J. Inst. Eng. India Ser. C
https://doi.org/10.1007/s40032-024-01104-5

ORIGINAL CONTRIBUTION

Many-Objective Multi-Verse Optimizer (MaOMVO): A Novel Algorithm for Solving Complex Many-Objective Engineering Problems

Kanak Kalita[1,2] · Pradeep Jangir[3,4] · Sundaram B. Pandya[5] · G. Shanmugasundar[6] · Jasgurpreet Singh Chohan[7] · Laith Abualigah[8,9,10,11,12,13]

Received: 23 March 2024 / Accepted: 2 October 2024

Abstract Research has demonstrated the effectiveness of metaheuristic algorithms in addressing optimization challenges involving multi-objectives. These algorithms excel in generating a range of solutions that not only approach the Pareto front but are also evenly distributed. Multi-objective algorithms are effective for problems with two or three objectives. However, their efficiency diminishes in scenarios with many objectives, as the diversity selection and convergence pressure become less effective. Balancing convergence and diversity in multi-objective optimization pose a significant challenge. In response, this study introduces a novel many-objective multi-verse optimizer algorithm named MaOMVO for addressing many-objective problems. It integrates reference point and niche preserve to improve convergence and diversity and employs an innovative information feedback mechanism technique for population renewal. The superiority of the MaOMVO algorithm is evident in tests with MaF problems having 5, 8 and 15 objectives, as well as five real-world problems (RWMaOP1—RWMaOP5). It outperforms four leading algorithms Many-objective moth flame optimization, many-objective particle swarm optimization, non-dominated sorting genetic algorithm-III and reference vector guided evolutionary algorithm in terms of generational distance by 70%, inverted generational distance by 52%, spacing by 46.66%, spread by 55.55%, hypervolume by 52% and running time by 52% with concave, convex and mixed pareto fronts, confirming its robustness in diverse optimization scenarios.

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

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

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

[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, Jordan

[5] Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392001, India

[6] Department of Mechanical Engineering, Sri Sairam Institute of Technology, Chennai 600044, India

[7] Department of Mechanical Engineering and University Centre for Research and Development, Chandigarh University, Mohali 140413, India

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

[9] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, India

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

[11] MEU Research Unit, Middle East University, Amman 11831, Jordan

[12] Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

[13] School of Engineering and Technology, Sunway University Malaysia, 27500 Petaling Jaya, Malaysia

Published online: 21 October 2024

Keywords Convergence · Diversity · Many-objective optimization · Metaheuristic algorithm · Many-Objective Multi-Verse Optimizer · MaOMVO · Reference point

Abbreviations

GD	Generational distance
IGD	Inverted generational distance
SP	Spread
SD	Standard deviation
HV	Hypervolume
RT	Runtime
MaOMVO	Many-objective multi-verse optimizer
MaOMFO	Many-objective moth flame optimization
MaOPSO	Many-objective particle swarm optimization
NSGA-III	Non-dominated sorting genetic algorithm-iii
RVEA	Reference vector guided evolutionary algorithm
IFM	Information feedback mechanism
MaOPs	Many-objective problems
POF	Pareto-optimal front
MOEA/DD	Multi-objective evolutionary algorithm based on decomposition based dominance
K-RVEA	Kriging reference vector guided evolutionary algorithm
I-DEBA	Indicator-based evolutionary algorithm
HypE	Hypervolume estimation algorithm
VaEA	Vector angle based evolutionary algorithm
θ-DEA	θ-dominance-based evolutionary algorithm
MPSO/D	Multi-objective particle swarm optimization based on decomposition
WS	Weighted sum method
TSM	Tchebycheff method
PBI	Penalty-based boundary interaction
DoD	Decomposition based dominance

Introduction

The domain of multi-objective optimization, particularly with an emphasis on evolutionary and swarm intelligence algorithms, has garnered significant interest from researchers. The focus is on crafting algorithms that are both powerful and capable, aimed at addressing issues of varying complexity. These developments predominantly leverage frameworks like Pareto-ranking [1], decomposition strategies [2], or methods based on specific indicators [3]. The core goal of these algorithms is to effectively tackle the optimization challenge as presented in Eq. (1). This equation becomes particularly relevant when dealing with more than three objectives.

Minimize subject to:

$$\mathrm{f}(x) = (f_1(x), \cdots, f_M(\mathrm{x})), \mathrm{x} \in \Omega, \tag{1}$$

where $\mathrm{f}(\mathrm{x})$ represents the goal function, $\mathrm{x} \in \Omega$ denotes the array of decision-making variables and Ω symbolizes the viable scope for exploration. Given the presence of numerous optimal solutions to Eq. (1), the Pareto Dominance concept is employed for their evaluation and comparison. The group of solutions that are optimal according to Pareto criteria forms the Pareto-Optimal (PO) set shown in Fig. 1.

In practical scenarios, it common to encounter problems with multiple objectives. Some examples are—optimizing the design of a bulldozer blade for soil cutting [4], structural topology optimization [5–7] and managing water resources [8], etc. Such complex problems necessitate the development of an algorithm capable of producing solutions that are either close to or directly on the PO frontier.

Many-objective evolutionary and swarm algorithms based on Pareto-ranking address the issue outlined in Eq. (1). They commence by assigning a Pareto rank to each solution, followed by the application of a mechanism that preserves diversity. A significant challenge with these algorithms is their diminished selection pressure in many-objective optimization scenarios, where most solutions are non-dominated [9]. In such cases, Pareto-ranking fails to effectively differentiate between solutions and selection hinges solely on the diversity preservation method. This often results in subpar convergence and diversity among solutions on the PO front [10]. To address this, researchers have introduced modified Pareto-ranking methods [11], including fuzzy $\alpha-$ dominance [12] and others. However, these modified approaches still struggle to produce a diverse set of solutions on the PO front.

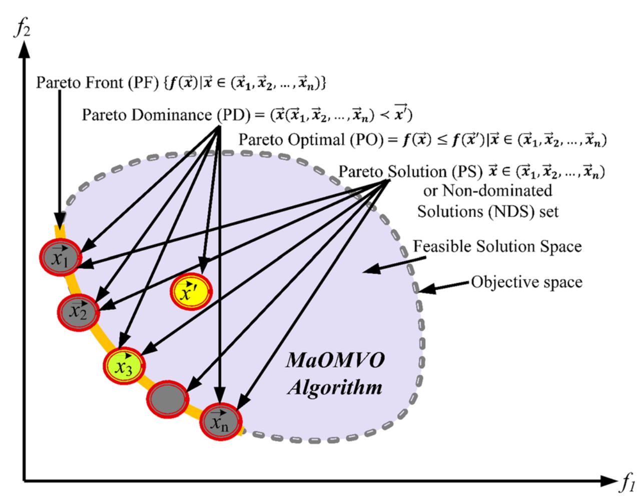


Fig. 1 Many-Objective all definitions in search space of MaO-Problem

Consequently, the focus has shifted towards devising more effective environmental selection methods. One successful approach involves a reference-lines-based framework, where solutions are selected via a series of reference lines or vectors. NSGA-III [1] employs this approach, introducing niche formation through solution association with reference lines. An effective environmental selection method is derived by favoring solutions from less populated areas. Building on this concept, algorithms such as MOEA/DD [13], VaEA [14] and various many-objective PSO approach [15], Many-Objective Moth Flame Optimization (MaOMFO) [16], Kriging Reference Vector Guided Evolutionary Algorithm (K-RVEA) [17].

Decomposition-based algorithms for many-objective optimization break down the problem outlined in Eq. (1) into numerous scalar optimization tasks, each addressing a single objective. These sub-tasks are crafted using combined functions and tackled concurrently. According to a study [18], one primary challenge in this approach is the creation of weight vectors that effectively aid in producing a varied range of solutions on the PO front. A further issue is the choice of combined functions, like the Weighted Sum (WS) method, Tchebycheff (TS) method and Penalty-based boundary interaction (PBI) method, which often struggle to accurately represent the full scope of the PO front. To overcome these hurdles, MOEA/DD [13] integrates dominance- and decomposition-based techniques. Conversely, REVA [19] implements an angle-penalized distance approach within a decomposed objective space, using reference lines and selecting a single solution from each sub-group. I-DEBA [20] improves θ upon the PBI method by prioritizing certain distances in the objective space. $\theta-$DEA [21] introduces a novel dominance relation utilizing the PBI method. These represent some of the decomposition-based algorithms that have been effective in generating a diverse array of solutions on the PO front.

In the realm of indicator-based algorithms, fitness is assigned to each solution based on specific indicators, aiming for convergence on the PO front while ensuring solution diversity. The hypervolume indicator is particularly prevalent for its effectiveness in many-objective optimization challenges. Its major drawback, however, is the significant computational time required, which grows exponentially with the increase in objectives. HypE [3] addresses this issue by calculating hypervolume through Monte-Carlo simulations. Other indicators, the unary epsilon indicator [22], have also seen application. Despite their strengths, these algorithms occasionally fall short in generating a well-diversified set of solutions on the PO front.

Decomposition-based and indicator-based algorithms typically allocate a combined fitness value to each solution in a population or swarm, aiming to achieve both convergence and diversity at the same time. On the other hand, Pareto-based algorithms assign a rank to each solution for convergence, followed by the application of a method to preserve diversity. Nevertheless, there exists a different category of algorithms that prioritize diversity before dominance. For instance, $\theta-$DEA [21], MPSO/D [23] and DoD [24] are examples of such algorithms. These methods create groups of solutions using a framework based on reference lines and then select a single solution from each group, employing either Pareto-ranking or a novel dominance relationship.

In the domain of many-objective optimization, various algorithms have been proposed, each with different strategies to handle the increasing computational demands. Many-Objective Evolutionary Algorithms (MOEAs) like NSGA-III [1] and RVEA [19] have introduced mechanisms such as reference points and angle-based selections to improve diversity, but they face challenges in computational efficiency, particularly with high-dimensional objectives. Swarm intelligence-based algorithms, including Many-Objective Particle Swarm Optimization (MaOPSO) [15] and MaOMFO [16], incorporate Pareto dominance and diversity preservation techniques, leading to increased computational time due to additional dominance checks. Decomposition-based approaches like MOEA/DD [13] decompose problems into scalar optimization sub-problems, yet the generation and management of weight vectors add to the computational overhead. Indicator-based algorithms such as HypE [3] utilize hypervolume to balance convergence and diversity, but their computational costs escalate with the number of objectives. Proposed MaOMVO algorithm distinguishes itself by integrating several unique strategies to reduce computational time while maintaining solution quality. The Information Feedback Mechanism (IFM) leverages historical data to enhance convergence, reducing the need for repeated dominance checks. The Reference Point-Based Selection strategy ensures well-distributed solutions across the objective space, minimizing the computational burden of maintaining diversity.

MVO is renowned for its rapid convergence capabilities, but ensuring diversity among the multi-verse in a universe remains a significant challenge in multi- and many-objective optimization contexts. Research indicates that the choice of verse in multi-objective MVO is pivotal, as it is essential to focus on non-dominated solutions while also preserving diversity among them.

In this research, a new strategy is introduced to enhance the balance between convergence and diversity in many-objective optimization by employing the Multi-Verse Optimizer (MVO) [25], along with innovative components such as an Information Feedback Mechanism, Reference Point-Based Selection and Association, Non-dominated

Sorting, Niche Preservation and Density Estimation in the Many-Objective Multi-Verse Optimizer (MaOMVO). The methodology involves selecting the MVO algorithm based on its efficacy in generating diverse, high-quality solutions for single-objective problems, thereby improving the MaOMVO ability to navigate and utilize the search space efficiently. An IFM addresses previous inefficiencies by preserving aggregated historical data of individuals through a weighted sum approach, enhancing convergence capabilities. The adoption of a reference point-based selection strategy ensures the selection of solutions that not only approach the optimal front but are also distributed across the entire objective spectrum, with solutions associated with the nearest reference point by perpendicular distance, marking well-represented areas. Furthermore, a niche preservation tactic is proposed for individuals at the boundaries of the search space, aimed at increasing diversity and mitigating congestion in dense regions, thus boosting the algorithm convergence rate. The inclusion of a density estimation approach ensures a well-distributed and extensive coverage of the population. The effectiveness of the newly developed MaOMVO is validated through comparisons with MaOMFO, MaOPSO, NSGA-III and RVEA algorithms across MaF1-MaF15 benchmark sets with 5, 8 and 15 objectives and five real-world (RWMaOP1- RWMaOP5) problems. The results from these experiments highlight MaOMVO capability to adeptly manage various problem types, underscoring its robust overall performance.

The paper organization includes an overview of MVO algorithm in Sect. "Multi-Verse Optimizer", a presentation of the proposed MaOMVO algorithm in Sect. "Proposed Many-Objective Multi-Verse Optimizer (MaOMVO)", experimental comparisons and evaluations in Sect. "Results and Discussions". Finally, Sect. "Conclusions" summarizes the findings and suggests avenues for future research.

Multi-Verse Optimizer

The MVO [25] is an algorithm inspired by the cosmological multi-verse theory, incorporating elements like white holes, black holes and wormholes shown in Fig. 2. This cosmological perspective views white and black holes as distinct astronomical entities with contrasting characteristics: white holes being outlets for matter and energy and black holes being their absorptive counterparts. Wormholes serve as conduits linking parallel universes, enabling instantaneous travel between different realms of time and space.

MVO uses these astrophysical concepts in a metaphorical sense shown in Fig. 3. Here, white and black holes symbolize the exploration and exploitation processes within a search space, while wormholes represent paths for optimal solutions across multiple universes. The algorithm introduces an 'expansion rate' to reflect the dynamic and evolving nature of the universe. This rate influences the behavior of objects in the multi-verse, guiding them towards stability through interaction with the white holes, black holes and wormholes.

The following principles guide the MVO algorithm (as per reference [25]):

- A higher expansion rate increases the likelihood of a white hole and decreases that of a black hole.
- Universes with greater expansion rates dispatch matter through white holes.
- Conversely, universes with lower expansion rates tend to acquire matter via black holes.
- Independently of the expansion rate, all entities can traverse wormholes, moving randomly towards the most optimal universe.

MVO process begins by creating various random universes. In each cycle, entities transition between universes based on their expansion rates, using white and black holes. Additionally, entities in any universe might be teleported to the most favorable universe through wormholes. This procedure repeats until specific criteria are satisfied.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} \tag{2}$$

MVO universe generation is described by Eq. (2). Here, 'n' represents the number of universes, each signifying a potential solution. 'd' denotes the matter content in each universe, symbolizing solution parameters. Universe updates follow Eq. (3):

$$XX_i^j = \begin{cases} XX_k^j, & r_1 < NI(XX_i) \\ XX_i^j, & r_1 \geq NI(XX_i) \end{cases} \tag{3}$$

where 'XX_i^j' is the $i-th$ universe $j-th$ matter, '$NI(XX_i)$' its normalized expansion rate and 'XX_k^j' the $k-th$ universe $j-th$ matter. A roulette mechanism, influenced by the expansion rate, determines white hole selection. Additionally, wormholes can alter universe contents at random, as shown in Eq. (4).

$$XX_i^j = \begin{cases} \begin{cases} XX_{best}^j + TDR \times ((ub - lb) \times r_4 + lb), & r_3 < 0.5 \\ XX_{best}^j - TDR \times ((ub - lb) \times r_4 + lb), & r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ XX_i^j, & r_2 \geq WEP \end{cases} \tag{4}$$

where r_2, r_3, r_4 are random numbers in the interval $[0,1]$. In the context of this model, the Travel Distance Rate (TDR)

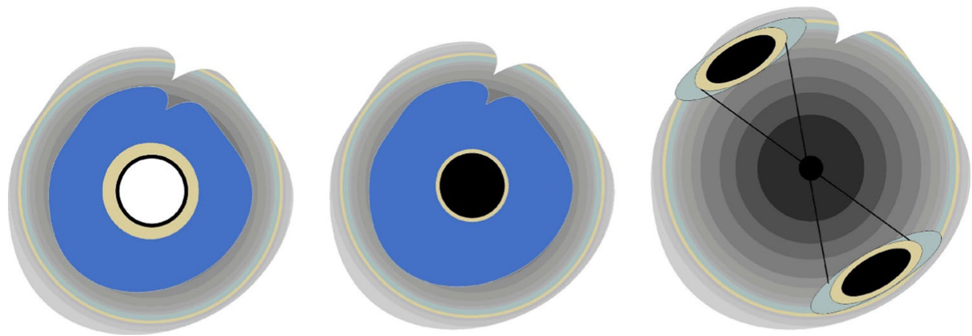


Fig. 2 A graphic overview of white holes, black holes and wormholes

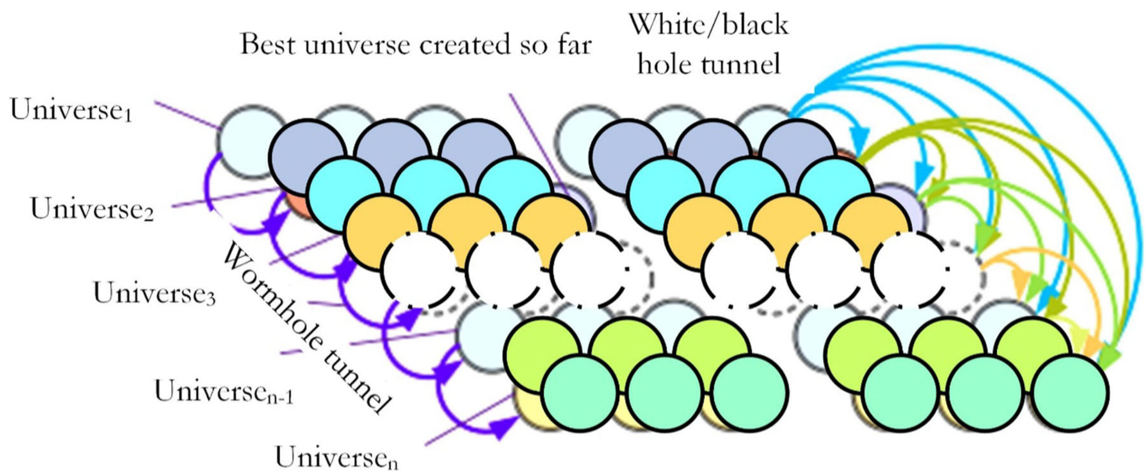


Fig. 3 MVO conceptual model

is a crucial parameter. It determines the extent to which an object can be instantaneously moved via a wormhole to the most favorable universe identified thus far. This rate of travel distance escalates progressively with each iteration. In parallel, the Wormhole Existence Probability (*WEP*) is another key metric. It quantifies the likelihood of wormholes being present within the universe, a probability that systematically increases with each iteration. The evolutionary patterns of both *TDR* and *WEP* are encapsulated in the following mathematical formulations:

$$WEP = 0.2 + l \times \left(\frac{0.8}{L}\right) \tag{5}$$

$$TDR = 1 - \left(\frac{l^{1/p}}{L^{1/p}}\right) \tag{6}$$

In Eq. (5) and Eq. (6) the algorithm efficiency depends on iteration-based precision development (p), with higher values ensuring quicker and more precise local searches and L represents the maximum iteration number.

Proposed Many-Objective Multi-Verse Optimizer (MaOMVO)

The MaOMVO algorithm begins by creating an initial population of N random solutions, M number of objectives, p number of partitions and generate a set of reference points by using Das and Dennis's method $H = \binom{M+p-1}{p}$, 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 MVO 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, we can calculate x_i^{t+1} according to u_i^{t+1} generated through the MVO algorithm and IFM in Eq. (7)

$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t; \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t}, \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t}, \partial_1 + \partial_2 = 1 \tag{7}$$

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 $(F_1, F_2, \ldots, F_l \ldots, F_w)$. 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 selection from the last front, F_l (described in Algorithm 1), incorporates a niche-preserving operator. Each member of P_{t+1} and F_l is normalized (as outlined in Algorithm 2) according to the current population spread to ensure uniformity in objective vectors and reference points. Subsequently, each member is linked to a specific reference point by the shortest perpendicular distance $(d())$ (introduced in Algorithm 3), creating a reference line from the origin to a designated reference point. A strategic niching approach (explained in Algorithm 5) is then applied to select members of F_l linked to under-represented reference points, with niche count ρ_i evaluated in P_{t+1}. Should the termination condition remain unmet, the process repeats otherwise, a new generation P_{t+1} is created and utilized to produce a subsequent population Q_{t+1}. This selection method introduces a computational complexity scaled as $\left(N^2 \log^{M-2} N\right)$ or $O\left(N^2 M\right)$.

