International Journal on Interactive Design and Manufacturing (IJIDeM)
https://doi.org/10.1007/s12008-024-02143-z

ORIGINAL ARTICLE

A novel many-objective symbiotic organism search algorithm for industrial engineering problems

Kanak Kalita[1,2] · Pradeep Jangir[3,4,5,6] · Ajay Kumar[7] · Sundaram B. Pandya[8] · Laith Abualigah[9]

Received: 22 April 2024 / Accepted: 5 October 2024
© The Author(s), under exclusive licence to Springer-Verlag France SAS, part of Springer Nature 2024

Abstract

The focus of multi-objective optimization is to derive a set of optimal solutions in scenarios with multiple and often conflicting objectives. However, the ability of multi-objective evolutionary algorithms in approaching the Pareto front and sustaining diversity within the population tends to diminish as the number of objectives grows. To tackle this challenge, this research introduces a novel Many-Objective Symbiotic Organism Search (MaOSOS) for many-objective optimization. In this method the concept of reference point, niche preservation and information feedback mechanism (IFM) are incorporated. Niche preservation aims to enhance selection pressure while preserving diversity by splitting the objective space. Reference point adaptation strategy effectively accommodates various Pareto front models to improve convergence. The IFM mechanism augments the likelihood of selecting parent solutions that exhibit both strong convergence and diversity. The efficacy of MaOSOS was validated through WFG1-WFG9 benchmark problems (with varied number of objectives ranging from 5 to 7) and five real-world engineering problems. Several metrics like GD, IGD, SP, SD, HV and RT metrics were used to assess the MaOSOS's efficacy. The extensive experiments establish the superior performance of MaOSOS in managing many-objective optimization tasks compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA.

Keywords Symbiotic organism search · Many-objective · Multi-objective · Convergence · Real world

1 Introduction

In the field of optimization, numerous real-world challenges encompass multiple objectives that often conflict with each other. Such challenges are commonly identified as multiobjective optimization problems (MOPs). When MOPs encompass four or more conflicting objectives, they are classified as many-objective optimization problems (MaOPs). The field of many-objective optimization has garnered significant attention, owing to its extensive practical applications,

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

Pradeep Jangir
pkjmtech@gmail.com

Ajay Kumar
dr.ajaykumarphd@gmail.com

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 600 062, India

2 Jadara University Research Center, Jadara University, Irbid, Jordan

3 Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602 105, India

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

5 Department of CSE, Graphic Era Hill University, Graphic Era Deemed To Be University, Dehradun 248002, Uttarakhand, India

6 University Centre for Research and Development, Chandigarh University, Gharuan 140413, Mohali, India

7 Department of Mechanical Engineering, JECRC University, Jaipur 303905, India

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

9 Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan

Published online: 01 November 2024

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

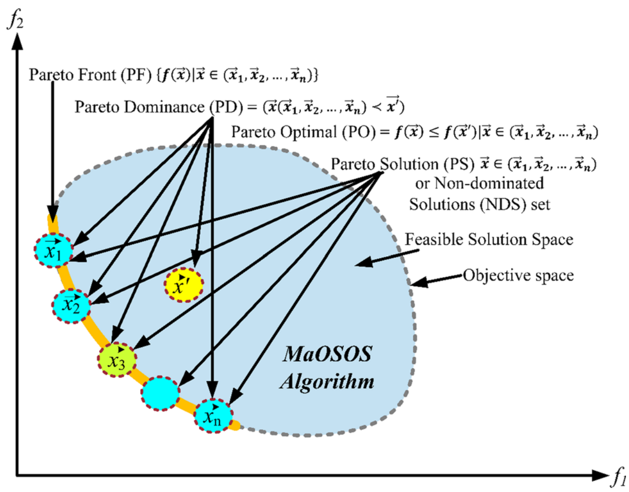


Fig. 1 Many-Objective all definitions in search space of MaO-Problem

as highlighted in sources [1, 2]. An unconstrained MOP, or MaOP, can be formally expressed as per Eq. (1) [3],

$$\begin{aligned} min\ F(x) &= (f_1(x),\ f_2(x),\ \ldots,\ f_m(x)), \\ subject\ to\ x &\in \ \Omega\ , \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)$ denotes a decision variable vector spanning multi-dimension within the decision space Ω and F(x) encapsulates m different objective functions. The inherent conflict among these objectives means there is no singular, optimal solution. Instead, a spectrum of solutions emerges, each signifying a compromise among varying objectives. In practical scenarios, several optimization challenges involve optimizing multiple, often conflicting, objectives concurrently. These challenges fall under the umbrella of Multi-Objective Optimization problems (MOPs). The nature of these objective functions being at odds implies that enhancing one objective might degrade others. Therefore, the solutions do not converge on a single optimum for the objective function, but rather form a collection of trade-off solutions, known as the Pareto-optimal Set (PS). The depiction of this PS within an objective space is referred to as a Pareto-optimal Front (PF) shown in Fig. 1.

Multiobjective evolutionary algorithms (MOEAs) stand out for their direct and derivative-free approach, proving to be a highly effective solution for tackling MOPs [4]. Various MOEAs, such as NSGA-II [5] and SPEA2 [6], have been developed specifically for this purpose. Within the spectrum of MOEAs, methods based on Pareto dominance are particularly renowned in the field. NSGA-II and SPEA2, both Pareto-based strategies, have shown commendable results in optimizing MOPs with two or three objectives. Nevertheless, their effectiveness significantly decreases when applied to MaOPs, mainly because the increase in objectives leads to most population solutions being nondominated [7]. To improve MOEAs' capability to effectively manage MaOPs, several innovative approaches have emerged recently [8].

The first group of strategies aims to redefine dominance relations to better guide the selection process toward the PF. Techniques like L-optimality [9] and preference order ranking [10] fall under this category. While these revised dominance relationships enhance MOEAs' convergence for MaOPs, they might reduce the diversity of the solutions. Conversely, grid-based algorithms like ϵ-MOEA [11] and GrEA [12] loosen these dominance relations. These algorithms leverage grid positions to differentiate individuals, ensuring a well-distributed population. However, the challenge lies in adaptively setting the grid size for various optimization problems.

The second group focuses on enhancing the algorithm's mechanism for maintaining diversity. For example, Li et al. [13] introduced a shift-based density estimation (SDE) strategy. This strategy balances the diversity and convergence aspects of individuals, effectively eliminating solutions that contribute to poor convergence. Similarly, the knee point-driven evolutionary algorithm (KnEA) [14] prioritizes knee points among nondominated solutions.

The third group encompasses a range of decomposition-based techniques. Within this group, two distinct methodologies are prevalent. The first approach consolidates objectives into several scalar functions, with each producing a singular scalar value. In the case of MOEA/D [15], a MOP is divided into multiple scalar optimization sub-challenges, which are then solved concurrently. Alternatively, the second approach segments a MaOP into a collection of smaller sub-MaOPs. Techniques like Many-Objective Gradient Based Optimizer (MaOGBO) [16], Many-Objective Jaya algorithm (MaOJAYA) [17], Many-Objective Teaching Learning Based Optimizer (MaOTLBO) [18], Many-Objective Sine Cosine Algorithm (MaOSCA) [19], Non-Dominated Sorting Genetic Algorithm-III (NSGA-III) [20] and MOEA/DD [21] partition the objective space into several subspaces using reference vectors, aiding in the management of nondominated solutions. A critical aspect for both methods is the configuration of weight vectors, a factor that greatly impacts the diversity within a population [22]. To address MaOPs with irregular PFs effectively, a range of adaptive, decomposition-based methods have been developed, including ESOEA [23] and A-NSGA-III [24].

The fourth category involves indicator-based methods. These approaches typically rely on a singular indicator to guide the evolution of the population, focusing on both convergence and diversity. Prominent indicators like hypervolume (HV) [25] and Inverted Generational Distance (IGD) [26] are often used. However, HV-based methods suffer from exponentially increasing computational complexity

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

with more objectives. For IGD-based strategies, the key challenge is acquiring a set of uniformly distributed reference points for IGD calculation.

Finally, the fifth category comprises objective reduction techniques. The fundamental concept here is to lessen the number of objectives while preserving as much information as possible. Pal et al. [27] introduced an approach using clustering-based objective reduction within differential evolution. Bandyopadhyay and Mukherjee [28] suggested periodically rearranging objectives based on their conflict levels, selecting the most α-conflicting ones for further exploration. Moreover, He et al. [29] devised a strategy aimed at swiftly directing the entire population towards a limited number of target points near the true PF, thereby minimizing the objective space.

While these advancements have significantly enhanced MOEAs' capability to handle MaOPs, each method has its limitations and it is universally acknowledged that no single optimization technique can be optimal for every problem [30]. A major challenge in addressing MaOPs is the high proportion of nondominated solutions, which complicates differentiating among them. Numerous strategies have been developed to alter the dominance relationship, thereby intensifying the environmental selection pressure [9, 12]. Additionally, a direct approach to amplify selection pressure is to prioritize individuals based on their convergence. However, relying solely on convergence indicators often results in capturing only a fraction of the PF [31]. Therefore, it is essential to implement mechanisms that improve the population's distribution. One intuitive method is to penalize the neighbors of a selected individual. Nevertheless, the execution of such penalization (identifying the neighboring individual and determining the extent of the penalty) is complex.

In this study, a novel information feedback mechanism (IFM) combining symbiotic organism search (SOS) [32], reference point, niche-based density estimation and perpendicular distance selection strategies is introduced. This approach focuses on balancing diversity and convergence effectively. The niche-based density estimation is used to pinpoint individuals lacking in diversity, while the angle-based selection method eliminates those with subpar convergence. This paper's key contributions are—.

1. The SOS algorithm was chosen as it was known to perform well in producing various high-quality solutions in single-objective problems. Thus, MaOSOS algorithm can exploit the search space efficiently through the use of the SOS operator's global search ability.
2. To overcome the limitations that caused the loss of important information, this paper proposes the IFM approach. It incorporates the historical information of individuals for the next generation through a weighted sum model. This ensures better convergence properties.
3. A reference point-based selection approach is proposed to control the selection process and ensure that the selected solutions are not only near the Pareto front but also spread out across the entire Pareto optimal surface. Using perpendicular distances to link each solution to the closest reference point, the strategy aims at finding the areas of the objective space that are well explored. Applying non-dominated sorting helps the algorithm to work only with solutions that are closer to the Pareto-optimal front and thus helps in convergence.
4. A niche preservation approach for boundary solutions is then introduced to diversify the population while eliminating those located dense in certain regions of the objective space, which in turn, accelerates the convergence of the algorithm. Furthermore, to enhance selection, a density estimation strategy is proposed to retain population density that is uniform and complete.
5. A large number of systematic simulation tests were performed with the use of WFG test suites to assess the performance of MaOSOS. In comparison with MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the performance of our algorithm was better in terms of the IGD, HV and SPREAD measures. In addition, MaOSOS showed high efficiency and potential when applied to five real-world MaOPs (RWMaOP1 – RWMaOP5).

The structure of this paper is organized in the following manner—Sect. 2 provides the overview of SOS algorithm. The detailed methodology of the proposed MaOSOS algorithm is then explored in Sect. 3. Section 4 comprises a comprehensive series of experiments to assess the algorithm's proficiency in tackling MaOPs, along with an analysis of the results. The paper concludes with Sect. 5, summarizing the findings.

2 Symbiotic organism search

In 2014, the Symbiotic Organisms Search (SOS) [32] algorithm was introduced by Cheng and Prayogo. This algorithm was specifically designed for optimizing a variety of mathematical benchmark functions and addressing complex engineering design optimization challenges. The SOS algorithm is inspired by the dynamics of symbiotic relationships found in nature, specifically focusing on mutualism, commensalism and parasitism among ecosystem organisms. The algorithm initiates by generating a random population of organisms within the search space, representing an ecosystem. This population is denoted as the X matrix, as outlined in Eq. (2). In this matrix, each organism possesses a distinct fitness value, signifying a potential solution to the optimization problem at hand.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

The initial set of organisms undergoes modifications through three key phases – mutualism, commensalism and parasitism. This process generates new organisms, which are then evaluated and refined based on their fitness. Following these phases, the algorithm selects the most promising organism to lead the next generation. This selection and updating of the superior organism continue iteratively until specific termination conditions are satisfied. These symbiotic interactions are crucial for the generation of new organisms and are detailed below.

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}; \quad \begin{matrix} n & = & Eco\_size \\ m & = & D \end{matrix} \tag{2}$$

Mutualism, exemplified by the interaction between flowering plants and honey bees. In this natural interaction, bees collect nectar from flowers, aiding in the pollination process, which benefits both parties. This mutualistic interaction is simulated in the SOS algorithm's mutualism phase. An X_i organism is chosen from the ecosystem's i^{th} row (referenced in Eq. (2)) to model this relationship, leading to the creation of new organisms X_i^{new}, X_j^{new} and MV based on Eqs. (3), (4) and (5).

$$X_i^{new} = X_i + rand(0,\ 1) \times \ (X_{best} - MV \times \ bf_1) \tag{3}$$

$$X_j^{new} = X_j + rand(0,\ 1) \times \ (X_{best} - MV \times \ bf_2) \tag{4}$$

$$MV = \frac{bf_1 + bf_2}{2} \tag{5}$$

The updating process of these organisms is detailed in Eqs. (6) and (7).

$$X_i = \begin{cases} X_i^{new} & \text{if } f\left(X_i^{new}\right) < f(X_i) \\ X_i & \text{otherwise} \end{cases} \tag{6}$$

$$X_j = \begin{cases} X_j^{new} & \text{if } f\left(X_i^{new}\right) < f(X_i) \\ X_j & \text{otherwise} \end{cases} \tag{7}$$

Commensalism is explored. A classic example of this is the relationship between remoras and larger marine animals like sharks. Remoras attach to these larger animals and feed on their leftovers, benefiting without harming the host. In the SOS algorithm, a similar process occurs where one organism benefits from the interaction without impacting the other. An organism X_j is selected randomly for this interaction X_i, leading to the X_i^{new} creation of a new organism as outlined in Eq. (8) and updated according to Eq. (6).

$$X_i^{new} = X_i + rand(-1,\ 1) \times \ (X_{best} - X_j) \tag{8}$$

Parasitism, a relationship where one organism benefits at the expense of the other, such as fleas or ticks on dogs. In the SOS algorithm, X_i this is represented by one organism acting as a parasite. A host organism X_j is selected randomly from the current ecosystem. The parasite organism generates a *Parasite Vector* (PV) by duplicating and modifying itself. The fitness values of both the PV and X_j the original organisms are then evaluated and the PV is either updated or discarded based on these evaluations, as detailed in Eq. (9).

$$X_j = \begin{cases} X_i^{PV} & \text{if } f\left(X_i^{PV}\right) < f\left(X_j\right) \\ X_j & \text{otherwise} \end{cases} \tag{9}$$

3 Proposed many-objective symbiotic organism search (MaOSOS)

The MaOSOS algorithm generates a set of reference points using the Das and Dennis's technique. The number of reference points H is determined by the formula, $H = \binom{M+p-1}{p}$, where M is the number of objectives, p is the number of partitions. The formula calculates the number of possible reference points based on the combination of objectives and partitions. This ensures that the set of reference points covers the objective space evenly. The current generation is t, x_i^t and x_i^{t+1} are the i^{th} individual at t and

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

$(t+1)$ generation. u_i^{t+1} is the i^{th} individual at the $(t+1)$ generation generated through the SOS 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}. Then, x_i^{t+1} is calculated according to u_i^{t+1} generated through the SOS algorithm and IFM as per Eq. (10)

$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t;\ \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t},$$
$$\partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t},\ \partial_1 + \partial_2 = 1 \tag{10}$$

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. ∂_1 and ∂_2 within the IFM are crucial in determining the contribution of current and historical information of the solutions during optimization. These coefficients are computed dynamically based on the fitness values of the individuals at each generation, as defined by Eq. (10). ∂_1 is calculated as the ratio of the fitness of the selected individual to the sum of fitness values of both current and selected individuals. This ensures that individuals with better fitness contribute more to the next generation while maintaining diversity through ∂_2, which considers the contribution of individuals with potentially diverse characteristics. The dynamic adaptation of these coefficients during the optimization process ensures that the mechanism remains robust across different types of problem landscapes.

