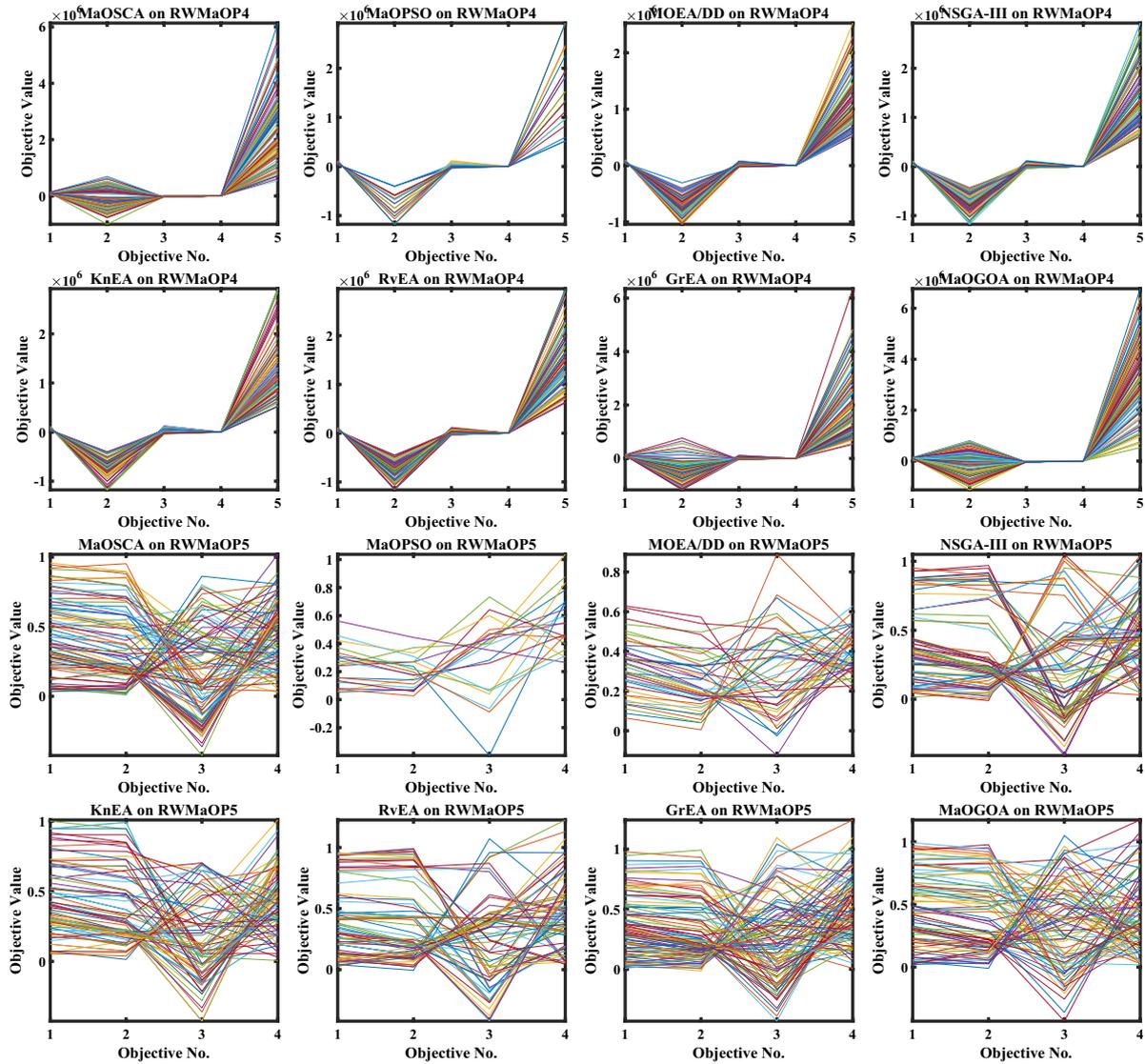


Fig. 12 (continued)