Algorithm 1 Generation t of MaOMVO algorithm with IFM Procedure

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

Output: Q_{t+1} = **MVO(** 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)**
MVO 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: $R_t = P_t \cup Q_t$

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

5: **repeat**

6: $S_t = S_t \cup F_i$ and $i = i+1$

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

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

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:
$[\boldsymbol{\pi}(\boldsymbol{s}), \boldsymbol{d}(\boldsymbol{s})] = \boldsymbol{Associate}\ (\boldsymbol{S_t}, \boldsymbol{Z^r})$**;** Brief Explanation in **Algorithm-3**
% $\pi(s)$: $closest\ reference\ point, d$: $distance\ between\ s\ and\ \pi(s)$

16: Compute niche count of reference point $j \in Z^r$:
$\boldsymbol{\rho_j} = \sum_{s \in s_t/F_l}((\boldsymbol{\pi}(\boldsymbol{s}) = \boldsymbol{j})\ , \mathbf{1} : \mathbf{0})$**;**

17: Choose K members one at a time F_l to construct
$\boldsymbol{P_{t+1}}$: $\boldsymbol{Niching}(\boldsymbol{K}, \boldsymbol{\rho_j}, \boldsymbol{\pi}, \boldsymbol{d}, \boldsymbol{Z^r}, \boldsymbol{F_l}, \boldsymbol{P_{t+1}})$ **; Represent in Algorithm-4**

18: **end if**

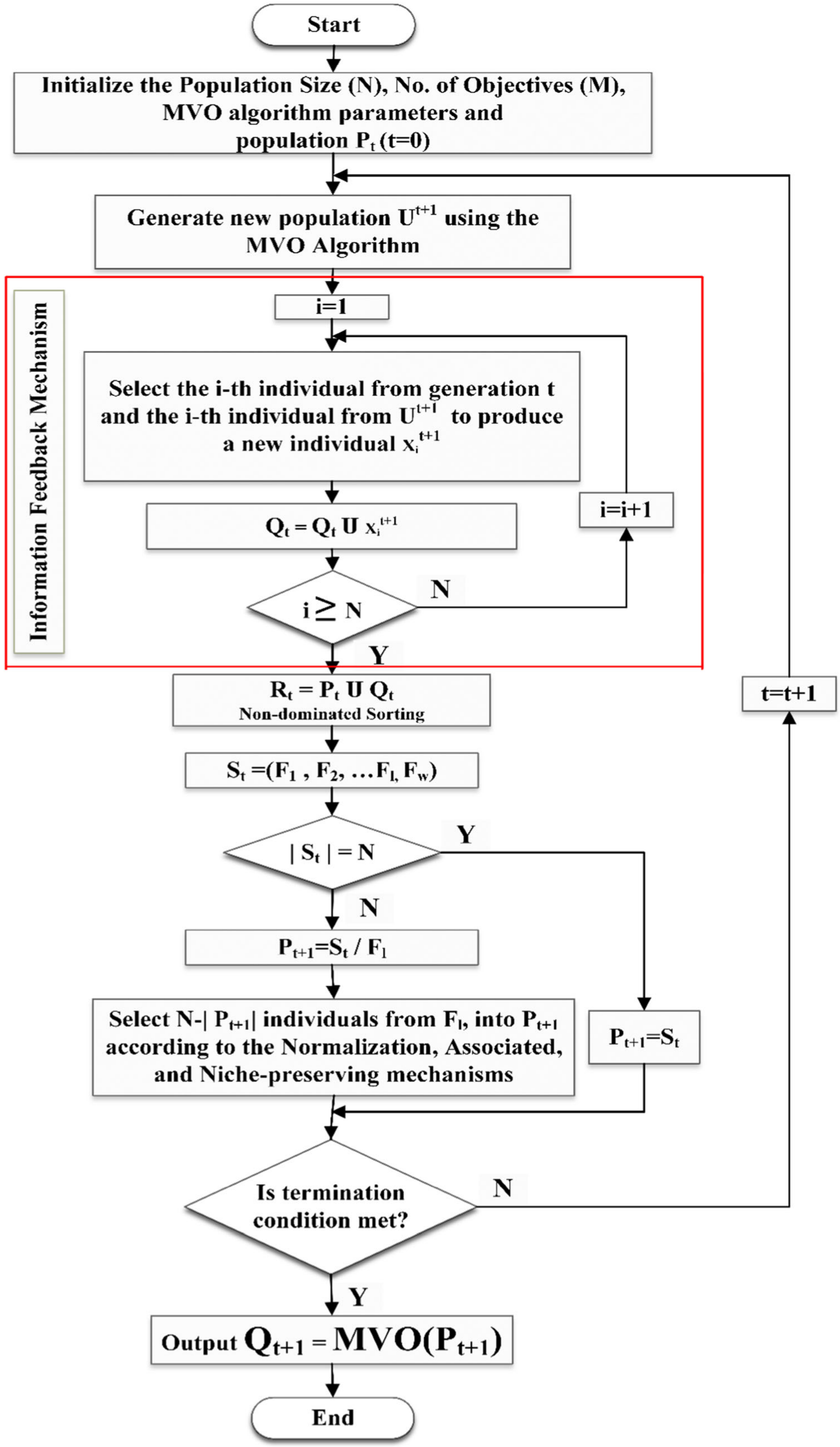
Algorithm 2 Normalize (f_n, S_t, Z_r/Z_a) procedure

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: $f_j^{'}(s) = f_j(s) - Z_j^{min}\ \forall s \in S_t$
4:	Compute extreme points: $Z^{j,max} = s$: argmin$_{s \in s_t}$ $ASF(s, w^j) = where\ w^j = (\epsilon 1, \ldots\ldots, \epsilon j)^T)$, $\epsilon = 10^{-6}, and\ w_j^j = 1$
5:	**end for**
6:	Compute intercepts a_j for j= 1, .., M
7:	Normalize objectives $f_i^n(X)$ using $f_i^n(X) = \frac{f_i^{'}(X)}{a_i - Z_i^{min}}$ $, for\ i = 1, 2, \ldots., M$
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** $d^{\perp}(s, w) = s - w^T s / \parallel w \parallel$
7:	**end for**
8:	Assign $\pi(s) = w$: $argmin_{W \in Z^r}\ d^{\perp}(s, w)$
9:	Assign $d(s) = d^{\perp}(s, \pi(s))$
10:	**end for**

Fig. 4 Flowchart of MaOMVO algorithm



Algorithm 4 Niching (K, ρ_j, π, d, Z_r, F_l, P_{t+1})

The flow chart of MaOMVO algorithm can be shown in Fig. 4.

Input: $K, \rho_j, \pi(s \in S_t), d(s \in S_t), Z^r, F_l,$

Output: P_{t+1}

```
k = 1
while k ≤ K do
    J_min = {j : argmin_{j ∈ Z^r} ρ_j}
    j̄ = random (J_min)
    I_j̄ = {s : π(s) = j̄, s ∈ F_l}
    if I_j̄ ≠ ϕ then
        if ρ_j̄ = 0 then
            P_{t+1} = P_{t+1} ∪ (s : argmin_{s ∈ I_j̄} d_s)
        else
            P_{t+1} = P_{t+1} ∪ random (I_j̄)
        end if
        ρ_j̄ = ρ_j̄ + 1, F_l = F_l/s
        k = k+1
    else
        Z^r = Z^r/{j̄}
    end if
end while
```

Results and Discussions

Experimental Settings

Benchmarks

In order to verify the effectiveness of the MaOMVO, the MaF1- MaF15 [26] benchmark and five real world engineering design (Appendix 1): Car cab design (RWMaOP1) [27], 10-bar truss structure (RWMaOP2) [28], Water and oil repellent fabric development (RWMaOP3) [29], Ultra-wideband antenna design (RWMaOP4) [30] and Liquid-rocket single element injector design (RWMaOP5) [31] problems are used in this paper. These problems were selected due to their diverse objective structures, realistic engineering constraints, complex decision variables, industrial relevance and variety of Pareto front shapes. Each problem has been extensively studied, providing benchmark data for rigorous comparison. The Car Cab Design problem involves optimizing safety and performance constraints; the 10-Bar Truss Structure problem focuses on structural optimization under stress and buckling constraints; the Water and Oil Repellent Fabric Development problem optimizes fabric properties; the Ultra-Wideband Antenna Design problem aims to enhance communication system parameters; and the Liquid-Rocket Single Element Injector Design problem addresses thermal and structural performance in aerospace applications. This selection ensures a comprehensive evaluation of MaOMVO capabilities, demonstrating its robustness, flexibility and practical utility in addressing complex many-objective optimization problems. The number of decision variables for the MaF problems is $k + M - 1$, M is the number of objective functions. k is set to 10 in MaF1-MaF6, k is set to 20 in MaF7-MaF15. The MaFs are specifically designed to tackle various types of Pareto fronts (PF), including concave, convex and mixed PFs and are capable of handling up to 15 objectives, whether they involve minimization or maximization tasks. These algorithms have been tested and proven effective on the MaF test problems, which encompass a wide range of properties and challenges. For instance, MaF1 features a linear PF, while MaF2 presents a concave PF. Other test problems such as MaF3 and MaF4 exhibit convex, multimodal and

Table 1 Properties of the quality indicators

Quality indicator[32]	Convergence	Diversity	Uniformity	Cardinality	Computational Burden
GD	✔				
SD		✔			
SP			✔		
RT					✔
IGD	✔	✔	✔		
HV	✔	✔	✔	✔	

concave, multimodal PFs, respectively. Some problems like MaF5 and MaF10 incorporate biases, whereas MaF6 and MaF8 deal with degenerate PFs. Additionally, the algorithms can address mixed, disconnected and multimodal PFs as seen in MaF7 and they effectively handle non-separable and deceptive PFs like those in MaF12. Large-scale problems, both linear and convex, are represented by MaF14 and MaF15, respectively.

Comparison Algorithms and Parameter Settings

In this study, the performance of MaOMVO by empirically comparing it with some state-of-the-art MOAs for MaOPs, namely, MaOPSO [15], MaOMFO [16], RVEA [19] and NSGA-III [1], will be verified.

Performance Measures

This paper adopts Generational distance (*GD*), Spread (*SD*), Spacing (*SP*), Run Time (*RT*), Inverse Generational distance (*IGD*) and Hypervolume (*HV*) quality indicator [32], shown in Table 1 and Fig. 5. GD measures how far the solutions obtained by an algorithm are from the true Pareto front, IGD evaluates the distance between the true Pareto front and the solutions obtained by the algorithm, SP measures the uniformity of the distribution of solutions along the Pareto front, SD assesses the extent to which the solutions obtained by the algorithm spread along the objectives, HV calculates the volume of the objective space dominated by the solutions obtained by the algorithm and RT measures the computational efficiency of the algorithm. GD and IGD are complementary metrics that provide insights into the proximity of the solutions to the true Pareto front and the quality of the approximation of the Pareto front, respectively. SP and SD ensure that the solutions are not only close to the true Pareto front but also well-distributed and spread across the objective space, preventing clustering and ensuring diversity. HV is a comprehensive metric that combines aspects of convergence and diversity, providing a single measure to evaluate the overall performance of the algorithm. RT is crucial for practical applications, ensuring that the algorithm is not only effective but also efficient in terms of computational resources.

Experimental Results on MaF Problems

Table 2 presents the Generational Distance (GD) metrics for various algorithms, including MaOMVO, MaOMFO, MaOPSO, NSGA-III and RVEA, across different problems (MaF1 to MaF15) with varying dimensions (M). The GD values are expressed in mean (std) format. In the context of MaF1, MaOMVO demonstrates superior efficiency, particularly in lower-dimensional problems (M = 5 and M = 8), with its performance marginally declining as the problem dimension increases (M = 15). This trend is notable in comparison to MaOMFO and MaOPSO, where MaOMVO consistently outperforms these algorithms in terms of lower mean GD values. For MaF2 and MaF3 problems, MaOMVO maintains a competitive edge, particularly in lower dimensions, suggesting its robustness in handling complex optimization landscapes. However, in higher dimensions (M = 15 for MaF3), its performance is overshadowed by the NSGA-III algorithm, indicating possible limitations in scaling. In problems MaF4 to MaF6, MaOMVO exhibits a mixed performance. While it shows promising results in MaF4, especially in lower dimensions, its efficiency diminishes in MaF5 and MaF6 as the problem complexity escalates. From Table 2, we can observe that MaOMVO outperforms 32 out of 45 best results, whereas MaOMFO, MaOPSO, NSGA-III and RVEA achieves 2, 0, 10 and 1 best results in terms of the GD values, respectively. Despite these limitations, MaOMVO establishes itself as a competitive algorithm, particularly in simpler problem landscapes shown in Fig. 6. This analysis shows the importance of algorithmic adaptability and scalability in diverse optimization scenarios.

Table 3, data reveals that the proportions of test problems where MaOMVO, MaOMFO, MaOPSO, NSGA-III and RVEA better among the 45 DTLZ test problems are 48.88%, 13.33%, 20%, % and 8.88%, highlighting its effectiveness in maintaining a balance between

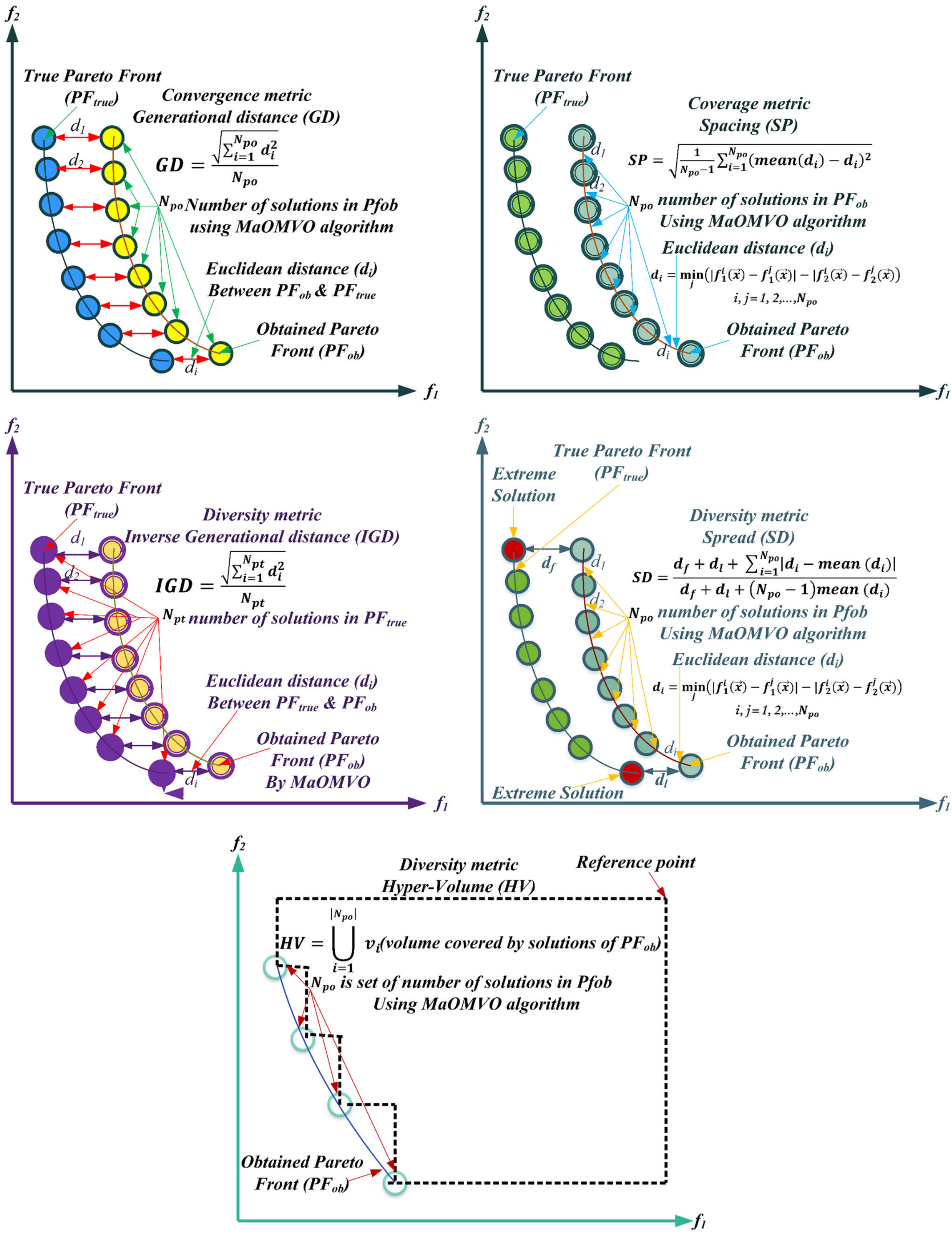


Fig. 5 Mathematical and Schematic view of the GD, IGD, SP, SD and HV metrics

Table 2 GD metric of various algorithms on MaF problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA-III	RVEA
MaF1	5	14	3.4947e − 3 (4.88e − 4) =	5.1185e − 3 (6.70e − 4) =	5.9541e − 3 (8.29e − 4) =	5.6249e − 3 (1.41e − 4) =	6.7280e − 3 (1.03e − 3)
	8	17	7.0493e − 3 (2.47e − 4) =	1.2041e − 2 (8.64e − 4) =	1.1552e − 2 (5.99e − 4) =	1.8387e − 2 (4.50e − 3) =	1.7389e − 2 (7.08e − 3)
	15	24	5.0294e − 3 (8.07e − 4) =	3.0512e − 2 (1.47e − 3) =	3.1497e − 2 (3.48e − 3) =	3.8056e − 2 (3.64e − 3) =	1.3669e − 2 (8.31e − 3)
MaF2	5	14	1.1241e − 2 (3.97e − 4) =	1.3412e − 2 (1.15e − 4) =	1.2833e − 2 (4.19e − 4) =	9.7805e − 3 (1.59e − 4) =	1.4007e − 2 (4.30e − 4)
	8	17	8.1944e − 3 (7.87e − 4) =	1.3096e − 2 (1.03e − 3) =	1.2143e − 2 (9.45e − 4) =	1.3163e − 2 (1.21e − 3) =	1.1820e − 2 (3.03e − 4)
	15	24	1.7390e − 2 (9.77e − 4) =	3.2674e − 2 (3.96e − 4) =	3.1745e − 2 (1.15e − 3) =	4.6386e − 2 (6.67e − 3) =	3.3806e − 2 (1.57e − 3)
MaF3	5	14	1.1552e + 5 (7.87e + 4) =	2.2193e + 8 (2.14e + 8) =	3.6805e + 7 (3.65e + 7) =	9.0528e + 6 (1.55e + 7) =	1.9668e + 5 (1.70e + 5)
	8	17	3.7829e + 6 (6.40e + 6) =	3.8498e + 9 (2.40e + 9) =	1.0934e + 10 (3.76e + 9) =	1.7992e + 8 (3.11e + 8) =	1.0859e + 12 (4.30e + 10)
	15	24	2.8173e + 2 (4.16e + 2) =	1.4810e + 9 (5.64e + 8) =	1.3863e + 9 (1.78e + 9) =	2.2381e + 3 (1.94e + 3) =	2.3141e + 12 (1.62e + 11)
MaF4	5	14	3.4717e + 1 (1.04e + 1) =	6.5554e + 1 (4.82) =	2.6396e + 1 (2.06e + 1) =	3.7814e + 1 (1.30e + 1) =	1.1909e + 2 (1.28e + 2)
	8	17	1.2783e + 2 (6.24e + 1) =	2.7577e + 2 (1.29e + 2) =	3.4087e + 2 (1.13e + 2) =	2.6163e + 2 (5.47) =	6.5127e + 2 (4.05e + 2)
	15	24	1.1061e + 4 (1.79e + 3) =	4.3414e + 4 (2.41e + 4) =	2.0920e + 4 (1.03e + 4) =	7.0253e + 4 (9.30e + 4) =	7.3075e + 4 (3.15e + 4)
MaF5	5	14	4.5255e − 2 (3.05e − 3) =	5.6319e − 2 (2.16e − 3) =	5.1787e − 2 (4.60e − 3) =	7.0912e − 2 (8.07e − 3) =	1.6329e − 1 (5.33e − 2)
	8	17	2.4567e − 1 (7.88e − 2) =	1.2291 (1.25e − 1) =	1.1757 (1.10e − 1) =	1.2472 (1.51e − 1) =	3.1519e + 1 (1.07)
	15	24	1.7385e + 1 (1.20e + 1) =	1.5956e + 2 (7.61e + 1) =	1.1703e + 2 (2.18e + 1) =	6.8274e + 1 (4.35e + 1) =	7.3411e + 3 (1.43e + 2)
MaF6	5	14	2.0055e − 4 (7.62e − 5) =	2.0380e − − 4 (3.53e − 5) =	2.3885e − − 4 (1.26e-4) =	4.4651e − 5 (1.61e − 5) =	1.6966e − 4 (3.43e − 5)
	8	17	1.6183e − 4 (1.03e − 4) =	5.2248 (9.05) =	5.3591 (9.28) =	1.4937e − 4 (1.89e − 4) =	1.2943e + 1 (1.13e + 1)
	15	24	1.9008e + 1 (5.83) =	1.9085e + 1 (4.52) =	1.4663e + 1 (1.22) =	7.9183e − 5 (3.05e − 5) =	4.4581e + 1 (6.22e − 1)
MaF7	5	24	2.9822e − 2 (1.01e − 2) =	4.8692e − 2 (2.11e − 2) =	5.0739e − 2 (1.77e − 2) =	5.3791e − 2 (2.10e − 2) =	4.6110e − 2 (1.05e − 2)
	8	27	1.4866e − 1 (6.36e − 2) =	3.0374e − 1 (9.51e − 2) =	3.6394e − 1 (1.25e − 1) =	2.8930e − 1 (6.08e − 2) =	2.8872 (1.13e − 1)
	15	34	2.7139e − 1 (1.89e − 1) =	1.0162 (8.76e − 1) =	4.3340e − 1 (1.80e − 1) =	2.6131e − 1 (6.50e − 2) =	1.1104e + 1 (7.57e − 1)
MaF8	5	2	8.5101e − 3 (6.19e − 3) =	3.3626e − 2 (6.58e − 3) =	7.9748e − 2 (4.01e − 2) =	6.0318e − 2 (8.58e − 2) =	2.8458e − 2 (1.67e − 2)
	8	2	2.5581e − 2 (1.77e − 2) =	6.1294e − 2 (5.33e − 2) =	3.1052e − 2 (2.52e − 2) =	2.7371e − 1 (4.38e − 1) =	1.7744e − 2 (5.41e − 3)
	15	2	1.0702e − 2 (7.40e − 3) =	3.3956e − 2 (3.15e − 3) =	5.3441e − 2 (2.43e − 2) =	2.5388e − 2 (5.20e − 3) =	1.4069e − 1 (1.63e − 1)
MaF9	5	2	1.4395e + 1 (1.71e + 1) =	1.2153e + 1 (1.01e + 1) =	2.1572e + 1 (1.83e + 1) =	1.0852e + 1 (1.35e + 1) =	1.5699e + 1 (1.97e + 1)
	8	2	1.4886 (5.58e − 1) =	7.9699e + 1 (4.02e + 1) =	2.2580e + 2 (1.12e + 2) =	1.8096 (3.72e − 1) =	2.4549e + 2 (1.53e + 2)
	15	2	2.2157 (1.24) =	1.2168e + 2 (9.70e + 1) =	9.5237e + 2 (1.57e + 3) =	1.9462e + 1 (7.50) =	4.8599e + 2 (9.79e + 1)

