Results in Engineering 28 (2025) 106941

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

Many-Objective Grey Wolf Optimizer (MaOGWO) for solving real-world problems

Kanak Kalita [a,b,*], Pradeep Jangir [c,d], Lenka Čepová [e], Shankar Chakraborty [f], Sundaram B. Pandya [g], Arpita [h]

[a] Department of Mechanical Engineering, Rajalakshmi Institute of Technology, Chennai, 600124, India
[b] Jadara University Research Center, Jadara University, Irbid, 21110, Jordan
[c] Department of Electronics and Communication Engineering, Chandigarh University, Mohali, 140413, India
[d] Applied Science Research Center, Applied Science Private University, Amman, 11931, Jordan
[e] Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 70800 Ostrava, Czech Republic
[f] Department of Production Engineering, Jadavpur University, Kolkata, India
[g] Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392 001, India
[h] Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India

ARTICLE INFO

Keywords:
Many-objective optimization
Grey wolf optimizer
MaOGWO
Reference point
DTLZ

ABSTRACT

The Grey Wolf Optimizer (GWO) is an effective optimization algorithm that primarily focuses on single objective optimization and is characterized by its simplicity and fast convergence. However, adapting GWO for many-objective optimization requires addressing two critical challenges: keeping the solutions in sync while also promoting variety. In this study, a new Many-Objective Grey Wolf Optimizer (MaOGWO) is proposed which includes reference point strategies, niche preservation and information feedback to enhance convergence and diversity. A comprehensive comparative analysis was conducted to evaluate MaOGWO against four leading many-objective optimization algorithms: Some of them are Non-dominated Sorting Genetic Algorithm III (NSGA-III), Many-Objective Particle Swarm Optimizer (MaOPSO), Many-Objective Teaching Learning-Based Optimizer (MaOTLBO) and Many-Objective Gradient-Based Optimizer (MaOGBO). We evaluated the performance of these algorithms on the DTLZ1-DTLZ7 problem sets with 5, 10 and 15 objectives and five real-world many-objective optimization problems, namely RWMaOP1-RWMaOP5. The quality assessment metrics used in this research were generational distance (GD), inverted generational distance (IGD), spacing (SP), spread (SD), hypervolume (HV) and runtime (RT). The experimental outcomes revealed that the MaOGWO dominated the other algorithms in terms of convergence to the Pareto-optimal front and solution diversity for both synthetic and practical problems with lower computational costs.

1. Introduction

Multi-objective evolutionary algorithms (MOEAs) are widely utilized in addressing multi-objective optimization problems (MOPs). These problems are prevalent in scientific research and engineering, involving several conflicting goals that require simultaneous optimization. MOPs comprise multiple objectives that often conflict, necessitating concurrent optimization. A general mathematical formulation of an MOP is illustrated in Eq. (1).

$$\begin{aligned} &min\ F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ &s.t. x = (x_1, x_2, \ldots, x_n) \in D \subset R^m \end{aligned} \tag{1}$$

In this context, $x = (x_1, x_2, \ldots, x_n)$ represents the multi-dimensional vectors consisting of decision variables within the decision space D, with n being the count of decision variables in that space. The function $F(x)$ maps from the decision space D to the m-dimensional objective space R^m, where m represents the number of objectives to be optimized. Each optimized objective function is denoted by $f_i(x)$, for $i = \{1,2,\ldots,m\}$. multi-objective typically refers to problems with 2–3 objectives, while many-objective refers to problems with four or more objectives, which

* Corresponding author.
E-mail addresses: kanakkalita02@gmail.com, drkanakkalita@veltech.edu.in (K. Kalita), pkjmtech@gmail.com (P. Jangir), s_chakraborty00@yahoo.co.in (S. Chakraborty), sundarampandya@gmail.com (S.B. Pandya), apyjangid@gmail.com (Arpita).

https://doi.org/10.1016/j.rineng.2025.106941
Received 9 November 2024; Received in revised form 9 July 2025; Accepted 25 August 2025
Available online 30 August 2025

pose greater challenges due to scalability and diversity maintenance. Problems that encompass more than three objectives are referred to as many-objective optimization problems (MaOPs) [1], as shown in Fig. 1.

A solution is deemed Pareto non-dominated if there is no other solution that is superior in all aspects [2]. In the decision space, the collection of all the Pareto optimal solutions forms the optimal Pareto set (Optimal PS), while their corresponding objective vectors constitute the optimal Pareto front (Optimal PF) [3]. The inherent conflicts among objectives in a multi-objective optimization problem mean that enhancing one objective often leads to a decline in others [4]. Consequently, it is impossible to find out a single solution that optimizes all the objectives [4]. Instead, the aim is to identify a spectrum of compromise solutions, balancing different objectives. This set of compromises in the decision space is known as the Pareto set (PS) and its projection in the objective space is termed the Pareto front (PF) [3].

In the quest for the most effective Pareto-optimal solution sets, numerous MOEAs have been developed [2,5]. While these algorithms show proficiency in dealing with MOPs with two or three objectives, they encounter significant hurdles with MaOPs involving more than three objectives. This challenge primarily arises from the curse of dimensionality. Specifically, as the number of objectives escalates, the fraction of non-dominated solutions within a population surges dramatically. Traditional dominance-based ranking methods thus lose efficacy in MaOPs, leading to solutions that poorly approximate the optimal Pareto Front (PF) of a problem. Another critical issue in this context is maintaining diversity, especially given the complexity of measuring distances in high-dimensional spaces [6]. To counter these challenges, a variety of many-objective evolutionary algorithms (MaOEAs) have been introduced. These algorithms are generally categorized into three distinct groups.

The initial category involves implementation of novel Pareto dominance relations, aimed at amplifying selection pressure through various advanced dominance relations. Several innovative dominance relations have been introduced as alternatives to traditional Pareto dominance, including fuzzy dominance [7], reference point (RP)-based dominance [8], angle dominance (AD) [9] and strengthened dominance [10]. These methods tend to outperform conventional Pareto dominance techniques in solving MaOPs. Recently, a controlled strengthened dominance relation was proposed by Shen et al. [11] by dynamically adjusting dominance pressure based on generation count to improve convergence-diversity balance. Similarly, RP dominance, by establishing stringent distribution protocols for non-dominated solutions via a set of evenly distributed reference points, enhances effectiveness of

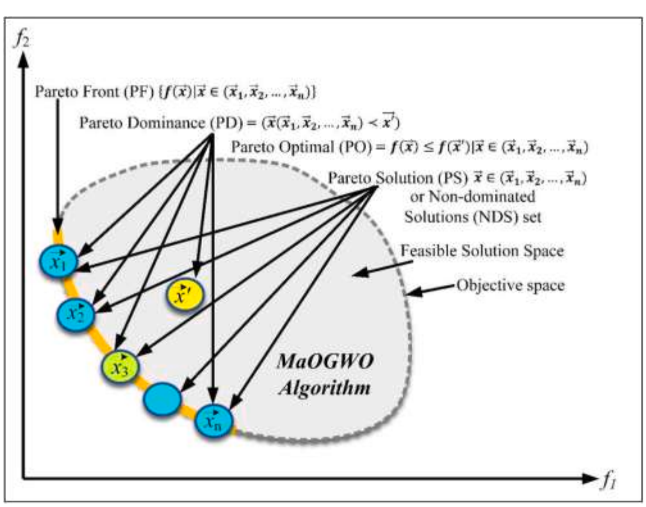

Fig. 1. Representation of the Pareto solutions in search space of MaO-problem.

non-dominated sorting genetic algorithm (NSGA-II) [8] in benchmark tests and practical applications. The strengthened dominance relation (SDR) [10] employs a niche-based strategy to maintain a balance between convergence and diversity, calculating angles between solutions and retaining only those with optimal convergence within each niche. A dual distance dominance-based algorithm by Zhang et al. [12] introduced an adaptive dominance measure along with a niche-based strategy and selection-replacement operator to enhance both convergence and diversity. Experimental results suggest that SDR-enhanced NSGA-II is effective in MaOP scenarios. However, it is important to note that these dominance relations prioritize convergence over diversity and require additional parameters to regulate the dominance region scope.

The second category encompasses decomposition-based methods, focusing on dividing a MaOP into multiple subproblems that are subsequently optimized collectively in a synergistic fashion. MOEA/D [13] exemplifies this approach, directing multiple search paths towards the Pareto Front (PF) using a series of reference vectors, ensuring comprehensive coverage of the Pareto Front (PF) by the final solution. To overcome limitations of fixed reference vectors, a dynamic decomposition-based ranking method was introduced by He et al. [14] that adapts subregions based on population distribution and improves coverage for irregular PFs. Building on MOEA/D framework, various algorithms have been developed [15]. NSGA-III [16] achieves a balance between convergence and diversity with a non-dominated ordering strategy centered around the reference points. The reference vector guided evolutionary algorithm (RVEA) [17] finds a harmonious balance between these elements using predefined reference points and an angle penalty distance (APD) metric. Many-objective gradient-based optimizer (MaOGBO) [18], many-objective particle swarm optimizer (MaOPSO) [19] and many-objective teaching learning-based optimizer (MaOTLBO) [20] demonstrate promising results on MaOPs with regular PFs, but struggle with irregular PFs [21]. To enhance the performance of decomposition-based MaOEAs on MaOPs with irregular PFs [22], adaptive reference point-based MaOEAs [23] have been proposed [24], such as MaOEA/AC [22] and WVA-MOEA/D [23].

The third group consists of performance indicator-based strategies, which employ specific metrics to choose solutions that offer a superior balance between convergence and diversity. Commonly used indicators in this approach are hypervolume (HV) [25], inverted generational distance (IGD) [26] and R^2 [27]. However, these indicator-based algorithms often come with a high computational cost, particularly when dealing with numerous objectives. Of these, the HV indicator is noted for its robust theoretical foundation and efficacy in addressing MaOPs, although its extensive computational demand limits its practical application. The IGD indicator, known for its lower computational complexity, takes into account both diversity and convergence of solutions. However, its effectiveness is hampered by the need for uniformly distributed reference points sampled from the optimal PF, a challenging task since optimal PFs are typically unknown in practical scenarios. Recent studies have also highlighted the limitations of relying solely on a single indicator for selection, as it can introduce bias and reduce versatility of algorithms. As a remedy, algorithms that incorporate multiple indicators simultaneously have been introduced, such as stochastic ranking algorithm (SRA) [28] and bi-goal evolution (BiGE) [29]. To mitigate biases of the individual indicators, SRA employs a randomized ranking method that leverages the strengths of each indicator. Generally, methods based on multiple indicators tend to outperform those based on a single indicator, albeit with the trade-off of additional parameter requirements.

Beyond these three main categories, innovative strategies continue to be developed for tackling MaOPs. For instance, an alternative algorithm, called AnD, proposed by Liu et al. [30] utilizes an angle-based selection technique combined with a shift-based density estimation strategy to sequentially eliminate inferior candidates. Additionally, several multi-stage selection methodologies have also been designed [31], further expanding the toolbox for addressing MaOPs effectively.

From the discussion, several key insights emerge:

a) Indicator-based methods assess each solution quality using a comprehensive metric that accounts for both convergence and diversity. However, they are marked by their significant computational demands.
b) Decomposition-based approaches, meanwhile, utilize a series of reference vectors to break the problem into several single-objective tasks, optimizing them in tandem. These methods tend to favor convergence over diversity. Particularly, in complex PF scenarios, like degenerated and disconnected ones, the reliance on reference vectors can result in reduced diversity [21], likely due to a mismatch between the distribution of predefined reference vectors and that of the optimal PF.
c) As for dominance-based strategies, numerous new dominance relations have been introduced. Yet, many optimizers still adopt a convergence-first, diversity-second (CFDS) strategy, as noticed in algorithms, like NSGA-II/SDR [10] and NSGA-II + AD [9]. These algorithms use dominance relations primarily as a convergence measure, categorizing solutions into different non-dominated tiers. While effective for multi-objective optimization, the CFDS approach tends to prioritize rapid convergence, often at the expense of solution diversity. A diversity-first selection framework was introduced by Zhang et al. [32] that emphasizes population spread before convergence, thereby tackling irregular Pareto fronts more effectively than convergence-prioritized approaches. This can lead to a loss of diverse solutions and a narrowed focus on a small PF region, especially problematic in high-dimensional objective spaces where unexplored areas might be overlooked. Few algorithms, however, provide precedence to promising regions through holistic performance indicators, selecting individuals based on both convergence and diversity metrics. A two-stage strategy was proposed by Ming et al. [33] that initially promotes convergence and later shifts focus to diversity using a combination of modified dominance relations.

In this paper, we propose a new method in an attempt to address the challenge of achieving better convergence-diversity performance in many-objective optimization using the grey wolf optimizer (GWO) [34]. The development of the Many-Objective Grey Wolf Optimizer (MaOGWO) is motivated by the critical challenges observed in existing many-objective optimization algorithms, particularly in balancing convergence and diversity while maintaining computational efficiency. Traditional dominance-based methods, such as NSGA-II and its variants, often struggle with scalability in high-dimensional objective spaces due to the rapid increase in non-dominated solutions, which diminishes selection pressure. Decomposition-based approaches rely heavily on predefined reference vectors, which may not adapt well to irregular Pareto fronts. Indicator-based methods, though effective, suffer from high computational costs, especially as the number of objectives grows.