Offspring population 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 $(F_1, F_2, \ldots, F_l \ldots, F_w)$. Beginning from F_1, all individuals in level 1 to l are added to S_t and remaining members of R_t are rejected. If $|S_t| = N$ then 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, niche-preserving operator is used. First, each population member of P_{t+1} and F_l is normalized (presented in Algorithm 2) by 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 by 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, niching strategy (described in Algorithm 5) improves the diversity of MaOSOS algorithm to choose those F_l members that are associated with the least represented reference points niche count ρ_i in P_{t+1}. Then, check is made to see if termination condition is met. If the termination condition is not satisfied, $t = t + 1$ than process is repeated and if it is satisfied, P_{t+1} is generated, it is then applied to generate a new population Q_{t+1} by SOS algorithm. The computational complexity of M-Objectives is $O\left(N^2 \log^{M-2} N\right)$ or $O\left(N^2 M\right)$, whichever is large. MaOSOS strategically choses reference points to maintain harmony between convergence and diversity. The adaptation of reference points in MaOSOS enables it to efficiently deal with difficult shapes of Pareto fronts like convex and/or discontinuous. To examine the objective space, it is divided into varied parts of the Pareto-optimal front. Using the perpendicular distance mechanism to find the shortest distance between solutions and reference points, even distribution of solutions throughout the Pareto-optimal surface is achieved. This increases variation by steering the algorithm toward neglected sections of the front. By integrating niche safeguarding and reference point modifications congestion is reduced in certain regions, often faced in many-objective optimization challenges.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

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

Output: Q_{t+1} = **SOS(** 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)**

SOS 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 is the set of x_i^{t+1})

3: $R_t = P_t \, U \, Q_t$

4: Different non-domination levels ($F_1, F_2, \ldots, F_l$) = Non-dominated-sort (R_t)

5: **repeat**

6: $S_t = S_t \, U \, F_i$ and $i = i+1$

7: **until** $| S_t | \geq N$

8: Last front to be included: $F_l = U_{i=1}^{l} F_i$

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 (f^n, 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:

$[\boldsymbol{\pi(s)}, \boldsymbol{d(s)}] = \boldsymbol{Associate}\ (\boldsymbol{S_t}, \boldsymbol{Z^r})$**;** Brief Explanation in **Algorithm-3**

% $\pi(s)$: $closest\ reference\ point, d$: $distance\ between\ s\ and\ \pi(s)$

16: Compute niche count of reference point $j \in Z^r$:

$\boldsymbol{\rho_j = \sum_{s \in s_t/F_l}((\pi(s) = j)\ , 1 : 0)}$**;**

17: Choose K members one at a time F_l to construct

$\boldsymbol{P_{t+1}} : \boldsymbol{Niching(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})}$ **; Represent in Algorithm-4**

18: **end if**

The flow chart of MaOSOS algorithm can be shown in Fig. 2.

4 Results and discussions

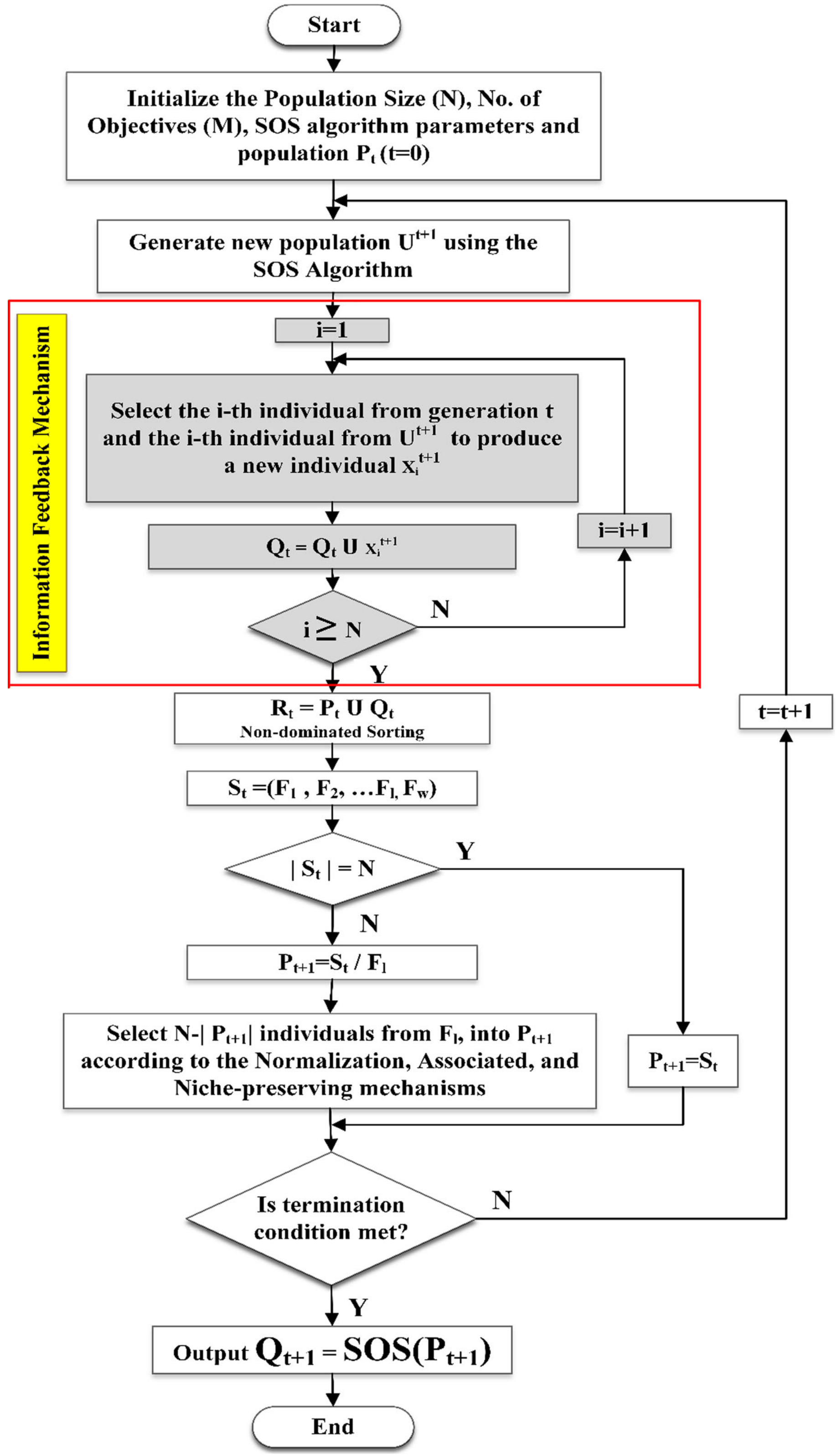
4.1 Problem description

In order to verify the effectiveness of the MaOSOS, the WFG1- WFG9 [33] benchmark (Appendix A) and five real world engineering design (Appendix B) namely Car cab design (RWMaOP1) [34], 10-bar truss structure (RWMaOP2) [35], Water and oil repellent fabric development (RWMaOP3) [36], Ultra-wideband antenna design (RWMaOP4) [37] and Liquid-rocket single element injector design (RWMaOP5) [38] problems are used in this paper. Choosing WFG1-WFG9 benchmark issues was based on their flair for creating numerous types of Pareto front shapes like concave and convex patterns representative of real-world MaOPs. The literature utilizes these benchmarks to measure the merit of many-objective algorithms since they incorporate critical challenges of convergence and diversity that indicate an algorithm's success in handling complex objective spaces. To evaluate the MaOSOS algorithm effectively in real-world scenarios five engineering issues were chosen that reflect actual optimization hurdles. In engineering design tasks like car cab and ultra-wideband antenna development optimizing various conflicting goals such as reducing weight and improving strength or performance is essential. These issues create an effective setting to measure the algorithm's efficiency and strength in addressing conflicting targets.

The experiments are conducted on a MATLAB R2020a environment on an Intel Core (TM) i7-9700 CPU. Each algorithm performs 30 times, the size of population N is set to 210, 276 and 240 for all the involved algorithms on $M = 5$, 6and7 objectives problems. The $MaxFEs$ is set to 1×10^5 for

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Fig. 2 Flowchart of MaOSOS algorithm



Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 1 Properties of the quality indicators

Quality indicator [40]	Convergence	Diversity	Uniformity	Cardinality	Computational Burden
GD	✓				
SD		✓			
SP			✓		
RT					✓
IGD	✓	✓	✓		
HV	✓	✓	✓	✓	

all the test instances. This paper adopts Generational distance (GD), Spread (SD), Spacing (SP), Run Time (RT), Inverse Generational distance (IGD) and Hypervolume (HV) quality indicator [39], shown in Table 1; Fig. 3.

4.2 Results on WFG problems

From Table 2, it is observed that MaOSOS outperforms 16 out of 27 best results, whereas MaOGBO, MaOJAYA, MaOTLBO and MaOSCA achieves 5, 1, 0 and 5 best results in terms of the GD values on WFG benchmark for 5-, 6- and 7-objectives. WFG1 problem with M = 5, MaOSOS achieves a GD of 0.094 ± 0.013, markedly better than its competitors. This trend is consistent across other problem sizes for WFG1, with MaOSOS maintaining a lower mean GD, indicative of better performance. Further examination reveals that MaOSOS outperforms other algorithms in specific scenarios, such as in WFG2 for M = 7, where it scores a GD of 0.041 ± 0.004. This is significantly lower than MaOGBO, MaOJAYA, MaOTLBO and MaOSCA. Throughout the other instances MaOSOS is seen to be consistently better than or at par with the rest. This leads to a reasonable assertion that MaOSOS is a competitive algorithm, thereby establishing its robustness in many-objective optimization tasks shown in Fig. 4.

Table 3 presents the Inverted Generational Distance (IGD) results for various algorithms, including MaOSOS, across a range of WFG problems. These results, detailed for different problem sizes (M), provide insights into each algorithm efficacy, with a lower mean IGD indicating better performance. In an overall assessment, MaOSOS achieves varying results across the WFG problems. Table 3 shows that MaOSOS does not consistently achieve the best IGD results, with its performance fluctuating depending on the problem and its dimension. However, it is important to note the specific problems where it exhibits competitive performance. For problem, in WFG2 with M = 5, MaOSOS records an IGD of 0.772 ± 0.006, which, while not the lowest but is competitive when compared to other algorithms like MaOGBO and MaOSCA. This pattern of near-best performance is observable in other problems, such as in WFG3 for M = 7, where MaOSOS achieves an IGD of 1.831 ± 0.1241. In Table 3, IGD value compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS is better in 24, 25, 21 and 27 out of 27 cases on WFG benchmark for 5-, 6- and 7-objectives. These proportions indicate that while MaOSOS does not universally lead in performance, it holds significant competitive strength in a notable subset of the WFG problems. This showcases its capability to handle a diverse range of many-objective optimization scenarios with varying degrees of effectiveness shown in Fig. 4.

Table 4 displays the Spacing (SP) SP values of MaOGBO, MaOJAYA, MaOTLBO and MaOSCA. This pattern of excellence is consistently observed in other problems, such as WFG2 with M = 6, where MaOSOS records an SP of 0.444 ± 0.034, again outperforming its competitors. In Table 4, SP value compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS worse in 3, 2, 0 and 2 out of 27 cases. These proportions highlight the considerable strength of MaOSOS in maintaining solution uniformity across a diverse range of many-objective optimization problems. Its ability to consistently achieve lower SP values in most of the test problems highlights its effectiveness in ensuring a well-distributed set of solutions shown in Fig. 4.

Table 5 elucidates the Spread (SD) results for MaOSOS and its counterparts across various WFG problems, with the metric measuring the extent of spread among the obtained solutions. A lower SD value indicates a better spread, signifying a more comprehensive exploration of the solution space. Within this framework, MaOSOS demonstrates its proficiency by achieving the best SD results in a significant number of test problems compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA. As can be seen from Table 5, MaOSOS achieves the best performance in terms of SD values, having obtained 20 best results, followed by MaOGBO, MaOJAYA, MaOTLBO and MaOSCA that have obtained 2, 1, 2 and 2 best results, respectively. The predominance of MaOSOS in securing the ability is crucial for ensuring that the solutions are not only optimal but also diverse, covering a wide range of possible outcomes shown in Fig. 4.