Table 2 continued

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA-III	RVEA
MaF10	5	14	1.0761e − 1 (2.14e − 2) =	1.0804e − 1 (6.01e − 3) =	1.2747e − 1 (1.32e − 2) =	1.2649e − 1 (2.47e − 2) =	1.3926e − 1 (3.99e-3)
	8	17	1.6116e − 1 (8.24e − 3) =	1.9556e − 1 (5.28e − 2) =	1.8956e − 1 (2.30e − 2) =	2.3489e − 1 (4.06e-3) =	2.3896e − 1 (2.94e − 2)
	15	24	1.8998e − 1 (4.73e − 2) =	4.6083e − 1 (5.47e − 2) =	3.8335e − 1 (5.66e − 2) =	4.4215e-1 (4.80e − 2) =	3.8273e − 1 (3.61e − 2)
MaF11	5	14	2.4411e-2 (5.28e − 3) =	2.3366e − 2 (2.54e − 3) =	2.8809e − 2 (2.70e-3) =	1.9613e − 2 (2.75e − 3) =	1.2813e − 1 (1.21e − 2)
	8	17	5.3659e − 2 (9.60e-3) =	7.1052e − 2 (1.25e − 2) =	8.4174e − 2 (2.82e − 2) =	5.6767e − 2 (4.13e − 3) =	3.0281e − 1 (3.60e − 2)
	15	24	1.3095e − 1 (1.08e − 2) =	4.0269e − 1 (6.08e − 2) =	4.1193e − 1 (9.62e − 2) =	3.2592e − 1 (1.15e − 1) =	1.1117 (2.36e − 1)
MaF12	5	14	4.8817e − 2 (5.39e − 3) =	5.1589e − 2 (6.48e − 3) =	4.9288e − 2 (3.74e − 3) =	5.1916e − 2 (5.61e − 3) =	6.3018e − 2 (1.36e − 2)
	8	17	2.1093e − 1 (7.88e − 3) =	2.0547e − 1 (1.18e − 2) =	2.3550e − 1 (3.41e − 3) =	2.5239e − 1 (2.64e − 2) =	2.7544e − 1 (1.61e − 2)
	15	24	2.3212e − 1 (5.70e − 2) =	6.2255e − 1 (3.08e − 2) =	5.0010e − 1 (3.26e − 2) =	9.2885e − 1 (1.30e − 1) =	9.6687e − 1 (2.62e − 2)
MaF13	5	5	6.5708e − 1 (1.00) =	1.5803e + 7 (2.60e + 7) =	9.7198e + 5 (1.40e + 6) =	3.1110e + 5 (5.38e + 5) =	1.6740e + 7 (2.90e + 7)
	8	5	5.6483e − 2 (1.76e − 2) =	9.6270e + 6 (1.18e + 7) =	4.2415e + 7 (5.33e + 7) =	3.8706e + 3 (4.90e + 3) =	1.0133e + 5 (1.74e + 5)
	15	5	8.7489e + 9 (1.52e + 10) =	2.2562e + 8 (1.22e + 8) =	6.6284e + 7 (8.83e + 7) =	1.0652e − 1 (1.16e − 1) =	3.5940e + 10 (6.22e + 10)
MaF14	5	100	1.0417e + 3 (5.16e + 2) =	2.2961e + 3 (8.34e + 2) =	1.1066e + 3 (3.32e + 2) =	5.1361e + 2 (5.00e + 2) =	3.5794e + 3 (5.09e + 3)
	8	160	6.9139e + 2 (1.20e + 3) =	3.3300e + 3 (1.55e + 3) =	3.7131e + 3 (2.35e + 3) =	1.5506e + 2 (1.38e + 2) =	7.4356e + 4 (8.32e + 4)
	15	300	8.3985e + 1 (6.75e + 1) =	1.9460e + 3 (9.38e + 2) =	2.9943e + 3 (8.13e + 2) =	3.8626e + 2 (4.16e + 2) =	5.0646e + 4 (1.12e + 4)
MaF15	5	100	1.1966e − 1 (3.69e − 2) =	1.6655 (1.55) =	1.1179 (4.96e − 1) =	4.4226e − 1 (5.14e − 1) =	1.1886e + 1 (1.09)
	8	160	3.4528e − 1 (2.32e − 2) =	5.0913 (1.44) =	6.7042 (5.48e − 1) =	4.3660e − 1 (8.50e − 2) =	2.8376e + 1 (3.75e-1)
	15	300	1.6541 (2.55e − 1) =	3.5352e + 1 (2.25e + 1) =	1.6472e + + 1 (8.45) =	2.3472 (6.38e − 1) =	6.8720e + 1 (4.97)

convergence and diversity, crucial in many-objective optimization. These percentages reveal that MaOMVO is particularly adept in scenarios with lower dimensions, as seen in problems like MaF1 and MaF2, where it outperforms other algorithms with a significant margin. However, as the complexity and dimensionality of the problems increase, as observed in MaF3, MaF14 and MaF15, MaOMVO performance becomes more varied. This suggests potential areas for enhancement, especially in high-dimensional problem spaces. It important to note that while MaOMVO exhibits competitive IGD values in problems like MaF4 and MaF5, it does not consistently outperform NSGA-III and RVEA across all dimensions. This indicates that for certain types of problem structures, these algorithms might be better suited, particularly where a balance of exploration and exploitation is critical. In lower-dimensional settings such as MaF6 and MaF7, MaOMVO performance is exemplary, indicating its efficiency in converging to optimal solutions. Conversely, in higher-dimensional problems (MaF8 to MaF10), its performance shows variability, pointing towards a need for adaptive mechanisms in the algorithm to maintain its efficacy in complex scenarios. In Table 3, IGD value compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO is better in 39, 41, 36 and 41 out of 45 cases. To summarize, MaOMVO establishes itself as a robust and competitive algorithm in the majority of the test cases shown in Fig. 6, especially in lower-dimensional problems. Its performance in higher-dimensional and more complex scenarios, however, indicates room for improvement. This

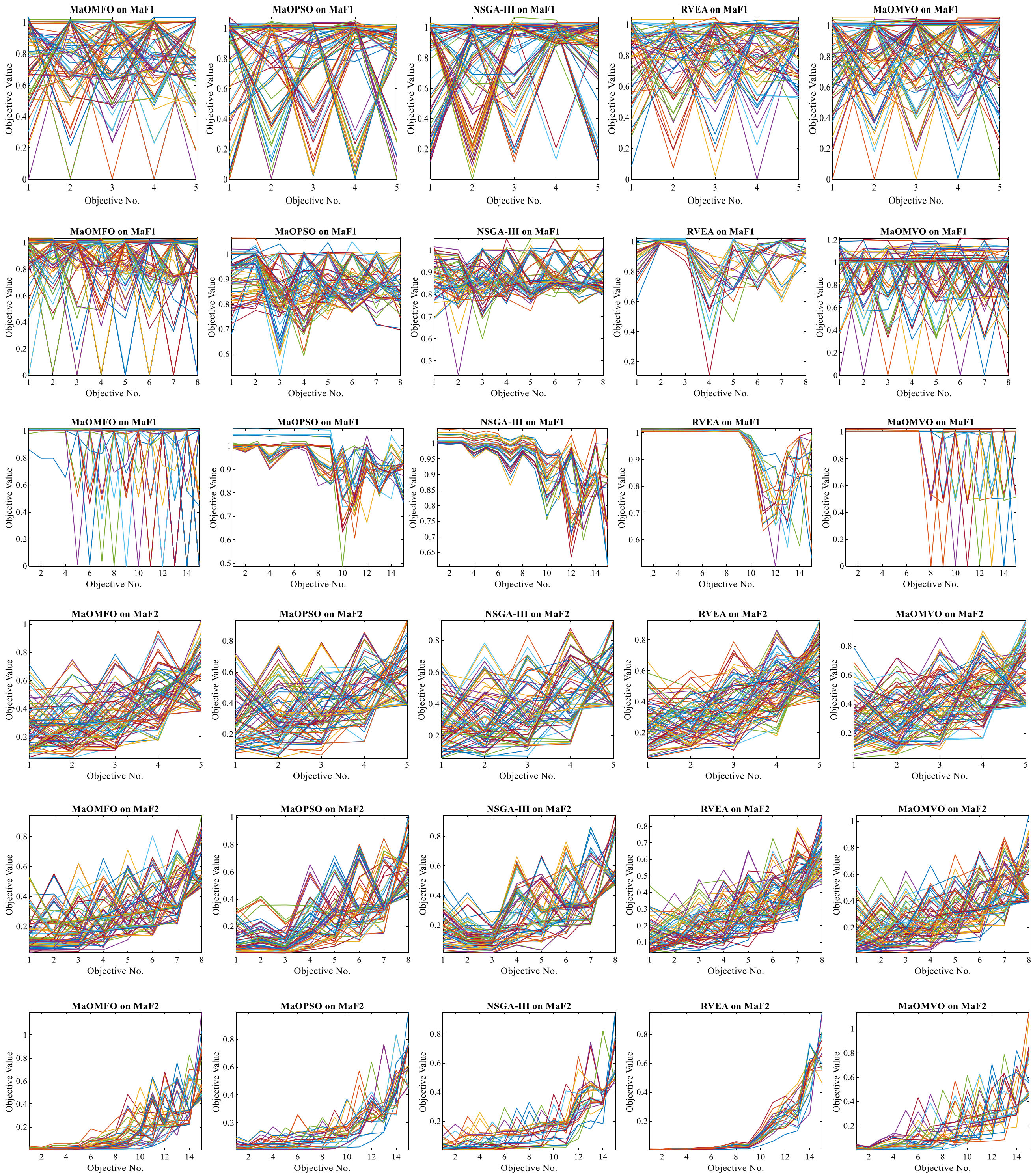


Fig. 6 Best Pareto optimal front obtained by different algorithms on MaF problems