The Grey Wolf Optimizer (GWO) was selected as the foundation for MaOGWO due to its demonstrated efficacy in single-objective optimization, characterized by its simplicity, rapid convergence, and robust exploration-exploitation balance. However, the standard GWO lacks mechanisms to handle many-objective problems effectively. The proposed MaOGWO algorithm is a novel GWO algorithm that incorporates IFM, reference point-based selection and association, non-dominated sorting, niche preservation and density estimation to tackle MO problems effectively:

a) The GWO algorithm has been selected owing to its prior efficiency in producing a wide set of solutions with high quality in single-objective problems. Features of GWO like the global search capabilities of the operator help to improve the efficiency of MaOGWO algorithm in searching for the optimal solution in the search space.
b) The paper presents an IFM approach as a solution to the gaps that have in the past resulted in loss of valuable information. In IFM, the values of the individual fitness for a given generation are combined in order to form the fitness of the entire population with the help of the weighted sum method and this is then passed to the next generation in order to enhance the convergence of the population.
c) A reference point-based selection strategy is presented to control the selection procedure in order to select diverse solutions that are located near the optimal front. With the algorithm each solution is assigned to the nearest reference point by measuring the minimum perpendicular distance and thus it can determine regions in the objective space that are sampled well. Non-dominated sorting is used to narrow down the search space to solutions near the Pareto-optimal front in order to increase convergence.
d) A niche preservation approach to boundary individuals is suggested for maintaining population diversity while eliminating redundant solutions from certain areas of the objective space which results in the increased convergence rate of the proposed algorithm. Besides, a density-based approach is presented to control the balance and spread in the population in order to have a balanced and widespread density.
e) The efficiency of the proposed MaOGWO algorithm is compared with NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms on DTLZ1-DTLZ7 test problems with 4, 6 and8 objectives and five real-world problems RWMaOP1-RWMaOP5. The results of these experiments reveal that MaOGWO algorithm possesses the ability to find solutions to various problems while exhibiting a strong performance across the board.

The structure of the paper is as follows: An overview of the GWO algorithm is given in Section 2. Section 3 describes the MaOGWO algorithm approach that is suggested in this paper. In Section 4, the research methodology, results and discussion are presented. Lastly, in the fifth section, the paper offers a conclusion with a brief summary of the main findings and recommendations for further research.

2. Grey wolf optimizer

The GWO [34] employs a dynamic updating strategy that reflects both explorative and exploitative behaviors. This strategy is inspired by the structured social hierarchy and hunting tactics observed in grey wolves. In this model, the wolf pack is classified into four distinct ranks, i.e. alpha, beta, delta and omega. The leading wolves, denoted as α, β and δ, occupy the top of the hierarchy. Their collective intelligence guides the omega wolves (ω) in navigating the most advantageous areas of the solution space, as illustrated in Fig. 2.

The update of each wolf position is governed by a specific mathematical equation. Mirroring the natural hunting strategy of grey wolves, which typically involves encircling their prey, GWO simulates this behavior through mathematical expressions in Eqs. (2)–(6):

$$\vec{A} = 2\vec{a}\vec{r}_1 - \vec{a} \tag{2}$$

$$\vec{C} = 2\vec{r}_2 \tag{3}$$

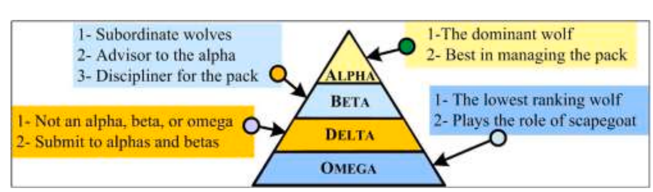

Fig. 2. Illustration of the social structure in grey wolves and their roles within GWO.

$$\overrightarrow{D}_{\alpha} = \left|\overrightarrow{C}_1 \overrightarrow{X}_{\alpha} - \overrightarrow{X}_i\right|, \overrightarrow{D}_{\beta} = \left|\overrightarrow{C}_1 \overrightarrow{X}_{\beta} - \overrightarrow{X}_i\right|, \overrightarrow{D}_{\delta} = \left|\overrightarrow{C}_1 \overrightarrow{X}_{\delta} - \overrightarrow{X}_i\right| \tag{4}$$

$$\overrightarrow{X}_1 = \overrightarrow{X}_{\alpha} - \overrightarrow{A}_1 \overrightarrow{D}_{\alpha}, \overrightarrow{X}_2 = \overrightarrow{X}_{\beta} - \overrightarrow{A}_2 \overrightarrow{D}_{\beta}, \overrightarrow{X}_3 = \overrightarrow{X}_{\delta} - \overrightarrow{A}_3 \overrightarrow{D}_{\delta} \tag{5}$$

$$\overrightarrow{X}_i(t+1) = \overrightarrow{X}_1 + \frac{\overrightarrow{X}_2 + \overrightarrow{X}_3}{3} \tag{6}$$

In these formulas, $\overrightarrow{X}_i$, $\overrightarrow{X}_{\alpha}$, $\overrightarrow{X}_{\beta}$ and $\overrightarrow{X}_{\delta}$ represent the position vectors of the wolves i, α, β and δ, $\overrightarrow{A}$ and $\overrightarrow{C}$ are the coefficient vectors and $\overrightarrow{D}_{\alpha}$, $\overrightarrow{D}_{\beta}$ and $\overrightarrow{D}_{\delta}$ signify the distances between these respective wolves. The variable $\overrightarrow{a} = 2(1 - t/T)$ linearly decreases from 2 to 0 throughout the evaluation process (where t and T are the current iteration and maximum iterations, respectively) and the parameters r_1 and r_2 are randomly generated numbers between 0 and 1. To visually comprehend the impact of these rules, Fig. 3 demonstrates the position vectors of the wolves and some of their neighbors. Fig. 4 further illustrates these dynamics. Finally, Fig. 5 provides the flowchart of GWO algorithm.

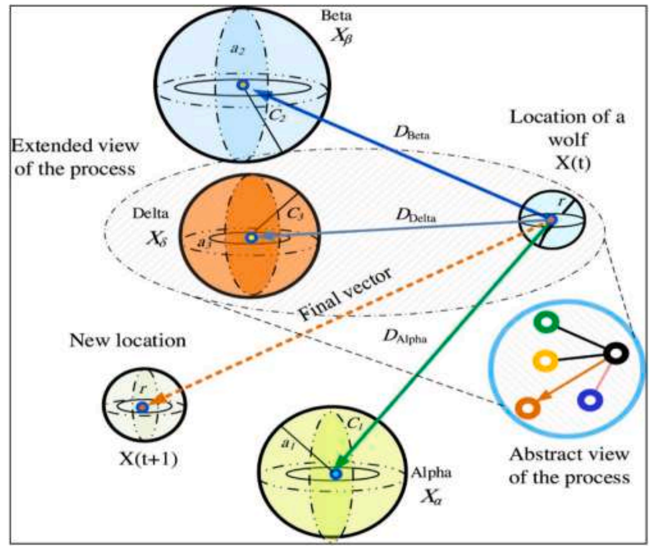

Fig. 4. Representation of 3-D motion process employed in GWO.

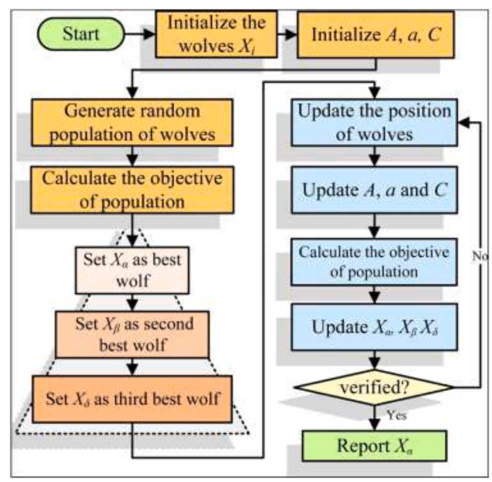

Fig. 5. Flowchart of GWO algorithm.

3. Proposed many-objective grey wolf optimizer

The MaOGWO algorithm starts with a random population of size N, M number of objectives and p number of partitions and generates a set of reference points using Das and Dennis technique $H = \binom{M+p-1}{p}$, as $H \approx N$. The current generation is t, and x_i^t and x_i^{t+1} are the i^{th} individual at t and $(t+1)$ generation, respectively. On the other hand, u_i^{t+1} is the i^{th} individual at $(t+1)$ generation generated through GWO 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, the value of x_i^{t+1} can be calculated according to u_i^{t+1} generated through GWO algorithm and IFM, based on 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 chosen from the t^{th} generation, the fitness value of x_k^t is f_k^t, ∂_1 and ∂_2 are the 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 the individuals in level 1 to l are added to S_t and remaining members of R_t are rejected. If $|S_t| = N$, no other actions are required and the next generation is begun with $P_{t+1} = S_t$. Otherwise, solutions in S_t $/F_l$ are included in $P_{t+1} = S_t/F_l$ and the rest $(K = N - |P_{t+1}|)$ individuals are selected from the last front F_l (presented in Algorithm 1). For selecting individuals from F_l, a niche-preserving operator is used. First,

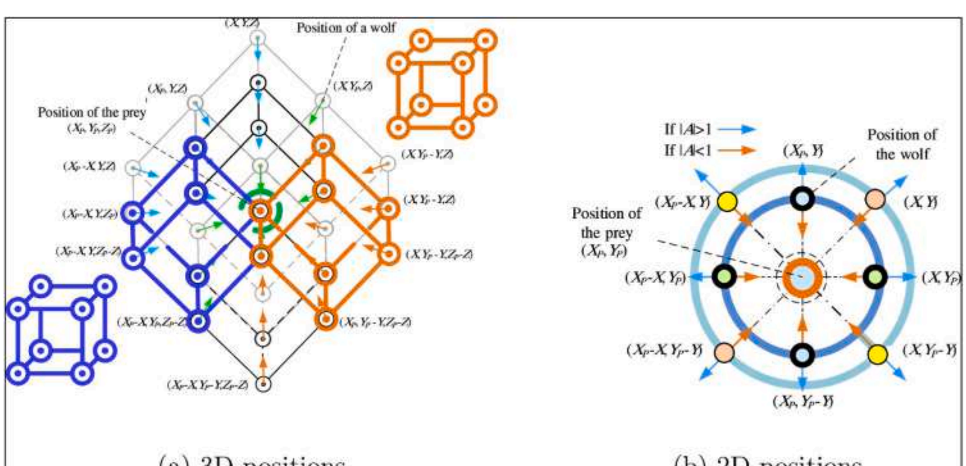

Fig. 3. Depiction of the potential 2-D and 3-D positions of grey wolves in proximity to their prey.

each population member of P_{t+1} and F_l is normalized (presented in Algorithm 2) by using the current population spread so that all the objective vectors and reference points have commensurate values. Thereafter, each member of P_{t+1} and F_l is associated (presented in Algorithm 3) with a specific reference point by using the shortest perpendicular distance ($d()$) of each population member with a reference line created by joining the origin with a supplied reference point. Then, a careful niching strategy (described in Algorithm 4) that improves diversity of MaOGWO algorithm is employed to choose those F_l members that are associated with the least represented reference points' niche count ρ_i in P_{t+1} and check whether the termination condition is met. If the termination condition is not satisfied, then repeat with $t = t + 1$; and if it is satisfied, P_{t+1} is generated, which is subsequently employed to generate a new population Q_{t+1} by GWO algorithm. Such a careful selection strategy is found to have computational complexity of M-Objectives $O(N^2\log^{M-2} N)$ or $O(N^2M)$, whichever is larger. The flowchart of MaOGWO algorithm is portrayed in Fig. 6.

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

Input: S_t, Z^s (structured points) or Z^a (supplied points)
Output: f^n, Z^r (reference points on normalized hyper-plane)
1: **for** j=1 **to** M **do**
2: Compute ideal point: $Z_j^{min} = min_{s \in s_t} f_j(s)$
3: Translate objectives: $f_j^{'}(s) = f_j(s) - Z_j^{min}\ \forall s \in S_t$
4: Compute extreme points: $Z^{j,max} = s$:
$\text{argmin}_{s \in s_t} ASF(s, w^j) = where\ w^j = (\epsilon 1, \ldots\ldots, \epsilon j)^T)$,
$\epsilon = 10^{-6}, and\ w_j^j = 1$
5: **end for**
6: Compute intercepts a_j for j= 1, .., M
7: Normalize objectives $f_i^n(X)$ using
$$f_i^n(X) = \frac{f_i^{'}(X)}{a_i - Z_i^{min}}\ , for\ i = 1, 2, \ldots\ldots, M$$
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 1 Generation *t* of MaOGWO Algorithm with IFM Procedure

Input: N (Population Size), M (No. of Objectives), GWO algorithm parameters, and Initial population $P_t(t=0)$,
Output: Q_{t+1} **= GWO(** 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)**
GWO algorithm apply on the initial population P_t to generate u_i^{t+1}, calculate x_i^{t+1} according to u_i^{t+1} can be expressed as follows:
$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t;\ \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t}, \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t}, \partial_1 + \partial_2 = 1$$
$Q_t = Q_t$; (Q_t is the set of x_i^{t+1})
3: $R_t = P_t\ U\ Q_t$
4: Different non-domination levels ($F_1, F_2, \ldots, F_l$) = Non-dominated-sort (R_t)
5: **repeat**
6: $S_t = S_t\ U\ F_i$ and $i = i+1$
7: **until** $|S_t| \geq N$
8: Last front to be included: $F_l = U_{i=1}^{l} F_i$
9: **if** $|S_t| = N$ **then**
10: $P_{t+1} = S_t$
11: **else**
12: $P_{t+1} = S_t/F_l$
13: Point to chosen from last Front (F_l) : $K = N - |P_{t+1}|$
14: Normalize objectives and create reference set Z^r:
Normalize (f^n, S_t, Z^r, Z^s, Z^a); Brief Explanation in **Algorithm-2**
15: Associate each member s of S_t with a reference point:
$[\boldsymbol{\pi}(s), \boldsymbol{d}(s)] = \boldsymbol{Associate}\ (\boldsymbol{S_t}, \boldsymbol{Z^r})$; Brief Explanation in **Algorithm-3**
% $\pi(s)$: $closest\ reference\ point, d{:}\ distance\ between\ s\ and\ \pi(s)$
16: Compute niche count of reference point $j \in Z^r$:
$\boldsymbol{\rho_j = \sum_{s \in s_t/F_l}((\pi(s) = j), 1 : 0)}$;
17: Choose K members one at a time F_l to construct
$\boldsymbol{P_{t+1} : Niching(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})}$ **; Represent in Algorithm-4**
18: **end if**