Table 9 Results of RT metric for RWMaOPs

Problem	M	D	MaOSCA	MaOPSO	MOEA/DD	NSGA-III	KnEA	RvEA	GrEA	MaOGOA
RWMaOP1	9	7	0.96	2.59	0.73	2.25	7.67	1.73	4.67	0.47
RWMaOP2	4	10	6.01	7.27	5.98	6.87	9.90	6.82	7.54	5.68
RWMaOP3	7	3	0.70	2.43	0.57	1.93	9.23	1.80	4.55	0.31
RWMaOP4	5	6	0.69	1.86	0.56	2.03	7.05	1.98	3.67	0.29
RWMaOP5	4	4	0.64	1.60	0.51	1.79	6.66	1.60	3.05	0.29

phases. The algorithm further integrates to IFM improve both the convergence and the spread of solutions. The efficacy of MaOGOA was validated by benchmarking it against four established algorithms—MaOSCA, MaOPSO, MOEA/DD, NSGA-III, KnEA, RvEA and GrEA across

test suite like WFG1-WFG9 with 5, 7 and 9 objectives. Results demonstrate MaOGOA robust capability to balance solution convergence and diversity in terms of IGD, HV and RT. Additionally, its application to the real-world RWMaOP1-RWMaOP5 problems showcased MaOGOA

competitive edge in practical scenarios. The standout feature of MaOGOA is its overall excellence, primarily due to the IFM mechanism that adeptly balances convergence, uniformity and diversity in solution distribution.

Future research directions exploring more potent techniques for many-objective optimization challenges and developing finer strategies for these issues. Investigating the application of MaOGOA preselection strategy in different contexts, such as multi-objective neural architecture search within deep neural networks for medical image segmentation, also presents a promising avenue for further study.

Supplementary Information The online version of this article (<https://doi.org/10.1007/s44196-024-00627-0>) contains supplementary material, which is available to authorized users.

Author Contributions Conceptualization: Kanak Kalita, Pradeep Jangir, and Sundaram B. Pandya; formal analysis: Kanak Kalita, Pradeep Jangir, and Sundaram B. Pandya; investigation: Kanak Kalita, Pradeep Jangir, and Sundaram B. Pandya; methodology: Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya, Robert Ćep, and Laith Abualigah; software: Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya, Robert Ćep, and Laith Abualigah; writing—Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya, Robert Ćep, and Laith Abualigah; writing—review and editing: Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya, Robert Ćep, and Laith Abualigah. All authors have read and agreed to the published version of the manuscript.

Funding This article was supported by the project Students Grant Competition SP2024/087, “Specific Research of Sustainable Manufacturing Technologies”, financed by the Ministry of Education, Youth and Sports of Czech Republic and Faculty of Mechanical Engineering VŠB-TUO.

Data Availability The data presented in this study are available through email upon request to the corresponding author.

Declarations

Conflict of interest The authors declare no conflict of interest.

Informed Consent Not applicable.

Institutional Review Board Statement Not applicable.

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

- Palakonda, V., Kang, J.-M.: Pre-DEMO: preference-inspired differential evolution for multi/many-objective optimization. *IEEE Trans. Syst. Man and Cybern. Syst.* **53**(12), 7618–7630 (2023). <https://doi.org/10.1109/TSMC.2023.3298690>
- Ishibuchi, H., Tsukamoto, N. and Nojima, Y., 2008, June. Evolutionary many-objective optimization: A short review. In: 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence) (pp. 2419–2426). IEEE.
- Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: a survey. *ACM Comput. Surv. (CSUR)* **48**(1), 1–35 (2015)
- Liu, S., Lin, Q., Li, J., Tan, K.C.: A survey on learnable evolutionary algorithms for scalable multiobjective optimization. *IEEE Trans. Evol. Comput.* **27**(6), 1941–1961 (2023). <https://doi.org/10.1109/TEVC.2023.3250350>
- Yuan, J., Liu, H.-L., Gu, F., Zhang, Q., He, Z.: Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions. *IEEE Trans. Evol. Comput.* **25**(1), 75–86 (2021). <https://doi.org/10.1109/TEVC.2020.2999100>
- Fonseca, C. M., Fleming, P. J., Zitzler, E., Deb, K., & Thiele, L. (2003). Evolutionary multicriterion optimization. In *Second International Conference, EMO 2003*. Springer.
- Wang, Y., Liu, Z., Wang, G.-G.: Improved differential evolution using two-stage mutation strategy for multimodal multi-objective optimization. *Swarm Evol. Comput.* **78**, 101232 (2023). <https://doi.org/10.1016/j.swevo.2023.101232>
- Torres, L.C.B., Castro, C.L., Rocha, H.P., Almeida, G.M., Braga, A.P.: Multi-objective neural network model selection with a graph-based large margin approach. *Inf. Sci.* **599**, 192–207 (2022). <https://doi.org/10.1016/j.ins.2022.03.019>
- Wang, Z., Qi, F., & Zou, L. (2022). ‘Continuous Evolution for Efficient Neural Architecture Search Based on Improved NSGA-III Algorithm.’ signal and information processing, networking and computers. In *Proceedings of the 8th International Conference on Signal and Information Processing, Networking and Computers (ICSINC)*. Springer Nature Singapore.
- Lin, A., Yu, P., Cheng, S., Xing, L.: One-to-one ensemble mechanism for decomposition-based multi-objective optimization. *Swarm Evol. Comput.* **68**, 101007 (2022). <https://doi.org/10.1016/j.swevo.2021.101007>
- Taghanaki, S.A., Kawahara, J., Miles, B., Hamarneh, G.: Pareto-optimal multi-objective dimensionality reduction deep auto-encoder for mammography classification. *Comput. Methods Programs Biomed.* **145**, 85–93 (2017). <https://doi.org/10.1016/j.cmpb.2017.04.012>
- Purshouse, R.C., Fleming, P.J.: On the evolutionary optimization of many conflicting objectives. *IEEE Trans. Evol. Comput.* **11**(6), 770–784 (2007). <https://doi.org/10.1109/TEVC.2007.910138>
- Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **18**(4), 602–622 (2013). <https://doi.org/10.1109/TEVC.2013.2281534>
- Celik, H., Karaboga, N.: Blind source separation with strength Pareto evolutionary algorithm 2 (SPEA2) using discrete wavelet transform. *Electronics* **12**(21), 4383 (2023). <https://doi.org/10.3390/electronics12214383>
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>
- He, Z., Yen, G.G., Zhang, J.: Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. *IEEE Trans. Evol.*