In Table 6 on the HV values, when respectively compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS is better in 27, 24, 24 and 17 out of 27

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

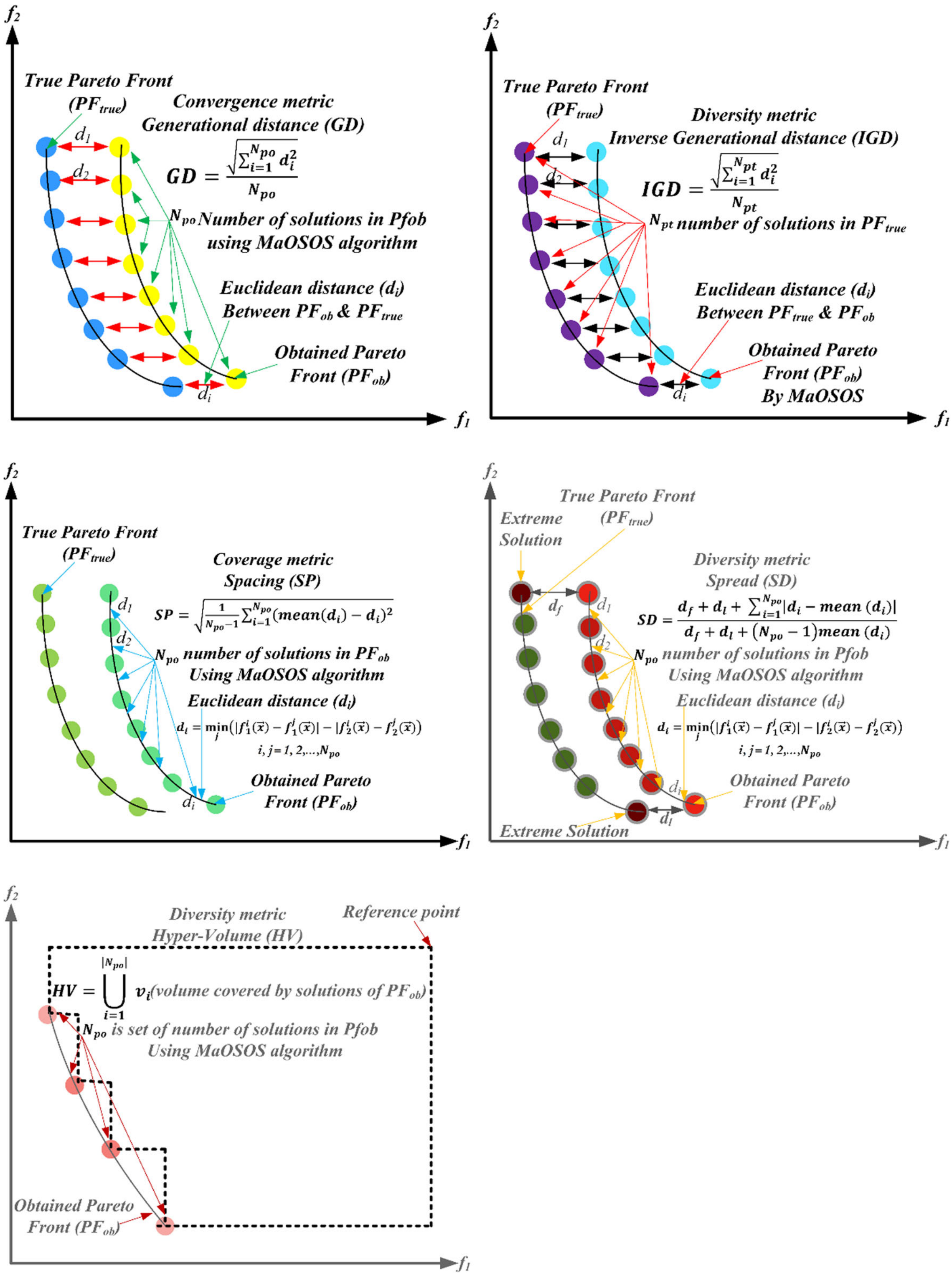


Fig. 3 Mathematical and Schematic view of the GD, IGD, SP, SD and HV metrics

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 2 Comparison of GD Metric on WFG problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
WFG1	5	14	**0.094 ± 0.013 =**	0.13 ± 0.015 =	0.129 ± 0.025 =	0.222 ± 0.145 =	0.118 ± 0.021
	6	15	**0.134 ± 0.015 =**	0.159 ± 0.025 =	0.137 ± 0.025 =	0.162 ± 0.007 =	0.156 ± 0.034
	7	16	**0.159 ± 0.006 =**	0.176 ± 0.032 =	0.176 ± 0.009 =	0.205 ± 0.012 =	0.17 ± 0.011
WFG2	5	14	0.024 ± 0.003 =	**0.02 ± 0.003 =**	0.027 ± 0.003 =	0.095 ± 0.006 =	0.027 ± 0.003
	6	15	0.041 ± 0.005 =	**0.025 ± 0.003 =**	0.052 ± 0.004 =	0.157 ± 0.033 =	0.04 ± 0.01
	7	16	**0.041 ± 0.004 =**	0.044 ± 0.015 =	0.06 ± 0.032 =	0.226 ± 0.009 =	0.044 ± 0.005
WFG3	5	14	0.279 ± 0.007 =	**0.165 ± 0.033 =**	0.267 ± 0.02 =	0.278 ± 0.006 =	0.319 ± 0.011
	6	15	0.377 ± 0.011 =	**0.216 ± 0.017 =**	0.304 ± 0.048 =	0.381 ± 0.006 =	0.321 ± 0.093
	7	16	0.481 ± 0.005 =	0.26 ± 0.018 =	**0.117 ± 0.027 =**	0.489 ± 0.006 =	0.242 ± 0.039
WFG4	5	14	**0.031 ± 0.002 =**	0.043 ± 0.001 =	0.036 ± 0.001 =	0.04 ± 0.003 =	0.033 ± 0.001
	6	15	**0.054 ± 0.004 =**	0.072 ± 0.005 =	0.064 ± 0.003 =	0.072 ± 0.004 =	0.055 ± 0.001
	7	16	**0.057 ± 0.004 =**	0.108 ± 0.006 =	0.075 ± 0.004 =	0.115 ± 0.002 =	0.071 ± 0.014
WFG5	5	14	**0.034 ± 0.001 =**	0.045 ± 0.001 =	0.042 ± 0.002 =	0.042 ± 0.002 =	0.035 ± 0.001
	6	15	**0.052 ± 0.003 =**	0.081 ± 0.002 =	0.078 ± 0.002 =	0.079 ± 0.002 =	0.059 ± 0.001
	7	16	**0.056 ± 0.001 =**	0.114 ± 0 =	0.095 ± 0.008 =	0.123 ± 0.006 =	0.073 ± 0.002
WFG6	5	14	**0.041 ± 0.002 =**	0.051 ± 0.002 =	0.047 ± 0.003 =	0.048 ± 0.008 =	0.045 ± 0.006
	6	15	**0.057 ± 0.003 =**	0.085 ± 0.006 =	0.075 ± 0.003 =	0.075 ± 0.008 =	0.061 ± 0.002
	7	16	**0.063 ± 0.003 =**	0.126 ± 0.006 =	0.081 ± 0.011 =	0.122 ± 0.003 =	0.071 ± 0.003
WFG7	5	14	0.041 ± 0.003 =	0.048 ± 0.005 =	0.042 ± 0.005 =	0.054 ± 0.002 =	**0.039 ± 0**
	6	15	0.064 ± 0.005 =	0.081 ± 0.004 =	0.075 ± 0.005 =	0.085 ± 0.001 =	**0.064 ± 0.008**
	7	16	**0.067 ± 0.006 =**	0.12 ± 0.004 =	0.082 ± 0.006 =	0.136 ± 0.009 =	0.075 ± 0.003
WFG8	5	14	**0.065 ± 0.003 =**	0.071 ± 0.002 =	0.069 ± 0.002 =	0.071 ± 0.002 =	0.067 ± 0.002
	6	15	0.103 ± 0.005 =	0.123 ± 0.002 =	0.109 ± 0.006 =	0.109 ± 0.007 =	**0.103 ± 0.009**
	7	16	0.128 ± 0.01 =	0.18 ± 0.007 =	0.126 ± 0.022 =	0.167 ± 0.004 =	**0.126 ± 0.019**
WFG9	5	14	0.051 ± 0.003 =	**0.05 ± 0.001 =**	0.052 ± 0.007 =	0.058 ± 0.001 =	0.051 ± 0.004
	6	15	0.083 ± 0.002 =	0.097 ± 0.001 =	0.092 ± 0.005 =	0.113 ± 0.015 =	**0.076 ± 0.005**
	7	16	**0.107 ± 0.014 =**	0.14 ± 0.002 =	0.127 ± 0.009 =	0.177 ± 0.002 =	0.109 ± 0.005

Bold indicates the best solution among the tested algorithms

cases and is only worse in 0%, 11.11%, 11.11% and 37.03% cases. Therefore, MaOSOS has a better balance between convergence and diversity for solving WFG benchmark for 5-, 6- and 7- objectives. This demonstrates its effectiveness in generating solutions that cover a larger area of the Pareto front shown in Fig. 4.

Table 7 presents the runtime (RT) results for MaOSOS compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA across various WFG problems. In Table 7, RT value compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS is better in 20, 27, 25 and 27 out of 27 cases. MaOSOS exhibits more efficient runtime results in 26/27 test problems. These proportions highlight the significant strength of MaOSOS in terms of computational efficiency. It consistently requires less time to find solutions compared to its counterparts, which is a crucial factor in many-objective optimization, especially for problems that require quick decision-making or are computationally intensive.

In terms of computational complexity, the proposed MaOSOS algorithm has a complexity of $O(N^2 \log^{M-2} N)$ or $O\left(N^2 M\right)$, where N is the population size and M is the number of objectives. This complexity arises from the niche-preserving operator, reference point-based selection and the use of the Information Feedback Mechanism (IFM) to maintain both convergence and diversity. These mechanisms, while increasing the computational burden, ensure that MaOSOS excels in solving problems with a higher number of objectives by balancing exploration and exploitation more effectively. On the other hand, MaOGBO has a simpler complexity of $O\left(NM^2\right)$, which is due to its gradient-based optimization approach that emphasizes convergence but may compromise diversity when applied to problems with more objectives. MaOJAYA, with a complexity of

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

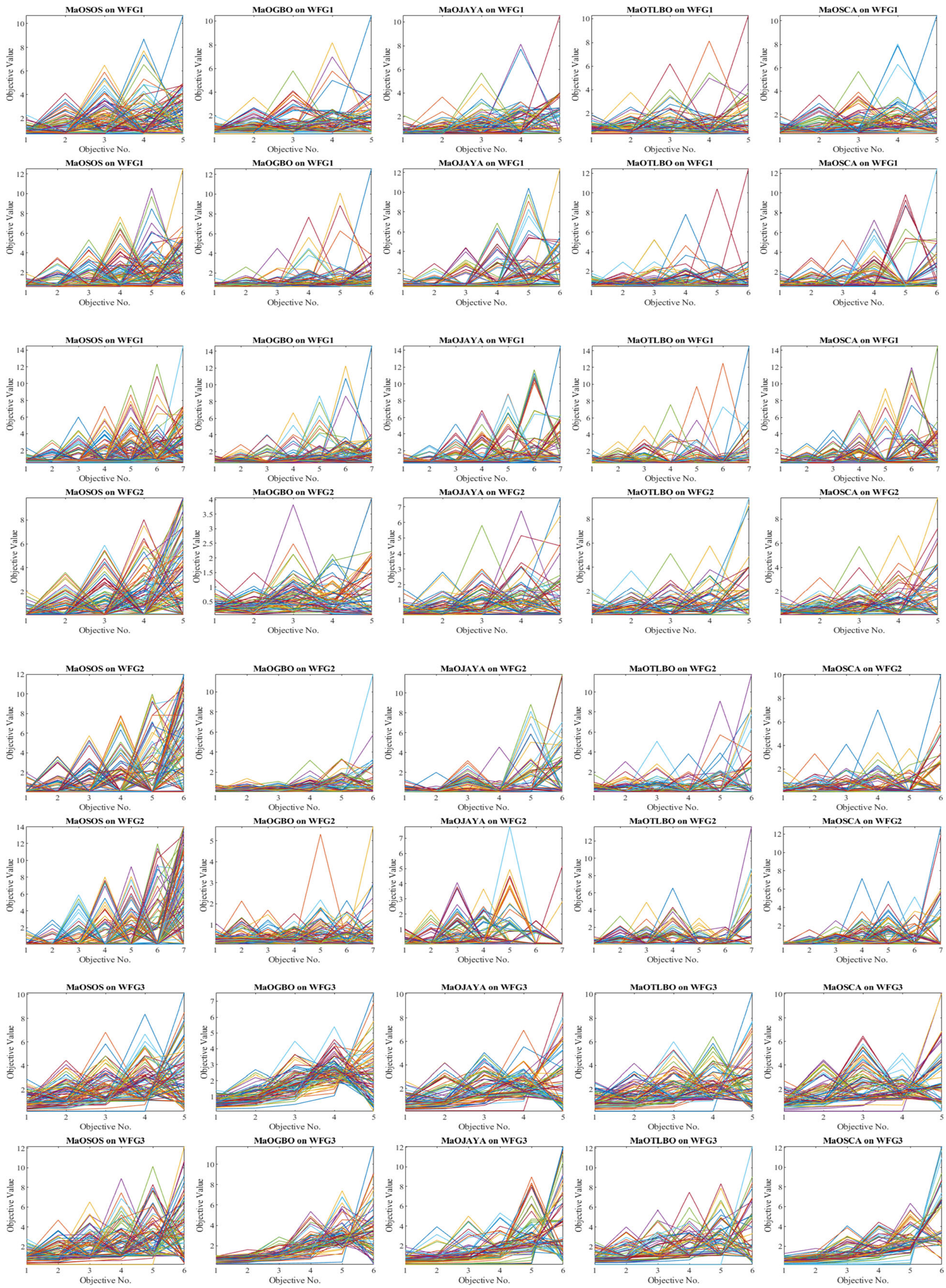


Fig. 4 Best Pareto optimal front obtained by different algorithms on WFG problems