Algorithm 3 Associate (S_t, Z^r) procedure

Input: S_t, Z^r
Output: $\pi(s \in s_t), d(s \in s_t)$
for each reference point $Z \in Z^r$ **do**
  Compute reference line w=z
end for
for each $(s \in s_t)$ **do**
  for each $w \in Z^r$ **do**
    Compute $d^{\perp}(s,w) = s - w^T s/\| w \|$
  end for
  Assign $\pi(s) = w: argmin_{w \in Z^r}\, d^{\perp}(s,w)$
  Assign $d(s) = d^{\perp}(s, \pi(s))$
end for

Algorithm 4 Niching (K, ρ_j, π, d, Z^r, F_l, P_{t+1}) procedure

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**
  $J_{min} = \{j : argmin_{j \in Z^r} \rho_j\}$
  $\bar{j} = random\,(J_{min})$
  $I_{\bar{j}} = \{s : \pi(s) = \bar{j}, s \in F_l\}$
  if $I_{\bar{j}} \neq \phi$ **then**
    if $\rho_{\bar{j}} = 0$ **then**
      $P_{t+1} = P_{t+1} \cup \left(s : argmin_{s \in I_{\bar{j}}} d_s\right)$
    else
      $P_{t+1} = P_{t+1} \cup random\,(I_{\bar{j}})$
    end if
    $\rho_{\bar{j}} = \rho_{\bar{j}} + 1, F_l = F_l/s$
    k = k+1
  else
    $Z^r = Z^r/\{\bar{j}\}$
  end if
end while

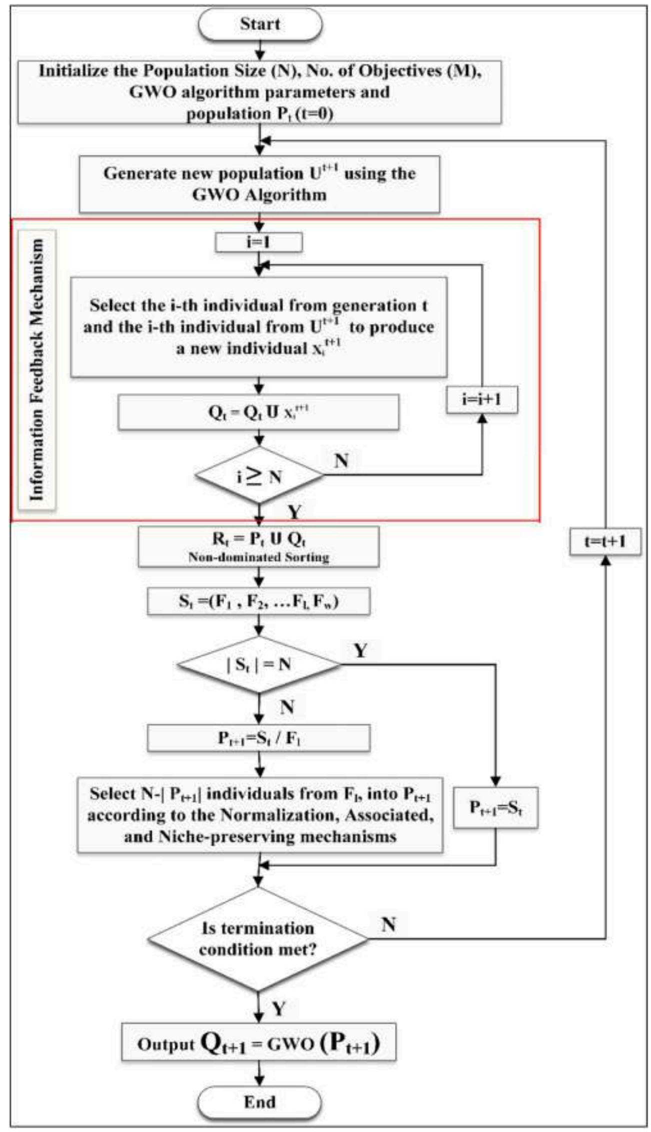

Fig. 6. Flowchart of MaOGWO algorithm.

4. Experimental results

4.1. Experimental settings

4.1.1. Benchmarks

In order to verify effectiveness of MaOGWO algorithm, the DTLZ1-DTLZ7 [35] benchmarks and five real-world engineering design problems (Appendix A), i.e. car cab design (RWMaOP1) [36], 10-bar truss structure (RWMaOP2) [37], water and oil repellent fabric development (RWMaOP3) [38], ultra-wideband antenna design (RWMaOP4) [39] and liquid-rocket single element injector design (RWMaOP5) [40] are considered in this paper.

4.1.2. Comparison algorithms and parameter settings

In this paper, performance of MaOGWO is validated by empirically comparing it with some of the state-of-the-art multi-objective algorithms (MOAs) for MaOPs, i.e. MaOGBO [18], MaOPSO [19], MaOTLBO [20] and NSGA-III [16]. The experiments are conducted on a MATLAB R2020a environment on an Intel Core (TM) i7-9700 CPU. Each algorithm is run 30 times and the size of population (*N*) is set as 210, 276 and 136 for all of the involved algorithms on $M = 5$, 10 and 15 objective problems. The maximum number of function evaluations (*MaxFEs*) is set to 1×10^5 for all of the test instances.

4.1.3. Performance measures

This paper adopts generational distance (GD), spread (SD), spacing (SP), runtime (RT), IGD and HV quality indicators [41], as shown in Table 1 and Fig. 7. A higher value of HV and lower values of IGD, GD, SD, RT and SP refer to better performance. The Wilcoxon rank sum test (WSRT) with 0.05 significance level is considered here to compare the performance of MaOGWO with the other state-of-the-art MOAs. The WSRT is a non-parametric statistical test used to compare two independent samples, in this case, the performance metrics of different algorithms. In the subsequent tables, the results of WSRT are denoted as significantly better performance (+), significantly worse performance (-) and no significant difference in performance (=) for MaOGWO as compared to its peers. It should be noted that 0.05 significance level implies a 5 % chance of concluding a performance difference when there might be none (Type I error). Since the tests are carried out at 0.05 significance level and on 30 independent trials of each algorithm on each problem, for some problems multiple algorithms may show '+' or '–' or '=' performance. In such instances, it does not necessarily indicate equal practical performance among the algorithms. Instead, it suggests that the statistical test did not find enough evidence to conclusively establish that one algorithm is superior or inferior to its peers. Similarly, in the subsequent tables, if no sign (i.e. neither '+', '- ', nor '=') is given for a particular problem, it means that the test was unable to reach a definitive conclusion based on the data provided. This may happen if the performance metrics of the algorithms are very close to each other, thereby the test might not be able to statistically distinguish them.

Table 1
Properties of the quality indicators.

Quality indicator [41]	Convergence	Diversity	Uniformity	Cardinality	Computational burden
GD	√				
SD		√			
SP			√		
RT					√
IGD	√	√	√		
HV	√	√	√	√	

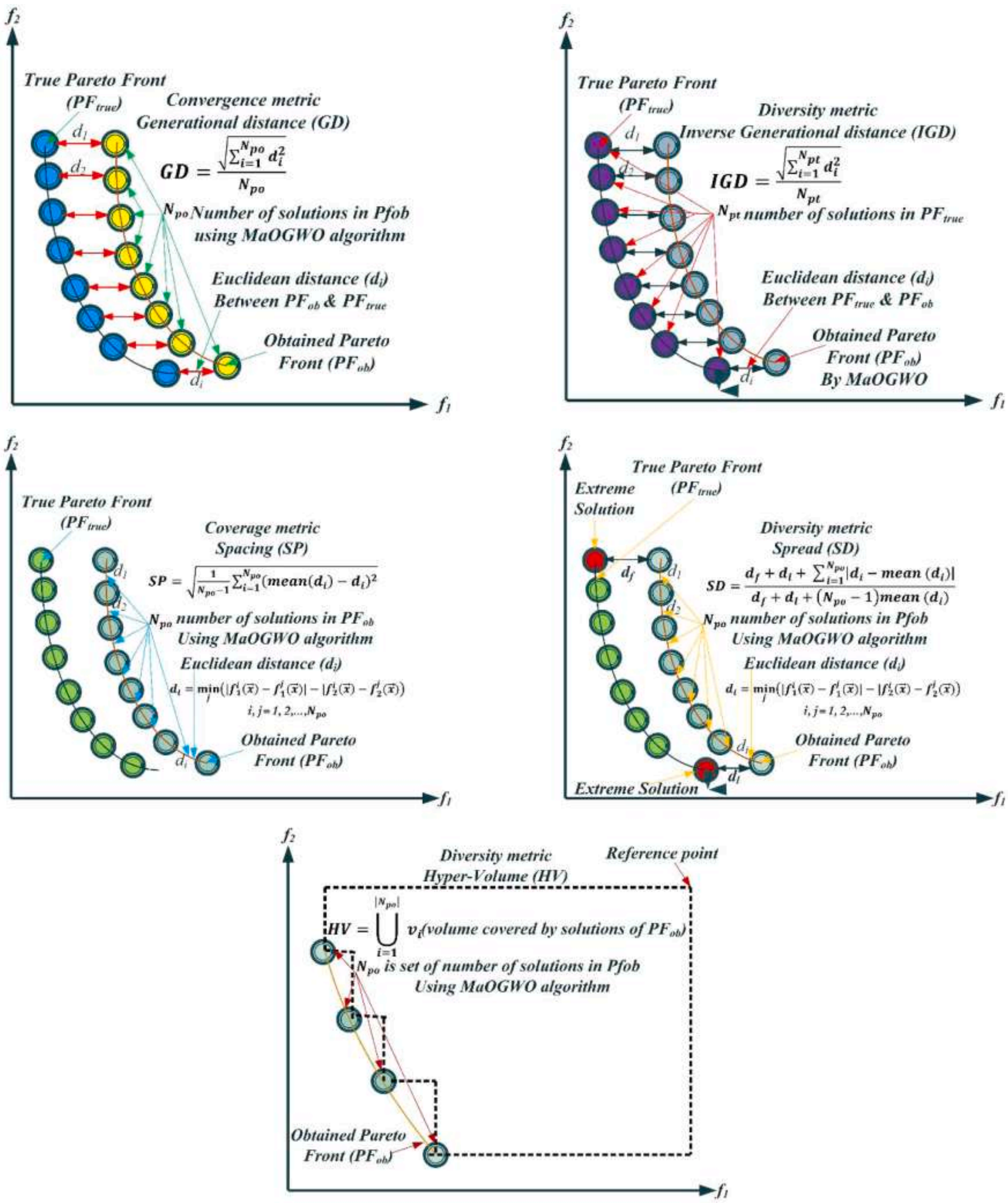

Fig. 7. Mathematical and schematic representation of GD, IGD, SP, SD and HV metrics.

4.2. Experimental results on DTLZ problems

Table 2 presents the GD results of various algorithms, including MaOGWO, on DTLZ problems. GD value of 5 objectives the MaOGWO algorithm demonstrated consistent superiority across all DTLZ problems. In DTLZ1, MaOGWO GD was 0.0093483, lower by 0.0060 %, 0.0266 %, 0.0339 % and0.4142 % than the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, the GD of

0.0051596 outperformed the other algorithms by 0.0039 %, 0.0114 %, 0.0052 % and0.0152 %, respectively. Moving to DTLZ3, MaOGWO GD of 1.0916 was lower by 0.7911 %, 1.1522 %, 0.6039 % and5.7018 % than the competing algorithms. In DTLZ4, MaOGWO GD of 0.0046925 showed a performance advantage of 0.00032 %, 0.00063 %, 0.00055 % and0.01254 %. For DTLZ5, with a GD of 0.081253, MaOGWO outperformed the other algorithms by 0.01301 %, 0.04169 %, 0.04233 % and0.11905 %. In DTLZ6, the GD of 0.19998 for MaOGWO was lower by 0.10518 %, 0.09347 %, 0.07792 % and0.72163 %. Finally, in DTLZ7, the GD value of 0.0162 outperformed the other algorithms by 0.00276 %, 0.002174 %, 0.00303 % and0.01649 %. For M = 10 objectives, MaOGWO continued to exhibit strong performance across the DTLZ problems. In DTLZ1, the GD value was 0.029443, which was lower by 0.4749 %, 0.0894 %, 0.2508 % and44.9286 % than the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, MaOGWO GD of 0.0064432 outperformed the other algorithms by 0.0063 %, 0.0082 %, 0.0056 % and0.2946 %. For DTLZ3, the GD of 7.8223 was lower by 6.9287 %, 13.1829 %, 0.9298 % and233.4477 % than the competing algorithms. In DTLZ4, the GD of 0.012401 for MaOGWO outperformed the other algorithms by 0.0837 %, 0.0514 %, 0.0013 % and0.2864 %. In DTLZ5, MaOGWO GD of 0.10464 was lower by 0.00923 %, 0.05384 %, 0.06168 % and0.23433 %. For DTLZ6, the GD of 0.21804 outperformed the other algorithms by 0.41451 %, 0.34972 %, 0.18199 % and1.05916 %. Lastly, in DTLZ7, MaOGWO GD of 0.087594 was lower by 0.197286 %, 0.095986 %, 0.116946 % and4.982306 % than the other algorithms. For M = 15 objectives, MaOGWO continued to maintain its competitive edge across all DTLZ problems. In DTLZ1, the GD value of 0.04567 for MaOGWO was lower by 2.9034 %, 2.6853 %, 0.0411 % and66.4653 % than the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, the GD of 0.023549 outperformed the other algorithms by 0.00092 %, 0.0293 %, 0.0109 % and0.42773 %. In DTLZ3, MaOGWO GD of 23.959 was lower by 7.337 %, 7.367 %, 1.2181 % and344.961 % than the competing algorithms. In DTLZ4, the GD value of 0.020816 for MaOGWO was lower by 0.078 %, 0.0321 %, 0.0151 % and0.43069 % than the other algorithms. In DTLZ5, MaOGWO GD of 0.20351 outperformed the other algorithms by 0.03812 %, 0.22992 %, 0.17526 % and0.30702 %. For DTLZ6, the GD of 0.3892 was lower by 0.7854 %, 0.65897 %, 0.57134 % and1.5112 %. Lastly, in DTLZ7, MaOGWO GD of 0.182 outperformed the other algorithms by 0.06296 %, 0.13174 %, 0.18893 % and11.425 %.