- Comput. **18**(2), 269–285 (2014). <https://doi.org/10.1109/TEVC.2013.2258025>
17. Hadka, D., Reed, P.: Borg: An auto-adaptive many-objective evolutionary computing framework. *Evol. Comput.* **21**(2), 231–259 (2013). https://doi.org/10.1162/EVCO_a_00075
 18. Yuan, Y., Xu, H., Wang, B., Yao, X.: A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **20**(1), 16–37 (2015). <https://doi.org/10.1109/TEVC.2015.2420112>
 19. Tian, Y., Cheng, R., Zhang, X., Su, Y., Jin, Y.: A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans. Evol. Comput.* **23**(2), 331–345 (2019). <https://doi.org/10.1109/TEVC.2018.2866854>
 20. Qiu, W., Zhu, J., Wu, G., Fan, M., Suganthan, P.N.: Evolutionary many-objective algorithm based on fractional dominance relation and improved objective space decomposition strategy. *Swarm Evol. Comput.* **60**, 100776 (2021). <https://doi.org/10.1016/j.swevo.2020.100776>
 21. Zhang, X., Tian, Y., Jin, Y.: A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **19**(6), 761–776 (2014). <https://doi.org/10.1109/TEVC.2014.2378512>
 22. Taherzohad-Javazm, F., Rankin, D., Coyle, D.: R2-HMEO: Hybrid multi-objective evolutionary algorithm based on the equilibrium optimizer and whale optimization algorithm IEEE Congress on Evolutionary Computation (CEC). IEEE Publ. (2022). <https://doi.org/10.1109/CEC55065.2022.9870371>
 23. Li, X., Song, Y., Gao, J., Zhang, B., Gui, L., Yuan, W., Li, Z., Han, S.: Multi-objective optimization method for reactor shielding design based on SMS-EMOA. *Ann. Nucl. Energy* **194**, 110097 (2023). <https://doi.org/10.1016/j.anucene.2023.110097>
 24. Hsieh, T.-J.: Performance indicator-based multi-objective reliability optimization for multi-type production systems with heterogeneous machines. *Reliab. Eng. Syst. Saf.* **230**, 108970 (2023). <https://doi.org/10.1016/j.res.2022.108970>
 25. Liu, S., Handing, W., Wen, Y., Wei, P.: Surrogate-assisted environmental selection for fast hypervolume-based many-objective optimization. *IEEE Trans. Evol. Comput.* **28**(1), 132–146 (2023). <https://doi.org/10.1109/TEVC.2023.3243632>
 26. Wang, B., Singh, H.K., Ray, T.: Adjusting normalization bounds to improve hypervolume based search for expensive multi-objective optimization. *Complex Intell. Syst.* **9**(2), 1193–1209 (2023). <https://doi.org/10.1007/s40747-021-00590-9>
 27. Emmerich, M., Beume, N., & Naujoks, B. (2005). An EMO algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, pp. 62–76
 28. Chen, H., Cheng, R., Wen, J., Li, H., Weng, J.: Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations. *Inf. Sci.* **509**, 457–469 (2020). <https://doi.org/10.1016/j.ins.2018.10.007>
 29. Liu, H., Gu, F., Zhang, Q.: Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Trans. Evol. Comput.* **18**(3), 450–455 (2013). <https://doi.org/10.1109/TEVC.2013.2281533>
 30. Zhang, Q., Li, H.: MOEA/D A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
 31. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **20**(5), 773–791 (2016). <https://doi.org/10.1109/TEVC.2016.2519378>
 32. Figueiredo, E.M.N., Ludermir, T.B., Bastos-Filho, C.J.A.: Many objective particle swarm optimization. *Inf. Sci.* **374**, 115–134 (2016). <https://doi.org/10.1016/j.ins.2016.09.026>
 33. Narayanan, R.C., Ganesh, N., Çep, R., Jangir, P., Chohan, J.S., Kalita, K.: A novel many-objective sine–cosine algorithm (MaOSCA) for engineering. *Appl. Math.* **11**(10), 2301 (2023)
 34. Li, K., Deb, K., Zhang, Q., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **19**(5), 694–716 (2014). <https://doi.org/10.1109/TEVC.2014.2373386>
 35. Yuan, Y., Xu, H., Wang, B., Zhang, B., Yao, X.: Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans. Evol. Comput.* **20**(2), 180–198 (2015). <https://doi.org/10.1109/TEVC.2015.2443001>
 36. Li, M., Yao, X.: What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evol. Comput.* **28**(2), 227–253 (2020). https://doi.org/10.1162/evco_a_00269
 37. Liu, Q., Jin, Y., Heiderich, M., Rodemann, T., & Yu, G. (2020). An adaptive reference vector guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems [Tech. rep.]. University of Surrey.
 38. Liu, S., Lin, Q., Wong, K.-C., Coello Coello, C.A.C., Li, J., Ming, Z., Zhang, J.: A self-guided reference vector strategy for many-objective optimization. *IEEE Trans. Cybern.* **52**(2), 1164–1178 (2020). <https://doi.org/10.1109/TCYB.2020.2971638>
 39. Cheng, X., Gong, W., Ming, F., Zhu, X.: Multimodal multi-objective optimization via determinantal pointprocess-assisted evolutionary algorithm. *Neural Comput. Appl.* **36**(3), 1381–1411 (2024). <https://doi.org/10.1007/s00521-023-09110-x>
 40. Wang, W., Dong, H., Wang, P., Wang, X., Shen, J.: A clustering-based surrogate-assisted evolutionary algorithm (CSMOEA) for expensive multi-objective optimization. *Soft. Comput.* **27**(15), 10665–10686 (2023). <https://doi.org/10.1007/s00500-023-08227-4>
 41. Wang, H., Jiao, L., Yao, X., Two.: Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **19**(4), 524–541 (2014)
 42. Chen, Y., Yuan, X., Cang, X.: A new gradient stochastic ranking-based multi-indicator algorithm for many-objective optimization. *Soft. Comput.* **23**(21), 10911–10929 (2019). <https://doi.org/10.1007/s00500-018-3642-7>
 43. Cai, X., Guo, W., Zhao, M., Cui, Z., Chen, J.: A knowledge graph-based many-objective model for explainable social recommendation. *IEEE Trans. Comput. Soc. Syst.* **10**(6), 3021–3030 (2023). <https://doi.org/10.1109/TCSS.2023.3283574>
 44. Ding, Z., Chen, L., Sun, D., Zhang, X.: A multi-stage knowledge-guided evolutionary algorithm for large-scale sparse multi-objective optimization problems. *Swarm Evol. Comput.* **73**, 101119 (2022). <https://doi.org/10.1016/j.swevo.2022.101119>
 45. Li, J., Wang, P., Dong, H., Shen, J.: A two-stage surrogate-assisted evolutionary algorithm (TS-SAEA) for expensive multi/many-objective optimization. *Swarm Evol. Comput.* **73**, 101107 (2022). <https://doi.org/10.1016/j.swevo.2022.101107>
 46. Liu, Q., Zou, J., Yang, S., Zheng, J.: A multiobjective evolutionary algorithm based on decision variable classification for many-objective optimization. *Swarm Evol. Comput.* **73**, 101108 (2022). <https://doi.org/10.1016/j.swevo.2022.101108>
 47. Li, D., Wang, L., Guo, W., Zhang, M., Hu, B., Wu, Q.: A particle swarm optimizer with dynamic balance of convergence and diversity for large-scale optimization. *Appl. Soft Comput.* **132**, 109852 (2023). <https://doi.org/10.1016/j.asoc.2022.109852>
 48. Yang, L., Hu, X., Li, K.: A vector angles-based many-objective particle swarm optimization algorithm using archive. *Appl. Soft Comput.* **106**, 107299 (2021). <https://doi.org/10.1016/j.asoc.2021.107299>