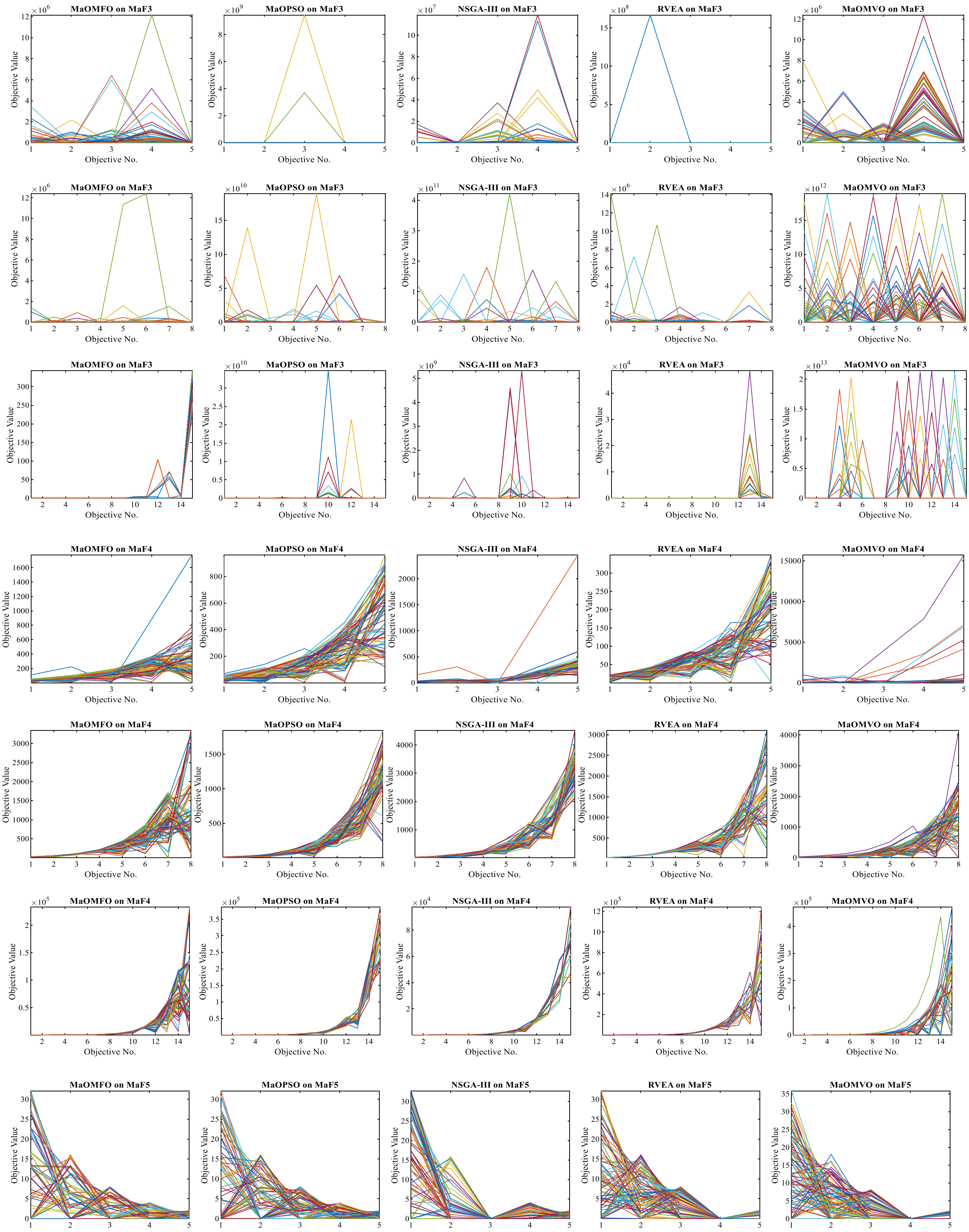


Fig. 6 continued

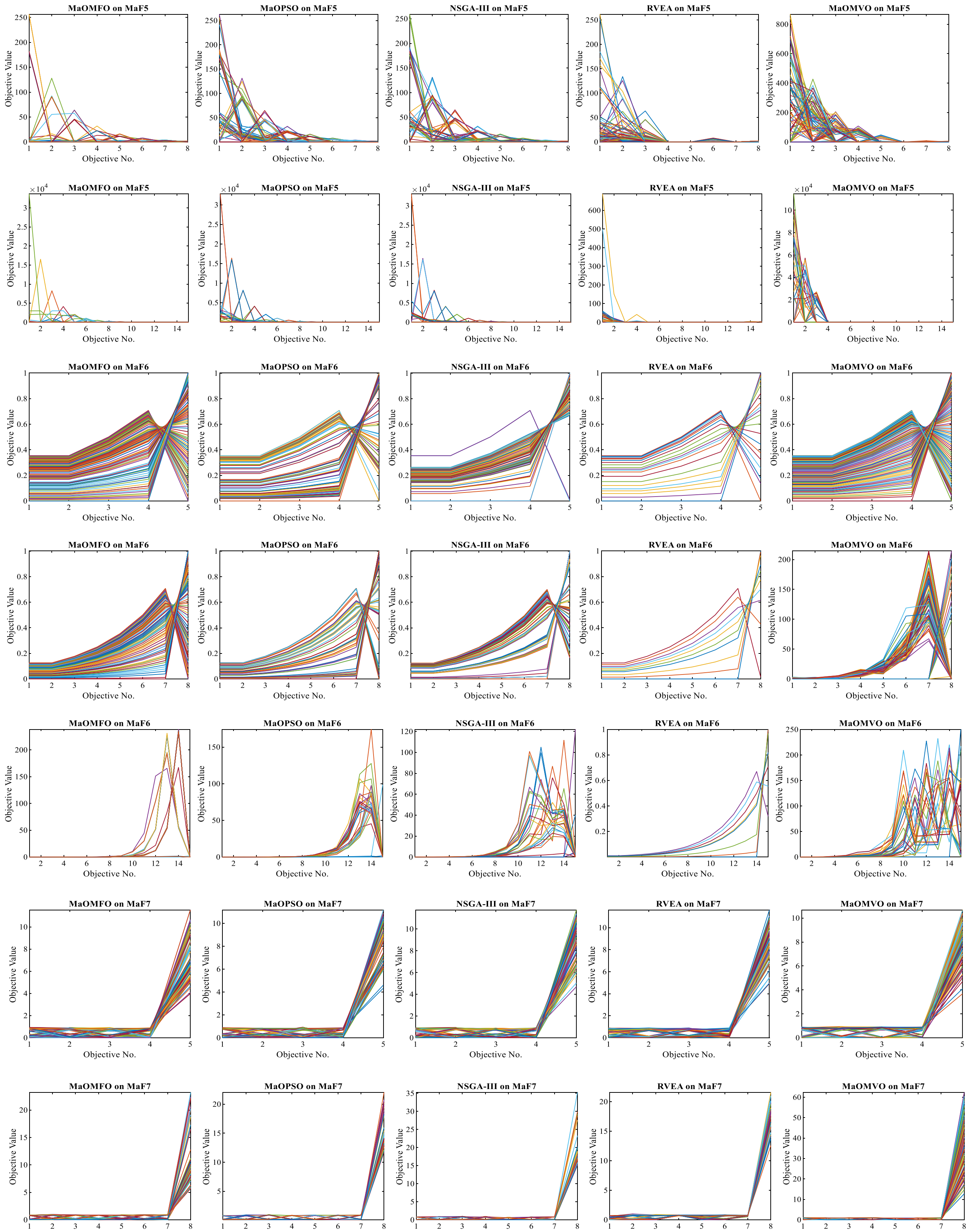


Fig. 6 continued

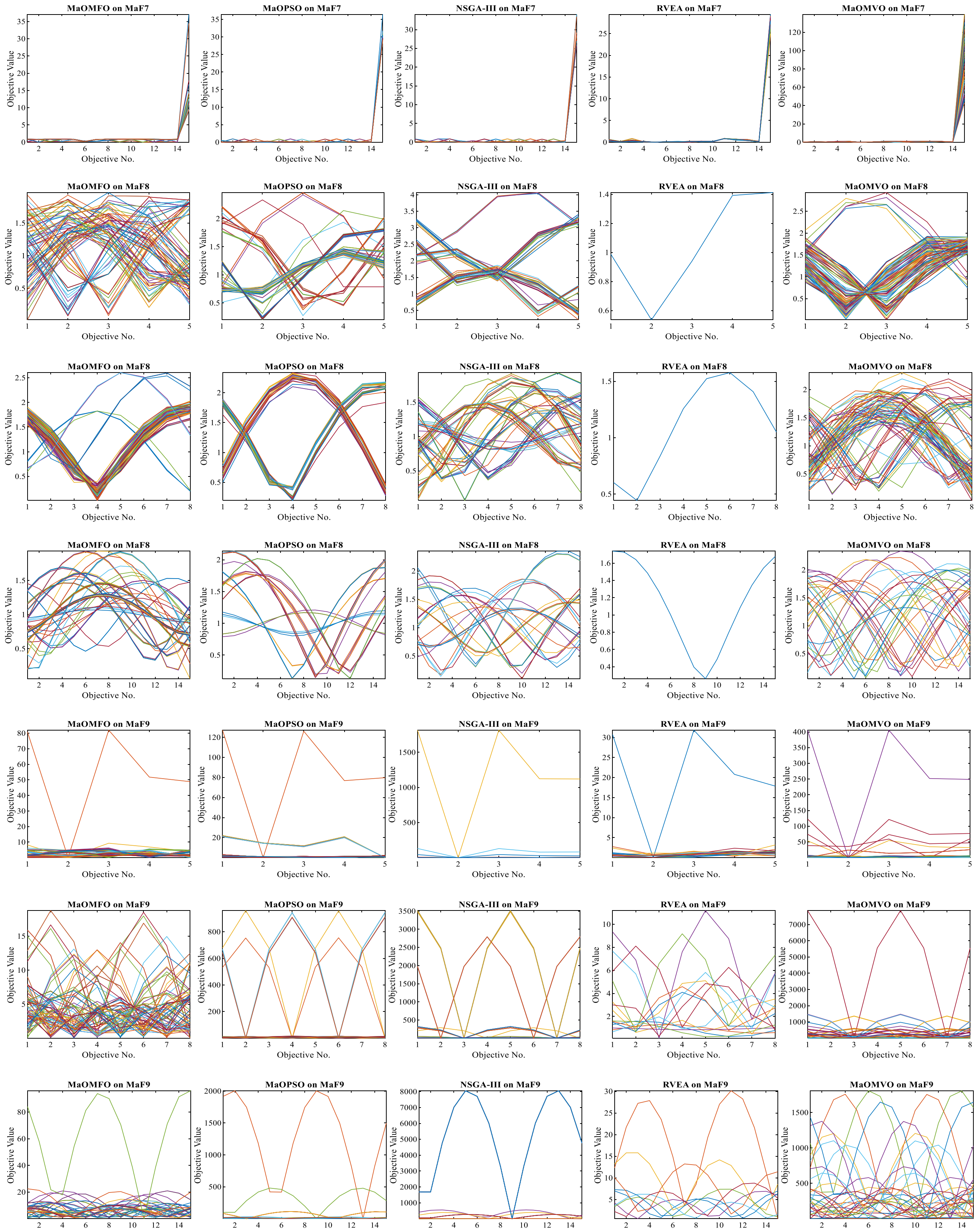


Fig. 6 continued

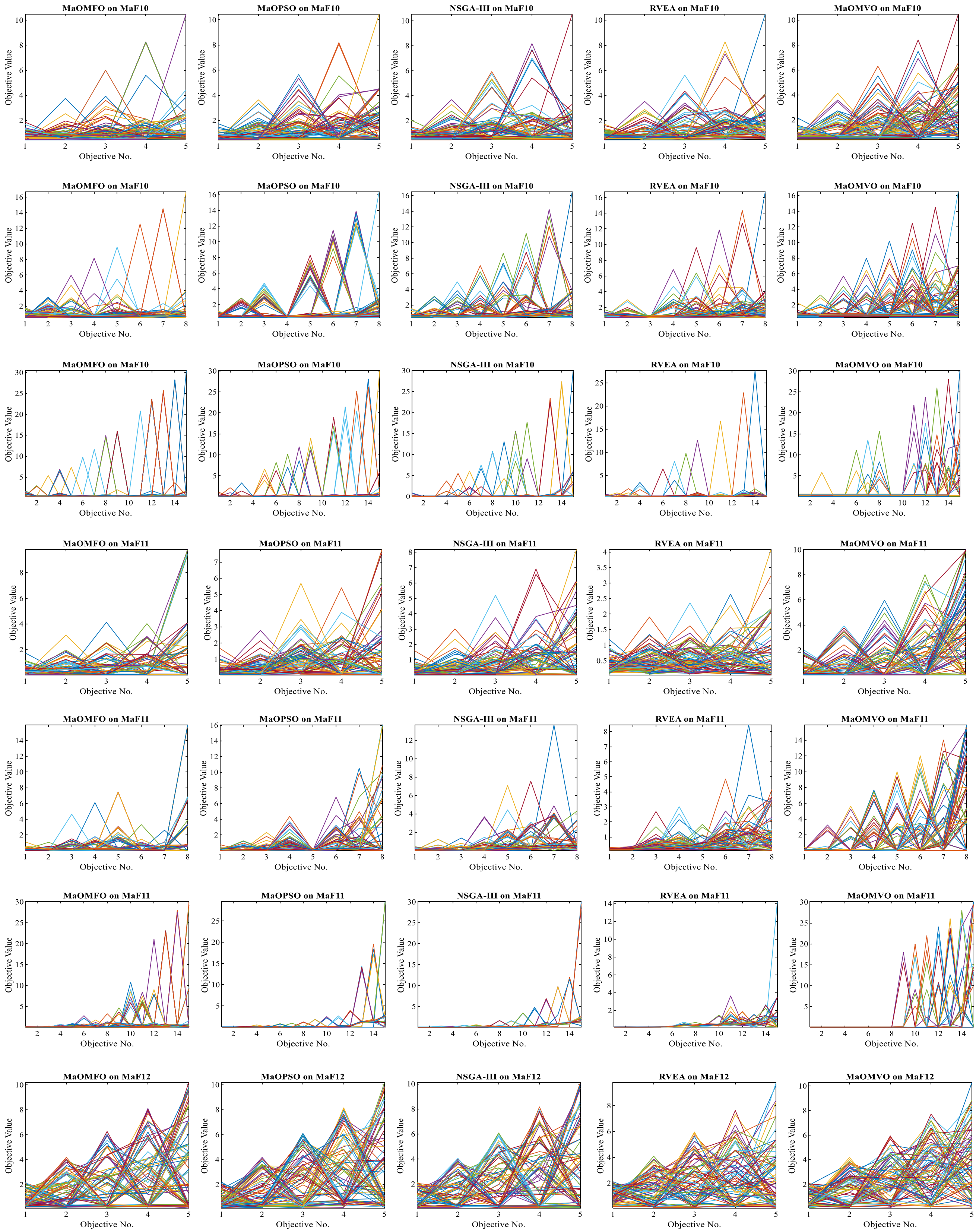


Fig. 6 continued

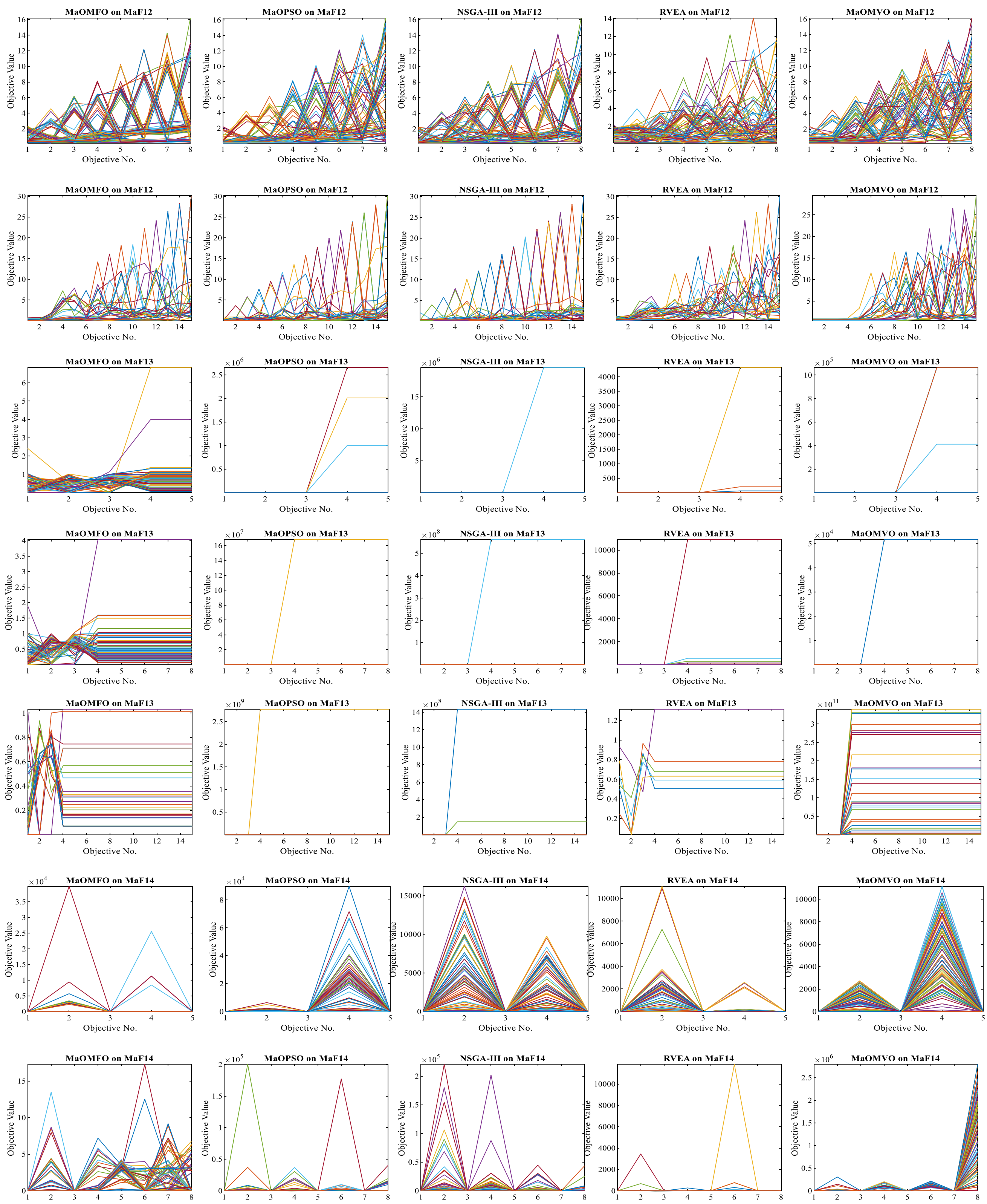


Fig. 6 continued

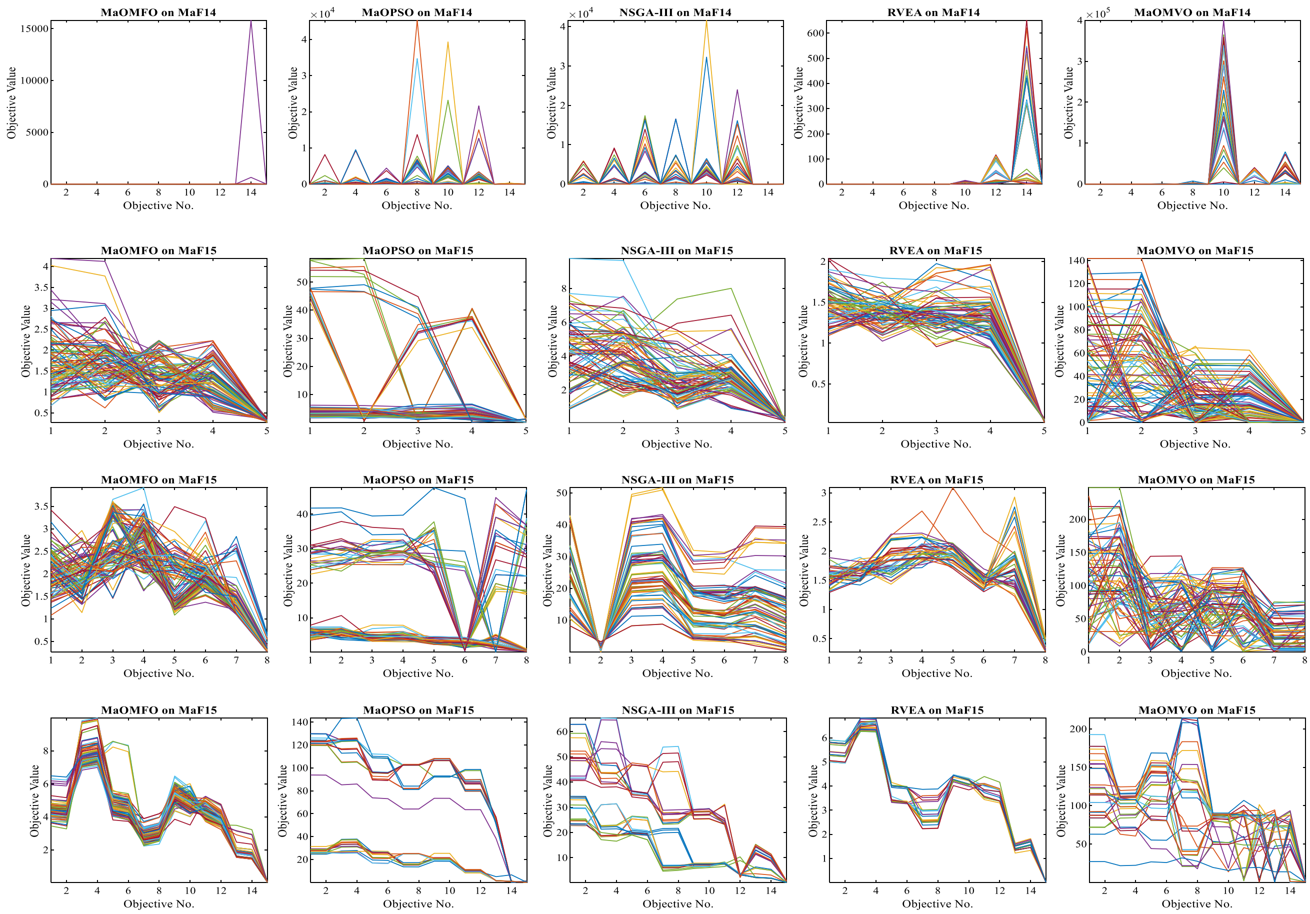


Fig. 6 continued

analysis is crucial for guiding future developments and applications of MaOMVO in many-objective optimization tasks.

Analyzing the Spacing (SP) metric results from Table 4, SP value compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO worse in 1, 3, 9 and 11 out of 45 cases. In problems like MaF1 and MaF2, MaOMVO distinctly outperforms its counterparts, especially in lower-dimensional setups, indicating its proficiency in evenly distributing solutions in simpler search spaces. However, as we move to more complex problems like MaF3 and higher dimensions, MaOMVO dominance wanes, suggesting areas where the algorithm could be enhanced for better performance. It is notable that in certain challenges, such as MaF4 and MaF5, where diversity and distribution of solutions are critical, MaOMVO does not always secure the lowest SP values. This implies that while MaOMVO is generally effective, there is potential for improvement in maintaining solution diversity across the Pareto front shown in Fig. 6.

In Table 5 MaOMVO achieves the best SD results in 25/45 test problems. This impressive performance of MaOMVO shows its effectiveness in maintaining a diverse spread of solutions across the Pareto front. MaOMVO particularly excels in problems like MaF1, MaF2 and MaF4, especially in environments with lower dimensions, indicating its proficiency in evenly distributing solutions in simpler problem spaces. It is notable that in certain challenges, such as MaF5 and MaF6, where maintaining a diverse solution set is crucial, MaOMVO does not consistently record the lowest SD values. This suggests that while MaOMVO is generally effective, there is room for improvement in its diversity maintenance across the Pareto front shown in Fig. 6.

Table 6, which presents the HV values are given in mean (standard deviation) format, with a higher mean indicating better performance. Based on the data from Table 6, on the HV values, when respectively compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO is better in 37, 36, 42 and 43 out of 45 cases and is only worse in 17.77%, 20.0%, 6.66% and 4.44% cases. outstanding performance highlights its effectiveness in not only covering a larger area of the Pareto front but also in maintaining a good balance between convergence and diversity. MaOMVO demonstrates significant effectiveness in problems like MaF1, MaF2 and MaF5, especially in

Table 3 IGD metric of various algorithms on MaF problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF1	5	14	1.5534e − 1 (2.19e − 3) =	2.2576e − 1 (1.25e − 2) =	2.3088e − 1 (2.04e − 2) =	1.5388e − 1 (6.28e − 3) =	1.5700e − 1 (1.97e − 3)
	8	17	2.6415e − 1 (9.97e − 4) =	3.0601e − 1 (6.58e − 3) =	3.0648e − 1 (3.81e − 3) =	3.7495e − 1 (2.75e − 2) =	2.9601e − 1 (1.25e − 2)
	15	24	4.3911e − 1 (4.31e − 3) =	4.0507e − 1 (2.00e − 2) =	4.1005e − 1 (6.20e − 3) =	4.9751e − 1 (2.92e − 2) =	5.6837e − 1 (8.11e − 3)
MaF2	5	14	1.2055e − 1 (2.74e − 3) =	1.3693e − 1 (1.26e − 3) =	1.4112e − 1 (6.07e − 3) =	1.0542e − 1 (8.87e − 4) =	1.3281e − 1 (2.12e − 3)
	8	17	1.9291e − 1 (2.17e − 3) =	2.4230e − 1 (2.07e − 2) =	2.4811e − 1 (2.37e − 2) =	2.7309e − 1 (4.61e − 2) =	1.8404e − 1 (3.33e − 3)
	15	24	3.5817e − 1 (7.46e − 3) =	3.6514e − 1 (1.84e − 2) =	3.6118e − 1 (5.14e − 2) =	5.4185e − 1 (5.71e − 2) =	3.2797e − 1 (5.14e − 2)
MaF3	5	14	5.9211e + 2 (4.16e + 2) =	1.1081e + 3 (5.06e + 2) =	6.6594e + 2 (6.78e + 2) =	4.2400e + 2 (3.37e + 2) =	3.1572e + 5 (2.45e + 5)
	8	17	2.6936e + 2 (1.08e + 2) =	1.6696e + 3 (1.16e + 2) =	4.8518e + 3 (3.18e + 3) =	1.1680e + 3 (2.52e + 2) =	1.6154e + 12 (3.88e + 11)
	15	24	8.0750e + 2 (8.85e + 2) =	2.2946e + 2 (2.27e + 2) =	9.4065e + 2 (3.11e + 2) =	2.8031e + 2 (3.09e + 2) =	2.2684e + 12 (8.72e + 11)
MaF4	5	14	1.7509e + 2 (7.40e + 1) =	2.3935e + 2 (2.86e + 1) =	9.4429e + 1 (7.20e + 1) =	1.7562e + 2 (7.01e + 1) =	1.5676e + 2 (4.05e + 1)
	8	17	7.1659e + 2 (4.15e + 2) =	8.8956e + 2 (4.09e + 2) =	1.7596e + 3 (6.87e + 2) =	9.7271e + 2 (2.05e + 2) =	1.1053e + 3 (4.79e + 2)
	15	24	6.9566e + 4 (1.15e + 4) =	1.7922e + 5 (9.86e + 4) =	8.9903e + 4 (4.96e + 4) =	2.1189e + 5 (2.90e + 5) =	1.7002e + 5 (8.86e + 4)
MaF5	5	14	2.4070 (1.15e − 2) =	2.4837 (4.90e − 2) =	4.1706 (1.87) =	2.6998 (6.81e − 2) =	2.6276 (1.18e − 1)
	8	17	3.6343e + 1 (3.45) =	2.1187e + 1 (1.28) =	2.3596e + 1 (7.79e − 1) =	2.7732e + 1 (2.17) =	6.8750e + 1 (9.77)
	15	24	5.1645e + 3 (7.10e + 2) =	3.8992e + 3 (7.46e + 2) =	4.4812e + 3 (5.74e + 1) =	6.4331e + 3 (4.72e + 2) =	8.8164e + 3 (4.54e + 3)
MaF6	5	14	6.2679e − 3 (3.72e − 4) =	1.7707e − 2 (2.21e − 3) =	5.1803e − 2 (3.40e − 2) =	2.7108e − 2 (4.21e − 3) =	5.4011e − 3 (2.00e − 4)
	8	17	6.7195e − 3 (4.54e − 4) =	9.1316e − 1 (1.54) =	7.1726e − 1 (1.19) =	2.7913e − 1 (3.97e − 1) =	4.0224 (4.21)
	15	24	7.0787e − 1 (3.17e − 1) =	8.9271e − 1 (1.52e − 1) =	4.6249 (4.69) =	2.2251e − 1 (2.57e − 1) =	1.8863e + 2 (5.93e + 1)
MaF7	5	24	3.8223e − 1 (2.87e − 2) =	4.8248e − 1 (6.05e − 2) =	4.7594e − 1 (3.79e − 2) =	5.7186e − 1 (6.59e − 2) =	3.9513e − 1 (3.35e − 2)
	8	27	1.2116 (2.07e − 1) =	2.9929 (1.12) =	3.8378 (8.81e − 1) =	2.4259 (2.20e − 1) =	3.4190 (1.47)
	15	34	3.8432 (1.65) =	1.3855e + 1 (4.99) =	8.6702 (1.03) =	6.0374 (3.41) =	1.7838e + 1 (6.53)
MaF8	5	2	2.7477e − 1 (8.09e − 2) =	6.0001e − 1 (1.45e − 1) =	8.6332e − 1 (4.09e − 1) =	1.2674 (3.92e − 1) =	7.2723e − 1 (2.54e − 1)
	8	2	1.0634 (7.25e − 1) =	1.2524 (6.80e − 1) =	6.4564e − 1 (2.56e − 1) =	1.7365 (5.82e − 1) =	5.3948e − 1 (3.29e − 1)
	15	2	5.7951e − 1 (8.32e − 2) =	9.0152e − 1 (1.01e − 1) =	7.4344e − 1 (8.10e − 2) =	2.3411 (4.00e − 1) =	1.0938 (8.98e − 1)
MaF9	5	2	9.3434e − 1 (4.76e − 1) =	4.1290e − 1 (1.05e − 1) =	4.2747e − 1 (1.74e − 1) =	3.8645e − 1 (8.84e − 2) =	9.4786e − 1 (5.17e − 2)
	8	2	2.7152 (1.06) =	4.0146 (5.19e − 1) =	2.6450 (1.81) =	1.6235 (1.40) =	4.9373 (7.15e − 1)
	15	2	8.8610 (6.66) =	1.0840e + 1 (9.06) =	9.8818 (8.06) =	9.0310 (6.48) =	2.1738e + 1 (7.75)
MaF10	5	14	1.1882 (1.62e − 1) =	1.0982 (1.25e − 2) =	1.2707 (9.40e − 2) =	1.2574 (1.70e − 1) =	1.3692 (3.93e − 2)
	8	17	1.8727 (7.62e − 2) =	1.9387 (3.38e − 1) =	1.9292 (1.15e − 1) =	2.1374 (3.18e − 2) =	2.2974 (1.18e − 1)
	15	24	2.7445 (1.18e − 1) =	2.4714 (1.58e − 1) =	2.5225 (4.40e − 2) =	2.8981 (6.49e − 2) =	5.5673 (1.43)