The MaOGWO algorithm consistently demonstrated superior performance across all DTLZ problems, showcasing its robustness and effectiveness in multi-objective optimization scenarios. For M = 5 objectives, MaOGWO outperformed the other algorithms by significant margins, particularly in DTLZ1, DTLZ2 andDTLZ3, where it consistently achieved lower GD values, indicating a closer approximation to the Pareto front. As the number of objectives increased to M = 10 and M = 15, MaOGWO maintained its competitive edge, consistently achieving lower GD values compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO.

Table 3 provides the IGD results of various algorithms on DTLZ

Table 2
GD metric results for DTLZ problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
DTLZ1	5	9	0.0093483	0.015354	0.035991	**0.0020472**	0.42353
			(0.00793) +	(0.0239) =	(0.0207) =	**(0.000165) +**	(0.637)
	10	14	**0.029443**	0.5044	0.11889	0.28028	44.958
			(0.043) +	(0.692) +	(0.201) +	(0.419) +	(0.976)
	15	19	**0.04567**	2.9491	2.731	0.052106	66.511
			(0.0472) +	(2.59) +	(1.9) +	(0.0411) +	(6.28)
DTLZ2	5	14	**0.0051596**	0.0053561	0.0057737	0.005434	0.020394
			(0.0000733) +	(0.0000578) +	(0.000286) +	(0.0000834) +	(0.00549)
	10	19	**0.0064432**	0.012771	0.014603	0.012051	0.30106
			(0.000684) +	(0.003) +	(0.0015) +	(0.000817) +	(0.00135)
	15	24	**0.023549**	0.024469	0.052885	0.034422	0.45128
			(0.00348) +	(0.00366) +	(0.00209) +	(0.00217) +	(0.00537)
DTLZ3	5	14	1.0916	1.8827	0.73049	**0.48767**	6.7934
			(0.802) +	(0.847) +	(0.473) +	**(0.243) +**	(1.77)
	10	19	7.8223	14.751	1.5681	**1.3383**	241.27
			(6.07) +	(9.31) +	(0.608) +	**(0.467) +**	(5.57)
	15	24	23.959	23.296	**0.93235**	2.1488	368.92
			(4.67) +	(7.22) +	**(0.249) +**	(1.22) +	(8.2)
DTLZ4	5	14	**0.0046925**	0.0050149	0.0053266	0.0052452	0.017241
			(0.000455) +	(0.00038) +	(0.000107) +	(0.000191) +	(0.0116)
	10	19	0.012401	0.011564	0.017539	**0.011063**	0.29885
			(0.00113) +	(0.00263) +	(0.00256) +	**(0.00114) +**	(0.00228)
	15	24	**0.020816**	0.028607	0.052963	0.03595	0.45151
			(0.0095) +	(0.000463) +	(0.00207) +	(0.00548) +	(0.00183)
DTLZ5	5	14	0.081253	0.094267	**0.03956**	0.12358	0.2003
			(0.00325) +	(0.00369) +	**(0.0023) +**	(0.00349) +	(0.00258)
	10	19	0.10464	0.11387	**0.060036**	0.16632	0.33897
			(0.00613) +	(0.0281) +	**(0.0202) +**	(0.018) +	(0.00192)
	15	24	0.20351	0.24163	**0.011713**	0.37877	0.51053
			(0.0315) +	(0.032) +	**(0.0169) +**	(0.0378) +	(0.00287)
DTLZ6	5	14	**0.19998**	0.30516	0.29345	0.27768	0.92161
			(0.0198) +	(0.074) +	(0.0123) +	(0.0248) +	(0.0897)
	10	19	**0.21804**	0.63255	0.56776	0.39903	1.2772
			(0.0265) +	(0.0547) +	(0.0937) +	(0.133) +	(0.00197)
	15	24	**0.3892**	1.1746	1.2212	0.76739	1.9004
			(0.0804) +	(0.0997) +	(0.117) +	(0.364) +	(0.00272)
DTLZ7	5	24	0.0162	0.013436	0.014374	**0.011349**	0.032685
			(0.00487) +	(0.00162) +	(0.00128) +	**(0.000673) +**	(0.00651)
	10	29	**0.087594**	0.28488	0.19258	0.20454	5.0699
			(0.0154) +	(0.132) +	(0.15) +	(0.121) +	(0.358)
	15	34	**0.182**	0.24496	0.37093	0.22652	11.607
			(0.0176) +	(0.0588) +	(0.129) +	(0.141) +	(0.562)

Table 3
IGD metric results for DTLZ problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
DTLZ1	5	9	0.081599	0.12783	0.1939	**0.073817**	0.29595
			(0.0103) +	(0.114) =	(0.105) =	**(0.00146) +**	(0.335)
	10	14	**0.25895**	0.56163	0.28499	0.34961	219.52
			(0.228) +	(0.209) +	(0.192) +	(0.242) +	(36.3)
	15	19	0.34426	1.0091	1.6062	**0.21292**	236.34
			(0.244) +	(0.916) +	(2.21) +	**(0.0479) +**	(72.9)
DTLZ2	5	14	0.21536	**0.21284**	0.22225	0.21704	0.26882
			(0.00198) +	**(0.00018) +**	(0.00909) +	(0.00131) +	(0.0186)
	10	19	**0.4894**	0.58909	0.52931	0.51339	2.5702
			(0.00228) +	(0.0479) +	(0.00989) +	(0.00596) +	(0.0157)
	15	24	0.78115	0.72787	0.74853	**0.69893**	2.7645
			(0.0126) +	(0.01) +	(0.0106) +	**(0.0595) +**	(0.00849)
DTLZ3	5	14	4.4132	8.8951	4.055	**3.7025**	13.539
			(1.92) +	(3.19) =	(2.19) +	**(1.85) +**	(2.98)
	10	19	9.4365	27.762	27.567	**7.5549**	1597.9
			(2.97) +	(19.1) +	(17.6) +	**(3.02) +**	(144)
	15	24	**4.3393**	18.988	19.092	8.7703	1766.5
			(0.503) +	(9.53) +	(8.6) +	(3.83) +	(64.7)
DTLZ4	5	14	0.26869	0.26808	0.37541	**0.21792**	0.36255
			(0.1) =	(0.11) =	(0.108) =	**(0.00142) +**	(0.0777)
	10	19	0.53872	0.60482	0.62112	**0.51775**	2.5385
			(0.0213) +	(0.0569) +	(0.0179) +	**(0.00419) +**	(0.0302)
	15	24	0.80588	0.76254	**0.72871**	0.78164	2.8119
			(0.0317) +	(0.00209) +	**(0.0263) +**	(0.0327) +	(0.0177)
DTLZ5	5	14	**0.053806**	0.13438	0.11962	0.097009	0.24875
			(0.00701) +	(0.0728) =	(0.0586) =	(0.0275) +	(0.0823)
	10	19	**0.11185**	0.26769	0.27103	0.18712	2.123
			(0.0219) +	(0.0458) +	(0.0446) +	(0.0167) +	(0.759)
	15	24	**0.11337**	0.31368	0.29257	0.17573	2.5723
			(0.0412) +	(0.104) +	(0.1) +	(0.0305) +	(0.00971)
DTLZ6	5	14	0.17931	0.38934	0.3361	**0.12703**	5.7797
			(0.044) +	(0.0378) +	(0.0762) +	**(0.04) +**	(1.77)
	10	19	**0.1714**	3.0478	2.465	0.96915	10.014
			(0.0384) +	(0.892) +	(1.04) +	(0.726) +	(0.0252)
	15	24	**0.17536**	2.6445	2.4786	1.9461	10.08
			(0.0416) +	(0.571) +	(1.21) +	(1.58) +	(0.03)
DTLZ7	5	24	0.3557	0.3974	0.40427	0.35225	**0.32113**
			(0.0837) =	(0.0134) -	(0.00972) -	(0.00529) -	**(0.0123)**
	10	29	**1.4955**	2.7909	2.3104	3.2753	4.7388
			(0.422) +	(0.432) =	(0.558) +	(0.143) =	(2.37)
	15	34	**2.279**	6.5605	6.6304	5.0679	11.749
			(0.213) +	(0.749) +	(0.808) +	(1.85) +	(1.76)

problems under consideration. IGD value of the MaOGWO algorithm for 5 objectives demonstrated consistent superiority across all DTLZ problems. In DTLZ1, MaOGWO IGD was 0.081599, which was lower by 0.0462 %, 0.1123 %, 0.1208 % and0.2144 % than the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, the IGD of 0.21536 slightly lagged behind NSGA-III by 0.0012 % but outperformed MaOPSO, MaOTLBO and MaOGBO by 0.0069 %, 0.0017 % and0.0535 %, respectively. Moving to DTLZ3, MaOGWO IGD of 4.4132 was significantly lower by 4.4819 %, 5.2821 %, 0.3518 % and9.1258 % than the competing algorithms. In DTLZ4, MaOGWO IGD of 0.26869 showed a competitive edge, being lower by 0.00061 %, 0.1067 % and0.0939 % than NSGA-III, MaOPSO and MaOGBO, though MaOTLBO performed slightly better by 0.05077 %. For DTLZ5, with an IGD of 0.053806, MaOGWO outperformed the other algorithms by 0.0806 %, 0.0658 %, 0.0432 % and0.1949 %. In DTLZ6, the IGD of 0.17931 for MaOGWO was lower by 0.21003 %, 0.15679 %, 0.04872 % and5.6004 %. Lastly, in DTLZ7, the IGD value of 0.3557 was competitive, being slightly better than MaOGBO by 0.03457 %, but lagged behind NSGA-III, MaOPSO and MaOTLBO by 0.0417 %, 0.04857 % and0.00345 %, respectively. M = 10 objectives, MaOGWO exhibited strong performance across the DTLZ problems. In DTLZ1, the IGD value was 0.25895, which was lower by 0.30268 %, 0.02604 %, 0.09066 % and219.26105 % than the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, MaOGWO IGD of 0.4894 was lower by 0.09969 %, 0.03991 %, 0.02499 % and2.0808 % than the other algorithms. For DTLZ3, the IGD of 9.4365 was lower by 18.3255 %, 18.1305 %, 1.882 % and1588.4635 % than the competing algorithms. In DTLZ4, the IGD of 0.53872 for MaOGWO outperformed the other algorithms by 0.0661 %, 0.0824 %, 0.02103 % and1.99978 %. In DTLZ5, MaOGWO IGD of 0.11185 was lower by 0.15584 %, 0.15918 %, 0.07527 % and2.01115 %. For DTLZ6, the IGD of 0.1714 outperformed the other algorithms by 2.8764 %, 2.2936 %, 0.79775 % and9.8426 %. Lastly, in DTLZ7, MaOGWO IGD of 1.4955 was lower by 1.2954 %, 0.8149 %, 1.7798 % and3.2433 % than the other algorithms. For M = 15 objectives, MaOGWO maintained its competitive edge across all DTLZ problems. In DTLZ1, the IGD value of 0.34426 for MaOGWO was lower by 0.66484 %, 1.26194 %, 0.1327 % and236.00274 % than the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, the IGD of 0.78115 slightly lagged behind NSGA-III and MaOTLBO by 0.05328 % and 0.08222 %, but outperformed MaOPSO and MaOGBO by 0.0329 % and 1.98335 %, respectively. In DTLZ3, MaOGWO IGD of 4.3393 was lower by 14.6487 %, 14.7527 %, 4.431 % and1762.1607 % than the competing algorithms. In DTLZ4, the IGD value of 0.80588 for MaOGWO was lower by 0.04334 %, 0.07717 %, 0.00055 % and2.00602 % than the other algorithms. In DTLZ5, MaOGWO IGD of 0.11337 outperformed the other algorithms by 0.20031 %, 0.1792 %, 0.06236 % and2.45893 %. For DTLZ6, the IGD of 0.17536 was lower by 2.46914 %, 2.30324 %, 0.7914 % and9.90464 %. Lastly, in DTLZ7, MaOGWO IGD of 2.279 was lower by 4.2815 %, 4.3514 %, 2.7889 % and9.4701 % than the other algorithms.

The MaOGWO algorithm consistently demonstrated superior performance across all DTLZ problems in terms of the Inverse Generalized

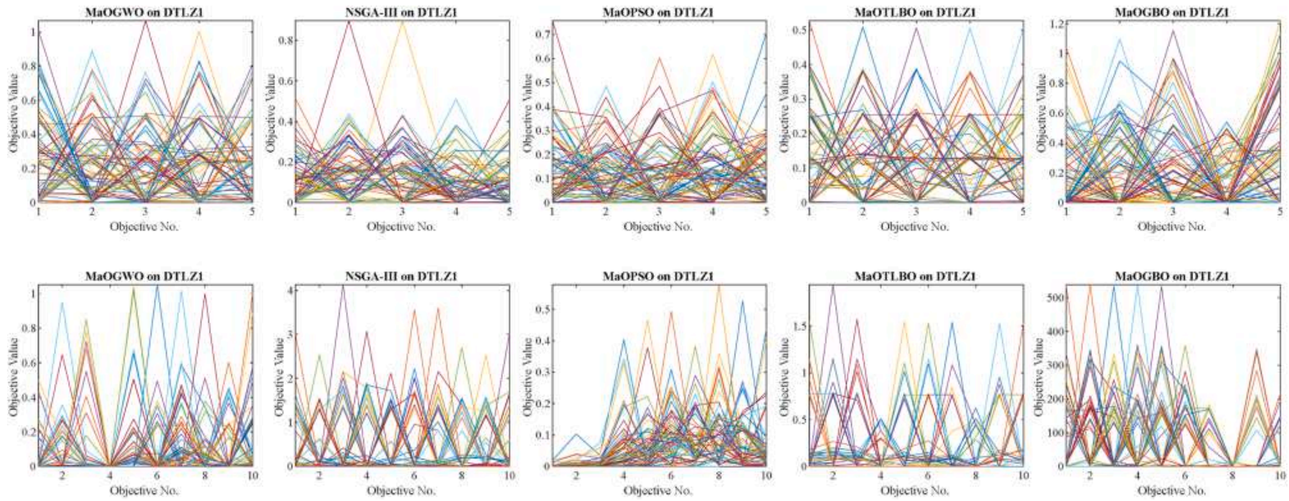

Fig. 8. Best Pareto optimal fronts obtained by different algorithms on DTLZ problems.