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

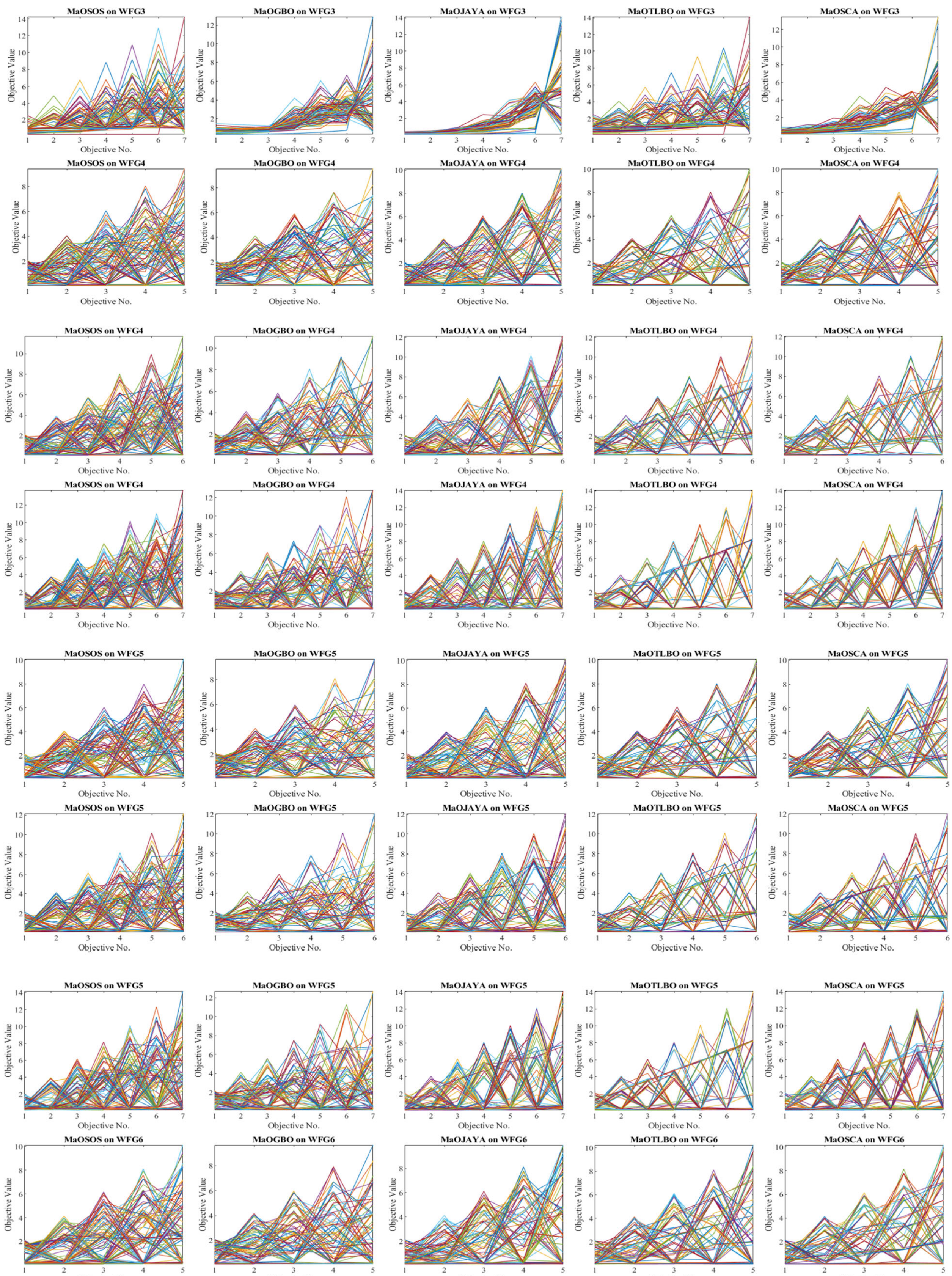


Fig. 4 continued

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

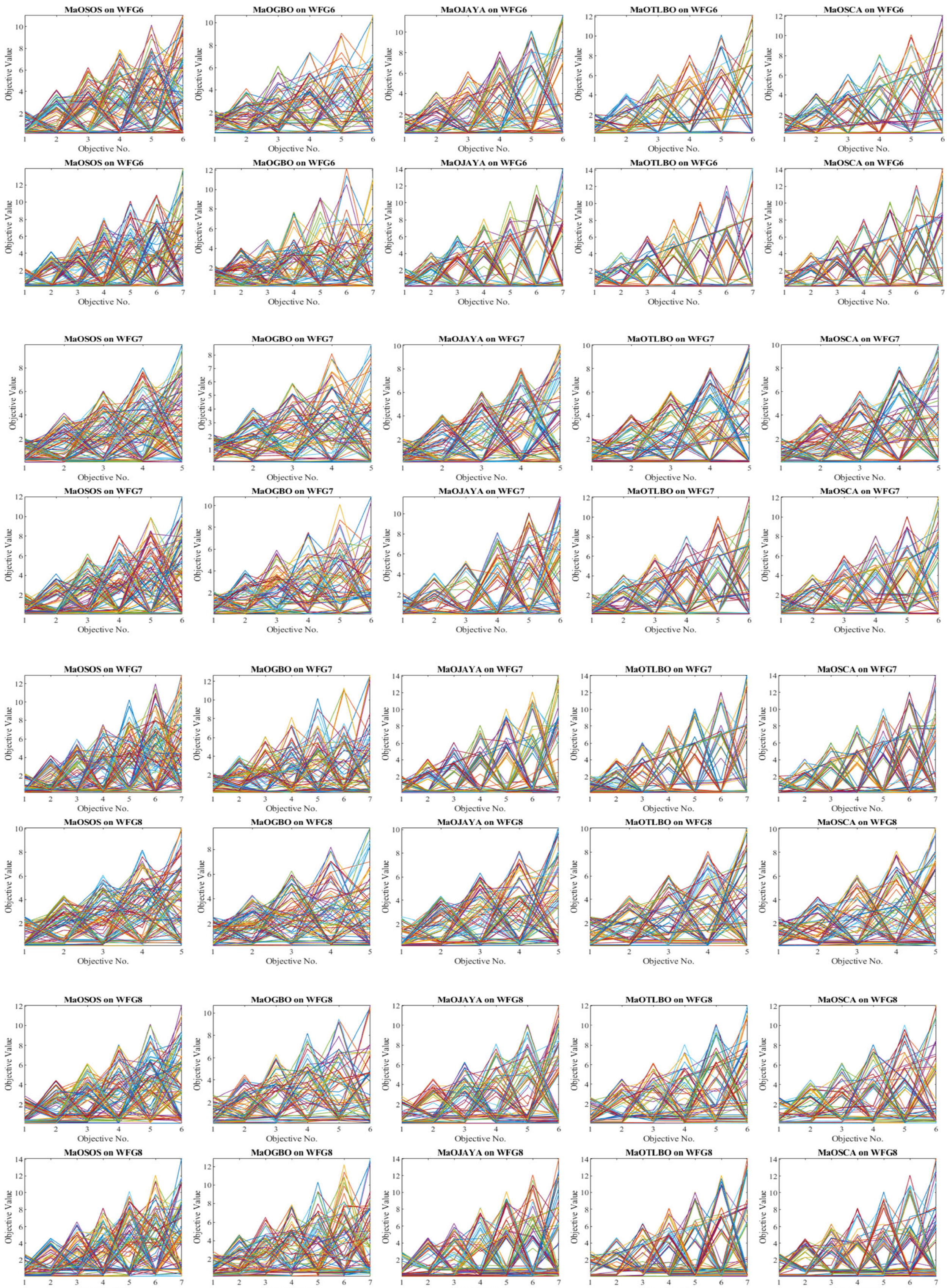


Fig. 4 continued

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

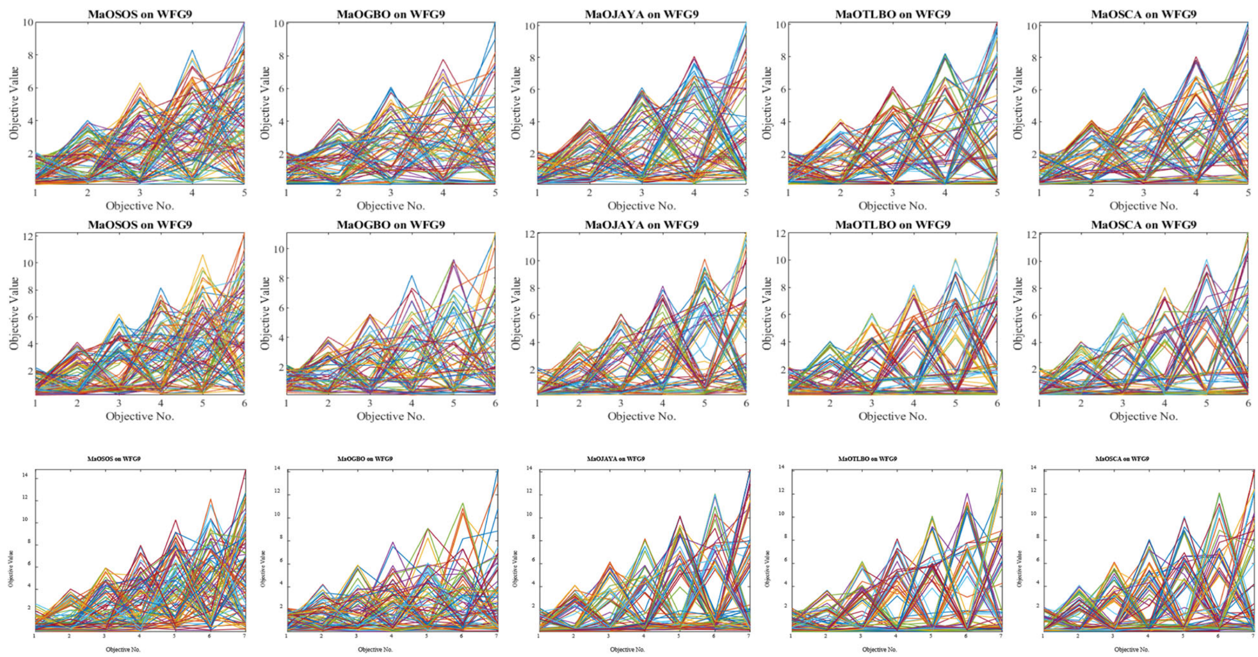


Fig. 4 continued

$O(NM\log N)$, focuses on a teaching-learning-based mechanism that reduces computational overhead but struggles with maintaining a diverse solution set in high-dimensional objective spaces. MaOTLBO, another teaching-learning-based optimizer, has a complexity of $O(NM^2\log N)$, making it faster in smaller problems but less efficient in handling larger objective spaces where diversity and convergence are critical. MaOSCA, which relies on a sine cosine mechanism for generating solutions, has a complexity of $O\left(NM^2\right)$, leading to reasonable performance in mid-sized problems but may face difficulties in balancing convergence and diversity as the number of objectives increases. When comparing these algorithms, MaOSOS proves to be the most robust for solving many-objective optimization problems, particularly when the number of objectives increases. While the complexity of MaOSOS is higher than that of MaOGBO, MaOJAYA and MaOTLBO, the additional computational cost is justified by its superior ability to handle diverse and complex Pareto fronts. The use of niche preservation, reference point-based selection and the IFM ensures that MaOSOS not only converges faster but also maintains a well-distributed set of solutions across the Pareto front, making it the most effective algorithm in high-dimensional objective spaces. Thus, despite its higher complexity, MaOSOS outperforms other algorithms in managing the trade-off between computational efficiency and solution quality.

The distribution of HV values across various runs for each algorithm is displayed by the box plots shown in Fig. 5 along with the strong consistency and durability of MaOSOS. These graphs allow simple distinction of any outliers and offer a clearer view of the algorithms' performances regarding variability. The narrow distribution and minimal outliers in the MaOSOS box plots on various WFG benchmarks indicate its reliability and durability in attaining superior hypervolume results. Each algorithm's progress toward the optimal Pareto front can be seen in the convergence curves regarding the HV metric over time shown in Fig. 6. These graphs demonstrate how rapidly MaOSOS moves toward a more robust HV value than various algorithms. This illustrates both the effectiveness of the algorithm and its ability to span a broader region of the Pareto front across various objective spaces in the WFG benchmark problems. In these curves MaOSOS showcases its superior function by rapidly converging and achieving a higher final HV value.

4.3 Experimental results on RWMaOP problems

Table 8 highlights the Spacing (SP) metric results for MaOSOS and its counterparts across a range of real-world many-objective optimization problems (RWMaOPs). This metric measures the distribution uniformity of solutions, with a lower mean value indicating better performance. In RWMaOP1, MaOSOS records a SP of 1.274 ± 0.213, which is higher than MaOGBO but significantly lower compared to MaOJAYA, MaOTLBO and MaOSCA. In RWMaOP2, MaOSOS achieves a perfect SP of 0 ± 0, substantially outperforming all other algorithms by a significant margin. In RWMaOP3, MaOSOS shows a SP of 15.556 ± 0.419, which is considerably better than the SP values achieved by other competing algorithms. In RWMaOP4, MaOSOS has a SP of

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 3 Comparison of IGD Metric on WFG problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
WFG1	5	14	1.858 ± 0.757 =	1.283 ± 0.098 =	1.274 ± 0.233 =	**0.993 ± 0.123 =**	1.206 ± 0.149
	6	15	1.684 ± 0.025 =	1.441 ± 0.181 =	1.379 ± 0.143 =	**1.307 ± 0.101 =**	1.53 ± 0.18
	7	16	2.176 ± 0.203 =	1.784 ± 0.416 =	1.81 ± 0.103 =	**1.626 ± 0.078 =**	1.775 ± 0.093
WFG2	5	14	0.772 ± 0.006 =	0.519 ± 0.013 =	0.619 ± 0.167 =	**0.496 ± 0.007 =**	0.506 ± 0.007
	6	15	1.117 ± 0.028 =	**0.684 ± 0.005 =**	0.756 ± 0.068 =	0.696 ± 0.007 =	0.811 ± 0.235
	7	16	1.551 ± 0.138 =	**0.84 ± 0.061 =**	1.203 ± 0.109 =	0.854 ± 0.042 =	0.86 ± 0.019
WFG3	5	14	0.725 ± 0.102 =	**0.652 ± 0.129 =**	0.704 ± 0.097 =	0.824 ± 0.058 =	0.901 ± 0.084
	6	15	1.25 ± 0.136 =	1.243 ± 0.316 =	**0.799 ± 0.045 =**	1.318 ± 0.086 =	1.217 ± 0.43
	7	16	1.831 ± 0.124 =	1.662 ± 0.938 =	**0.746 ± 0.085 =**	1.639 ± 0.101 =	0.987 ± 0.019
WFG4	5	14	**1.165 ± 0.012 =**	1.262 ± 0.012 =	1.273 ± 0.029 =	1.223 ± 0.002 =	1.231 ± 0.005
	6	15	**1.803 ± 0.029 =**	1.964 ± 0.013 =	2.062 ± 0.005 =	1.956 ± 0.011 =	1.952 ± 0.006
	7	16	**2.532 ± 0.011 =**	2.604 ± 0.036 =	2.703 ± 0.011 =	2.651 ± 0.013 =	2.714 ± 0.119
WFG5	5	14	**1.141 ± 0.007 =**	1.244 ± 0.011 =	1.259 ± 0.005 =	1.207 ± 0.003 =	1.2 ± 0.004
	6	15	**1.793 ± 0.024 =**	1.97 ± 0.025 =	2.003 ± 0.027 =	1.933 ± 0.007 =	1.921 ± 0.015
	7	16	**2.508 ± 0.011 =**	2.624 ± 0.099 =	2.66 ± 0.048 =	2.633 ± 0.011 =	2.578 ± 0.019
WFG6	5	14	**1.181 ± 0.016 =**	1.294 ± 0.007 =	1.282 ± 0.013 =	1.232 ± 0.003 =	1.248 ± 0.022
	6	15	**1.813 ± 0.018 =**	2.025 ± 0.019 =	2.117 ± 0.047 =	1.952 ± 0.002 =	1.965 ± 0.007
	7	16	**2.538 ± 0.01 =**	2.657 ± 0.037 =	2.773 ± 0.072 =	2.661 ± 0.015 =	2.63 ± 0.016
WFG7	5	14	**1.205 ± 0.013 =**	1.285 ± 0.016 =	1.308 ± 0.004 =	1.241 ± 0.003 =	1.243 ± 0.003
	6	15	**1.826 ± 0.01 =**	1.956 ± 0.035 =	2.076 ± 0.064 =	1.988 ± 0.016 =	1.983 ± 0.013
	7	16	**2.585 ± 0.051 =**	2.599 ± 0.035 =	2.694 ± 0.032 =	2.636 ± 0.018 =	2.604 ± 0.009
WFG8	5	14	**1.266 ± 0.011 =**	1.331 ± 0.005 =	1.345 ± 0.025 =	1.282 ± 0.01 =	1.301 ± 0.017
	6	15	**1.93 ± 0.018 =**	2.061 ± 0.016 =	2.121 ± 0.021 =	2.029 ± 0.032 =	2.172 ± 0.159
	7	16	2.744 ± 0.058 =	2.786 ± 0.054 =	2.804 ± 0.187 =	**2.69 ± 0.022 =**	2.797 ± 0.284
WFG9	5	14	**1.205 ± 0.021 =**	1.252 ± 0.021 =	1.269 ± 0.049 =	1.226 ± 0.008 =	1.22 ± 0.016
	6	15	**1.926 ± 0.091 =**	1.967 ± 0.024 =	2.006 ± 0.04 =	1.932 ± 0.012 =	1.928 ± 0.014
	7	16	2.773 ± 0.025 =	2.645 ± 0.051 =	2.563 ± 0.006 =	**2.558 ± 0.018 =**	2.569 ± 0.078

Bold indicates the best solution among the tested algorithms

48,941 ± 895, which, while higher than MaOGBO, is lower than the results of MaOJAYA, MaOTLBO and MaOSCA. In RWMaOP5, MaOSOS achieves an SP of 0.036 ± 0.003, demonstrating superior uniformity in solution distribution compared to its counterparts. In Table 8, SP value compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS is better in 3, 5, 5 and 5 out of 5 cases. These results indicate that MaOSOS excels in maintaining a uniform distribution of solutions shown in Fig. 7 across various complex real-world many-objective problems. Especially in scenarios such as the 10-bar truss structure and water and oil repellent fabric development, MaOSOS performance in terms of SP is notably superior, reflecting its efficacy in diverse optimization contexts.