Table 3 continued

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF11	5	14	5.0123e − 1 (4.01e − 3) =	5.1203e − 1 (1.32e − 2) =	5.0085e − 1 (1.26e − 2) =	5.1034e − 1 (1.73e − 2) =	8.3152e − 1 (4.77e − 2)
	8	17	1.0979 (3.99e − 2) =	1.2210 (3.08e − 2) =	1.1371 (1.60e − 2) =	1.1420 (4.53e − 2) =	2.4125 (7.47e − 2)
	15	24	2.1662 (1.54e − 1) =	2.6322 (2.49e − 1) =	2.6919 (1.32e − 1) =	3.5917 (2.09) =	8.8881 (4.30e − 1)
MaF12	5	14	1.2361 (2.88e − 2) =	1.2611 (1.15e − 2) =	1.2123 (9.62e − 3) =	1.2625 (3.97e − 2) =	1.2702 (6.75e − 2)
	8	17	3.6135 (5.93e − 2) =	3.6067 (1.38e − 1) =	3.5932 (2.01e − 2) =	3.7740 (8.97e − 2) =	3.8481 (1.83e − 1)
	15	24	1.1169e + 1 (2.72e − 1) =	1.1162e + 1 (3.35e − 1) =	1.1449e + 1 (1.76e − 1) =	1.0354e + 1 (5.36e − 1) =	1.2219e + 1 (1.33e − 1)
MaF13	5	5	1.8474e − 1 (2.48e − 2) =	2.8618e − 1 (4.84e − 2) =	2.7073e − 1 (4.58e − 2) =	9.9764e − 1 (5.34e − 1) =	2.4983e − 1 (5.50e − 2)
	8	5	2.3794e − 1 (2.07e − 2) =	3.7923e − 1 (4.09e − 2) =	4.1281e − 1 (5.78e − 2) =	6.4169e − 1 (1.37e − 1) =	3.5870e − 1 (2.07e − 1)
	15	5	3.9206e − 1 (3.84e − 2) =	9.0408e − 1 (2.37e − 1) =	7.5238e − 1 (9.01e − 2) =	1.0626 (2.17e − 1) =	1.4015 (4.66e − 1)
MaF14	5	100	3.2676 (7.72e − 1) =	1.1794e + 1 (4.43) =	1.2407e + 1 (1.93) =	5.4611 (2.12) =	2.1076e + 1 (5.66)
	8	160	3.0910 (8.62e − 1) =	6.3331e + 1 (8.36e + 1) =	3.1549e + 1 (1.68e + 1) =	9.2351 (5.56) =	2.7111e + 4 (1.49e + 4)
	15	300	6.7077 (1.66) =	2.1193e + 1 (9.43) =	1.0267e + 1 (7.56) =	9.6281 (2.87) =	4.2364e + 4 (2.67e + 4)
MaF15	5	100	1.2968 (1.63e − 1) =	3.3710 (7.99e − 1) =	3.3600 (4.62e − 1) =	1.6783 (5.05e − 1) =	2.5562e + 1 (5.86)
	8	160	3.2458 (6.10e − 2) =	8.4394 (1.08) =	1.0451e + 1 (3.25) =	2.9639 (4.44e − 1) =	9.3629e + 1 (1.43e + 1)
	15	300	1.5437e + 1 (2.41) =	5.6742e + 1 (1.21e + 1) =	4.5220e + 1 (9.49) =	1.0169e + 1 (3.51) =	1.0212e + 2 (1.94e + 1)

lower-dimensional settings, where it excels in covering a larger area of the Pareto front. However, in more complex problems like MaF3 and higher dimensions, the effectiveness of MaOMVO becomes less pronounced, pointing out potential areas for improvement. It noteworthy that in challenges like MaF4 and MaF6, where a broader Pareto front coverage is crucial, MaOMVO does not consistently achieve the highest HV values. This indicates that while MaOMVO is generally effective, there is a potential for improvement in its ability to cover a more extensive area of the Pareto front shown in Fig. 6, especially in complex problem spaces.

Table 7, which presents the runtime (RT) metric results for MaOMVO and other algorithms across various many-objective optimization problems, provides insights into the computational efficiency of these algorithms. In Table 7, RT value compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO is better in 44, 40, 32 and 42 out of 45 cases. The shorter running time of MaOMVO does not come at the cost of performance degradation. Instead, it indicates higher search efficiency, enabling MaOMVO to reach satisfactory solutions faster than its counterparts. Thus, the experimental results from Table 7 clearly illustrate that MaOMVO not only excels in running speed but also maintains high efficiency in problem-solving. Its ability to perform computations faster than other algorithms like MaOMFO, MaOPSO, NSGA-III and RVEA, without compromising on the quality of the solutions, is a testament to its robustness and suitability for scenarios where time efficiency is important.

Experimental Results on RWMaOP Problems

Analyzing the Spacing (SP) metric from Table 8 for MaOMVO across a range of real-world many-objective optimization problems (RWMaOPs), we gain insights into the algorithm performance in terms of solution distribution uniformity. In RWMaOP1, MaOMVO shows a mean SP value of 1.1789 (SD: 0.186), which is significantly lower than those of MaOMFO, MaOPSO, NSGA-III and RVEA. This indicates MaOMVO superior ability in evenly spreading out solutions, particularly in the context of car cab design. MaOMVO records a mean SP value of 10,625 (SD: 8,470), which is higher compared to other algorithms. Despite this, MaOMVO performance in this engineering problem suggests a need for improvement in maintaining

Table 4 SP metric of various algorithms on MaF problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF1	5	14	4.5452e − 2 (5.39e − 3)	1.3403e − 1 (2.78e − 2) =	1.0514e − 1 (9.29e − 3) =	8.7241e − 2 (5.88e − 3) =	1.8742e − 1 (1.24e − 2) =
	8	17	9.0376e − 2 (4.22e − 2)	1.2597e − 1 (8.35e − 3) =	1.2140e − 1 (8.88e − 3) =	9.7680e − 2 (1.25e − 2) =	2.8171e − 1 (3.77e − 3) =
	15	24	7.7106e − 2 (2.96e − 2)	1.5690e − 1 (8.88e − 3) =	1.3265e − 1 (2.71e − 2) =	9.0669e − 2 (5.52e − 2) =	4.1910e − 1 (3.13e − 2) =
MaF2	5	14	1.4435e − 1 (6.16e − 3) =	1.2076e − 1 (9.32e − 3) =	1.2005e − 1 (6.38e − 3) =	4.7415e − 2 (6.66e − 3) =	3.8591e − 2 (2.90e − 3)
	8	17	2.1602e − 1 (5.91e − 3) =	1.4798e − 1 (2.72e − 2) =	1.3979e − 1 (1.05e − 2) =	5.4290e − 2 (1.82e − 3) =	6.0302e − 2 (1.77e − 3)
	15	24	2.5921e − 1 (4.53e − 2) =	2.3614e − 1 (2.92e − 2) =	2.1623e − 1 (8.28e − 2) =	6.9475e − 2 (2.00e − 2) =	1.0069e − 1 (1.18e − 2)
MaF3	5	14	6.8156e + 5 (4.89e + 5) =	1.8497e + 9 (2.10e + 9) =	3.3098e + 8 (3.45e + 8) =	7.1190e + 7 (1.22e + 8) =	4.5758e + 5 (2.08e + 5)
	8	17	5.4208e + 7 (9.19e + 7) =	2.6805e + 10 (1.99e + 10) =	6.5347e + 10 (1.46e + 10) =	1.6971e + 9 (2.94e + 9) =	6.8132e + 11 (1.08e + 10)
	15	24	1.9701e + 2 (3.34e + 2) =	6.3745e + 9 (3.51e + 9) =	7.2277e + 9 (1.11e + 10) =	5.9119e + 3 (6.06e + 3) =	1.3785e + 12 (1.51e + 11)
MaF4	5	14	2.4839e + 1 (8.74) =	3.1441e + 2 (3.38e + 2) =	1.2517e + 2 (1.84e + 2) =	9.0590e + 1 (8.72e + 1) =	6.2148e + 2 (1.00e + 3)
	8	17	1.3881e + 2 (2.59e + 1) =	1.6415e + 3 (2.63e + 3) =	2.2090e + 2 (6.40e + 1) =	1.7623e + 2 (7.91e + 1) =	4.2250e + 3 (5.21e + 3)
	15	24	1.8410e + 4 (1.66e + 3) =	1.8610e + 4 (9.26e + 3) =	1.0968e + 4 (5.16e + 3) =	2.7470e + 4 (3.80e + 4) =	6.4636e + 4 (4.03e + 4)
MaF5	5	14	2.1150 (3.42e − 2) =	1.6936 (8.35e − 2) =	1.6626 (1.95e − 1) =	1.6427 (5.64e − 1) =	8.2189e − 1 (1.71e − 1)
	8	17	1.3877e + 1 (4.19) =	1.7239e + 1 (2.74) =	1.3268e + 1 (1.46) =	2.3256e + 1 (5.57) =	2.2995e + 1 (3.17)
	15	24	3.8246e + 3 (1.04e + 2) =	1.6049e + 3 (1.00e + 3) =	1.6505e + 3 (2.69e + 2) =	1.3041e + 3 (1.87e + 3) =	3.6929e + 3 (9.77e + 2)
MaF6	5	14	1.6683e − 2 (8.12e − 4) =	2.5148e − 2 (7.30e − 3) =	3.9265e − 2 (2.27e − 2) =	5.2268e − 2 (6.90e − 3) =	8.7610e − 3 (1.97e − 3)
	8	17	2.1281e − 2 (1.87e − 4) =	1.2784 (2.17) =	1.0471 (1.78) =	7.7244e − 2 (6.71e − 2) =	8.4113 (7.69)
	15	24	1.4720e + 1 (9.23) =	2.1868e + 1 (1.95) =	1.4953e + 1 (8.34) =	2.1042e − 1 (1.47e − 1) =	3.8465e + 1 (1.08e + 1)
MaF7	5	24	3.3561e − 1 (1.91e − 2) =	3.2066e − 1 (2.15e − 2) =	3.2120e − 1 (1.40e − 2) =	1.8742e − 1 (3.26e − 2) =	1.3599e − 1 (8.82e − 3)
	8	27	2.6761e − 1 (4.34e − 2) =	4.9785e − 1 (7.16e − 2) =	6.0151e − 1 (2.38e − 1) =	6.7243e − 1 (3.37e − 2) =	5.8393e − 1 (4.01e − 2)
	15	34	3.9501e − 1 (1.25e − 1) =	1.0764 (6.07e − 1) =	1.4430 (7.22e − 1) =	3.7966 (2.98) =	8.3133e − 1 (1.77e − 1)
MaF8	5	2	1.5566e − 1 (2.43e − 2) =	2.0285e − 1 (1.67e − 1) =	4.5984e − 1 (3.53e − 1) =	NaN (NaN)	6.3272e − 2 (3.36e − 2)
	8	2	5.1317e − 1 (5.08e − 1) =	4.8830e − 1 (4.12e − 1) =	2.7090e − 1 (1.10e − 1) =	NaN (NaN)	2.9280e − 1 (1.19e − 1)
	15	2	6.6034e − 1 (3.99e − 1) =	1.0067 (5.97e − 1) =	1.2015 (6.66e − 1) =	NaN (NaN)	1.9459 (2.68)
MaF9	5	2	2.2667e + 2 (3.71e + 2) =	1.8235e + 2 (1.59e + 2) =	2.0176e + 2 (3.38e + 2) =	9.6315e + 1 (1.28e + 2) =	2.3418e + 2 (3.20e + 2)
	8	2	5.5901 (3.32) =	2.7813e + 2 (3.45e + 2) =	4.7293e + 1 (7.60e + 1) =	5.4567 (3.21) =	1.5929e + 3 (1.76e + 3)
	15	2	3.5398e + 1 (3.35e + 1) =	1.0135e + 3 (1.66e + 3) =	3.2987e + 2 (1.45e + 2) =	4.5143e + 1 (1.15e + 1) =	2.7891e + 3 (1.30e + 3)

Table 4 continued

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF10	5	14	9.5284e − 1 (1.16e − 1) =	8.2585e − 1 (4.56e − 2) =	9.1536e − 1 (1.17e − 1) =	7.6450e − 1 (1.01e − 1) =	5.0247e − 1 (9.29e − 2)
	8	17	2.0454 (3.51e − 1) =	1.7221 (8.75e − 2) =	1.7378 (1.10e − 1) =	1.8412 (4.01e − 2) =	9.7994e − 1 (1.43e − 1)
	15	24	5.0812 (9.47e − 1) =	3.9727 (2.54) =	5.3134 (5.22e − 1) =	6.0111 (8.34e − 1) =	2.9594 (3.85e − 1)
MaF11	5	14	6.9225e − 1 (1.48e − 1) =	4.5365e − 1 (1.07e − 1) =	6.0264e − 1 (1.90e − 2) =	3.2298e − 1 (4.88e − 2) =	3.7896e − 1 (9.62e − 2)
	8	17	9.3316e − 1 (3.36e − 1) =	1.0582 (3.48e − 1) =	1.4183 (2.44e − 1) =	9.0802e − 1 (3.64e − 1) =	7.9646e − 1 (2.10e − 1)
	15	24	2.0414 (9.26e − 1) =	1.4209 (6.63e − 1) =	1.8395 (6.17e − 1) =	2.1776 (7.09e − 1) =	2.0226 (2.67e − 1)
MaF12	5	14	4.2173e − 1 (1.24e − 2)	8.5358e − 1 (8.66e − 2) =	8.2629e − 1 (7.13e − 2) =	7.8075e − 1 (3.10e − 2) =	1.2261 (1.52e − 2) =
	8	17	1.1225 (2.40e − 1)	1.9101 (1.05e − 1) =	2.1981 (3.84e − 1) =	2.5357 (1.98e − 1) =	2.4661 (2.05e − 1) =
	15	24	3.8013 (3.45e − 1)	9.4081 (1.88) =	7.2273 (1.15) =	7.4482 (2.81) =	9.8442 (1.98) =
MaF13	5	5	3.5835 (4.98) =	3.1069e + 7 (3.73e + 7) =	9.3478e + 6 (1.27e + 7) =	1.0777e + 6 (1.86e + 6) =	7.2055e + 6 (1.24e + 7)
	8	5	8.0928e − 1 (5.38e − 1) =	1.4345e + 8 (1.59e + 8) =	1.4919e + 8 (1.59e + 8) =	2.6118e + 4 (3.27e + 4) =	1.6506e + 6 (2.83e + 6)
	15	5	3.0307e + 11 (5.25e + 11) =	3.2078e + 9 (2.49e + 9) =	1.1461e + 9 (1.49e + 9) =	1.0157 (1.58) =	2.9041e + 10 (5.03e + 10)
MaF14	5	100	5.4378e + 3 (4.91e + 3) =	2.0018e + 3 (6.54e + 2) =	1.0574e + 3 (4.24e + 2) =	7.9000e + 2 (4.22e + 2) =	7.3519e + 2 (9.97e + 2)
	8	160	7.3536e + 2 (6.66e + 2) =	1.2282e + 4 (1.34e + 4) =	9.8889e + 3 (9.71e + 3) =	3.2303e + 3 (5.59e + 3) =	1.9086e + 4 (1.22e + 4)
	15	300	4.4284e + 2 (6.23e + 2) =	4.8657e + 3 (2.22e + 3) =	5.4068e + 3 (2.73e + 3) =	5.1558e + 2 (8.67e + 2) =	1.3883e + 4 (9.18e + 3)
MaF15	5	100	5.6535e − 1 (6.89e − 1) =	1.2913 (5.32e − 1) =	4.3471 (4.99) =	2.6703e − 1 (1.07e − 1) =	6.0782 (4.79e − 1)
	8	160	3.2682e − 1 (1.73e − 2) =	6.7429 (3.28) =	7.2824 (4.28) =	5.9689e − 1 (1.31e − 1) =	2.2504e + 1 (1.31)
	15	300	6.0997e − 1 (4.50e − 2) =	4.0832e + 1 (3.02e + 1) =	1.2779e + 1 (7.27) =	8.4165e − 1 (6.68e − 2) =	5.3196e + 1 (7.82)

uniform solution distribution for large-scale structural designs. With a mean SP value of 16.218 (SD: 2.60), MaOMVO demonstrates better performance than MaOMFO and MaOPSO, but it is outperformed by NSGA-III and RVEA. This result suggests MaOMVO intermediate capability in generating uniformly distributed solutions in the context of material science optimization. MaOMVO has a mean SP value of 58,271 (SD: 9,550), indicating its weaker performance in this domain. This high SP value suggests a concentration of solutions, which is less desirable for complex engineering problems like antenna design. In this problem, MaOMVO achieves a mean SP value of 0.04318 (SD: 0.00377), In Table 8, SP value compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO is better in 5, 5, 4 and 4 out of 5 cases showcasing its excellent performance with the most uniform solution distribution among all compared algorithms shown in Fig. 7.

From Table 9, In RWMaOP1, HV of 2.1753e-3 (SD: 1.60e-4), outperforming MaOMFO, MaOPSO, NSGA-III and RVEA. This indicates its superior ability to cover a larger area of the Pareto front with diverse solutions. In RWMaOP2, HV of 8.1461e-2 (SD: 5.40e-4), MaOMVO shows better performance than most of its competitors, except RVEA. In RWMaOP3, MaOMVO records an HV of 1.5889e-2 (SD: 5.94e-4), indicating competitive performance but slightly outperformed by MaOMFO and MaOPSO. It suggests MaOMVO capacity to generate a comprehensive set of solutions is on par with these algorithms. In RWMaOP4, MaOMVO HV value of 5.4170e-1