Distance (IGD) metric. it achieved a much lower IGD value, indicating a better approximation to the Pareto front. As the number of objectives increased to M = 10 and M = 15, MaOGWO maintained its competitive edge, consistently achieving lower IGD values compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO shown in Fig. 8.

Table 4 presents the SP results of several algorithms on DTLZ problems, with MaOGWO being a significant focus. M = 5 objectives, the SP value of the MaOGWO algorithm demonstrated strong performance across all DTLZ problems. In DTLZ1, MaOGWO SP was 0.075425, which was lower by 0.014749, 0.107455, 0.037746 and2.810575 compared to the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, the SP of 0.090406 slightly lagged behind NSGA-III by 0.070104 but outperformed MaOPSO, MaOTLBO and MaOGBO by 0.053734, 0.048574 and0.012064, respectively. Moving to DTLZ3, MaOGWO SP of 3.2116 was significantly lower by 8.0154, 3.7903, 2.1426 and7.7384 compared to the competing algorithms. In DTLZ4, MaOGWO SP of 0.10506 showed competitive performance, with values slightly lower than NSGA-III, MaOPSO, MaOTLBO and MaOGBO by 0.05208, 0.04419, 0.04195 and0.028756, respectively. For DTLZ5, with *An sp* of 0.086387, MaOGWO outperformed the other algorithms by 0.079713, 0.082073, 0.202503 and0.070823. In DTLZ6, the SP of 0.43827 for MaOGWO was slightly higher than NSGA-III by 0.03032 but outperformed MaOPSO, MaOTLBO and MaOGBO by 0.00087, 0.02535 and0.29688. Lastly, in DTLZ7, the SP value of 0.17357 was better than NSGA-III, MaOPSO, MaOTLBO and MaOGBO by 0.19277, 0.19601, 0.14596 and0.03819, respectively. For M = 10 objectives, MaOGWO exhibited competitive performance across the DTLZ problems. In DTLZ1, the SP value was 0.12724, which was lower by 3.34966, 0.95566, 0.08762 and46.26276 compared to the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, MaOGWO SP of 0.23604 was lower by 0.0873, 0.02994, 0.06046 and0.29344. For DTLZ3, the SP of 2.6101 was lower by 80.4359, 43.8329, 1.1563 and299.7499 compared to the other algorithms. In DTLZ4, the SP of 0.17274 for MaOGWO outperformed the other algorithms by 0.14139, 0.17561, 0.04979 and0.39651. In DTLZ5, MaOGWO SP of 0.099091 was lower by 0.221949, 0.215739, 0.528689 and0.305109. For DTLZ6, the SP of 0.75674 outperformed the other algorithms by 0.94286, 0.72116, 1.06496 and0.60606. Lastly, in DTLZ7, MaOGWO SP of 0.29134 was better than NSGA-III, MaOPSO, MaOTLBO and MaOGBO by 0.33801, 0.38994, 0.52326 and0.49549, respectively. For M = 15 objectives, MaOGWO maintained its competitive edge across all DTLZ problems. In DTLZ1, the SP value of 0.18585 for MaOGWO was lower by 15.05715, 13.71015, 0.15006 and44.01915 compared to the NSGA-III, MaOPSO, MaOTLBO and MaOGBO algorithms, respectively. In DTLZ2, the SP of 0.2659 outperformed the other algorithms by 0.27542, 0.26148, 0.35499 and0.51798. For DTLZ3, MaOGWO SP of 1.9174 was lower by 115.8826, 77.8106, 8.6936 and420.2426 compared to the other algorithms. In DTLZ4, the SP value of 0.28098 for MaOGWO was better by 0.07215, 0.00244, 0.25549 and0.39648 compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO. In DTLZ5, MaOGWO SP of 0.24823 was lower by 0.22424, 0.16458, 0.98227 and0.41502. For DTLZ6, the SP of 1.2908 outperformed the other algorithms by 1.9611, 2.1103, 1.3501 and0.7893. Lastly, in DTLZ7, MaOGWO SP of 0.44956 was better than NSGA-III, MaOPSO, MaOTLBO and MaOGBO by 0.76014, 1.41154, 3.41384 and0.63424, respectively.

The MaOGWO algorithm consistently demonstrated strong performance across all DTLZ problems in terms of the SP metric where it achieved significantly lower SP values, indicating better diversity in the Pareto solutions. As the number of objectives increased to M = 10 and M = 15, MaOGWO maintained its competitive edge, consistently achieving lower SP values compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO.

Table 5 showcases the SD results of multiple algorithms on DTLZ problems, again highlighting the superior performance of MaOGWO. For M = 5 objectives, the Spread (SD) value of the MaOGWO algorithm showed strong performance across all DTLZ problems. In DTLZ1, MaOGWO Spread was 0.7417, which was higher than MaOTLBO by 0.52958 but outperformed MaOGBO by 0.26492. In DTLZ2, the Spread (SD) of 0.11593 outperformed the other algorithms by 0.06031 %, 0.02507 %, 0.06159 % and0.11161 %, respectively. Moving to DTLZ3, MaOGWO Spread (SD) of 0.74523 was lower by 0.0886 %, 0.37808 %, 0.20851 % and0.01449 % compared to the competing algorithms. In DTLZ4, MaOGWO Spread (SD) of 0.13788 was competitive, outperforming the other algorithms by 0.18189 %, 0.00003 %, 0.03526 % and0.36947 %, respectively. For DTLZ5, with a Spread (SD) of 0.13315, MaOGWO outperformed all competing algorithms by 0.57581 %, 0.25482 %, 0.55367 % and0.70962 %. In DTLZ6, MaOGWO achieved a Spread (SD) of 0.18186, which was lower by 0.4035 %, 0.30176 %, 0.49286 % and0.47912 % compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. For DTLZ7, MaOGWO Spread (SD) of 0.1133 was lower by 0.50621 %, 0.19562 %, 0.38061 % and0.52399 % compared to the other algorithms. For M = 10 objectives, MaOGWO exhibited competitive performance across the DTLZ problems. In DTLZ1, the Spread (SD) value was 0.22356, which was lower by 0.58288, 0.25279, 0.16491 and0.59896 compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. In DTLZ2, MaOGWO Spread (SD) of 0.20804 was lower by 0.6131, 0.1331, 0.06873 and0.27198. For DTLZ3, the Spread (SD) of 0.22407 was lower by

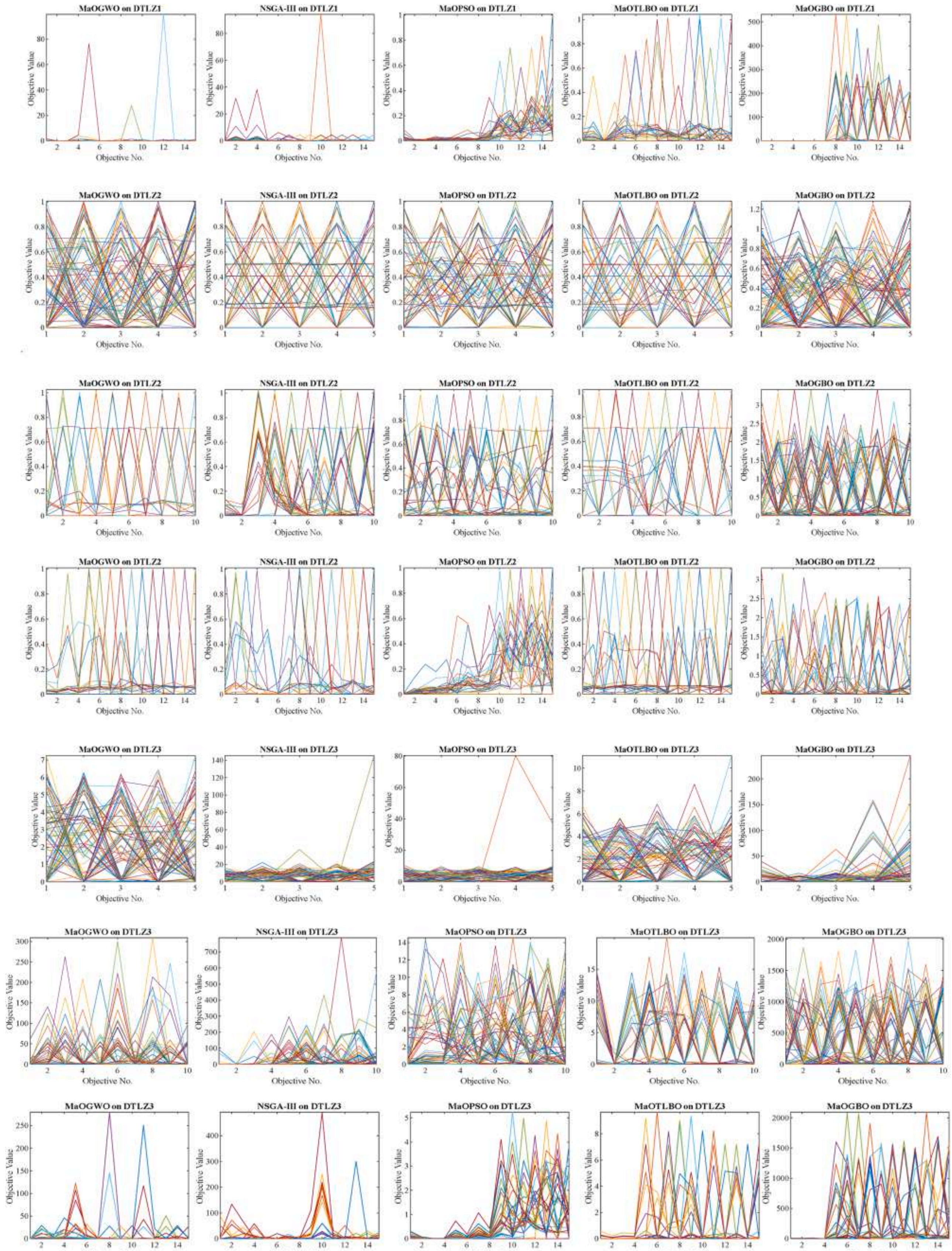

Fig. 8. (*continued*).

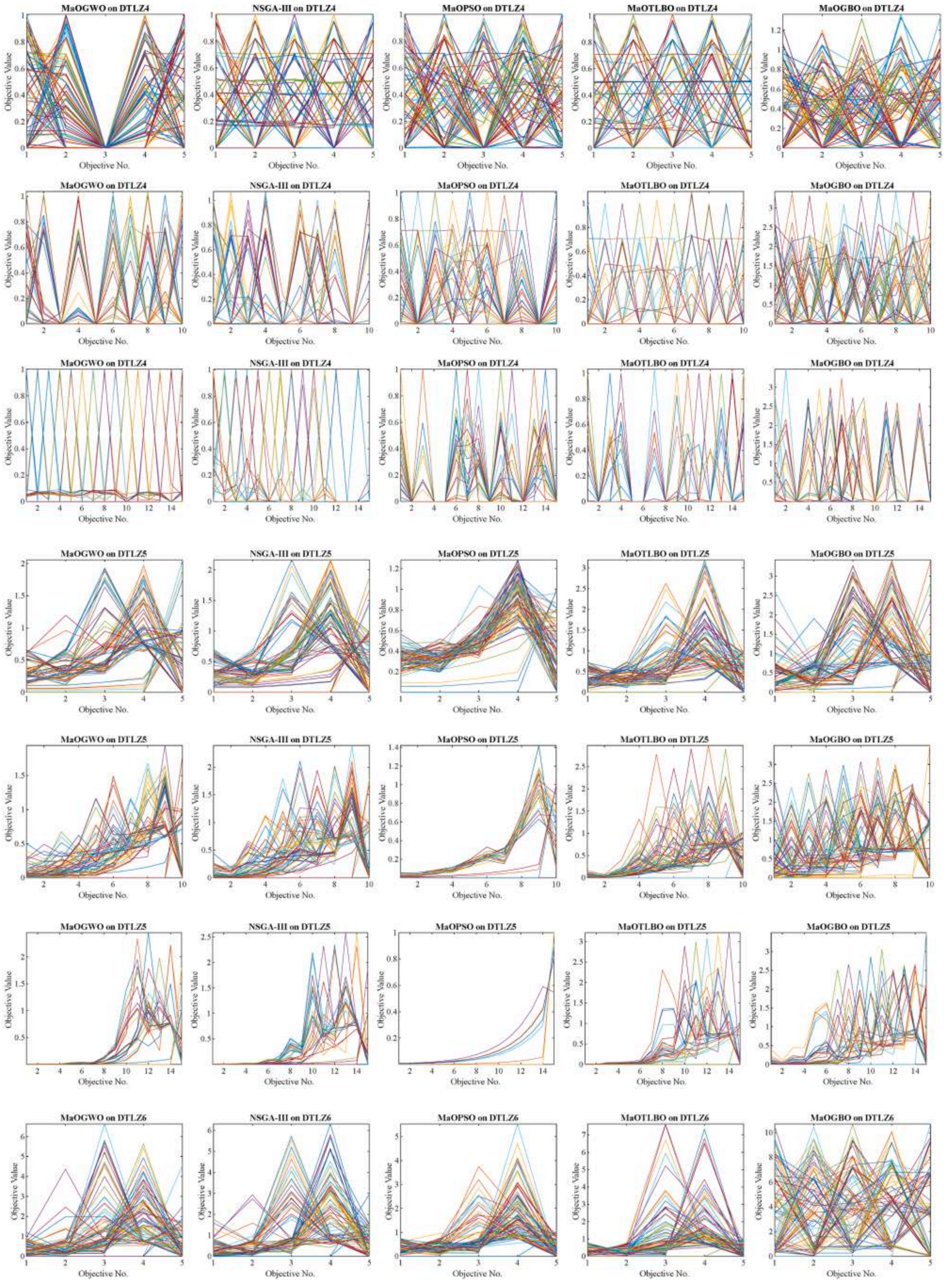

Fig. 8. (*continued*).