From Table 9, the overall performance of MaOSOS in terms of the Hypervolume (HV) metric across RWMaOP1, MaOSOS achieves an HV of 0.002 ± 0, which is substantially higher than that of MaOGBO and outperforms MaOJAYA, MaOTLBO and MaOSCA. For RWMaOP2, MaOSOS records an HV of 0.081 ± 0.001, surpassing MaOGBO, MaOJAYA, MaOTLBO and significantly better than MaOSCA. In RWMaOP3, MaOSOS has an HV of 0.016 ± 0, showing a competitive performance compared to MaOGBO and better than MaOJAYA, MaOTLBO and MaOSCA. For RWMaOP4, MaOSOS HV is 0.543 ± 0.001, which is higher than all other compared algorithms. In RWMaOP5, MaOSOS achieves an HV of 0.531 ± 0.003, indicating a slightly lower performance compared to some counterparts but still competitive. In Table 9 on the HV values, when respectively compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS is better in 4, 5, 5 and 4 out of 5 cases and is only worse in 20%, 0%, 0% and 20% cases. These results emphasize the capability of

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 4 Comparison of SP Metric on WFG problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
WFG1	5	14	**0.439 ± 0.29 =**	0.888 ± 0.108 =	0.836 ± 0.009 =	0.996 ± 0.119 =	0.926 ± 0.124
	6	15	**0.701 ± 0.028 =**	1.18 ± 0.066 =	1.099 ± 0.057 =	1.504 ± 0.233 =	1.109 ± 0.023
	7	16	**0.837 ± 0.086 =**	1.424 ± 0.061 =	1.254 ± 0.108 =	1.85 ± 0.092 =	1.293 ± 0.191
WFG2	5	14	**0.369 ± 0.065 =**	0.388 ± 0.092 =	0.582 ± 0.241 =	0.546 ± 0.04 =	0.644 ± 0.086
	6	15	**0.444 ± 0.034 =**	0.608 ± 0.251 =	0.794 ± 0.072 =	1.006 ± 0.107 =	0.708 ± 0.162
	7	16	**0.54 ± 0.023 =**	0.862 ± 0.247 =	0.72 ± 0.198 =	0.929 ± 0.239 =	0.854 ± 0.138
WFG3	5	14	**0.279 ± 0.043 =**	0.391 ± 0.053 =	0.691 ± 0.115 =	0.728 ± 0.058 =	0.578 ± 0.048
	6	15	0.437 ± 0.13 =	**0.428 ± 0.129 =**	0.688 ± 0.057 =	1.125 ± 0.104 =	0.666 ± 0.131
	7	16	0.652 ± 0.073 =	0.726 ± 0.172 =	**0.552 ± 0.114 =**	1.393 ± 0.091 =	0.748 ± 0.115
WFG4	5	14	**0.398 ± 0.018 =**	0.745 ± 0.026 =	0.942 ± 0.041 =	0.803 ± 0.067 =	0.88 ± 0.056
	6	15	**0.625 ± 0.074 =**	1.076 ± 0.075 =	1.352 ± 0.149 =	1.194 ± 0.067 =	1.217 ± 0.032
	7	16	**0.907 ± 0.125 =**	1.741 ± 0.134 =	1.736 ± 0.167 =	1.533 ± 0.124 =	1.548 ± 0.144
WFG5	5	14	**0.418 ± 0.034 =**	0.715 ± 0.023 =	0.885 ± 0.069 =	0.791 ± 0.041 =	0.851 ± 0.042
	6	15	1.189 ± 0.096 =	**0.624 ± 0.042 =**	1.193 ± 0.112 =	1.14 ± 0.033 =	1.179 ± 0.044
	7	16	1.535 ± 0.163 =	**0.853 ± 0.02 =**	1.573 ± 0.123 =	1.383 ± 0.018 =	1.382 ± 0.051
WFG6	5	14	**0.446 ± 0.019 =**	0.739 ± 0.052 =	0.831 ± 0.121 =	0.831 ± 0.004 =	0.812 ± 0.049
	6	15	**0.636 ± 0.06 =**	1.304 ± 0.041 =	1.388 ± 0.052 =	1.143 ± 0.096 =	1.208 ± 0.008
	7	16	**0.931 ± 0.023 =**	1.569 ± 0.057 =	1.568 ± 0.086 =	1.48 ± 0.068 =	1.434 ± 0.136
WFG7	5	14	0.913 ± 0.028 =	0.715 ± 0.027 =	**0.436 ± 0.028 =**	0.808 ± 0.026 =	0.912 ± 0.011
	6	15	**0.564 ± 0.042 =**	1.144 ± 0.085 =	1.512 ± 0.191 =	1.416 ± 0.048 =	1.239 ± 0.029
	7	16	**0.799 ± 0.028 =**	1.544 ± 0.025 =	1.876 ± 0.083 =	1.648 ± 0.071 =	1.528 ± 0.093
WFG8	5	14	**0.407 ± 0.042 =**	0.719 ± 0.05 =	0.822 ± 0.058 =	0.987 ± 0.05 =	1.023 ± 0.039
	6	15	1.777 ± 0.174	1.077 ± 0.084 =	1.203 ± 0.047 =	1.35 ± 0.225 =	**0.714 ± 0.04 =**
	7	16	1.968 ± 0.045	1.522 ± 0.099 =	1.99 ± 0.197 =	2.095 ± 0.2 =	**1.063 ± 0.057 =**
WFG9	5	14	**0.369 ± 0.036 =**	0.723 ± 0.099 =	0.836 ± 0.058 =	0.864 ± 0.053 =	0.792 ± 0.026
	6	15	**0.509 ± 0.044 =**	1.222 ± 0.101 =	1.148 ± 0.077 =	1.283 ± 0.093 =	1.163 ± 0.111
	7	16	**0.788 ± 0.042 =**	1.44 ± 0.029 =	1.679 ± 0.056 =	1.615 ± 0.038 =	1.481 ± 0.098

Bold indicates the best solution among the tested algorithms

MaOSOS in effectively covering a larger area of the Pareto front compared to its counterparts shown in Fig. 7.

From Table 10, the overall running time of MaOSOS is notably the least in several of the real-world many-objective optimization problems (RWMaOPs), showcasing its computational efficiency. The specific running times and comparisons are as follows. In RWMaOP1, MaOSOS runtime is 1.026 s, accounting for only 8.6%, 34.5%, 6.1% and 34.8% of the runtimes of MaOGBO (11.897 s), MaOJAYA (2.971 s), MaOTLBO (16.651 s) and MaOSCA (2.959 s), respectively. For RWMaOP2, with a runtime of 11.69 s, MaOSOS is more efficient, taking 70.1%, 87.9%, 59.7% and 89.5% of the time taken by MaOGBO, MaOJAYA, MaOTLBO and MaOSCA. In RWMaOP3, MaOSOS runtime is 11.514 s, which is significantly more efficient than MaOTLBO and MaOSCA and comparable to MaOGBO and MaOJAYA. For RWMaOP4, MaOSOS has a runtime of 9.868 s, demonstrating faster processing by taking only a fraction of the time taken by MaOTLBO and MaOSCA. In RWMaOP5, with a runtime of 0.887 s, MaOSOS is significantly more efficient than all other algorithms, especially MaOTLBO and MaOSCA, indicating a higher speed. In Table 10, RT value compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA, the proposed MaOSOS is better in 2, 5, 3 and 2 out of 5 cases. These results clearly show that MaOSOS has a faster running speed compared to its counterparts in most of the RWMaOPs. The lower running times of MaOSOS, especially in cases like RWMaOP1 and RWMaOP5, suggest that it not only provides efficient solutions but also does so in a considerably shorter amount of time. This efficiency, as demonstrated in Table 10, proves that MaOSOS has higher search efficiency in solving real-world many-objective problems.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 5 Comparison of SD Metric on WFG problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
WFG1	5	14	**0.495 ± 0.438 =**	0.542 ± 0.016 =	0.554 ± 0.038 =	0.7 ± 0.047 =	0.688 ± 0.038
	6	15	**0.223 ± 0.027 =**	0.665 ± 0.093 =	0.704 ± 0.066 =	0.932 ± 0.101 =	0.763 ± 0.1
	7	16	**0.234 ± 0.008 =**	0.73 ± 0.058 =	0.737 ± 0.036 =	1.03 ± 0.037 =	0.762 ± 0.052
WFG2	5	14	**0.136 ± 0.015 =**	0.583 ± 0.03 =	0.647 ± 0.143 =	0.414 ± 0.021 =	0.484 ± 0.038
	6	15	**0.119 ± 0.005 =**	0.698 ± 0.023 =	0.709 ± 0.097 =	0.631 ± 0.063 =	0.679 ± 0.179
	7	16	**0.118 ± 0.015 =**	0.723 ± 0.042 =	0.84 ± 0.143 =	0.685 ± 0.151 =	0.647 ± 0.01
WFG3	5	14	**0.131 ± 0.007 =**	0.333 ± 0.021 =	0.676 ± 0.031 =	0.425 ± 0.061 =	0.626 ± 0.054
	6	15	**0.126 ± 0.018 =**	0.379 ± 0.026 =	0.679 ± 0.015 =	0.527 ± 0.069 =	0.665 ± 0.094
	7	16	0.411 ± 0.033 =	**0.136 ± 0.006 =**	0.833 ± 0.049 =	0.511 ± 0.067 =	0.704 ± 0.096
WFG4	5	14	0.211 ± 0.02 =	**0.092 ± 0.002 =**	0.34 ± 0.041 =	0.255 ± 0.021 =	0.259 ± 0.01
	6	15	**0.1 ± 0.013 =**	0.234 ± 0.015 =	0.345 ± 0.025 =	0.257 ± 0.007 =	0.233 ± 0.003
	7	16	**0.103 ± 0.004 =**	0.269 ± 0.02 =	0.422 ± 0.039 =	0.336 ± 0.061 =	0.393 ± 0.156
WFG5	5	14	**0.09 ± 0.009 =**	0.215 ± 0.02 =	0.353 ± 0.021 =	0.265 ± 0.01 =	0.284 ± 0.006
	6	15	**0.087 ± 0.01 =**	0.272 ± 0.021 =	0.363 ± 0.034 =	0.26 ± 0.018 =	0.247 ± 0.005
	7	16	**0.089 ± 0.008 =**	0.292 ± 0.023 =	0.425 ± 0.033 =	0.339 ± 0.02 =	0.308 ± 0.009
WFG6	5	14	0.304 ± 0.036 =	0.213 ± 0.001 =	**0.065 ± 0.014 =**	0.252 ± 0.006 =	0.269 ± 0.02
	6	15	**0.08 ± 0.006 =**	0.271 ± 0.018 =	0.415 ± 0.016 =	0.233 ± 0.017 =	0.24 ± 0.012
	7	16	**0.087 ± 0.013 =**	0.299 ± 0.01 =	0.422 ± 0.043 =	0.356 ± 0.027 =	0.315 ± 0.042
WFG7	5	14	**0.085 ± 0.012 =**	0.211 ± 0.009 =	0.373 ± 0.008 =	0.263 ± 0.013 =	0.292 ± 0.012
	6	15	**0.093 ± 0.004 =**	0.244 ± 0.011 =	0.484 ± 0.142 =	0.331 ± 0.055 =	0.26 ± 0.008
	7	16	**0.101 ± 0.008 =**	0.271 ± 0.014 =	0.47 ± 0.042 =	0.379 ± 0.025 =	0.323 ± 0.006
WFG8	5	14	0.279 ± 0.011 =	0.216 ± 0.006 =	0.297 ± 0.024 =	**0.079 ± 0.011 =**	0.318 ± 0.017
	6	15	0.275 ± 0.047 =	0.226 ± 0.003 =	0.347 ± 0.046 =	**0.074 ± 0.004 =**	0.539 ± 0.189
	7	16	**0.086 ± 0.012 =**	0.231 ± 0.009 =	0.512 ± 0.162 =	0.408 ± 0.038 =	0.503 ± 0.186
WFG9	5	14	**0.103 ± 0.012 =**	0.22 ± 0.016 =	0.351 ± 0.022 =	0.298 ± 0.034 =	0.292 ± 0.015
	6	15	0.283 ± 0.02	0.281 ± 0.019 =	0.341 ± 0.036 =	0.324 ± 0.009 =	**0.094 ± 0.009 =**
	7	16	0.348 ± 0.024	0.29 ± 0.01 =	0.396 ± 0.017 =	0.372 ± 0.024 =	**0.117 ± 0.003 =**

Bold indicates the best solution among the tested algorithms

In the context of many-objective optimization, the performance of MaOSOS, as concluded from Tables 2, 3, 4, 5, 6, 7, 8, 9 and 10, demonstrates its significant strengths and weaknesses across various test problems. Utilizing Wilcoxon rank sum test for a comprehensive assessment, MaOSOS exhibits a diverse range of efficiencies. Based on the Wilcoxon rank-sum test, MaOSOS obtains the best score of 1.59, which means that our proposed algorithm outperforms MaOGBO, MaOJAYA, MaOTLBO and MaOSCA achieves 7.08, 16.09, 8.85 and 17.7. Thus, MaOSOS shows better overall performance compared to MaOGBO, MaOJAYA, MaOTLBO and MaOSCA.

Across the different test scenarios, MaOSOS outperforms competitors like MaOGBO, MaOJAYA, MaOTLBO and MaOSCA in most of the test problems. Specifically, MaOSOS shows 96%, 64% and 73% significantly better values than MaOGBO, MaOJAYA and MaOTLBO, respectively, across a combined total of WFG and RWMaOPs problems.

In earlier evaluations the performance of MaOSOS in relation to other algorithms was depicted using symbols (+, -, =) to denote worse and better as well as equal. Using a Wilcoxon signed-rank test on every performance metric (GD, IGD, SP, SD, HV and RT), the statistical significance of the variations between MaOSOS and the competing algorithms (MaOGBO, MaOJAYA, MaOTLBO and MaOSCA) is investigated. The results from the test established that the distinctions emphasized by the + symbol (representing better performance of MaOSOS) are statistically significant in most situations.