Table 5 SD metric of various algorithms on MaF problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF1	5	14	9.0793e − 1 (1.02e − 1) =	7.9451e − 1 (1.45e − 1) =	6.9819e − 1 (8.24e − 2) =	3.3243e − 1 (1.22e − 2) =	6.5618e − 2 (5.24e − 3)
	8	17	1.0261 (3.74e − 2) =	7.6846e − 1 (9.03e − 2) =	7.8428e − 1 (2.62e − 2) =	7.6572e − 1 (3.37e − 2) =	8.8549e − 2 (6.07e − 2)
	15	24	1.8019 (4.63e − 2) =	1.0032 (1.65e − 2) =	9.8903e − 1 (6.13e − 3) =	1.0144 (5.47e − 2) =	8.6430e − 2 (5.17e − 2)
MaF2	5	14	6.1541e − 1 (5.30e − 2) =	7.6957e − 1 (3.28e − 2) =	8.1270e − 1 (7.99e − 2) =	1.6082e − 1 (1.15e − 2) =	1.1757e − 1 (8.72e − 3)
	8	17	1.1029e − 1 (4.18e − 3)	7.8307e − 1 (5.30e − 2) =	7.7945e − 1 (7.01e − 3) =	2.7393e − 1 (5.07e − 2) =	8.2081e − 1 (4.28e − 2) =
	15	24	4.8174e − 1 (4.43e − 2)	1.0244 (2.10e − 2) =	9.6649e − 1 (7.37e − 2) =	9.3206e − 1 (5.30e − 2) =	1.3163 (9.77e − 2) =
MaF3	5	14	1.2959 (1.85e − 1)	2.0316 (6.49e − 2) =	1.8757 (3.26e − 1) =	1.7727 (4.03e − 1) =	1.3524 (3.74e − 2) =
	8	17	2.3330e − 1 (1.66e − 2)	1.8251 (1.10e − 1) =	1.7462 (4.65e − 2) =	1.9378 (2.67e − 1) =	2.0207 (1.01e − 1) =
	15	24	5.8510e − 1 (7.21e − 2)	3.2696 (1.03e − 1) =	3.2699 (1.42e − 1) =	2.8050 (2.14) =	1.0782 (1.09e − 1) =
MaF4	5	14	8.2789e − 1 (9.19e − 2) =	9.4880e − 1 (3.57e − 1) =	8.3778e − 1 (3.38e − 1) =	5.1713e − 1 (2.79e − 2) =	7.4830e − 1 (7.12e − 1)
	8	17	8.8152e − 1 (3.92e − 2) =	1.1746 (5.85e − 1) =	8.2768e − 1 (5.02e − 2) =	6.5854e − 1 (6.41e − 2) =	8.1388e − 1 (4.56e − 1)
	15	24	1.0542 (5.80e − 2) =	1.0139 (1.68e − 2) =	1.0382 (5.03e − 3) =	1.0169 (2.24e − 2) =	7.9913e − 1 (7.03e − 2)
MaF5	5	14	5.6048e − 1 (1.22e − 2) =	3.9095e − 1 (1.57e − 2) =	7.8149e − 1 (3.73e − 1) =	3.6797e − 1 (3.78e − 2) =	1.1897e − 1 (8.37e − 3)
	8	17	1.4428e − 1 (3.27e − 2)	6.1694e − 1 (2.07e − 2) =	5.9359e − 1 (6.17e − 2) =	1.0188 (1.33e − 1) =	1.3920 (4.70e − 2) =
	15	24	3.2657e − 1 (1.34e − 1)	1.4353 (1.59e − 1) =	1.7022 (7.28e − 2) =	1.0982 (1.24e − 1) =	2.0633 (8.38e − 2) =
MaF6	5	14	1.6483e − 1 (2.14e − 2)	1.0912 (3.89e − 2) =	1.1660 (1.91e − 1) =	5.2541e − 1 (8.04e − 2) =	7.8770e − 1 (5.17e − 2) =
	8	17	2.2432e − 1 (6.63e − 2)	1.0652 (1.56e − 1) =	9.6106e − 1 (1.79e − 1) =	1.3450 (4.50e − 1) =	1.1022 (7.86e − 2) =
	15	24	6.2616e − 1 (1.08e − 1)	1.9222 (1.53e − 1) =	1.3469 (2.08e − 1) =	1.9289 (1.20) =	1.7516 (3.79e − 1) =
MaF7	5	24	1.5433e − 1 (2.88e − 2)	5.7568e − 1 (6.33e − 2) =	5.0472e − 1 (4.83e − 2) =	3.7720e − 1 (8.90e − 2) =	6.6889e − 1 (1.11e − 2) =
	8	27	1.9902e − 1 (3.31e − 2)	5.5618e − 1 (8.11e − 2) =	5.7996e − 1 (2.27e − 2) =	4.2822e − 1 (3.57e − 2) =	9.8820e − 1 (2.90e − 2) =
	15	34	7.5338e − 1 (1.31e − 1)	1.0709 (5.79e − 2) =	1.1943 (1.59e − 1) =	8.8133e − 1 (3.58e − 2) =	1.1487 (8.69e − 2) =
MaF8	5	2	9.3900e − 1 (1.07e − 1) =	1.2306 (1.83e − 1) =	1.2160 (1.71e − 1) =	NaN (NaN)	6.4623e − 1 (2.50e − 1)
	8	2	1.1286 (9.50e − 2) =	1.1047 (6.67e − 2) =	1.0803 (7.66e − 2) =	NaN (NaN)	5.4447e − 1 (1.56e − 1)
	15	2	1.2717 (1.99e − 1) =	1.1725 (1.32e − 1) =	1.2476 (2.07e − 1) =	NaN (NaN)	9.5770e − 1 (5.31e − 1)
MaF9	5	2	1.7128 (3.87e − 1) =	2.0585 (2.70e − 2) =	1.8924 (1.33e − 1) =	2.2300 (1.14e − 1) =	1.5972 (6.44e − 1)
	8	2	1.2868 (1.21e − 1) =	1.8296 (4.47e − 1) =	1.6211 (3.73e − 1) =	1.0851 (3.43e − 1) =	1.7732 (2.16e − 1)
	15	2	1.5916 (2.84e − 1) =	2.4722 (9.83e − 1) =	2.9304 (4.92e − 1) =	4.3533 (5.90) =	1.4401 (3.09e − 1)

Table 5 continued

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF10	5	14	7.9805e − 1 (9.80e − 2) =	6.7323e − 1 (3.75e − 2) =	6.7863e − 1 (4.65e − 2) =	5.7213e − 1 (6.28e − 2) =	2.2878e − 1 (3.30e − 2)
	8	17	2.2706e − 1 (2.32e − 2)	8.3695e − 1 (4.49e − 2) =	8.6503e − 1 (1.29e − 1) =	7.9783e − 1 (6.16e − 2) =	1.2719 (4.58e − 2) =
	15	24	5.8829e − 1 (4.82e − 2)	1.1777 (1.81e − 1) =	1.3819 (5.61e − 2) =	1.3356 (1.06e − 1) =	1.6342 (5.43e − 2) =
MaF11	5	14	1.2914e − 1 (3.52e − 2)	5.1622e − 1 (2.29e − 2) =	4.7099e − 1 (3.81e − 2) =	5.6109e − 1 (1.60e − 2) =	6.7945e − 1 (6.72e − 2) =
	8	17	1.5464e − 1 (2.73e − 2)	8.5681e − 1 (7.17e − 2) =	8.6589e − 1 (1.13e − 1) =	7.9612e − 1 (3.76e − 2) =	9.9706e − 1 (4.40e − 2) =
	15	24	5.3570e − 1 (8.90e − 2)	1.0344 (1.38e − 2) =	1.0559 (2.49e − 2) =	1.0194 (1.24e − 2) =	1.1447 (1.56e − 1) =
MaF12	5	14	1.1318e − 1 (2.60e − 3)	3.6795e − 1 (6.13e − 2) =	2.7104e − 1 (3.08e − 2) =	2.5697e − 1 (4.70e − 3) =	5.8982e − 1 (2.04e − 3) =
	8	17	7.4428e − 1 (7.93e − 3) =	3.7540e − 1 (2.12e − 2) =	4.3963e − 1 (6.28e − 2) =	4.1337e − 1 (1.49e − 2) =	1.3986e − 1 (2.78e − 2)
	15	24	1.4666 (3.12e − 2) =	1.0639 (1.17e − 1) =	1.0307 (5.77e − 2) =	7.5267e − 1 (2.49e − 1) =	4.9576e − 1 (2.05e − 2)
MaF13	5	5	1.3354 (3.82e − 1) =	2.0583 (3.82e − 2) =	2.0745 (2.43e − 2) =	5.6404 (3.79) =	1.7597 (4.95e − 1)
	8	5	1.1491 (4.65e − 2) =	2.1979 (3.60e − 2) =	2.2185 (4.88e − 4) =	5.0848 (1.86) =	1.7431 (7.67e − 1)
	15	5	1.8234 (4.40e − 1) =	3.8008 (1.14e − 1) =	3.7748 (7.97e − 2) =	1.2595 (3.25e − 1) =	2.3842 (1.18)
MaF14	5	100	1.8068 (1.99e − 1) =	9.5830e − 1 (1.34e − 1) =	9.0038e − 1 (1.01e − 1) =	1.3202 (1.17e − 1) =	1.9210e − 1 (3.28e − 2)
	8	160	1.3561 (4.91e − 1) =	1.2364 (4.46e − 1) =	1.0567 (2.36e − 1) =	1.3049 (6.53e − 1) =	1.7951e − 1 (2.96e − 2)
	15	300	4.6595e − 1 (2.37e − 2)	1.9592 (1.81e − 1) =	2.1179 (3.24e − 1) =	2.1905 (1.19) =	1.8145 (4.09e − 1) =
MaF15	5	100	1.8835e − 1 (1.11e − 2)	7.4657e − 1 (1.79e − 1) =	9.0078e − 1 (2.16e − 1) =	6.6079e − 1 (1.58e − 1) =	7.1622e − 1 (1.34e − 2) =
	8	160	2.1277e − 1 (1.30e − 2)	8.5496e − 1 (1.11e − 1) =	8.5116e − 1 (1.14e − 1) =	8.0854e − 1 (7.97e − 2) =	6.8865e − 1 (2.60e − 2) =
	15	300	5.4936e − 1 (3.58e − 2)	1.0538 (4.90e − 2) =	1.0437 (7.26e − 2) =	1.0116 (2.54e − 2) =	8.9416e − 1 (2.73e − 2) =

(SD: 5.76e-3) is higher than that of MaOPSO and NSGA-III, but marginally lower than MaOMFO and RVEA, demonstrating its competence in achieving a diverse solution set. In this problem, In RWMaOP5, MaOMVO achieves an HV of 5.4069e-1 (SD: 7.26e-3), which is comparable to the other algorithms. Therefore, MaOMVO has a better divergence for solving RWMaOPs. In Table 9 on the HV values, when respectively compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO is better in 4, 4, 5 and 5 out of 5 cases and is only worse in 38.09%, 9.52%, 9.52% and 4.76% cases. This suggests a balanced performance in maintaining both coverage and diversity of the Pareto front shown in Fig. 7.

From Table 10, the overall running time of MaOMVO is observed to be significantly lower compared to other algorithms across various RWMaOPs, indicating its superior computational efficiency. In RWMaOP1, MaOMVO runtime is 4.7818 s (SD: 0.91), accounting for only 27%, 45%, 41% and 96% of the runtimes of MaOMFO (17.491 s), MaOPSO (10.627 s), NSGA-III (11.715 s) and RVEA (4.9501 s), respectively. This demonstrates MaOMVO significantly faster running speed. In RWMaOP2, a runtime of 8.125 s (SD: 0.316), MaOMVO operates at about 50%, 83% and 72% of the speeds of MaOMFO (16.232 s), MaOPSO (9.8068 s) and RVEA (11.188 s), highlighting its efficiency. In RWMaOP3, MaOMVO records a runtime of 1.6577 s (SD: 0.4), which is significantly faster, operating at approximately 13%, 23% and 36% of the speeds of MaOMFO (13.069 s), MaOPSO (6.9385 s) and RVEA (4.6233 s). In RWMaOP4, MaOMVO runtime of 1.3095 s (SD: 0.403) is markedly quicker, functioning at about 13%, 22% and 34% of the runtimes of MaOMFO (10.132 s), MaOPSO (5.8930 s) and RVEA (3.7994 s). In RWMaOP5, MaOMVO completes its

Table 6 HV metric of various algorithms on MaF problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF1	5	14	7.5417e − 3 (1.58e − 4) =	4.0867e − 3 (2.13e − 4) =	3.7271e − 3 (4.91e − 4) =	7.2382e − 3 (2.54e − 4) =	6.4508e − 3 (2.98e − 4)
	8	17	1.4691e − 5 (2.12e − 6) =	1.8231e − 5 (2.45e − 6) =	1.8688e − 5 (6.10e − 7) =	6.3141e − 6 (1.10e − 6) =	2.8269e − 6 (1.72e − 6)
	15	24	0.0000 (0.00) =	8.5509e − 13 (2.16e − 13) =	6.4663e − 13 (1.16e − 13) =	1.4332e − 13 (5.49e − 14) =	1.6436e − 13 (1.18e − 13)
MaF2	5	14	1.5739e − 1 (4.58e − 4) =	1.5229e − 1 (1.35e − 3) =	1.5267e − 1 (2.25e − 3) =	1.6258e − 1 (1.05e − 3) =	1.4398e − 1 (1.34e − 3)
	8	17	1.3880e − 1 (4.34e − 3) =	1.6858e − 1 (8.57e − 3) =	1.5185e − 1 (1.19e − 2) =	1.2992e − 1 (8.29e − 3) =	1.5329e − 1 (6.59e − 3)
	15	24	1.2240e − 1 (1.21e − 2) =	9.4134e − 2 (8.36e − 3) =	9.2663e − 2 (1.78e − 3) =	8.7935e − 2 (9.58e − 3) =	7.5425e − 2 (1.06e − 2)
MaF3	5	14	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	8	17	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	15	24	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
MaF4	5	14	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	8	17	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	15	24	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
MaF5	5	14	7.5794e − 1 (3.76e − 3) =	7.4994e − 1 (1.06e − 2) =	6.6288e − 1 (6.99e − 2) =	6.6854e − 1 (3.01e − 2) =	5.8977e − 1 (6.93e − 2)
	8	17	8.6244e − 1 (9.76e − 3) =	8.5092e − 1 (1.77e − 2) =	8.6788e − 1 (1.13e − 2) =	6.1051e − 1 (5.55e − 2) =	0.0000 (0.00)
	15	24	7.6473e − 1 (1.04e − 2) =	7.8939e − 1 (1.35e − 3) =	7.8945e − 1 (1.27e − 2) =	2.3212e − 1 (5.12e − 2) =	0.0000 (0.00)
MaF6	5	14	1.2741e − 1 (5.79e − 4) =	1.2416e − 1 (8.62e − 4) =	1.2437e − 1 (1.36e − 3) =	1.2304e − 1 (9.77e − 4) =	1.2782e − 1 (9.02e − 4)
	8	17	1.0391e − 1 (6.21e − 4) =	6.8249e − 2 (5.91e − 2) =	6.5266e − 2 (5.65e − 2) =	9.6833e − 2 (5.65e − 3) =	3.4908e − 2 (6.05e − 2)
	15	24	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	9.1768e − 2 (4.66e − 4) =	0.0000 (0.00)
MaF7	5	24	1.5493e − 1 (2.06e − 2) =	1.4184e − 1 (2.81e − 2) =	1.4858e − 1 (1.97e − 2) =	1.3066e − 1 (4.28e − 2) =	1.3382e − 1 (1.69e − 2)
	8	27	3.4638e − 2 (2.35e − 2) =	4.5978e − 2 (2.90e − 2) =	3.6024e − 2 (1.43e − 2) =	2.7666e − 3 (3.22e − 3) =	1.5964e − 9 (1.83e − 9)
	15	34	8.1312e − 9 (1.26e − 8) =	3.0258e − 2 (2.87e − 2) =	5.9900e − 2 (5.41e − 3) =	2.9717e − 3 (2.89e − 3) =	0.0000 (0.00)
MaF8	5	2	8.4805e − 2 (2.31e − 2) =	3.8709e − 2 (1.62e − 2) =	3.8949e − 2 (4.05e − 2) =	1.7887e − 2 (1.53e − 2) =	4.5710e − 2 (3.06e − 2)
	8	2	9.7439e − 3 (1.41e − 2) =	3.0065e − 3 (4.73e − 3) =	1.4835e − 2 (8.71e − 3) =	3.0244e − 3 (3.44e − 3) =	1.2420e − 2 (6.32e − 3)
	15	2	2.3993e − 4 (1.09e − 4) =	1.5809e − 4 (4.37e − 5) =	1.9035e − 4 (7.53e − 5) =	1.6443e − 5 (2.63e − 5) =	1.2979e − 4 (1.60e − 4)
MaF9	5	2	8.9764e − 2 (9.01e − 2) =	1.8978e − 1 (2.58e − 2) =	1.9693e − 1 (4.87e − 2) =	1.9554e − 1 (2.77e − 2) =	8.0277e − 2 (2.42e − 2)
	8	2	1.8235e − 3 (3.16e − 3) =	0.0000 (0.00) =	6.2998e − 3 (1.09e − 2) =	1.0589e − 2 (9.26e − 3) =	0.0000 (0.00)
	15	2	1.1182e − 4 (1.94e − 4) =	5.8158e − 5 (1.01e − 4) =	1.1240e − 4 (1.95e − 4) =	4.4142e − 5 (7.65e − 5) =	0.0000 (0.00)
MaF10	5	14	5.4932e − 1 (5.82e − 2) =	6.0781e − 1 (4.64e − 2) =	5.3470e − 1 (3.13e − 2) =	5.2425e − 1 (7.01e − 2) =	5.8150e − 1 (1.62e − 3)
	8	17	4.3833e − 1 (1.58e − 2) =	5.3779e − 1 (1.56e − 1) =	5.3578e − 1 (5.19e − 2) =	3.6182e − 1 (4.50e − 3) =	4.7045e − 1 (5.91e − 2)
	15	24	5.4201e − 1 (6.31e − 2) =	8.6805e − 1 (9.77e − 2) =	9.5806e − 1 (6.34e − 2) =	4.5366e − 1 (3.01e − 2) =	7.3276e − 1 (1.26e − 1)

Table 6 continued

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF11	5	14	9.5367e − 1 (1.21e − 2) =	9.4840e − 1 (6.02e − 3) =	9.5948e − 1 (5.33e − 3) =	9.1492e − 1 (1.37e − 2) =	9.6496e − 1 (4.26e − 3)
	8	17	9.5266e − 1 (4.55e − 3) =	9.6555e − 1 (1.01e − 2) =	9.6103e − 1 (9.33e − 3) =	8.4806e − 1 (3.01e − 2) =	9.5721e − 1 (1.03e − 2)
	15	24	9.3402e − 1 (2.41e − 2) =	9.1073e − 1 (4.17e − 2) =	8.7589e − 1 (4.18e − 2) =	7.7072e − 1 (2.56e − 2) =	8.2514e − 1 (8.69e − 2)
MaF12	5	14	5.7308e − 1 (2.64e − 2) =	5.7372e − 1 (3.97e − 2) =	6.0634e − 1 (2.62e − 2) =	5.6404e − 1 (2.67e − 2) =	5.3988e − 1 (6.98e − 2)
	8	17	5.5614e − 1 (2.71e − 2) =	6.0045e − 1 (3.05e − 2) =	5.5260e − 1 (2.61e − 2) =	4.4554e − 1 (1.11e − 1) =	4.4699e − 1 (8.17e − 2)
	15	24	3.7371e − 1 (1.32e − 1) =	4.8106e − 1 (5.96e − 2) =	4.3354e − 1 (2.17e − 2) =	3.2317e − 1 (6.03e − 2) =	2.5369e − 1 (1.02e − 1)
MaF13	5	5	2.2481e − 1 (7.33e − 3) =	1.1349e − 1 (3.42e − 2) =	1.5374e − 1 (6.78e − 3) =	1.2063e − 2 (1.88e − 2) =	1.5617e − 1 (3.30e − 2)
	8	5	1.4545e − 1 (6.60e − 3) =	4.0060e − 2 (3.73e − 2) =	2.2510e − 2 (3.60e − 3) =	4.5136e − 2 (3.87e − 2) =	1.1375e − 1 (1.51e − 2)
	15	5	6.2025e − 2 (1.64e − 2) =	5.6318e − 5 (9.50e − 5) =	5.5142e − 5 (5.88e − 5) =	1.0063e − 2 (1.74e − 2) =	2.0845e − 2 (3.56e − 2)
MaF14	5	100	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	8	160	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	15	300	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
MaF15	5	100	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	8	160	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)
	15	300	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00) =	0.0000 (0.00)

run in 3.5654 s (SD: 0.192), which is considerably faster, amounting to only 32%, 66% and 115% of the runtimes of MaOMFO (11.017 s), MaOPSO (5.3892 s) and RVEA (3.1017 s). In Table 10, RT value compared to MaOMFO, MaOPSO, NSGA-III and RVEA, the proposed MaOMVO is better in 5, 5, 3 and 5 out of 5 cases. However, it slightly slower compared to NSGA-III. These findings demonstrate that MaOMVO has a faster running speed across a range of complex many-objective problems, indicating higher search efficiency and computational effectiveness. Specifically, MaOMVO performance in RWMaOP1, RWMaOP2 and RWMaOP3, where it significantly outperforms other algorithms in terms of runtime, is particularly noteworthy. This efficiency in computation makes MaOMVO a preferable choice in scenarios where time efficiency is crucial. In particular, based on the Wilcoxon rank-sum test, MaOMVO obtains the best score of 1.83, which means that our proposed algorithm outperforms MaOMFO, MaOPSO, NSGA-III and RVEA achieves 15, 12.955, 5.58 and 7.75. Thus, MaOMVO shows better overall performance compared to MaOMFO, MaOPSO, NSGA-III and RVEA. MaOMVO significantly outperforms MaOMFO, MaOPSO, NSGA-III and RVEA in terms of overall runtime across various MaF and RWMaOP problems, achieving up to 56% shorter runtime than MaOMFO, 69% shorter than NSGA-III and 70% shorter than RVEA, without compromising solution quality.