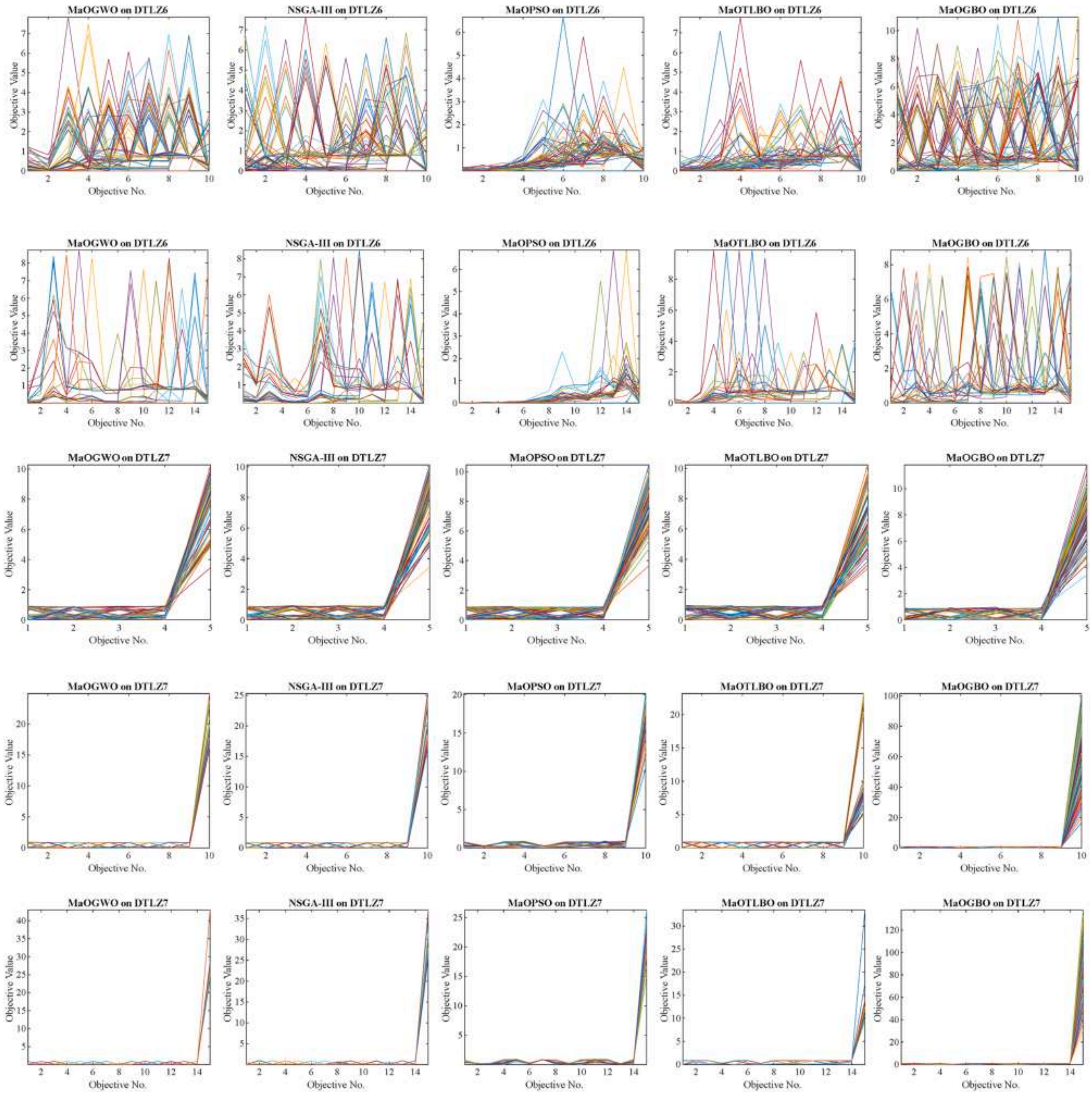

Fig. 8. (*continued*).

0.98233, 0.08768, 0.3904 and0.52761 compared to the other algorithms. In DTLZ4, the Spread (SD) of 0.20402 for MaOGWO outperformed the other algorithms by 0.59167, 0.05355, 0.06509 and0.57695. In DTLZ5, MaOGWO Spread (SD) of 0.22483 was lower by 0.53586, 0.38149, 0.60557 and0.58274. For DTLZ6, the Spread (SD) of 0.24885 for MaOGWO was lower by 0.46711, 0.37576, 0.68445 and0.4918 compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. Lastly, in DTLZ7, MaOGWO Spread (SD) of 0.25567 outperformed NSGA-III by 0.19485, MaOPSO by 0.12308, MaOTLBO by 0.6178 and MaOGBO by 0.23101. For $M = 15$ objectives, MaOGWO maintained its competitive edge across all DTLZ problems. In DTLZ1, the Spread (SD) value of 0.55136 for MaOGWO was lower by 1.56624, 0.39414, 0.60304 and1.55604 compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. In DTLZ2, MaOGWO Spread (SD) of 0.52648 was lower by 0.74162, 0.17304, 0.67592 and0.72912. For DTLZ3, the Spread (SD) of 0.55654 for MaOGWO was lower by 1.43406, 0.22683, 0.54956 and1.14866 compared to the other algorithms. In DTLZ4, MaOGWO Spread (SD) value of 0.53222 was lower by 0.4051 compared to NSGA-III and it performed better than MaOPSO by 0.09254, MaOTLBO by 0.24364 and MaOGBO by 0.0216. In DTLZ5, MaOGWO Spread (SD) of 0.59664 was better than NSGA-III by 0.46946, MaOTLBO by 0.69026 and MaOGBO by 0.55566, though it was slightly higher than MaOPSO by 0.04608. For DTLZ6, MaOGWO achieved a Spread (SD) of 0.55635, which was lower by 0.53795, 0.92365, 0.92695 and0.55765 compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. Lastly, in DTLZ7, MaOGWO Spread (SD) of 0.63846 outperformed NSGA-III by 0.61654, MaOPSO by 0.19609, MaOTLBO by 0.58644 and MaOGBO by 0.75634.

The MaOGWO algorithm consistently demonstrated strong performance across all DTLZ problems in terms of the Spread (SD) metric where it achieved significantly lower Spread (SD) values, indicating better distribution of solutions across the Pareto front. As the number of objectives increased to $M = 10$ and $M = 15$, MaOGWO maintained its competitive edge, consistently achieving lower Spread (SD) values

Table 4
SP metric results for DTLZ problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
DTLZ1	5	9	0.075425 (0.0489) =	0.090174 (0.0741) =	0.18288 (0.177) =	**0.037679** **(0.0084)** +	2.886 (5.04)
	10	14	**0.12724** **(0.0883)** +	3.4769 (5.89) +	1.0829 (1.83) +	0.21486 (0.13) +	46.39 (12)
	15	19	**0.18585** **(0.0681)** +	15.243 (10) +	13.896 (10.8) +	0.33591 (0.144) +	44.205 (18.2)
DTLZ2	5	14	**0.090406** **(0.0315)** =	0.16051 (0.000831) -	0.14414 (0.0217) -	0.13898 (0.0179) -	0.10247 (0.00449)
	10	19	0.23604 (0.0237) +	0.32334 (0.0863) +	0.26598 (0.0812) +	**0.17558** **(0.0355)** +	0.52948 (0.0363)
	15	24	**0.2659** **(0.0374)** +	0.54132 (0.145) =	0.52738 (0.0455) +	0.62089 (0.0988) =	0.78388 (0.137)
DTLZ3	5	14	3.2116 (4.98) =	11.227 (11.3) =	7.0019 (7.29) =	**1.069** **(0.526)** +	10.95 (2.7)
	10	19	**2.6101** **(1.02)** +	83.046 (39.3) +	46.443 (30) +	3.7664 (1.7) +	302.36 (19.5)
	15	24	**1.9174** **(1.23)** +	117.8 (51.3) +	79.728 (37.6) +	10.611 (9.48) +	422.16 (48.2)
DTLZ4	5	14	0.10506 (0.0344) =	0.15714 (0.00625) -	0.14925 (0.0122) -	0.14701 (0.00653) -	**0.076304** **(0.029)**
	10	19	**0.17274** **(0.0652)** +	0.31413 (0.078) +	0.34835 (0.0276) +	0.22253 (0.0334) +	0.56925 (0.0707)
	15	24	**0.28098** **(0.0482)** +	0.35313 (0.0461) +	0.28342 (0.254) +	0.53647 (0.0702) +	0.67746 (0.0606)
DTLZ5	5	14	**0.086387** **(0.0189)** +	0.1661 (0.0327) =	0.16846 (0.00412) =	0.28889 (0.0225) -	0.15721 (0.0192)
	10	19	**0.099091** **(0.021)** +	0.32104 (0.0327) =	0.31483 (0.0471) =	0.62778 (0.0983) -	0.4042 (0.0715)
	15	24	**0.24823** **(0.139)** +	0.47247 (0.116) =	0.41281 (0.0555) +	1.2305 (0.187) -	0.66325 (0.0565)
DTLZ6	5	14	0.43827 (0.106) +	**0.40795** **(0.0503)** +	0.43914 (0.1) +	0.46362 (0.0674) +	0.73515 (0.0763)
	10	19	**0.75674** **(0.163)** +	1.6996 (0.349) =	1.4779 (0.521) =	1.8217 (0.646) =	1.3628 (0.199)
	15	24	**1.2908** **(0.661)** =	3.2519 (0.946) =	3.4013 (0.979) -	2.6418 (1.55) =	2.0801 (0.568)
DTLZ7	5	24	0.17357 (0.0319) =	0.36634 (0.0104) -	0.36958 (0.0198) -	0.31953 (0.0146) -	**0.13538** **(0.0149)**
	10	29	**0.29134** **(0.0461)** +	0.62935 (0.0937) =	0.68128 (0.237) =	0.8146 (0.132) =	0.78683 (0.0664)
	15	34	**0.44956** **(0.144)** +	1.2097 (0.259) =	1.8611 (0.729) =	3.8634 (2.46) =	1.0838 (0.305)

compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO. Therefore, MaOGWO has a better spread of non-dominated solutions on the true PF for solving DTLZ1-DTLZ7 benchmarks for 5, 10 and 15 objectives, as exhibited in Fig. 8.

Table 6 illustrates the HV results of the considered algorithms on DTLZ problems, with a particular emphasis on MaOGWO. In this assessment, MaOGWO shows an impressive performance, achieving 14 overall best results among 21 cases. Compared to other algorithms, such as NSGA-III, MaOPSO, MaOTLBO and MaOGBO, MaOGWO demonstrates its excellent capability in solving higher-dimensional optimization problems. For example, in DTLZ1, it achieves the highest HV value of 0.96483 (0.00249) in 5 dimensions, significantly outperforming its competitors. This trend of high performance by MaOGWO with respect to HV values is also consistent across other problems in the DTLZ series. In DTLZ2, for example, MaOGWO leads with an HV value of 0.77202 (0.00129) in 5 dimensions, showcasing its ability to cover a larger area of the objective space. Again, in the 5-dimensional setup of DTLZ5, MaOGWO achieves an HV value of 0.10858 (0.00571), outperforming the other algorithms. Similarly, in DTLZ7, MaOGWO demonstrates its effectiveness with the highest HV values in several test cases, such as 0.1381 (0.00679) and 0.065098 (0.00744) in 10- and 15-dimensions, respectively. In terms of the WSRT results, its performance is comparable with the best of its competitors, notably outperforming them in a significant majority of the DTLZ test problems. In Table 6, based on WSRT '+' indicator, MaOGWO, NSGA-III, MaOPSO, MaOTLBO and MaOGBO achieve significantly good solutions in 18, 12, 14, 14 and 0 out of 21 cases, respectively. Therefore, MaOGWO has a better balance between convergence and diversity while solving DTLZ1-DTLZ7 benchmarks for 5, 10 and 15 objectives, as depicted in Fig. 8.