The performance advantage of MaOSOS over MaOGBO and MaOJAYA originates from essential features in its architecture. The use of the IFM helps MaOSOS harmonize

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 6 Comparison of HV Metric on WFG problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
WFG1	5	14	5.0987e−1 (4.86e−2)	5.3465e−1 (6.47e−2)	**6.3122e−1 (5.58e−2)**	5.1555e−1 (4.12e−2)	5.1084e−1 (4.74e−2)
	6	15	4.9977e−1 (5.84e−2)	4.2082e−1 (2.75e−2)	5.1270e−1 (4.88e−2)	5.2959e−1 (1.01e−1)	**5.7951e−1 (3.65e−2)**
	7	16	4.5263e−1 (5.69e−2)	4.5214e−1 (9.53e−2)	4.3285e−1 (1.40e−2)	4.9229e−1 (6.04e−2)	**5.1812e−1 (3.81e−2)**
WFG2	5	14	9.5478e−1 (1.09e−2)	9.0354e−1 (1.77e−2)	9.5893e−1 (1.18e−2)	9.3832e−1 (6.18e−3)	**9.6869e−1 (5.89e−3)**
	6	15	9.5056e−1 (6.06e−3)	8.9701e−1 (3.30e−2)	**9.7048e−1 (8.00e−3)**	9.3691e−1 (1.14e−2)	9.5880e−1 (4.87e−3)
	7	16	**9.5935e−1 (7.74e−3)**	9.1328e−1 (2.52e−2)	9.0335e−1 (9.81e−2)	9.4311e−1 (1.17e−2)	9.5533e−1 (2.75e−3)
WFG3	5	14	4.7342e−2 (5.51e−3)	1.2492e−2 (9.24e−3)	**8.072e−2 (2.57e−2)**	5.6515e−2 (1.30e−2)	4.3743e−2 (3.15e−2)
	6	15	4.6036e−3 (7.97e−3)	0.000e + 0 (0.00e + 0)	7.4309e−3 (6.45e−3)	**3.3855e−2 (2.12e−2)**	1.3960e−2 (1.34e−2)
	7	16	0.000e + 0 (0.00e + 0)	0.000e + 0 (0.00e + 0)	0.000e + 0 (0.00e + 0)	**2.7872e−2 (2.18e−2)**	0.000e + 0 (0.00e + 0)
WFG4	5	14	**7.0269e−1 (7.17e−3)**	6.4105e−1 (5.13e−3)	6.4437e−1 (1.23e−2)	6.7384e−1 (7.19e−3)	6.9225e−1 (3.92e−3)
	6	15	7.1235e−1 (6.51e−2)	6.5530e−1 (2.30e−4)	6.5269e−1 (4.47e−3)	6.9508e−1 (1.44e−2)	**7.4092e−1 (3.92e−3)**
	7	16	7.7154e−1 (7.28e−3)	6.7192e−1 (1.38e−2)	6.5928e−1 (3.10e−2)	7.5691e−1 (1.20e−2)	**7.7160e−1 (1.39e−2)**
WFG5	5	14	**6.8019e−1 (7.94e−3)**	6.1048e−1 (4.98e−3)	6.2350e−1 (4.85e−3)	6.4916e−1 (8.23e−3)	6.7816e−1 (2.47e−3)
	6	15	**7.2162e−1 (3.74e−3)**	6.3056e−1 (1.21e−2)	6.3796e−1 (1.23e−2)	6.7071e−1 (8.53e−3)	7.0366e−1 (1.18e−2)
	7	16	**7.5671e−1 (1.08e−2)**	6.2268e−1 (9.50e−3)	6.0290e−1 (5.45e−3)	7.1625e−1 (1.16e−2)	7.4791e−1 (1.58e−2)
WFG6	5	14	**6.5259e−1 (2.04e−2)**	5.6491e−1 (1.15e−2)	6.1072e−1 (2.46e−2)	6.1216e−1 (2.70e−2)	6.4945e−1 (1.23e−2)
	6	15	6.9245e−1 (7.90e−4)	5.5192e−1 (2.29e−2)	6.2044e−1 (2.58e−2)	6.3227e−1 (8.24e−3)	**7.0539e−1 (1.44e−2)**
	7	16	7.2804e−1 (2.29e−3)	5.1890e−1 (5.39e−2)	6.0521e−1 (1.94e−2)	7.0452e−1 (1.14e−2)	**7.3006e−1 (2.51e−2)**
WFG7	5	14	**6.8966e−1 (1.62e−2)**	6.2292e−1 (7.91e−3)	5.8310e−1 (8.04e−3)	6.7561e−1 (2.49e−2)	6.8115e−1 (6.13e−3)
	6	15	**7.3421e−1 (1.63e−2)**	6.3957e−1 (5.75e−3)	6.2361e−1 (2.34e−2)	6.9158e−1 (2.15e−2)	7.1980e−1 (3.50e−2)
	7	16	**7.8211e−1 (1.26e−2)**	6.1576e−1 (6.33e−2)	5.5062e−1 (4.80e−2)	7.6495e−1 (9.52e−3)	7.4908e−1 (4.71e−2)
WFG8	5	14	5.6029e−1 (1.44e−2)	5.1589e−1 (1.65e−2)	5.2761e−1 (1.70e−2)	5.3437e−1 (1.81e−2)	**5.8748e−1 (6.24e−3)**
	6	15	5.8766e−1 (2.43e−3)	5.0719e−1 (3.23e−2)	5.2608e−1 (1.49e−2)	5.8551e−1 (8.04e−3)	**6.1404e−1 (1.64e−2)**
	7	16	5.8098e−1 (2.02e−2)	4.7540e−1 (7.69e−2)	5.6784e−1 (2.17e−2)	5.7979e−1 (3.61e−2)	**6.5334e−1 (8.72e−3)**
WFG9	5	14	**6.3994e−1 (1.73e−2)**	5.6950e−1 (3.29e−2)	5.5061e−1 (8.13e−3)	5.7007e−1 (5.38e−2)	5.9394e−1 (1.96e−2)
	6	15	6.1747e−1 (3.72e−2)	5.6409e−1 (3.10e−2)	4.6304e−1 (1.62e−2)	**6.3514e−1 (7.78e−3)**	6.0832e−1 (1.90e−2)

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 6 (continued)

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
	7	16	**6.9211e−1 (3.22e−2)**	5.4649e−1 (1.11e−2)	4.8455e−1 (7.92e−2)	6.7937e−1 (1.63e−2)	6.2836e−1 (3.24e−2)

Bold indicates the best solution among the tested algorithms

Table 7 Comparison of RT Metric on WFG problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
WFG1	5	14	**0.764 ± 0.083**	0.817 ± 0.118 =	1.066 ± 0.121 =	5.677 ± 0.313 =	2.078 ± 1.54 =
	6	15	0.707 ± 0.043	**0.706 ± 0.027 =**	1.067 ± 0.103 =	4.86 ± 0.151 =	2.838 ± 0.135 =
	7	16	0.722 ± 0.127	**0.72 ± 0.162 =**	1.012 ± 0.156 =	6.744 ± 0.189 =	3.329 ± 0.102 =
WFG2	5	14	**0.632 ± 0.026**	0.768 ± 0.018 =	0.949 ± 0.081 =	5.574 ± 0.175 =	2.995 ± 0.297 =
	6	15	**0.68 ± 0.033**	0.787 ± 0.007 =	1.308 ± 0.385 =	5.395 ± 0.022 =	3.053 ± 0.084 =
	7	16	0.944 ± 0.205	**0.791 ± 0.025 =**	1.111 ± 0.098 =	6.807 ± 0.132 =	3.208 ± 0.025 =
WFG3	5	14	0.723 ± 0.067	**0.707 ± 0.025 =**	1.121 ± 0.159 =	7.435 ± 0.137 =	4.677 ± 0.098 =
	6	15	1.77 ± 0.304	**0.705 ± 0.027 =**	2.045 ± 0.122 =	6.942 ± 0.062 =	5.167 ± 0.047 =
	7	16	1.733 ± 0.165	**0.723 ± 0.019 =**	1.889 ± 0.095 =	8.523 ± 0.104 =	5.653 ± 0.125 =
WFG4	5	14	**0.659 ± 0.048**	0.858 ± 0.012 =	0.89 ± 0.111 =	6.808 ± 0.052 =	4.83 ± 0.059 =
	6	15	**0.735 ± 0.043**	0.853 ± 0.019 =	1.117 ± 0.145 =	6.86 ± 0.499 =	5.181 ± 0.106 =
	7	16	**0.82 ± 0.131**	0.88 ± 0.064 =	1.01 ± 0.024 =	7.755 ± 0.062 =	5.688 ± 0.469 =
WFG5	5	14	**0.598 ± 0.003**	0.833 ± 0.042 =	0.833 ± 0.044 =	6.654 ± 0.038 =	4.429 ± 0.037 =
	6	15	**0.679 ± 0.032**	0.834 ± 0.014 =	0.957 ± 0.058 =	6.248 ± 0.08 =	4.899 ± 0.092 =
	7	16	**0.687 ± 0.036**	0.84 ± 0.014 =	0.948 ± 0.051 =	7.685 ± 0.105 =	5.169 ± 0.119 =
WFG6	5	14	**0.614 ± 0.062**	0.814 ± 0.02 =	0.834 ± 0.035 =	5.924 ± 0.147 =	3.692 ± 0.088 =
	6	15	5.592 ± 0.027 =	0.818 ± 0.012 =	0.95 ± 0.077 =	**0.658 ± 0.043**	4.255 ± 0.109 =
	7	16	6.893 ± 0.104 =	0.845 ± 0.024 =	0.969 ± 0.094 =	**0.688 ± 0.015**	4.682 ± 0.075 =
WFG7	5	14	**0.606 ± 0.002**	0.852 ± 0.023 =	0.879 ± 0.095 =	7.203 ± 0.066 =	4.999 ± 0.053 =
	6	15	**0.691 ± 0.055**	0.876 ± 0.023 =	0.963 ± 0.051 =	6.6 ± 0.098 =	5.452 ± 0.058 =
	7	16	**0.729 ± 0.023**	0.88 ± 0.046 =	0.977 ± 0.043 =	8.016 ± 0.118 =	5.783 ± 0.042 =
WFG8	5	14	**0.598 ± 0.02**	0.829 ± 0.025 =	0.891 ± 0.046 =	5.607 ± 0.065 =	3.506 ± 0.063 =
	6	15	**0.702 ± 0.085**	0.825 ± 0.016 =	0.958 ± 0.013 =	5.384 ± 0.122 =	3.911 ± 0.04 =
	7	16	**0.769 ± 0.113**	0.843 ± 0.073 =	1.026 ± 0.153 =	6.537 ± 0.05 =	4.343 ± 0.063 =
WFG9	5	14	**0.64 ± 0.005**	1.085 ± 0.212 =	1.063 ± 0.222 =	7.366 ± 0.265 =	4.921 ± 0.708 =
	6	15	**0.757 ± 0.015**	0.853 ± 0.017 =	1.044 ± 0.04 =	6.785 ± 0.184 =	5.115 ± 0.201 =
	7	16	0.864 ± 0.098	**0.864 ± 0.051 =**	1.015 ± 0.051 =	8.307 ± 0.223 =	5.65 ± 0.055 =

Bold indicates the best solution among the tested algorithms

convergence with diversity efficiently. MaOSOS diverges from typical methods by employing IFM to utilize past data for improving its ability to explore while also preserving population diversity. This avoids the algorithm entering local maxima commonly encountered by MaOGBO and MaOTLBO that emphasize convergence more than population diversity. In MaOSOS the strategy for niche preservation maintains diversity on the Pareto front better than MaOJAYA. With reference point selection as a feature MaOSOS achieves a more even distribution of solutions on the Pareto front unlike MaOSCA which frequently encounters consistency issues in the solution set. Tenacious performance of MaOSOS in irregular and complex Pareto fronts arises from its adaptability enabling search adjustments compared to MaOGBO designed for normal fronts. In many-objective optimization tasks, MaOSOS demonstrates its robustness through its reliable performance across multiple metrics including HV and IGD.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