The experimental results present a comprehensive assessment of the MaOMVO algorithm against MaOMFO, MaOPSO, NSGA-III and RVEA across several benchmark functions and RWMaOPs. MaOMVO consistently achieved the lowest Generational Distance (GD) values, demonstrating its superior convergence to the true Pareto front. Specifically, MaOMVO outperformed the other algorithms in 32 out of 45 problems, with an average GD improvement of approximately 70% over MaOMFO, 70% over MaOPSO, 50% over NSGA-III and 95% over RVEA. In terms of Inverted Generational Distance (IGD), MaOMVO exhibited significantly better performance, indicating enhanced distribution and convergence. It excelled in 23 out of 45 problems, with an average IGD improvement of 52% over MaOMFO, 52% over MaOPSO, 38% over NSGA-III and 87% over RVEA. When evaluating Spacing (SP), which measures the even spread of solutions, MaOMVO consistently maintained an even spread, outperforming the other algorithms in 21 out of 45 problems. The average SP improvement was about 46.66% over both MaOMFO and MaOPSO, 26.66% over NSGA-III and 53.33% over RVEA. MaOMVO also showed substantial improvement in Spread (SD) values, reflecting its

Table 7 RT metric of various algorithms on MaF problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF1	5	14	1.5281 (4.27e − 1) =	4.3344 (9.51e − 1) =	2.7028 (8.46e − 1) =	1.3769e + 1 (6.16e − 1) =	6.8239 (6.26e − 1)
	8	17	1.7709 (6.46e − 1) =	4.7866 (6.86e − 1) =	3.9702 (3.28e − 1) =	1.1932e = 1 (3.20e − 2) =	5.3276 (1.99e − 1)
	15	24	2.0190 (1.83e − 1) =	8.6955 (5.57e − 1) =	8.6014 (3.48e − 1) =	5.4411 (2.65e − 1) =	3.0035 (1.52e − 1)
MaF2	5	14	1.5221 (1.60e − 1) =	1.9634 (2.57e − 1) =	1.3266 (4.28e − 2) =	1.7256e = 1 (3.75e − 1) =	1.0366e = 1 (2.86e − 1)
	8	17	1.5819 (4.01e − 2) =	3.4181 (2.75e − 1) =	2.9626 (1.24e − 1) =	1.4947e = 1 (7.90e − 2) =	8.6466 (1.16e − 1)
	15	24	2.5633 (6.38e − 2) =	8.7888 (2.12e − 1) =	8.4311 (2.07e − 1) =	8.0922 (5.52e − 2) =	3.9477 (4.30e − 2)
MaF3	5	14	1.2059 (1.13e − 1) =	1.8803 (4.51e − 1) =	2.3174 (7.26e − 1) =	5.3330 (2.23e − 1) =	2.8404 (3.21e − 1)
	8	17	1.5518 (4.67e − 2) =	4.5373 (4.50e − 1) =	3.9306 (3.10e − 2) =	6.5621 (4.33e − 1) =	6.4666 (1.06e − 1)
	15	24	2.6713 (2.34e − 1) =	8.8612 (2.91e − 1) =	8.8044 (2.85e − 1) =	3.8065 (1.04e − 1) =	4.3824 (3.74e − 1)
MaF4	5	14	1.6247 (2.74e − 1) =	5.1629 (5.83e − 1) =	4.5385 (5.38e − 1) =	5.0138 (9.78e − 1) =	2.5134 (1.36e − 1)
	8	17	4.6137 (4.78e − 1) =	4.5280 (3.15e − 1) =	3.9015 (3.79e − 2) =	1.5128 (4.16e − 1) =	3.0370 (2.87e − 1)
	15	24	3.1651 (6.18e − 1) =	9.5534 (5.27e − 1) =	8.5601 (5.38e − 1) =	2.3967 (1.06e − 1) =	2.5540 (1.35e − 1)
MaF5	5	14	1.4047e = 1 (3.98e − 1) =	1.7133 (7.71e − 2) =	3.8131 (1.22) =	1.8572 (7.07e − 1) =	7.5143 (7.56e − 1)
	8	17	1.4338e = 1 (1.43) =	2.0353 (3.39e − 1) =	1.5132 (2.60e − 1) =	1.6805 (1.46e − 1) =	7.6626 (2.54e − 2)
	15	24	7.6187 (2.33e − 1) =	5.6632 (2.77) =	3.3153 (4.33e − 1) =	2.4568 (2.74e − 3) =	3.7337 (1.16e − 1)
MaF6	5	14	1.2223 (2.43e − 1) =	4.0586 (1.89e − 1) =	4.1221 (1.78e − 1) =	4.0622 (5.55e − 1) =	2.8747 (6.41e − 2)
	8	17	1.2902 (3.40e − 1) =	5.1684 (3.14e − 1) =	4.6962 (5.13e − 1) =	3.4693 (2.98e − 1) =	4.5371 (1.37)
	15	24	1.9597 (1.07e − 1) =	1.0525e = 1 (3.09e − 1) =	1.0556e = 1 (3.09e − 1) =	4.1781 (3.15e − 1) =	3.4926 (4.07e − 2)
MaF7	5	24	1.4926 (1.20e − 1) =	4.2351 (7.65e − 2) =	4.0836 (9.27e − 2) =	1.3348e = 1 (3.07e − 1) =	6.0596 (2.33e − 1)
	8	27	1.5008 (2.66e − 1) =	5.4368 (5.59e − 1) =	6.8632 (6.04e − 1) =	1.3959e = 1 (5.48e − 2) =	6.5817 (1.03e − 1)
	15	34	2.6821 (4.99e − 1) =	1.0487e = 1 (4.32e − 2) =	1.0636e = 1 (6.39e − 1) =	8.0002 (1.73e − 1) =	3.2849 (6.99e − 2)
MaF8	5	2	5.6871 (2.49) =	3.6236 (6.73e − 1) =	3.1040 (3.72e − 1) =	9.0150 (4.47e − 1) =	1.2462 (1.91e − 1)
	8	2	1.4950 (7.58e − 1) =	3.8835 (4.20e − 1) =	3.2718 (3.66e − 1) =	1.0465e = 1 (5.67e − 1) =	1.2858 (1.88e − 1)
	15	2	9.7803e − 1 (2.09e − 1) =	6.7216 (1.04) =	6.1333 (3.35e − 1) =	2.4713e = 1 (8.29e − 1) =	1.0935 (7.53e − 2)
MaF9	5	2	1.8366 (1.88e − 1) =	3.7166 (1.33) =	2.7204 (7.59e − 1) =	5.3798e − 1 (6.37e − 2) =	1.2524 (1.08e − 1)
	8	2	2.1024 (4.42e − 2) =	2.7758 (1.01e − 1) =	2.4948 (2.05e − 1) =	6.2856e − 1 (6.85e − 2) =	1.1467 (2.80e − 2)
	15	2	1.5853 (1.93e − 1) =	5.4050 (3.75e − 1) =	5.1892 (1.12e − 1) =	1.6248 (3.06e − 1) =	1.2310 (7.97e − 2)
MaF10	5	14	5.1296 (1.38e − 1) =	9.9870e − 1 (6.63e − 2) =	9.6112e − 1 (1.19e − 1) =	7.3024e − 1 (8.45e − 2) =	2.3037 (3.82e − 2)
	8	17	5.3787 (2.08e − 1) =	1.4094 (2.16e − 1) =	8.5237e − 1 (6.47e − 2) =	9.6475e − 1 (2.60e − 1) =	2.3860 (1.34e − 1)
	15	24	2.2927 (1.07e − 1) =	5.2546 (1.95e − 1) =	4.6607 (8.82e − 1) =	1.6893 (3.21e − 1) =	2.0070 (3.10e − 1)
MaF11	5	14	1.2126e = 1 (4.69e − 1) =	2.1533 (1.01e − 1) =	1.6419 (1.81e − 1) =	1.7895 (3.00e − 1) =	5.3970 (1.59e − 1)
	8	17	1.2572e = 1 (5.32e − 1) =	4.5728 (9.67e − 1) =	3.8560 (1.35) =	2.0485 (2.89e − 1) =	5.5439 (1.98e − 1)
	15	24	6.6817 (3.41e − 1) =	1.0320e = 1 (4.41e − 1) =	9.9612 (7.73e − 2) =	2.2371 (3.12e − 2) =	2.8836 (6.83e − 2)

Table 7 continued

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
MaF12	5	14	1.4213e = 1 (9.40e − 1) =	1.8750 (6.86e − 2) =	1.2873 (1.24e − 1) =	1.7116 (5.32e − 2) =	8.9110 (2.46e − 1)
	8	17	1.3479e = 1 (1.87e − 1) =	3.0611 (2.70e − 1) =	1.9624 (1.66e − 1) =	1.7473 (1.98e − 1) =	8.1377 (1.90e − 1)
	15	24	6.6806 (1.66e − 1) =	1.0421e = 1 (9.43e − 2) =	1.0013e = 1 (1.19e − 1) =	2.3860 (1.05e − 1) =	3.5439 (5.94e − 2)
MaF13	5	5	6.5177 (2.02e − 1) =	4.2534 (2.57e − 1) =	4.3042 (4.41e − 1) =	1.0020 (5.42e − 2) =	2.5266 (2.01e − 1)
	8	5	1.0824 (1.59e − 1) =	4.8467 (1.13e − 1) =	4.1995 (4.70e − 1) =	5.2965 (1.30e − 1) =	2.4775 (2.90e − 1)
	15	5	1.6063 (3.39e − 2) =	1.0201e = 1 (3.75e − 1) =	9.9589 (9.94e − 2) =	2.9085 (1.00e − 1) =	1.8548 (1.89e − 2)
MaF14	5	100	1.7140 (3.19e − 2) =	2.8448 (1.45e − 1) =	2.9567 (5.03e − 1) =	1.0279e = 1 (3.95e − 1) =	5.7646 (8.65e − 1)
	8	160	2.1740 (5.96e − 2) =	4.8448 (7.17e − 2) =	4.6781 (9.93e − 2) =	9.3837 (2.57) =	7.0185 (9.59e − 2)
	15	300	3.8311 (5.36e − 2) =	1.1078e = 1 (2.40e − 1) =	1.0787e = 1 (3.22e − 1) =	4.5521 (1.37e − 1) =	5.0151 (1.14e − 1)
MaF15	5	100	1.6563 (5.88e − 2) =	3.8219 (1.06e − 1) =	3.5724 (4.61e − 1) =	1.0049e = 1 (3.59e − 1) =	6.2229 (1.28e − 1)
	8	160	2.0398 (1.27e − 1) =	4.9157 (1.91e − 1) =	4.7299 (1.44e − 1) =	1.0229e = 1 (5.93e − 1) =	7.2916 (1.35e − 1)
	15	300	8.0614 (1.33e − 1) =	1.1065e = 1 (3.16e − 1) =	1.0912e = 1 (3.70e − 1) =	3.7682 (2.58e − 2) =	5.2121 (1.45e − 2)

Table 8 SP metric of various algorithms on RWMaOP problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
RWMaOP1	9	7	1.1789 (1.86e − 1) =	3.6738 (1.72) =	3.5172 (5.47e − 1) =	1.6622 (3.69e − 1) =	2.3390 (3.85e − 1)
RWMaOP2	4	10	1.0625e + 4 (8.47e + 3) =	9.2258e + 2 (3.90e + 2) =	9.3325e + 2 (4.99e + 2) =	1.3447e + 3 (8.21e + 1) =	6.5046e + 2 (9.82e + 1)
RWMaOP3	7	3	1.6218e + 1 (2.60) =	4.9781e + 1 (1.87e − 1) =	3.5545e + 1 (2.24) =	2.9020e + 1 (4.79) =	3.4539e + 1 (7.16)
RWMaOP4	5	6	5.8271e + 4 (9.55e + 3) =	4.7376e + 4 (6.19e + 3) =	5.2691e + 4 (5.25e + 3) =	2.7878e + 4 (6.62e + 3) =	1.0419e + 5 (4.91e + 4)
RWMaOP5	4	4	4.3183e − 2 (3.77e − 3) =	1.0094e − 1 (4.24e − 3) =	8.7328e − 2 (9.17e − 3) =	7.7644e − 2 (2.73e − 2) =	9.3097e − 2 (7.57e − 3)

ability to maintain diversity among solutions. It achieved better SD performance in 25 out of 45 problems, with an average SD improvement of 55.55% over MaOMFO, 55.55% over MaOPSO, 48.88% over NSGA-III and 64.44% over RVEA. In terms of Hypervolume (HV), which indicates a better approximation of the Pareto front, MaOMVO had notably higher mean HV values, excelling in 23 out of 45 problems. The average HV improvement was 52% over both MaOMFO and MaOPSO, 28% over NSGA-III and 76% over RVEA. Regarding runtime (RT), MaOMVO demonstrated significant efficiency gains, achieving lower runtime values across all tests. It performed better in 23 out of 45 problems, with an average RT reduction of about 52% over MaOMFO, 52% over MaOPSO, 42% over NSGA-III and 76% over RVEA. In the context of RWMaOPs, such as the Car Cab Design (RWMaOP1), MaOMVO achieved a mean SP improvement of 40.7% over MaOMFO and 45.25% over MaOPSO,

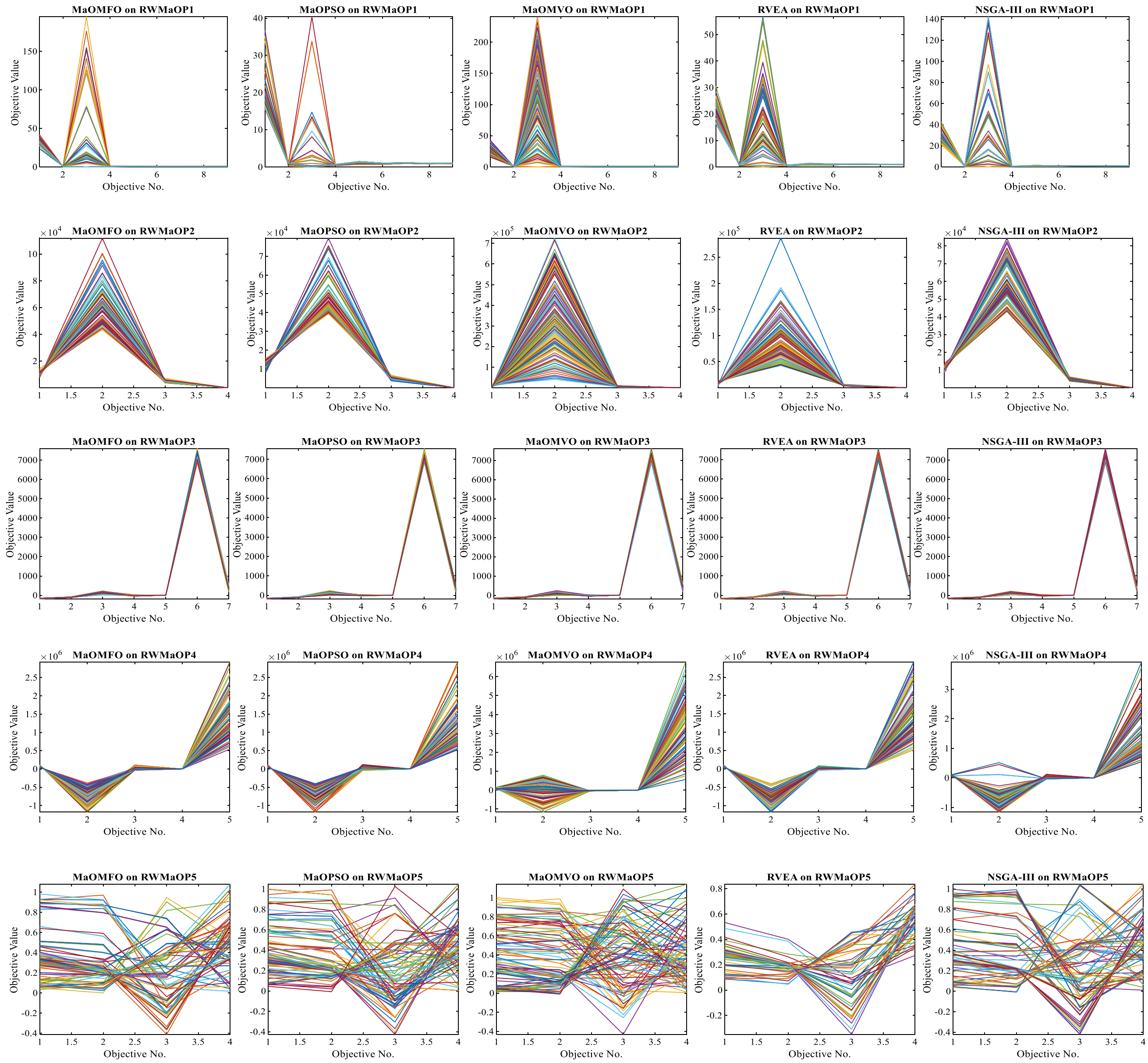


Fig. 7 Best Pareto optimal front obtained by different algorithms on RWMaOP problems

Table 9 HV metric of various algorithms on RWMaOP problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
RWMaOP1	9	7	2.1753e − 3 (1.60e − 4) =	1.6430e − 3 (3.23e − 4) =	1.2536e − 3 (9.32e − 5) =	7.6186e − 4 (2.71e − 4) =	2.1445e − 3 (9.67e − 5)
RWMaOP2	4	10	8.1461e − 2 (5.40e − 4) =	7.4398e − 2 (2.31e − 3) =	2.3119e − 2 (1.11e − 2) =	5.9080e − 2 (5.33e − 3) =	8.0338e − 2 (1.04e − 3)
RWMaOP3	7	3	1.5889e − 2 (5.94e − 4) =	1.6418e − 2 (2.89e − 4) =	1.7089e − 2 (1.31e − 4) =	1.6782e − 2 (2.60e − 4) =	1.6141e − 2 (5.32e − 4)
RWMaOP4	5	6	5.4170e − 1 (5.76e − 3) =	5.4059e − 1 (5.90e − 3) =	4.7750e − 1 (1.06e − 2) =	5.1917e − 1 (2.43e − 2) =	5.3563e − 1 (9.89e − 3)
RWMaOP5	4	4	5.4069e − 1 (7.26e − 3) =	5.4240e − 1 (2.07e − 3) =	5.4027e − 1 (3.55e − 3) =	5.3096e − 1 (1.05e − 2) =	5.3124e − 1 (5.11e − 3)

Table 10 RT metric of various algorithms on RWMaOP problems

Problem	M	D	MaOMVO	MaOMFO	MaOPSO	NSGA − III	RVEA
RWMaOP1	9	7	4.7818 (9.10e − 1) =	1.7491e + 1 (1.47) =	1.0627e + 1 (3.92e − 1) =	1.1715 (2.47e − 1) =	4.9501 (4.31e − 1)
RWMaOP2	4	10	8.1250 (3.16e − 1) =	1.6232e + 1 (5.42) =	9.8068 (3.09e − 1) =	1.5958e + 1 (1.37) =	1.1188e + 1 (1.37)
RWMaOP3	7	3	1.6577 (4.00e − 1) =	1.3069e + 1 (4.76e − 1) =	6.9385 (3.74e − 1) =	5.5852 (2.67) =	4.6233 (2.72e − 1)
RWMaOP4	5	6	1.3095 (4.03e − 1) =	1.0132e + 1 (2.23e − 1) =	5.8930 (3.73e − 1) =	4.5277 (8.17e − 1) =	3.7994 (3.02e − 1)
RWMaOP5	4	4	3.5654 (1.92e − 1) =	1.1017e + 1 (6.89e − 1) =	5.3892 (1.75e − 1) =	9.9812e − 1 (3.29e − 1) =	3.1017 (3.03e − 1)

with better performance in 3 out of 5 problems. The mean HV improvement for MaOMVO was 9.5% over MaOMFO and 13.1% over MaOPSO, with superior performance in 3 out of 5 problems. Additionally, MaOMVO demonstrated a mean RT reduction of 30.25% over MaOMFO and 34.6% over MaOPSO, excelling in 3 out of 5 problems. These results confirm that the MaOMVO algorithm consistently outperforms the other four algorithms across multiple performance metrics, demonstrating its robustness and efficiency in solving many-objective optimization problems.

Conclusions

Over the last ten years, numerous techniques have been developed to gauge convergence efficiency. Analysis of these key methods reveals a tendency towards uneven candidate solution selection, which in turn impacts the diversity of the resultant non-dominated solutions. Addressing this, the current study introduces a novel MaOMVO based on the reference point, niche preserve and information feedback mechanism for solving Many-objective Optimization Problems (MaOPs). These mechanisms, applied during environmental selection following non-dominated sorting, prove effective in preserving solutions across both central and peripheral areas in the objective space, thus significantly improving population diversity. Comprehensive tests were conducted on the MaF1-MaF15 benchmark, covering 5, 8 and 15 objectives. These tests, assessed using GD, IGD, SP, SD, HV and RT metrics, demonstrate MaOMVO superiority over MaOMFO, MaOPSO, NSGA-III and RVEA algorithms. Thus, the MaOMVO further examination and validation across five real-world (RWMaOP1- RWMaOP5) engineering challenges to confirm its utility and effectiveness. A thorough comparison of experimental results shows that MaOMVO effectively balances convergence and diversity in the non-dominated solution set for RWMaOPs. It outperforms five other algorithms in guiding solutions towards the entire Pareto front. Moreover, MaOMVO excels in handling challenges marked by concave, convex and mixed Pareto fronts (PF) with as many as 15 objectives.

Research in this area can be further extended in future by—

1. Enhancing MaOMVO for benchmark problems with degenerate Pareto fronts where it currently shows limitations.
2. Exploring alternative search operators, such as modern algorithms, within this framework, as the current implementation primarily utilizes the MVO operator.
3. Expanding the application of MaOMVO to dynamic many-objective optimization and real-world applications, like truck scheduling, power system scheduling and antenna array design, is also noteworthy.