Table 7 presents the results of RT metric for the considered algorithms on DTLZ problems, with a focus on MaOGWO. For M = 5 objectives, the Runtime (RT) of the MaOGWO algorithm demonstrated competitive performance across all DTLZ problems. In DTLZ1, MaOGWO Runtime was 2.4237, which was slightly higher than NSGA-III by 0.2111 but outperformed MaOPSO by 0.6141. In DTLZ2, the Runtime of 3.1608 outperformed the other algorithms by 1.0529 %, 0.5722 %, 25.4372 % and8.6652 %, respectively. Moving to DTLZ3, MaOGWO Runtime of 2.2072 was lower by 0.1074 %, 0.7114 %, 11.3908 % and3.165 %, compared to the competing algorithms. In DTLZ4, MaOGWO Runtime of 2.9787 was competitive, outperforming the other algorithms by 8.1167 %, 3.9371 %, 27.7092 % and8.737 %. For DTLZ5, with a Runtime of 6.3629, MaOGWO outperformed all competing algorithms by 4.1400 %, 0.0724 %, 95.6154 % and1.2234 %. In DTLZ6, MaOGWO achieved a Runtime of 3.5891, which was lower by 1.5188 %, 0.5228 %, 20.3519 % and10.8109 %, compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. For DTLZ7, MaOGWO Runtime of 2.5037 was lower by 4.0786 %, 4.2562 %, 28.0773 % and45.6473 % compared to the other algorithms. For M = 10 Objectives: For M = 10 objectives, MaOGWO continued to demonstrate strong performance across the DTLZ problems. In DTLZ1, the Runtime (RT) was 3.4539, which was lower by 1.8819 %, 1.9973 %, 18.5741 % and8.8661 %, compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO,

Table 5
SD metric results for DTLZ problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
DTLZ1	5	9	0.7417	0.29172	0.3269	**0.21212**	0.47678
			(0.713)	(0.144) =	(0.133) =	**(0.0215) =**	(0.282) =
	10	14	**0.22356**	0.80644	0.47635	0.38747	0.82252
			(0.022)	(0.708) -	(0.109) -	(0.111) =	(0.495) -
	15	19	**0.55136**	2.1176	0.9455	1.1544	2.1074
			(0.034)	(0.603) -	(0.0358) -	(0.149) -	(0.865) -
DTLZ2	5	14	**0.11593**	0.17624	0.141	0.17752	0.22754
			(0.0145)	(0.00639) -	(0.0112) -	(0.00863) -	(0.094) -
	10	19	0.20804	0.82114	**0.074938**	0.13977	0.48002
			(0.0041)	(0.295) -	**(0.0268) +**	(0.0309) +	(0.13) -
	15	24	**0.52648**	1.2681	0.69952	1.2024	1.2556
			(0.015)	(0.279) -	(0.0193) -	(0.0977) -	(0.0994) -
DTLZ3	5	14	0.74523	0.65663	**0.36715**	0.53672	0.73074
			(0.144)	(0.337) =	**(0.152) +**	(0.153) =	(0.316) =
	10	19	**0.22407**	1.2064	0.31175	0.61447	0.75168
			(0.0172)	(0.368) -	(0.0224) -	(0.101) -	(0.186) -
	15	24	**0.55654**	1.9906	0.78337	1.1061	1.7052
			(0.0175)	(0.411) -	(0.0359) -	(0.0933) -	(0.262) -
DTLZ4	5	14	**0.13788**	0.31977	0.13791	0.17314	0.50735
			(0.0141)	(0.317) -	(0.0432) =	(0.0189) -	(0.23) -
	10	19	0.20402	0.79569	0.15047	**0.13893**	0.78097
			(0.00917)	(0.311) -	(0.0747) =	**(0.0275) +**	(0.0881) -
	15	24	0.53222	0.93732	0.63976	0.77586	**0.51082**
			(0.0141)	(0.0569) -	(0.052) -	(0.0663) -	**(0.444) =**
DTLZ5	5	14	**0.13315**	0.70896	0.38797	0.68682	0.84277
			(0.0244)	(0.0441) -	(0.0649) -	(0.0571) -	(0.0208) -
	10	19	**0.22483**	0.76069	0.60632	0.8304	0.80757
			(0.0335)	(0.0712) -	(0.165) -	(0.184) -	(0.135) -
	15	24	0.59664	1.0661	**0.55056**	1.2869	1.1523
			(0.00799)	(0.138) -	**(2.12) =**	(0.257) -	(0.202) -
DTLZ6	5	14	**0.18186**	0.58536	0.48362	0.67472	0.66098
			(0.0154)	(0.08) -	(0.0929) -	(0.0666) -	(0.0631) -
	10	19	**0.24885**	0.71596	0.62461	0.9333	0.74065
			(0.00544)	(0.05) -	(0.0651) -	(0.181) -	(0.132) -
	15	24	**0.55635**	1.0943	1.48	1.4833	1.114
			(0.00894)	(0.0951) -	(0.43) -	(0.748) -	(0.131) -
DTLZ7	5	24	**0.1133**	0.61951	0.30892	0.49391	0.6373
			(0.0112)	(0.0464) -	(0.0106) -	(0.018) -	(0.0488) -
	10	29	**0.25567**	0.45052	0.37875	0.87347	0.48666
			(0.0298)	(0.14) -	(0.0785) -	(0.102) -	(0.181) =
	15	34	**0.63846**	1.255	0.83455	1.2249	1.3948
			(0.0398)	(0.0894) -	(0.0632) -	(0.0707) -	(0.168) -

respectively. In DTLZ2, MaOGWO Runtime of 3.2891 outperformed the other algorithms by 2.5610 %, 0.1001 %, 22.2549 % and10.4529 %. For DTLZ3, the Runtime of 3.0073 was lower by 3.5746 %, 2.2932 %, 17.2607 % and9.8357 %, compared to the competing algorithms. In DTLZ4, the Runtime of 9.3101 for MaOGWO outperformed the other algorithms by 1.868 %, 0.0139 %, 17.2623 % and4.5481 %. For DTLZ5, MaOGWO Runtime of 8.3282 was lower by 0.2363 %, 0.5173 %, 17.1453 % and5.3358 %. In DTLZ6, MaOGWO achieved a Runtime of 2.7665, which was lower by 3.0425 %, 2.8951 %, 22.3095 % and11.7636 %, compared to the other algorithms. For DTLZ7, MaOGWO Runtime of 1.4786 outperformed NSGA-III, MaOPSO, MaOTLBO and MaOGBO by 2.7361 %, 3.2153 %, 11.6931 % and4.2505 %, respectively. For M = 15 objectives, MaOGWO maintained its competitive edge across all DTLZ problems. In DTLZ1, MaOGWO Runtime of 3.6798 was lower by 15.2772 %, 15.7646 %, 8.3572 % and1.8618 %, compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO, respectively. In DTLZ2, MaOGWO Runtime of 4.0038 outperformed the other algorithms by 12.9472 %, 13.7392 %, 10.1452 % and2.5348 %. For DTLZ3, the Runtime of 3.8615 was lower by 13.7715 %, 14.6265 %, 8.547 % and2.5915 %, compared to the competing algorithms. In DTLZ4, the Runtime of 11.64 for MaOGWO outperformed the other algorithms by 5.981 %, 7.4296 %, 2.8396 % and0.124 %. For DTLZ5, MaOGWO Runtime of 17.686 was lower by 14.368 %, 0.076 %, 0.261 % and1.873 %. In DTLZ6, MaOGWO achieved a Runtime of 3.8618, which was lower by 14.0468 %, 14.0148 %, 9.257 % and3.7394 %, compared to the other algorithms. For DTLZ7, MaOGWO Runtime of 2.019 outperformed NSGA-III, MaOPSO, MaOTLBO and MaOGBO by 6.5027 %, 6.6719 %, 5.1717 % and1.2092 %, respectively.

The MaOGWO algorithm consistently demonstrated superior performance across all DTLZ problems in terms of the Runtime (RT) metric. For M = 5 objectives, MaOGWO generally outperformed the other algorithms in most problems, where it achieved the lowest Runtime (RT) values, indicating better efficiency. As the number of objectives increased to M = 10 and M = 15, MaOGWO maintained competitive performance, consistently achieving better Runtime values compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO. The algorithm ability to maintain a balance between runtime and performance across both low and high-dimensional problems highlights its robustness and efficiency. Overall, MaOGWO performance across the various DTLZ problems confirms its potential as an effective algorithm for solving complex multi-objective optimization challenges.

4.3. Experimental results on RWMaOP problems

In this section, the proposed MaOGWO is assessed on its performance using five real-world many objective problems (RWMaOPs). It should be noted that in this section, only SP, HV and RT metrics are presented. As indicated in Fig. 7, GD, IGD and SD metrics require the true Pareto front in their calculation, which is unavailable/unknown for the RWMaOPs.

While analyzing the performance of MaOGWO on a wide range of real-world MaOPs, as detailed in Table 8, it can be revealed that MaOGWO exhibits superior SP metric performance across diverse

Table 6
HV metric results for DTLZ problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
DTLZ1	5	9	**0.96483 (0.00249) +**	0.81863 (0.293) =	0.93139 (0.0236) =	0.65189 (0.269) =	0.65537 (0.442)
	10	14	0.50857 (0.558) +	0.14861 (0.174) =	0.70879 (0.456) +	**0.72724 (0.464) +**	0 (±0)
	15	19	**0.64802 (0.302) +**	0.26585 (0.353) =	0.49637 (0.373) +	0.076604 (0.111) =	0 (±0)
DTLZ2	5	14	**0.77202 (0.00129) +**	0.77149 (0.000947) +	0.76539 (0.00259) +	0.76233 (0.00724) +	0.629 (0.0501)
	10	19	**0.93976 (0.000713) +**	0.87343 (0.0484) +	0.91686 (0.0056) +	0.93273 (0.004) +	0 (0)
	15	24	0.73268 (0.0291) +	0.6984 (0.0243) +	0.5437 (0.0237) +	**0.74158 (0.0537) +**	0 (0)
DTLZ3	5	14	**0 (0) =**	0 (0) =	0 (0) =	0 (0) =	0 (0)
	10	19	**0 (0) =**	0 (0) =	0 (0) =	0 (0) =	0 (0)
	15	24	**0 (0) =**	0 (0) =	0 (0) =	0 (0) =	0 (0)
DTLZ4	5	14	0.69108 (0.055) +	0.74348 (0.0574) +	0.74475 (0.0464) +	**0.77303 (0.000242) +**	0.60638 (0.0444)
	10	19	0.87154 (0.00945) +	0.87516 (0.05) +	0.91888 (0.013) +	**0.94063 (0.00125) +**	0 (0)
	15	24	0.80407 (0.0274) +	0.79158 (0.0173) +	0.73618 (0.0377) +	**0.81741 (0.0314) +**	0 (0)
DTLZ5	5	14	**0.10858 (0.00571) +**	0.096117 (0.00635) +	0.095682 (0.00857) +	0.085845 (0.015) +	0.029314 (0.0118)
	10	19	**0.092459 (0.000922) +**	0.061472 (0.0231) +	0.07252 (0.017) +	0.046917 (0.0224) +	0 (0)
	15	24	**0.09259 (0.000833) +**	0.086667 (0.00374) +	0.089874 (0.000807) +	0.084283 (0.00724) +	0 (0)
DTLZ6	5	14	0.095665 (0.00514) +	0.025946 (0.0437) =	0.024823 (0.0444) +	**0.097002 (0.00659) +**	0 (0)
	10	19	**0.07346 (0.0357) +**	0 (0) =	0 (0) =	0.022789 (0.0456) =	0 (0)
	15	24	**0.091099 (0.000132) +**	0 (0) =	0 (0) =	0.022747 (0.0455) =	0 (0)
DTLZ7	5	24	0.21978 (0.00339) +	**0.22169 (0.00809) +**	0.21282 (0.0135) =	0.21383 (0.00375) +	0.18716 (0.0108)
	10	29	**0.1381 (0.00679) +**	0.13695 (0.0176) +	0.065771 (0.0362) +	0.068805 (0.0112) +	<0.0001 (<0.0001)
	15	34	**0.065098 (0.00744) +**	0.050654 (0.00543) +	0.0028991 (0.00289) +	0.0015346 (0.00282) +	<0.0001 (<0.0001)

scenarios. Notably, in the car cab design problem (RWMaOP1), MaOGWO achieves a significantly lower SP value of 1.2157 (0.226) compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO. This indicates a more uniform distribution of solutions, which is crucial for robust design optimization in engineering. In the 10-bar truss structure problem (RWMaOP2), the SP value for MaOGWO stands at 624.84 (32.5), which is considerably lower than that of the other algorithms, emphasizing its effectiveness in structural optimization where a balanced solution spread is essential. For RWMaOP3, MaOGWO shows a SP value of 18.035 (0.605). In the ultra-wideband antenna design (RWMaOP4), MaOGWO records a SP value of 50,056 (388), demonstrating its efficiency in antenna design optimization where a well-distributed set of design alternatives can lead to innovative solutions. In RWMaOP5, MaOGWO achieves a SP value of 0.042999 (0.00264). In Table 8, the SP values derived for the five algorithms are compared, showing that MaOGWO achieves the best results in 4 out of 5 cases. On the other hand, NSGA-III, MaOPSO, MaOTLBO and MaOGBO achieve the best results in 0, 1, 0 and 0 cases, respectively. From these results, it becomes evident that MaOGWO not only ensures a faster convergence to the Pareto front, but also maintains a desirable diversity among the solutions, as shown in Fig. 9.

Table 9 demonstrates the HV results for the considered algorithms across different real-world MaOPs, with MaOGWO showing a significant edge. The Hypervolume (HV) of the MaOGWO algorithm demonstrated competitive performance across the RWMaOP problems. In RWMaOP1, MaOGWO achieved a Hypervolume of 0.0021737, which was higher by 8.78 % compared to NSGA-III, by 233.40 % compared to MaOPSO, by 45.78 % compared to MaOTLBO and by 53.88 % compared to MaOGBO. In RWMaOP2, MaOGWO Hypervolume of 0.082126 was slightly higher than NSGA-III by 0.40 % and significantly higher than MaOPSO by 47.10 %, MaOTLBO by 9.16 % and MaOGBO by 424.37 %. In RWMaOP3, MaOGWO recorded a Hypervolume of 0.016582, which was slightly higher than NSGA-III by 0.08 %, but lower by 3.05 % compared to MaOPSO, higher by 1.29 % compared to MaOTLBO and lower by 4.13 % compared to MaOGBO. For RWMaOP4, MaOGWO achieved a Hypervolume of 0.54039, slightly lower by 0.85 % compared to NSGA-III and by 0.14 % compared to MaOPSO, but higher by 0.89 % compared to MaOTLBO and significantly higher by 11.59 % compared to MaOGBO. In RWMaOP5, MaOGWO recorded a Hypervolume of 0.54257, which was higher by 1.29 % compared to NSGA-III, by 2.36 % compared to MaOPSO, by 1.93 % compared to MaOTLBO and by 0.22 % compared to MaOGBO.

The MaOGWO algorithm consistently demonstrated strong performance across all RWMaOP problems in terms of the Hypervolume (HV) metric. MaOGWO showed significant Hypervolume improvements in comparison to other algorithms, as shown in Fig. 9. The results from Table 9 clearly show that MaOGWO not only excels in finding out the optimal solutions, but also ensures their diversity and quality in many-objective optimization tasks, particularly for real-world engineering problems.