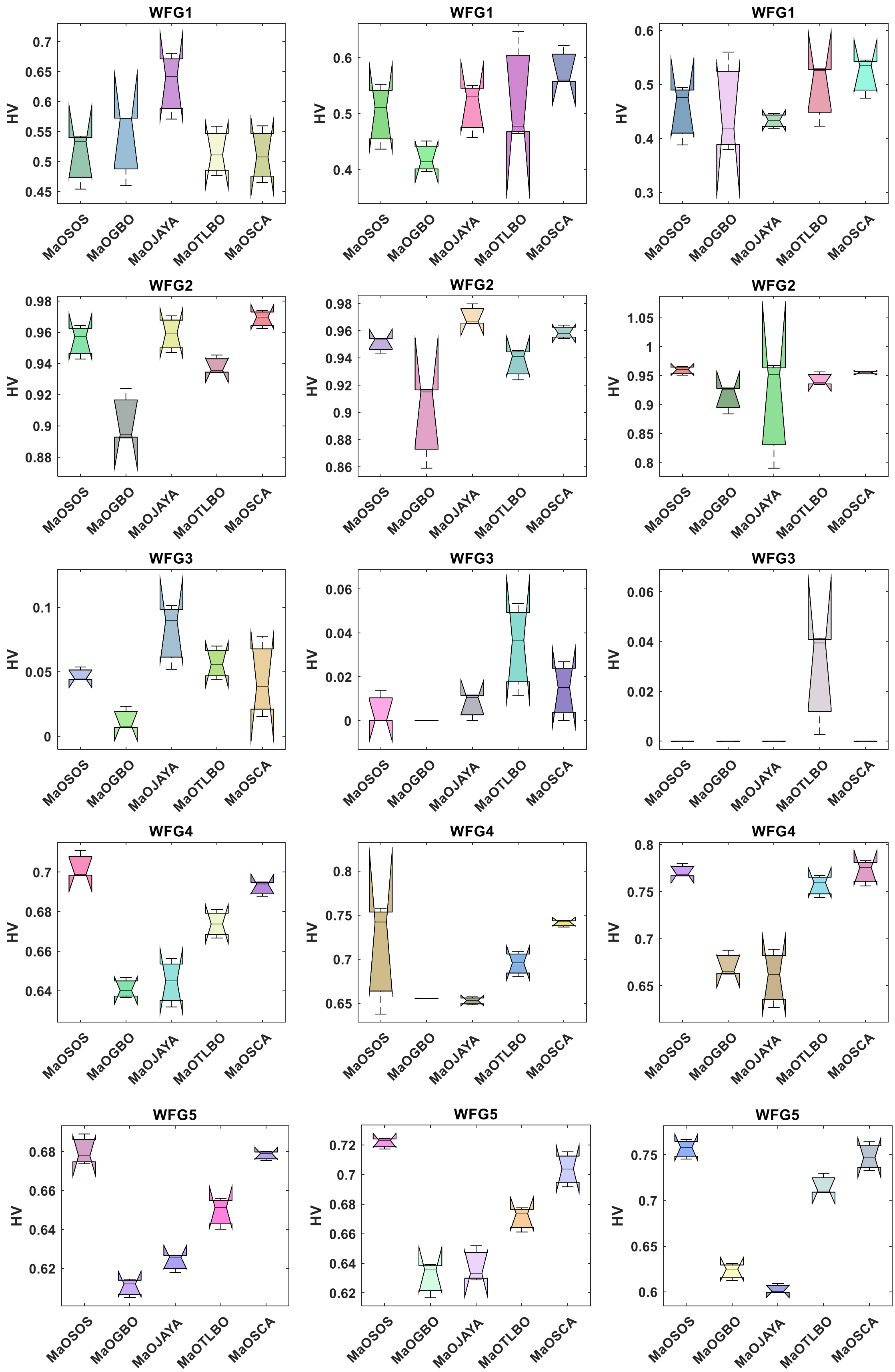


Fig. 5 Box plot obtained on HV metric by different algorithms on WFG problems

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

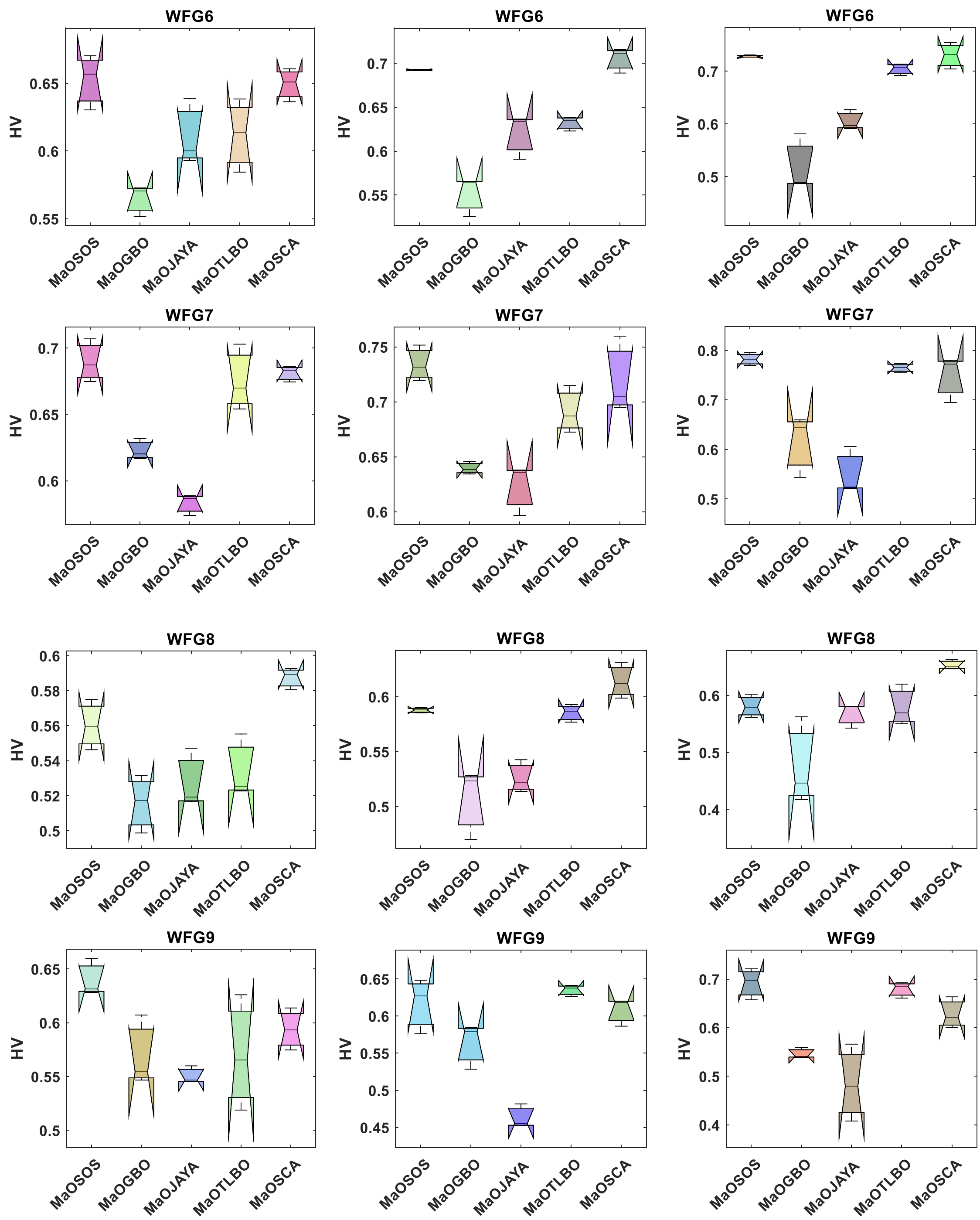


Fig. 5 continued

5 Conclusions

Previous many-objective optimization algorithms often overlooked valuable population data from earlier stages, leading to a potential loss of crucial insights. To address this limitation, we introduced the Many-Objective Symbiotic Organism Search (MaOSOS), building upon the SOS algorithm and incorporating innovative IFMs. These mechanisms preserve historical data and systematically select individuals, enhancing exploratory capabilities and rectifying the neglect of critical information. By leveraging SOS's convergence properties, MaOSOS demonstrates significant performance improvements. Comprehensive comparisons between MaOSOS and leading algorithms like MaOGBO, MaOJAYA, MaOTLBO, and MaOSCA on complex optimization challenges—specifically the WFG1-WFG9 test suite up to

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

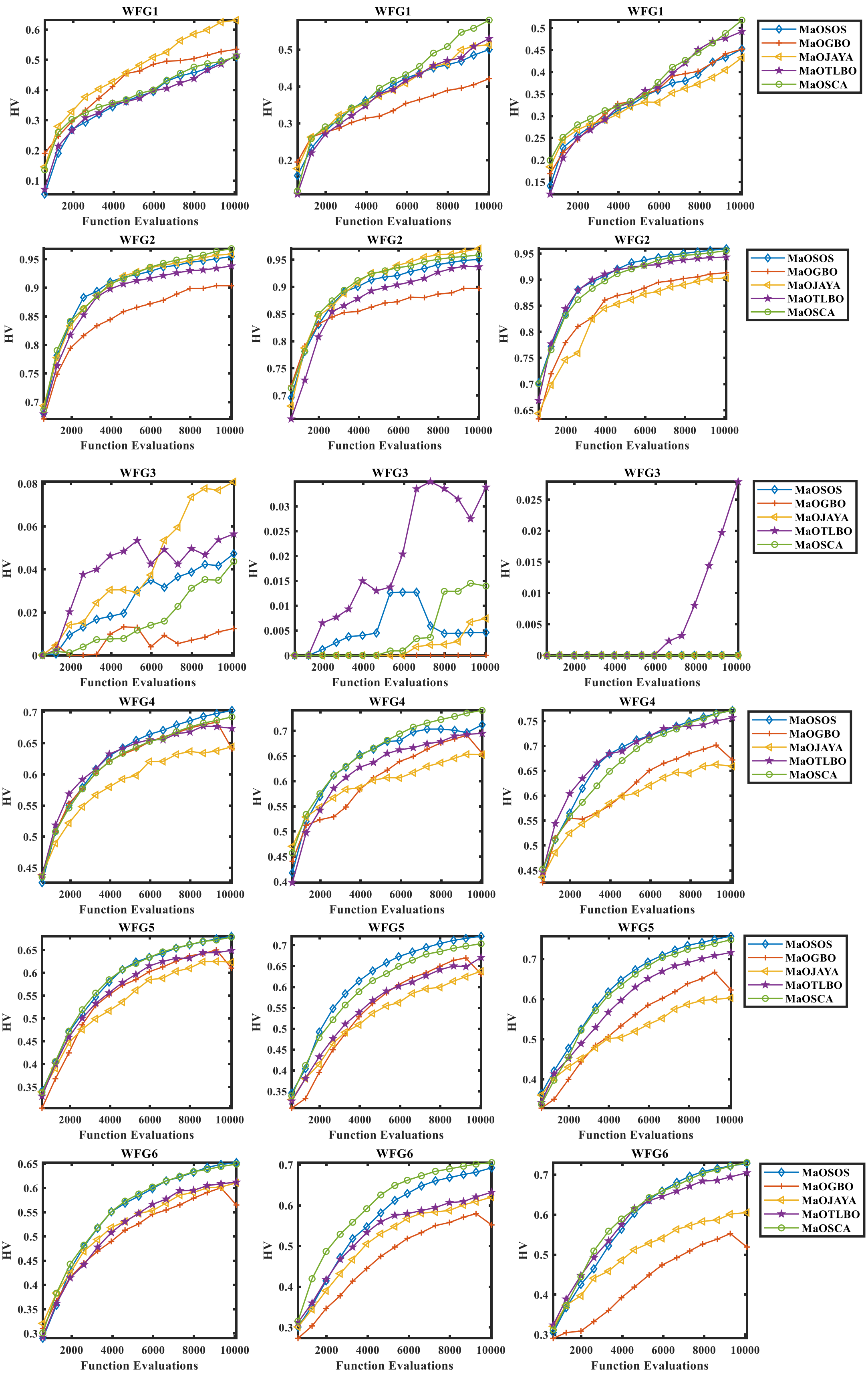


Fig. 6 Convergence curve obtained on HV metric by different algorithms on WFG problems

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

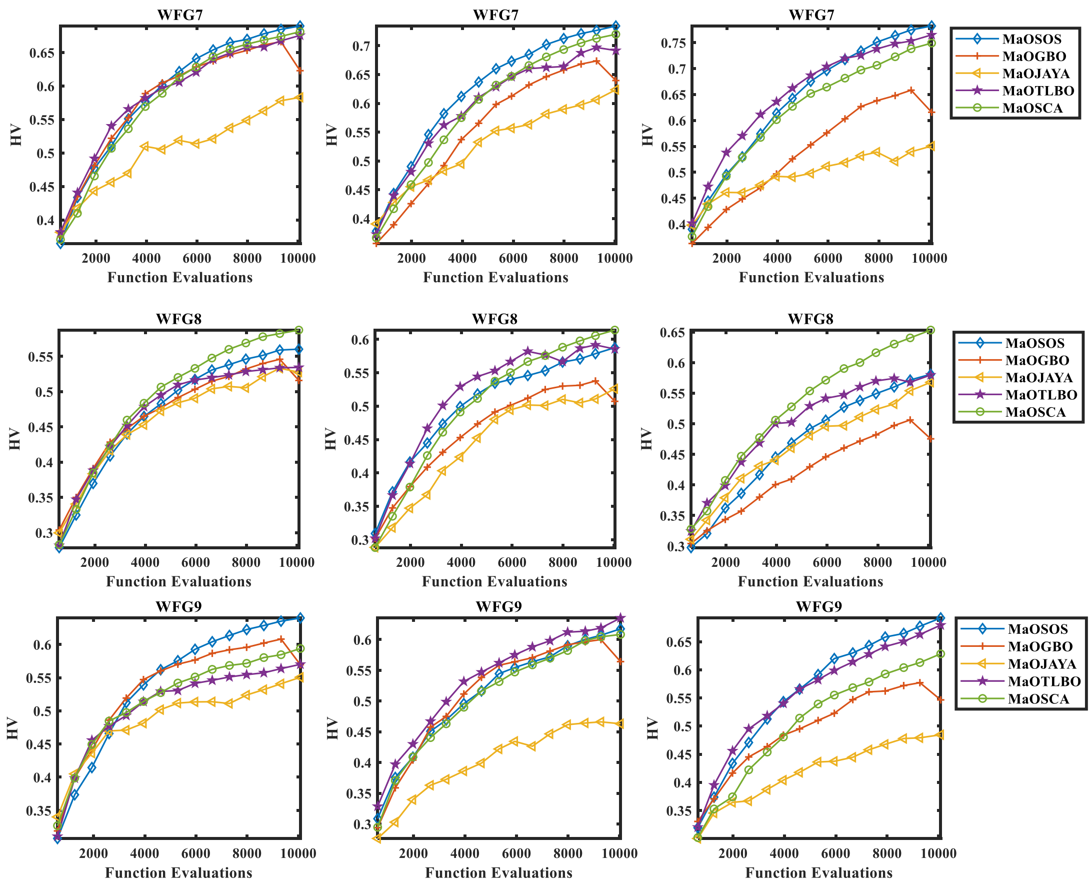


Fig. 6 continued

Table 8 Comparison of SP Metric on RWMaOP problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
RWMaOP1	9	7	1.274 ± 0.213 =	**1.202 ± 0.153 =**	3.403 ± 1.09 =	3.711 ± 2.22 =	3.275 ± 1.23
RWMaOP2	4	10	**0 ± 0 =**	1335.9 ± 109 =	780.95 ± 314 =	1258.9 ± 457 =	871.39 ± 178
RWMaOP3	7	3	**15.556 ± 0.419 =**	31.674 ± 3.24 =	36.132 ± 6.67 =	41.686 ± 5.25 =	35.442 ± 2.06
RWMaOP4	5	6	48,941 ± 895 =	**35,828 ± 5040 =**	46,866 ± 10,200 =	45,686 ± 3740 =	63,810 ± 13,100
RWMaOP5	4	4	**0.036 ± 0.003 =**	0.087 ± 0.012 =	0.105 ± 0.019 =	0.109 ± 0.004 =	0.088 ± 0.026

Bold indicates the best solution among the tested algorithms

Table 9 Comparison of HV Metric on RWMaOP problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
RWMaOP1	9	7	**0.002 ± 0**	0.001 ± 0 =	0.002 ± 0 =	0.001 ± 0 =	0.001 ± 0 =
RWMaOP2	4	10	**0.081 ± 0.001**	0.068 ± 0.008 =	0.081 ± 0 =	0.073 ± 0.002 =	0.018 ± 0.009 =
RWMaOP3	7	3	0.016 ± 0	0.017 ± 0.001 =	0.017 ± 0.001 =	0.016 ± 0 =	**0.018 ± 0 =**
RWMaOP4	5	6	**0.543 ± 0.001**	0.532 ± 0.022 =	0.539 ± 0.003 =	0.54 ± 0.005 =	0.487 ± 0.015 =
RWMaOP5	4	4	0.531 ± 0.003	**0.547 ± 0.018 =**	0.54 ± 0.006 =	0.546 ± 0.006 =	0.546 ± 0.002 =