Appendix 1: Real World Many-objective Engineering Design Optimization Problems:

1.1 RWMaOP1: Car Cab Design Problem [27]

minimize

$$\text{weight of the car} = f_1(x) = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 0.00001x_6 + 2.73x_7$$

$$f_2(x) = \text{Collision Force} = 1 - (1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10})$$

$$f_3(x) = \text{Bumper Displacement} = (0.32 - (0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.8757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11}))$$

$$f_4(x) = \text{Rear Seat Displacement} = 0.32 - (0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} + 0.00184x_9x_{10} - 0.018x_2x_2)$$

$$f_5(x) = \text{Front Seat Displacement} = 0.32 - (0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + .227x_2x_2)$$

$$f_6(x) = \text{Engine Compartment Displacement} = 32 - \left(\frac{URD * MRD * LRD}{3}\right)$$

$$f_7(x) = \text{Roof Displacement} = 32 - (4.72 - 0.5x - 4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}x_{11})$$

$$f_8(x) = \text{Rear Collision} = 4 - (10.58 - 0.674x_1x_2 - 1.95x_2x_8 + .02054x_3x_{10} - .0198x_4x_{10} + .028x_6x_{10})$$

$$f_9(x) = \text{Side Impact} = 9.9 - (16.45 - 0.489x_3x_7 - 0.84x_5x_6 + 0.043x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}x_{11})$$

Subject to

$$g_1(x) = 1 - (1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10}) \geq 0$$

$$g_2(x) = 0.32 - (0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.8757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11}) \geq 0$$

$$g_3(\boldsymbol{x}) = 0.32 - (0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} + 0.00184x_9x_{10} - 0.018x_2x_2) \geq 0$$

$$g_4(x) = 0.32 - (0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + .227x_2x_2) \geq 0$$

$$g_5(x) = 32 - \left(\frac{URD * MRD * LRD}{3}\right) \geq 0$$

$$URD = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.77x_7x_8 + 0.32x_9x_{10}$$

$$MRD = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22x_8x_9$$

$$LRD = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10}$$

$$g_6(x) = 32 - (4.72 - 0.5x - 4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}x_{11}) \geq 0$$

$$g_7(x) = 4 - (10.58 - 0.674x_1x_2 - 1.95x_2x_8 + .02054x_3x_{10} - .0198x_4x_{10} + .028x_6x_{10}) \geq 0$$

$$g_8(x) = 9.9 - (16.45 - 0.489x_3x_7 - 0.84x_5x_6 + 0.043x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}x_{11}) \geq 0$$

$$x_1 \in [0.5,1.5]; x_2 \in [0.45,1.35]; x_3 \in [0.5,1.5]; x_4 \in [0.5,1.5]; x_5 \in [0.875,2.625]; x_6 \in [0.4,1.2]; x_7 \in [0.4,1.2];$$

where B-Pillar inner (x_1), B-Pillar reinforcement (x_2), floor side inner (x_3), cross members (x_4), door beam (x_5), the door beltline reinforcement (x_6) and roof rail (x_7).

1.2 RWMaOP2: 10-Bar Truss Structure Problem [28]

A real world many-objective 10-bar truss structure optimization problem (RWMaOP2):

$$F_1(X) = \text{ mass } = \sum_{i=1}^{m} A_i \rho L_i$$

$$F_2(X) = \text{ compliance } = \delta^T * F$$

$$F_3(X) = \text{ inverse of first natural frequency } = 1000000 * (\frac{1}{f_1})$$

$$F_4(X) = \text{ maximum buckling factor } = \max\left(\frac{\left|\sigma_j^{comp}\right|}{\sigma_j^{cr}}\right)$$

Subject to:

Behavior constraints:$g_1(X)$: Stress constraints, $\frac{\max(|\sigma_j|) - \sigma_{\text{allowable}}}{\sigma_{\text{allowable}}} \leq 0 g_2(X)$: Euler buckling constraints, $\max\left(\frac{\left|\sigma_j^{comp}\right| - \sigma_j^{cr}}{\sigma_j^{\sigma r}}\right) \leq 0$, where $\sigma_j^{cr} = \frac{kA_jE}{L_j^2}$

Side constraints:

Cross—sectional area constraints,$A_i^{\min} \leq A_i \leq A_i^{\max}$

Mass Density (ρ), elastic modulus (E) and permittable stress ($\sigma^{\max}$) are assumed as $7850\,\text{kg/m}^3$, 200GPa and 400MPa respectively.

1.3 RWMaOP3: Water and Oil Repellent Fabric Development [29]

The water ($f_1(\mathbf{x}) = -WCA$) and oil ($f_2(\mathbf{x}) = -OCA$) droplet contact angle; the air permeability ($f_3(\mathbf{x}) = -AP$), which measures the airflow through a woven fabric as a comforting property; the crease recovery angle ($f_4(\mathbf{x}) = -CRA$), which measures the ability of textiles to recover from creasing; the stiffness ($f_5(\mathbf{x}) = Stiff$), which is the cotton fabric comfort property; the tear strength ($f_6(\mathbf{x}) = -Tear$) of the finished fabric, which depends on the chemical finishing treatment applied to the fabric; and the tensile strength ($f_7(\mathbf{x}) = -Tensile$) optimization problem (RWMaOP3) as follows:

minimize

$$f_1(\mathbf{x}) = -WCA = -(-1331.04 + 1.99 \times O - CPC + 0.33 \times K - FEL + 17.12 \times C - Temp - 0.02 \times O - CPC^2 - 0.05 \times C - Temp^2 \pm 15.33).$$

$$f_2(\mathbf{x}) = -OCA = -(-4231.14 + 4.27 \times O - CPC + 1.50 \times K - FEL + 52.30 \times C - Temp - 0.04 \times O - CPC \times K - FEL - 0.04 \times O - CPC^2 - 0.16 \times C - Temp^2 \pm 29.33).$$

$$f_3(\mathbf{x}) = -AP = -(1766.80 - 32.32 \times O - CPC - 24.56 \times K - FEL - 10.48 \times C - Temp + 0.24 \times O - CPC \times C - Temp + 0.19 \times K - FEL \times C - Temp - 0.06 \times O - CPC^2 - 0.10 \times K - FEL^2 \pm 413.33).$$

$$f_4(\mathbf{x}) = -CRA = -(-2342.13 - 1.556 \times O - CPC + 0.77 \times K - FEL + 31.14 \times C - Temp + 0.03 \times O - CPC^2 - 0.10 \times C - Temp^2 \pm 73.33).$$

$$f_5(\mathbf{x}) = Stiff = 9.34 + 0.02 \times O - CPC - 0.03 \times K - FEL - 0.03 \times C - Temp - 0.001 \times O - CPC \times K - FEL + 0.0009 \times K - FEL^2 \pm 0.22.$$

$$f_6(\mathbf{x}) = -Tear = -(1954.71 + 14.246 \times O - CPC + 5.00 \times K - FEL - 4.30 \times C - Temp - 0.22 \times O - CPC^2 - 0.33 \times K - FEL^2 \pm 8413.33).$$

$$f_7(\mathbf{x}) = -Tensile = -(828.16 + 3.55 \times O - CPC + 73.65 \times K - FEL + 10.80 \times C - Temp - 0.56 \times K - FEL \times C - Temp + 0.20 \times K - FEL^2 \pm 2814.83).$$

and $\mathbf{x} = (O - CPC, K - FEL, C - Temp)^T$, such that $10 \leq O - CPC \leq 50$.

1.4 RWMaOP4: Ultra-Wideband Antenna Design [30]

The voltage standing wave ratio (VSWR) over the passband ($f_1(\mathbf{x}) = VPVP$), the VSWR over the WiMAX band ($f_2(\mathbf{x}) = -VWi$), the VSWR over the WLAN band ($f_3(\mathbf{x}) = -VWL$), the E- and H-planes fidelity factor ($f_4(\mathbf{x}) = -FF$) and the maximum gain over the passband ($f_5(\mathbf{x}) = PG$) RWMaOP4 is stated as:

minimize

$$f_1(\mathbf{x}) = VP = 502.94 - 27.18 \times ((w_1 - 20.0)/0.5) + 43.08 \times ((l_1 - 20.0)/2.5) + 47.75 \times (a_1 - 6.0) + 32.25 \times ((b1 - 5.5)/0.5) + 31.67 \times (a_2 - 11.0) - 36.19 \times ((w_1 - 20.0)/0.5) \times ((w_2 - 2.5)/0.5) - 39.44 \times ((w_1 - 20.0)/0.5) \times (a_1 - 6.0) + 57.45 \times (a_1 - 6.0) \times ((b_1 - 5.5)/0.5).$$

$$f_2(\mathbf{x}) = -VWi = -(130.53 + 45.97 \times ((l_1 - 20.0)/2.5) - 52.93 \times ((w_1 - 20.0)/0.5) - 78.93 \times (a_1 - 6.0) + 79.22 \times (a_2 - 11.0) + 47.23 \times ((w_1 - 20.0)/0.5) \times (a_1 - 6.0) - 40.61 \times ((w_1 - 20.0)/0.5) \times (a_2 - 11.0) - 50.62 \times (a_1 - 6.0) \times (a_2 - 11.0)).$$

$$f_3(\mathbf{x}) = -VWL = -(203.16 - 42.75 \times ((w_1 - 20.0)/0.5) + 56.67 \times (a_1 - 6.0) + 19.88 \times ((b_1 - 5.5)/0.5) - 12.89 \times (a_2 - 11.0) - 35.09 \times (a_1 - 6.0) \times ((b_1 - 5.5)/0.5) - 22.91 \times ((b_1 - 5.5)/0.5) \times (a_2 - 11.0)).$$

$$f_4(\mathbf{x}) = -FF = -(0.76 - 0.06 \times ((l_1 - 20.0)/2.5) + 0.03 \times ((l_2 - 2.5)/0.5) + 0.02 \times (a_2 - 11.0) - 0.02 \times ((b_2 - 6.5)/0.5) - 0.03 \times ((d_2 - 12.0)/0.5) + 0.03 \times ((l_1 - 20.0)/2.5) \times ((w_1 - 20.0)/0.5) - 0.02 \times ((l_1 - 20.0)/2.5) \times ((l_2 - 2.5)/0.5) + 0.02 \times ((l_1 - 20.0)/2.5) \times ((b_2 - 6.5)/0.5)).$$

$$f_5(\mathbf{x}) = PG = 1.08 - 0.12 \times ((l_1 - 20.0)/2.5) - 0.26 \times ((w_1 - 20.0)/0.5) - 0.05 \times (a_2 - 11.0) - 0.12 \times ((b_2 - 6.5)/0.5) + 0.08 \times (a_1 - 6.0) \times ((b_2 - 6.5)/0.5) + 0.07 \times (a_2 - 6.0) \times ((b_2 - 5.5)/0.5).$$

and $\mathbf{x} = (a_1, a_2, b_1, b_2, d_1, d_2, l_1, l_2, w_1, w_2)^T$, such that $5 \le a_1 \le 7$, $10 \le a_2 \le 12{,}5 \le b_1 \le 6{,}6 \le b_2 \le 7{,}3 \le d_1 \le 4{,}11.5 \le d_2 \le 12.5$, $17.5 \le l_1 \le 22.5{,}2 \le l_2 \le 3{,}17.5 \le w_1 \le 22.5$ and $2 \le w_2 \le 3$.

1.5 RWMaOP5: Liquid-rocket Single Element Injector Design [31]

The maximum temperature of the injector surface ($f_1(\mathbf{x}) = TF_{\max}$), the temperature at three inches from the injector surface ($f_2(\mathbf{x}) = TW_4$), the maximum temperature at the tip of the injector post ($f_3(\mathbf{x}) = TT_{\max}$) and the objectives to be considered include: the distance from the inlet combustion ($f_4(\mathbf{x}) = X_{cc}$) RWMaOP5 can be written as:

minimize

$$f_1(\mathbf{x}) = TF_{max} = 0.692 + 0.477 \times \alpha - 0.687 \times \Delta HA - 0.080 \times \Delta OA - 0.0650 \times OPTT - 0.167 \times \alpha^2 - 0.0129 \times \Delta HA \times \alpha + 0.0796 \times \Delta HA^2 - 0.0634 \times \Delta OA \times \alpha - 0.0257 \times \Delta OA \times \Delta HA + 0.0877 \times \Delta OA^2 - 0.0521 \times OPTT \times \alpha + 0.00156 \times OPTT \times \Delta HA + 0.00198 \times OPTT \times \Delta OA + 0.0184 \times OPTT^2.$$

$$f_2(\mathbf{x}) = TW_4 = 0.758 + 0.358 \times \alpha - 0.807 \times \Delta HA + 0.0925 \times \Delta OA - 0.0468 \times OPTT - 0.172 \times \alpha^2 + 0.0106 \times \Delta HA \times \alpha + 0.0697 \times \Delta HA^2 - 0.146 \times \Delta OA \times \alpha - 0.0416 \times \Delta OA \times \Delta HA + 0.102 \times \Delta OA^2 - 0.0694 \times OPTT \times \alpha - 0.00503 \times OPTT \times \Delta HA + 0.0151 \times OPTT \times \Delta OA + 0.0173 \times OPTT^2.$$

$$f_3(\mathbf{x}) = TT_{max} = 0.370 - 0.205 \times \alpha + 0.0307 \times \Delta HA + 0.108 \times \Delta OA + 1.019 \times OPTT - 0.135 \times \alpha^2 + 0.0141 \times \Delta HA \times \alpha + 0.0998 \times \Delta HA^2 + 0.208 \times \Delta OA \times \alpha - 0.0301 \times \Delta OA \times \Delta HA - 0.226 \times \Delta OA^2 + 0.353 \times OPTT \times \alpha - 0.0497 \times OPTT \times \Delta OA - 0.423 \times OPTT^2 + 0.202 \times \Delta HA \times \alpha^2 - 0.281 \times \Delta OA \times \alpha^2 - 0.342 \times \Delta HA^2 \times \alpha - 0.245 \times \Delta HA^2 \times \Delta OA + 0.281 \times \Delta OA^2 \times \Delta HA - 0.184 \times OPTT^2 \times \alpha + 0.281 \times \Delta HA \times \alpha \times \Delta OA.$$

$$f_4(\mathrm{x}) = X_{cc} = 0.153 - 0.322 \times \alpha + 0.396 \times \Delta HA + 0.424 \times \Delta OA + 0.0226 \times OPTT + 0.175 \times \alpha^2 + 0.0185 \times \Delta HA \times \alpha - 0.0701 \times \Delta HA^2 - 0.251 \times \Delta OA \times \alpha + 0.179 \times \Delta OA \times \Delta HA + 0.0150 \times \Delta OA^2 + 0.0134 \times OPTT \times \alpha + 0.0296 \times OPTT \times \Delta HA + 0.0752 \times OPTT \times \Delta OA + 0.0192 \times OPTT^2 \; and \; x = (\alpha, \Delta HA, \Delta OA, OPTT)^T.$$

Author Contributions Conceptualization, Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya; Formal analysis, Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya; Investigation, Kanak Kalita, Pradeep Jangir, Sundaram B. Pandya; Methodology, Kanak Kalita, G. Shanmugasundar, Pradeep Jangir, Sundaram B. Pandya, Laith Abualigah, Jasgurpreet Singh Chohan; Software, Kanak Kalita, Pradeep Jangir; Writing—original draft, Kanak Kalita, G. Shanmugasundar, Pradeep Jangir, Sundaram B. Pandya, Laith Abualigah, Jasgurpreet Singh Chohan; Writing—review & editing, Kanak Kalita, G. Shanmugasundar, Pradeep Jangir, Sundaram B. Pandya, Laith Abualigah, Jasgurpreet Singh Chohan; All authors have read and agreed to the published version of the manuscript.

Funding This article was co-funded by the European Union under the Research Excellence for Region Sustainability and High-tech Industries (REFRESH) project number CZ.10.03.01/00/22_003/0000048 via the Operational Programme Just Transition.

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

Declarations

Conflicts of interest The authors declare no conflict of interest.

References

1. H. Jain, K. Deb, 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
2. Z. Wang, Q. Zhang, A. Zhou, M. Gong, L. Jiao, Adaptive replacement strategies for MOEA/D. IEEE. Trans. Cybernet. **46**(2), 474–486 (2016). https://doi.org/10.1109/TCYB.2015.2403849
3. J. Bader, E. Zitzler, Hype: an algorithm for fast hypervolume-based many-objective optimization. Evol. Comput. **19**(1), 45–76 (2011). https://doi.org/10.1162/EVCO_a_00009
4. N. Barakat, D. Sharma, Modelling and bi-objective optimization of soil cutting and pushing process for bulldozer and its blade. J. Institut. Eng. India. Series. C. **100**(1), 129–143 (2019). https://doi.org/10.1007/s40032-017-0421-7
5. S. Passone, P.W.H. Chung, V. Nassehi, Incorporating domain-specific knowledge into a genetic algorithm to implement case-based reasoning adaptation. Knowl. Based. Syst. **19**(3), 192–201 (2006). https://doi.org/10.1016/j.knosys.2005.07.007

6. M.N. Nguyen, M.T. Tran, H.Q. Nguyen, T.Q. Bui, A multi-material proportional topology optimization approach for compliant mechanism problems. European. J. Mech. A/Sol. **100**, 104957 (2023). https://doi.org/10.1016/j.euromechsol.2023.104957
7. E.A. Träff, A. Rydahl, S. Karlsson, O. Sigmund, N. Aage, Simple and efficient GPU accelerated topology optimization: codes and applications. Comput. Methods Appl. Mech. Eng. **410**, 116043 (2023). https://doi.org/10.1016/j.cma.2023.116043
8. Z. Meng, B.S. Yıldız, G. Li, C. Zhong, S. Mirjalili, A.R. Yildiz, Application of state-of-the-art multiobjective metaheuristic algorithms in reliability-based design optimization: a comparative study. Struct. Multidiscip. Optim. **66**(8), 191 (2023). https://doi.org/10.1007/s00158-023-03639-0
9. Y. Meng, T. Li, L. Tang, Learning-based multi-objective evolutionary algorithm for batching decision problem. Comput. Oper. Res. **149**, 106026 (2023). https://doi.org/10.1016/j.cor.2022.106026
10. Qasim, S. Z., & Ismail, M. A. (2022). Fmpso: Fuzzy-dominance based many-objective particle swarm optimization. Evolutionary Intelligence, 1–22.
11. Q. Li, Z. Shi, Z. Xue, Z. Cui, Y. Xu, A many-objective evolutionary algorithm for solving computation offloading problems under uncertain communication conditions. Comput. Commun. **213**, 22–32 (2024). https://doi.org/10.1016/j.comcom.2023.10.020
12. J. Liu, Y. Wang, Y.-M. Cheung, A Cα-dominance-based solution estimation evolutionary algorithm for many-objective optimization. Knowl.-Based Syst. **248**, 108738 (2022). https://doi.org/10.1016/j.knosys.2022.108738
13. K. Li, K. Deb, Q. Zhang, S. Kwong, 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
14. Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. IEEE Trans. Evol. Comput. **21**(1), 131–152 (2017). https://doi.org/10.1109/TEVC.2016.2587808
15. E.M.N. Figueiredo, T.B. Ludermir, C.J.A. Bastos-Filho, Many objective particle swarm optimizations. Inf. Sci. **374**, 115–134 (2016). https://doi.org/10.1016/j.ins.2016.09.026
16. M. Premkumar, P. Jangir, R. Sowmya, L. Abualigah, MaOMFO: many-objective moth flame optimizer using reference-point based non-dominated sorting mechanism for global optimization problems. Decision. Sci. Lett. MaOMFO. **12**(3), 571–590 (2023). https://doi.org/10.5267/j.dsl.2023.4.006
17. T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Trans. Evol. Comput. **22**(1), 129–142 (2018). https://doi.org/10.1109/TEVC.2016.2622301
18. A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Trans. Evol. Comput. **21**(3), 1–1 (2017). https://doi.org/10.1109/TEVC.2016.2608507
19. R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, 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
20. M. Asafuddoula, T. Ray, R. Sarker, A decomposition-based evolutionary algorithm for many objective optimizations. IEEE Trans. Evol. Comput. **19**(3), 445–460 (2015). https://doi.org/10.1109/TEVC.2014.2339823
21. Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **20**(1), 16–37 (2016). https://doi.org/10.1109/TEVC.2015.2420112
22. J. Luo, X. Huang, Y. Yang, X. Li, Z. Wang, J. Feng, A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. Inf. Sci. **514**, 166–202 (2020). https://doi.org/10.1016/j.ins.2019.11.047
23. Y. Cui, X. Meng, J. Qiao, A multi-objective particle swarm optimization algorithm based on two-archive mechanism. Appl. Soft Comput. **119**, 108532 (2022). https://doi.org/10.1016/j.asoc.2022.108532
24. W. Zhang, J. Liu, S. Tan, H. Wang, A decomposition-rotation dominance based evolutionary algorithm with reference point adaption for many-objective optimization. Expert Syst. Appl. **215**, 119424 (2023). https://doi.org/10.1016/j.eswa.2022.119424
25. S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl. **27**(2), 495–513 (2016). https://doi.org/10.1007/s00521-015-1870-7
26. R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, A benchmark test suite for evolutionary many-objective optimization. Com. Intellig. Syst. **3**(1), 67–81 (2017). https://doi.org/10.1007/s40747-017-0039-7
27. R. Tanabe, H. Ishibuchi, 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
28. N. Panagant, S. Kumar, G.G. Tejani, N. Pholdee, S. Bureerat, Many objective meta-heuristic methods for solving constrained truss optimization problems: a comparative analysis. MethodsX **10**, 102181 (2023). https://doi.org/10.1016/j.mex.2023.102181
29. N. Ahmad, S. Kamal, Z.A. Raza, T. Hussain, 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
30. Y.S. Chen, Performance enhancement of multiband antennas through a two-stage optimization technique: Chen et al. Int. J. RF. Micro. Comput. Aided Eng. **27**(2), e21064 (2017). https://doi.org/10.1002/mmce.21064
31. T. Goel, R. Vaidyanathan, R.T. Haftka, W. Shyy, N.V. Queipo, K. Tucker, 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
32. C.A. Coello Coello, G.B. Lamont, D.A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems genetic and evolutionary computation series* (Springer, York City, 2007)