49. Liu, Y., Gong, D., Sun, J., Jin, Y.: A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. Cybern.* **47**(9), 2689–2702 (2017). <https://doi.org/10.1109/TCYB.2016.2638902>
50. Yang, S., Li, M., Liu, X., Zheng, J.: A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **17**(5), 721–736 (2013)
51. Li, M., Yang, S., Liu, X.: Bi-goal evolution for many-objective optimization problems. *Artif. Intell.* **228**, 45–65 (2015)
52. Li, M., Yang, S., Liu, X.: Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans. Evol. Comput.* **18**(3), 348–365 (2013)
53. Saremi, S., Mirjalili, S., Lewis, A.: Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **105**, 30–47 (2017). <https://doi.org/10.1016/j.advengsoft.2017.01.004>
54. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **10**(5), 477–506 (2006). <https://doi.org/10.1109/TEVC.2005.861417>
55. Tanabe, R., Ishibuchi, H.: An easy-to-use real-world multi-objective optimization problem suite. *Appl. Soft Comput.* **89**, 106078 (2020). <https://doi.org/10.1016/j.asoc.2020.106078>
56. Panagant, N., Kumar, S., Tejani, G.G., Pholdee, N., Bureerat, S.: Many objective meta-heuristic methods for solving constrained truss optimisation problems: a comparative analysis. *MethodsX* **10**, 102181 (2023). <https://doi.org/10.1016/j.mex.2023.102181>
57. Ahmad, N., Kamal, S., Raza, Z.A., Hussain, T.: Multi-objective optimization in the development of oil and water repellent cellulose fabric based on response surface methodology and the desirability function. *Mate. Res. Express* **4**(3), 035302 (2017). <https://doi.org/10.1088/2053-1591/aa5f6a>
58. Chen, Y.-S.: Performance enhancement of multiband antennas through a two-stage optimization technique. *Int. J. RF Microw. Comput. Aided Eng.* **27**(2), e21064 (2017). <https://doi.org/10.1002/mmce.21064>
59. Goel, T., Vaidyanathan, R., Haftka, R.T., Shyy, W., Queipo, N.V., Tucker, K.: Response surface approximation of Pareto optimal front in multi-objective optimization. *Comput. Methods Appl. Mech. Eng.* **196**(4–6), 879–893 (2007). <https://doi.org/10.1016/j.cma.2006.07.010>
60. Li, M., Yao, X.: Quality evaluation of solution sets in multiobjective optimisation: a survey. *ACM Comput. Surv. (CSUR)* **52**(2), 1–38 (2019). <https://doi.org/10.1145/3300148>
61. Li, M., Chen, T., Yao, X.: How to evaluate solutions in Pareto-based search-based software engineering: a critical review and methodological guidance. *IEEE Trans. Softw. Eng.* **48**(5), 1771–1799 (2020). <https://doi.org/10.1109/TSE.2020.3036108>