Bold indicates the best solution among the tested algorithms

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

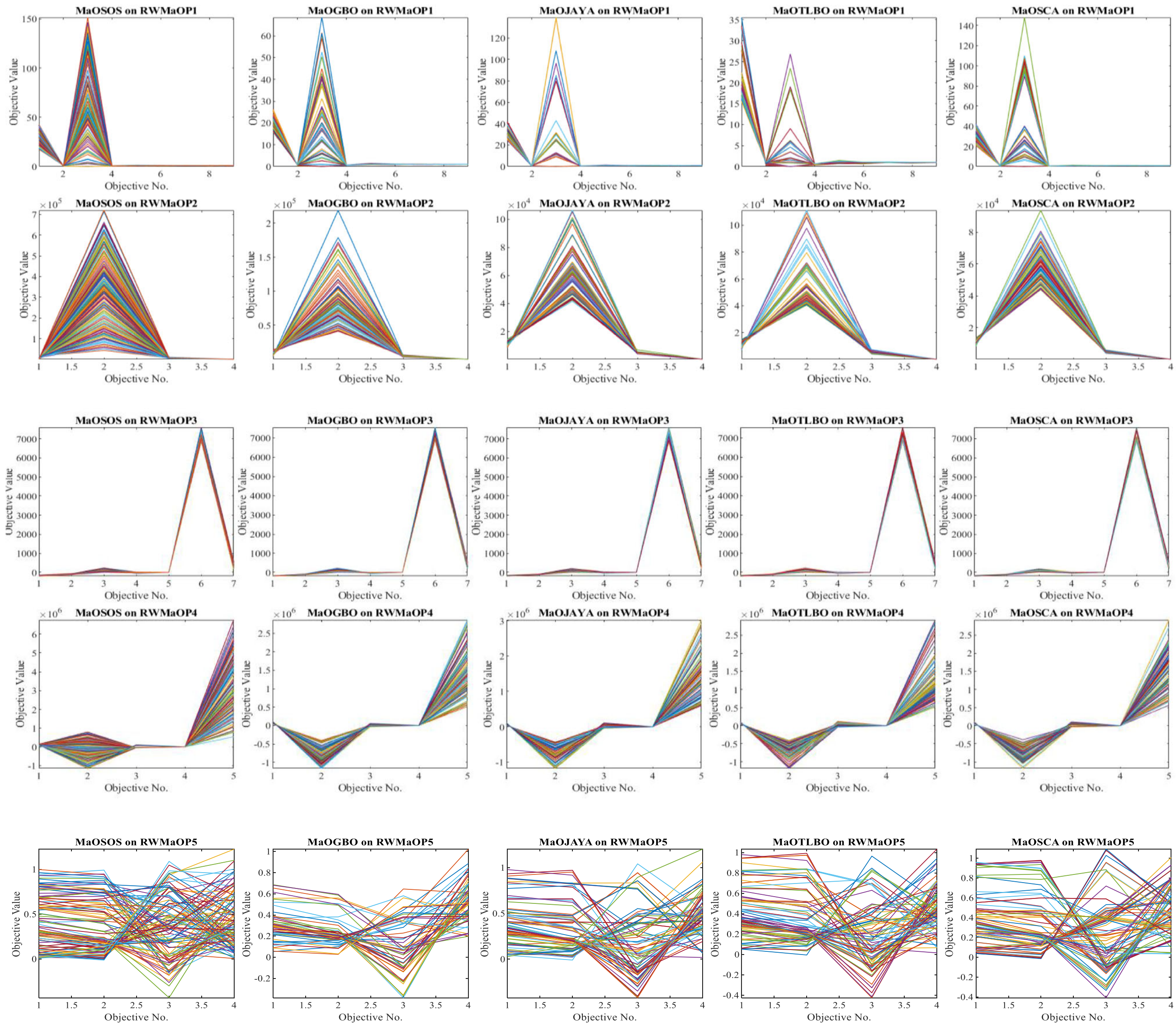


Fig. 7 Best Pareto optimal front obtained by different algorithms on RWMaOP problems

seven objectives—reveal MaOSOS's superior capability in addressing varying Pareto Front (PF) shapes, particularly convex PF challenges. Additionally, on real-world problems (RWMaOP1–RWMaOP5), MaOSOS outperformed contemporary many-objective algorithms across metrics such as GD, IGD, SP, SD, HV, and RT, showcasing superior convergence and diversity.

An important limitation of MaOSOS is its heavy reliance on fixed reference points, which may not suffice for different optimization landscapes. Its computational complexity increases when applied to real-world issues with a large number of objectives or decision variables. Balancing convergence and diversity remains challenging, as some problem instances may emphasize one over the other. Future research can enhance the IFM for dynamic adaptability to problem characteristics, potentially improving efficiency in solving high-dimensional many-objective problems. Merging MaOSOS with optimization strategies like machine learning and heuristic techniques may accelerate convergence and assure higher solution quality.

Further investigation into the applications of MaOSOS in other real-world scenarios, especially in industrial engineering with dynamic and conflicting aims, could lead to a better understanding of its effectiveness in tackling real-world engineering challenges [41, 42]. For instance, MaOSOS shows usefulness in key processes including robotic assembly sequence planning [43, 44]. Industry 4.0, which often presents challenging optimization tasks with

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 10 Comparison of RT Metric on RWMaOP problems

Problem	M	D	MaOSOS	MaOGBO	MaOJAYA	MaOTLBO	MaOSCA
RWMaOP1	9	7	**1.026 ± 0.077 =**	11.897 ± 0.271 =	2.971 ± 0.134 =	16.651 ± 0.378 =	2.959 ± 0.153
RWMaOP2	4	10	**11.69 ± 0.207 =**	16.65 ± 1.1 =	13.264 ± 0.111 =	19.575 ± 0.34 =	13.011 ± 0.203
RWMaOP3	7	3	11.514 ± 0.124 =	**0.969 ± 0.052 =**	2.792 ± 0.044 =	19.011 ± 0.458 =	2.833 ± 0.072
RWMaOP4	5	6	**9.868 ± 0.295 =**	0.998 ± 0.011 =	2.982 ± 0.144 =	14.941 ± 0.2 =	3.029 ± 0.118
RWMaOP5	4	4	**0.887 ± 0.02 =**	8.194 ± 0.166 =	2.752 ± 0.104 =	13.777 ± 0.189 =	2.435 ± 0.255

Bold indicates the best solution among the tested algorithms

numerous objectives, pertains closely to these fields [45]. Thanks to its new reference point strategy and niche preservation methods, MaOSOS effectively handles these problems.

To incorporate dynamic many-objective optimization problems (MaOPs), MaOSOS needs modifications in the IFM to reflect changing objectives over time by monitoring past solution performances and adjusting reference points accordingly. Niche preservation methods can be adapted to account for changing objective spaces, keeping the population varied and adaptable to new circumstances. Modifying cooperative relationships within MaOSOS can facilitate regular combination of solutions in response to new conditions, possibly integrating predictive models to foresee changes in the optimization environment. Researchers can explore these avenues to enhance MaOSOS's performance in dynamic optimization situations, making it suitable for practical applications where goals frequently shift.

Acknowledgements The authors thank Hazem Migdady and Mohammad Sh. Daoud for their help with the first draft of the manuscript.

Funding No Funding.

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.

References

1. Gao, C., Yin, Z., Wang, Z., Li, X., Li, X.: Multilayer network community detection: A novel multi-objective evolutionary algorithm based on consensus prior information [feature]. IEEE Comput. Intell. Mag. **18**(2), 46–59 (2023). https://doi.org/10.1109/MCI.2023.3245729
2. Harif, S., Azizyan, G., Dehghani Darmian, M., Givehchi, M.: Selecting the best location of water quality sensors in water distribution networks by considering the importance of nodes and contaminations using NSGA-III (case study: Zahedan water distribution network, Iran). Environ. Sci. Pollut. Res. Int. **30**(18), 53229–53252 (2023). https://doi.org/10.1007/s11356-023-26075-5
3. Zhou, A., Wang, Y., Zhang, J.: Objective extraction via fuzzy clustering in evolutionary many-objective optimization. Inf. Sci. **509**, 343–355 (2020). https://doi.org/10.1016/j.ins.2018.11.032
4. Mejía-de-Dios, J.-A., Rodríguez-Molina, A., Mezura-Montes, E.: Multiobjective bilevel optimization: A survey of the state-of-the-art. IEEE Trans. Syst. Man. Cybernetics: Syst. **53**(9), 5478–5490 (2023). https://doi.org/10.1109/TSMC.2023.3271125
5. Deb, K., et al.: (September 18–20, 'A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.' Parallel Problem Solving from Nature PPSN. Proceedings: 6th International Conference Paris, France, VI. (2000)
6. Springer: (2000)
7. Li, X., Li, X., Wang, K., Yang, S.: A strength pareto evolutionary algorithm based on adaptive reference points for solving irregular fronts. Inf. Sci. **626**, 658–693 (2023). https://doi.org/10.1016/j.ins.2023.01.073
8. Wang, Y., Zhang, Q., Wang, G.-G.: Improving evolutionary algorithms with information feedback model for large-scale many-objective optimization. Appl. Intell. **53**(10), 11439–11473 (2023). https://doi.org/10.1007/s10489-022-03964-9
9. Anosri, S., Panagant, N., Champasak, P., Bureerat, S., Thipyopas, C., Kumar, S., Pholdee, N., Yıldız, B.S., Yildiz, A.R.: A comparative study of state-of-the-art metaheuristics for solving many-objective optimization problems of fixed wing unmanned aerial vehicle conceptual design. Arch. Comput. Methods Eng. **30**(6), 3657–3671 (2023). https://doi.org/10.1007/s11831-023-09914-z
10. Jameel, M., Abouhawwash, M.: A reference point-based evolutionary algorithm solves multi and many-objective optimization problems: Method and validation. Comput. Intell. Neurosci. 4387053. (2023). https://doi.org/10.1155/2023/4387053
11. Wang, Y., Yang, Y.: Particle swarm optimization with preference order ranking for multi-objective optimization. Inf. Sci. **179**(12), 1944–1959 (2009). https://doi.org/10.1016/j.ins.2009.01.005
12. Deb, K., Mohan, M., Mishra, S.: Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions. Evolution. Comput. **13**(4), 501–525 (2005). https://doi.org/10.1162/106365605774666895
13. Li, M., Zheng, J., Shen, R., Li, K., Yuan, Q.: A grid-based fitness strategy for evolutionary many-objective optimization. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 463–470 (2010). https://doi.org/10.1145/1830483.1830570
14. 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). https://doi.org/10.1109/TEVC.2013.2262178

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

15. 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
16. Zhang, Q., Hui, L.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007). https://doi.org/10.1109/TEVC.2007.892759
17. Premkumar, M., Jangir, P., Sowmya, R., Elavarasan, R.M.: Many-objective gradient-based optimizer to solve optimal power flow problems: Analysis and validations. Eng. Appl. Artif. Intell. **106**, 104479 (2021). https://doi.org/10.1016/j.engappai.2021.104479
18. Mane, S.U., Narsingrao, M.R., Patil, V.C.: A many-objective Jaya algorithm for many-objective optimization problems. Decis. Sci. Lett. **7**(4), 567–582 (2018). https://doi.org/10.5267/j.dsl.2017.11.001
19. Jangir, P., Manoharan, P., Pandya, S., Sowmya, R.: MaOTLBO: Many-objective teaching-learning-based optimizer for control and monitoring the optimal power flow of modern power systems. Int. J. Industrial Eng. Comput. MaOTLBO. **14**(2), 293–308 (2023). https://doi.org/10.5267/j.ijiec.2023.1.003
20. Narayanan, R.C., Ganesh, N., Čep, R., Jangir, P., Chohan, J.S., Kalita, K.: A novel many-objective sine–cosine algorithm (MaOSCA) for engineering. Appl. Math. **11**(10), 2301 (2023)
21. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014). https://doi.org/10.1109/TEVC.2013.2281535
22. 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 (2015). https://doi.org/10.1109/TEVC.2014.2373386
23. Han, Y., Sun, L.: An enhanced strength Pareto evolutionary algorithm based on reference direction for many-objective optimization. In: IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). IEEE Publications (2021). https://doi.org/10.1109/AEECA52519.2021.9574152
24. 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
25. 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 (2014). https://doi.org/10.1109/TEVC.2013.2281534
26. Li, W., Zhang, T., Wang, R., Huang, S., Liang, J.: Multimodal multi-objective optimization: Comparative study of the state-of-the-art. Swarm Evol. Comput. **77**, 101253 (2023). https://doi.org/10.1016/j.swevo.2023.101253
27. Cheng, P., Jeng-Shyang Pan, L., Li, Y., Tang, Chunlun Huang: &. A survey of performance assessment for multiobjective optimizers Fourth International Conference on Genetic and Evolutionary Computing. IEEE Publications (2010). https://doi.org/10.1109/ICGEC.2010.91
28. Pal, M., Saha, S., Bandyopadhyay, S.: DECOR: Differential evolution using clustering based objective reduction for many-objective optimization. Inf. Sci. **423**, 200–218 (2018). https://doi.org/10.1016/j.ins.2017.09.051
29. Bandyopadhyay, S., Mukherjee, A.: An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution. IEEE Trans. Evol. Comput. **19**(3), 400–413 (2014). https://doi.org/10.1109/TEVC.2014.2332878
30. He, Z., Yen, G.G.: Many-objective evolutionary algorithm: Objective space reduction and diversity improvement. IEEE Trans. Evol. Comput. **20**(1), 145–160 (2015). https://doi.org/10.1109/TEVC.2015.2433266
31. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893
32. Liu, Y., Gong, D., Sun, J., Jin, Y.: A many-objective evolutionary algorithm using a one-by-one selection strategy. IEEE Trans. Cybernetics. **47**(9), 2689–2702 (2017). https://doi.org/10.1109/TCYB.2016.2638902
33. Cheng, M.Y., Prayogo, D.: Symbiotic organisms search: A new metaheuristic optimization algorithm. Comput. Struct. **139**, 98–112 (2014). https://doi.org/10.1016/j.compstruc.2014.03.007
34. 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
35. 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
36. 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
37. 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. Mater. Res. Express. **4**(3), 035302 (2017). https://doi.org/10.1088/2053-1591/aa5f6a
38. Chen, Y.-S.: Performance enhancement of multiband antennas through a two-stage optimization technique. Int. J. RF Microwave Comput. Aided Eng. **27**(2) (2017). article e21064 https://doi.org/10.1002/mmce.21064
39. 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
40. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving multi-objective Problems. Genetic And Evolutionary Computation Series, 2nd edn. Springer (2007)
41. Gunji, A.B., Deepak, B.B.B.V.L., Bahubalendruni, C.R., Biswal, D.B.B.: An optimal robotic assembly sequence planning by assembly subsets detection method using teaching learning-based optimization algorithm. IEEE Trans. Autom. Sci. Eng. **15**(3), 1369–1385 (2018)
42. Bahubalendruni, M.R., Deepak, B.B.V.L., Biswal, B.B.: An advanced immune based strategy to obtain an optimal feasible assembly sequence. Assembly Autom. **36**(2), 127–137 (2016)
43. Eswaran, M., kumar Inkulu, A., Tamilarasan, K., Bahubalendruni, M.R., Jaideep, R., Faris, M.S., Jacob, N.: Optimal layout planning for human robot collaborative assembly systems and visualization through immersive technologies. Expert Syst. Appl. **241**, 122465 (2024)
44. Champatiray, C., Bahubalendruni, M.R., Mahanta, G.B., Pham, T., D., Mahapatra, R.N.: Enhancing efficiency and accuracy in robotic assembly task planning through tool integration using a hybrid class topper optimisation algorithm. In: Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 09544062241264708 (2024)

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

45. Gulivindala, A.K., Bahubalendruni, M.V.A.R., Chandrasekar, R., Ahmed, E., Abidi, M.H., Al-Ahmari, A.: Automated disassembly sequence prediction for industry 4.0 using enhanced genetic algorithm. Computers Mater. Continua. **69**(2), 2531–2548 (2021)

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

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

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com