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

Authors and Affiliations

Kanak Kalita^{1,2} · Pradeep Jangir^{3,4,5} · Robert Čep⁶ · Sundaram B. Pandya⁷ · Laith Abualigah^{8,9,10,11}

✉ Kanak Kalita
drkanakkalita@veltech.edu.in; kanakkalita02@gmail.com

Pradeep Jangir
pkjmttech@gmail.com

Robert Čep
robert.cep@vsb.cz

Sundaram B. Pandya
sundarampandya@gmail.com

Laith Abualigah
aligah.2020@gmail.com

¹ Department of Mechanical Engineering, Vel Tech Rangarajan Dr, Sagunthala R&D Institute of Science and Technology, Avadi 600062, India

² University Centre for Research & Development, Chandigarh University, Mohali 140413, India

³ Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India

⁴ Jadara University Research Center, Jadara University, Irbid 21110, Jordan

⁵ Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

⁶ Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 70800 Ostrava, Czech Republic

⁷ Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392 001, India

⁸ Computer Science Department, Al Al-Bayt University, Mafraq 25113, Jordan

⁹ MEU Research Unit, Middle East University, Amman 11831, Jordan

¹⁰ Centre for Research Impact & Outcome, Chitkara University, Rajpura 140401, India

¹¹ Artificial Intelligence and Sensing Technologies (AIST) Research Center, University of Tabuk, 71491 Tabuk, Saudi Arabia