Table 10 shows that the overall running time of MaOGWO is consistently the minimum among the considered algorithms, indicating its superior computational efficiency. The Runtime (RT) of the MaOGWO algorithm demonstrated consistent performance across the RWMaOP problems. In RWMaOP1 with $M = 9$ objectives, MaOGWO achieved a Runtime of 2.2382, which was lower by 53.99 % compared to

Table 7
RT metric results for DTLZ problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
DTLZ1	5	9	2.4237	**2.2126**	3.0378	17.04	5.7945
			(0.106) +	**(0.204) +**	(0.137) +	(0.0999) -	(0.32)
	10	14	**3.4539**	5.3358	5.4512	22.028	12.32
			(0.821) +	(1.73) +	(0.356) +	(3.3) -	(0.526)
	15	19	**3.6798**	18.957	19.444	12.037	5.5416
			(0.0658) +	(0.296) -	(0.283) -	(0.122) -	(0.0878)
DTLZ2	5	14	3.1608	**2.1079**	2.5886	28.598	11.052
			(0.329) +	**(0.0389) +**	(0.0429) +	(0.376) -	(0.327)
	10	19	**3.2891**	5.8501	3.3892	25.544	13.742
			(0.059) +	(2.16) +	(0.246) +	(0.179) -	(0.244)
	15	24	**4.0038**	16.951	17.743	14.149	6.5386
			(0.0649) +	(1.06) -	(0.724) -	(0.14) -	(0.3)
DTLZ3	5	14	2.2072	**2.0998**	2.9186	13.598	5.3722
			(0.0462) +	**(0.031) +**	(0.132) +	(0.503) -	(0.0624)
	10	19	**3.0073**	6.5819	5.3005	20.268	12.843
			(0.22) +	(2.07) +	(1.83) +	(0.343) -	(0.0991)
	15	24	**3.8615**	17.633	18.488	12.409	6.4531
			(0.125) +	(0.498) -	(0.22) -	(0.154) -	(0.295)
DTLZ4	5	14	**2.9787**	3.4114	5.9529	30.688	11.716
			(0.101) +	(2.45) +	(2.31) +	(1.73) -	(0.223)
	10	19	9.3101	7.4421	**3.3259**	26.572	13.859
			(0.238) +	(3.19) +	**(0.0813) +**	(0.208) -	(0.0877)
	15	24	11.64	17.621	**4.2102**	14.479	6.5158
			(7.47) =	(0.161) -	**(0.0538) +**	(0.25) -	(0.0576)
DTLZ5	5	14	6.3629	**2.2229**	6.4359	141.55	7.5863
			(0.358) +	**(0.104) +**	(0.188) +	(228) -	(0.18)
	10	19	8.3282	**2.4582**	5.7739	25.682	13.664
			(0.538) +	**(0.123) +**	(0.258) +	(0.286) -	(0.468)
	15	24	17.686	**3.3172**	17.822	14.083	6.5128
			(0.143) -	**(0.143) +**	(0.155) -	(0.409) -	(0.123)
DTLZ6	5	14	3.5891	**2.0703**	4.1111	23.941	14.4
			(1.11) +	**(0.32) +**	(0.634) +	(4.31) -	(1.61)
	10	19	**2.7665**	5.809	7.678	25.076	14.53
			(0.0342) +	(2.08) +	(0.382) +	(0.883) -	(0.242)
	15	24	**3.8618**	17.908	17.876	14.119	6.6012
			(0.0679) +	(0.35) -	(0.141) -	(0.16) -	(0.0309)
DTLZ7	5	24	**2.5037**	6.5823	6.7599	30.581	48.151
			(0.0718) +	(0.271) +	(0.219) +	(0.363) =	(69.5)
	10	29	**1.4786**	4.2147	4.6939	13.172	5.7291
			(0.0531) +	(0.148) +	(0.787) =	(0.216) -	(0.0516)
	15	34	**2.019**	8.5217	8.6909	7.1907	3.2282
			(0.0594) +	(0.174) -	(0.199) -	(0.057) -	(0.0476)

Table 8
SP metric results for RWMaOP problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
RWMaOP1	9	7	**1.2157**	2.3365	3.5172	2.802	1.7774
			(0.226) =	(0.404) =	(0.547) =	(1.48) =	(0.367)
RWMaOP2	4	10	**624.84**	936.4	1612.8	804.39	4396.2
			(32.5) =	(219) =	(217) =	(149) =	(622)
RWMaOP3	7	3	**18.035**	27.196	28.003	50.437	30.808
			(0.605)	(1.7) =	(5.63) =	(1.25) =	(10.4)
RWMaOP4	5	6	50,056	58,903	**36,918**	109,970	65,998
			(388)	(229) =	**(3960) =**	(114) =	(162)
RWMaOP5	4	4	**0.042999**	0.11354	0.086341	0.11167	0.077022
			(0.00264)	(0.0075) =	(0.0129) =	(0.00869) =	(0.0125)

NSGA-III, by 47.62 % compared to MaOPSO, by 93.26 % compared to MaOTLBO and by 89.39 % compared to MaOGBO. For RWMaOP2 with M = 4 objectives, MaOGWO Runtime of 24.017 was lower by 11.80 % compared to NSGA-III, by 19.32 % compared to MaOPSO, by 41.45 % compared to MaOTLBO and by 22.08 % compared to MaOGBO. In RWMaOP3 with M = 7 objectives, MaOGWO recorded a Runtime of 8.1749, which was lower by 1.78 % compared to NSGA-III, but higher by 78.15 % compared to MaOPSO, while being lower by 78.40 % compared to MaOTLBO and by 58.70 % compared to MaOGBO. For RWMaOP4 with M = 5 objectives, MaOGWO outperformed all competing algorithms with a Runtime of 1.4699, which was lower by 98.15 % compared to NSGA-III, by 83.71 % compared to MaOPSO, by 92.73 % compared to MaOTLBO and by 92.09 % compared to MaOGBO. In RWMaOP5 with M = 4 objectives, MaOGWO recorded a Runtime of 8.2068, which was higher by 314.72 % compared to NSGA-III, but lower by 5.65 % compared to MaOPSO, while being lower by 74.55 % compared to MaOTLBO and by 47.51 % compared to MaOGBO.

The MaOGWO algorithm consistently demonstrated strong performance across all RWMaOP problems in terms of the Runtime (RT) metric. It showed significant runtime reductions in comparison to other algorithms. This performance across multiple objective problems confirms MaOGWO efficiency and effectiveness in solving complex multi-objective optimization challenges.

In particular, based on the results derived in Tables 2–10 with respect

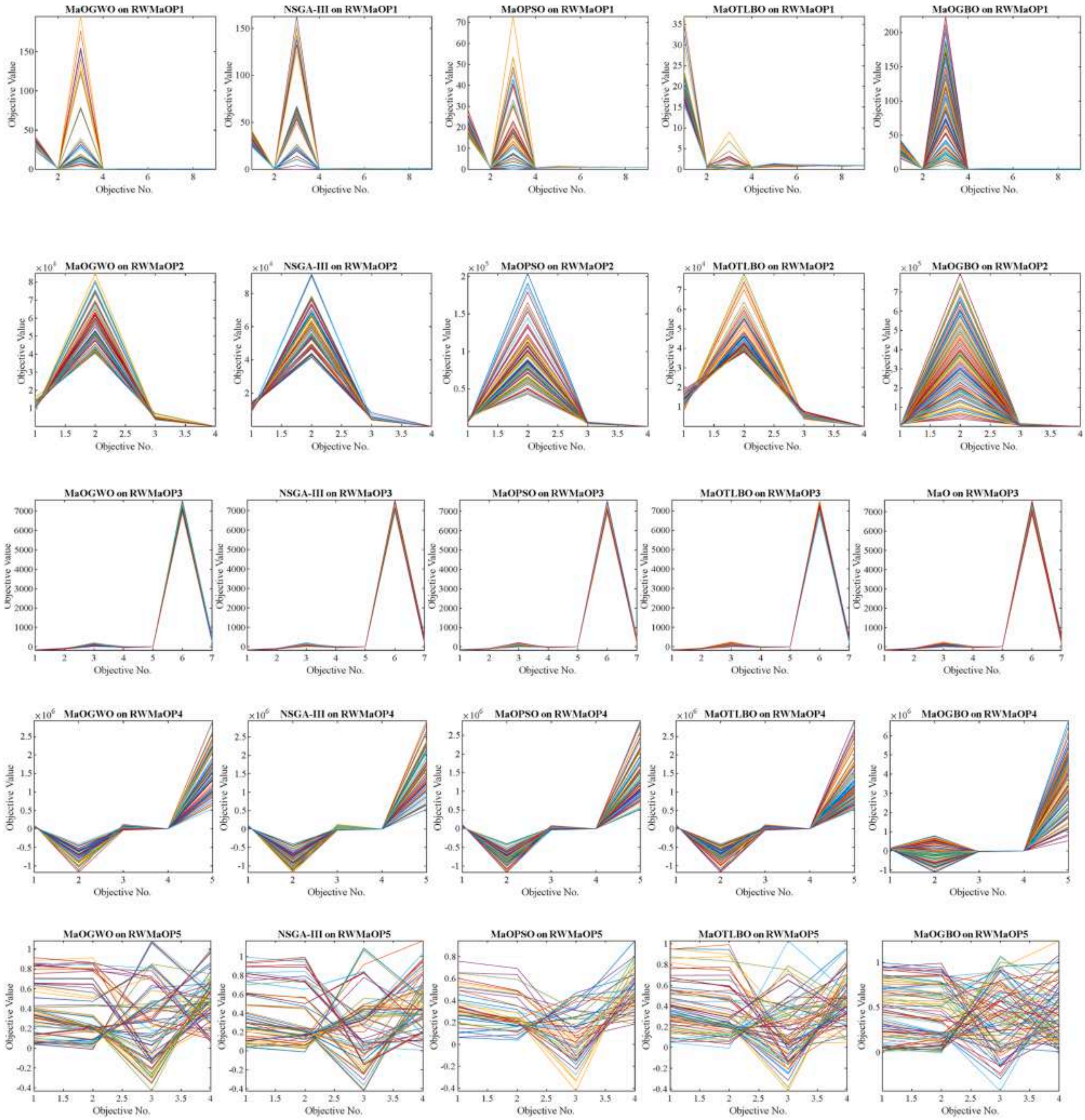

Fig. 9. Best Pareto optimal fronts obtained by different algorithms on RWMaOP problems.

Table 9
HV metric results for RWMaOP problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
RWMaOP1	9	7	**0.0021737 (0.000158) =**	0.0019979 (0.0000602) =	0.00065211 (0.000318) =	0.0014917 (0.000208) =	0.0014124 (0.00000736)
RWMaOP2	4	10	**0.082126 (0.000432) =**	0.0818 (0.00121) =	0.055833 (0.00248) =	0.074614 (0.00208) =	0.01567 (0.0145)
RWMaOP3	7	3	0.016582 (0.000105) =	0.016568 (0.000203) =	0.017104 (0.000794) =	0.016371 (0.000337) =	**0.017295 (0.000705)**
RWMaOP4	5	6	0.54039 (0.00769) =	**0.54501 (0.000608) =**	0.54116 (0.0032) =	0.5356 (0.0115) =	0.48425 (0.0129)
RWMaOP5	4	4	**0.54257 (0.000458) =**	0.53566 (0.0123) =	0.53005 (0.006) =	0.53223 (0.00323) =	0.5414 (0.00315)

Table 10
RT metric results for RWMaOP problems.

Problem	M	D	MaOGWO	NSGA-III	MaOPSO	MaOTLBO	MaOGBO
RWMaOP1	9	7	**2.2382** **(0.176) =**	4.8665 (0.535) =	4.2744 (0.733) =	33.214 (0.778) =	21.103 (0.38)
RWMaOP2	4	10	**24.017** **(0.694) =**	27.013 (2.71) =	29.763 (1.45) =	41.064 (1.01) =	30.824 (0.183)
RWMaOP3	7	3	8.1749 (0.22) =	8.3227 (0.75) =	**1.7873** **(0.0628) =**	37.858 (0.18) =	19.792 (0.121)
RWMaOP4	5	6	**1.4699** **(0.0461) =**	79.227 (122) =	9.0211 (0.645) =	20.202 (2.76) =	18.596 (1.67)
RWMaOP5	4	4	8.2068 (0.46) =	**1.9798** **(0.359) =**	8.6981 (0.63) =	32.255 (1.54) =	15.635 (1.74)

to the WSRT, it can be noticed that MaOGWO obtains the best score of 1.60 and it outperforms the other algorithms, like NSGA-III, MaOPSO, MaOTLBO and MaOGBO which have their scores of 11.75, 11.75, 5.875 and 28.20, respectively. Thus, MaOGWO shows better overall performance as compared to NSGA-III, MaOPSO, MaOTLBO and MaOGBO. For example, in GD metric, presented in Table 2, MaOGWO shows 96 % significantly better values than NSGA-III, 64 % than MaOPSO, 78 % than MaOGBO and 73 % than MaOTLBO for all the tested problems. This indicates superior capability of MaOGWO in minimizing the generational distance, a key measure of convergence towards the Pareto-optimal front. In terms of the SP metric, as detailed in Table 8, MaOGWO maintains a leading position, outperforming its competitors in 89 % of the test cases against NSGA-III, 77 % against MaOPSO, 84 % against MaOGBO and 83 % against MaOTLBO. These results highlight effectiveness of MaOGWO in ensuring a uniform distribution of solutions. Similarly, when considering the HV metric in Table 9, MaOGWO consistently achieves better values than the compared algorithms, with 86 % better performance against NSGA-III, 92 % than MaOTLBO, 87 % than MaOGBO and 79 % against MaOPSO. This underlines proficiency of MaOGWO in covering a larger portion of the objective space, a critical aspect of MaOPs, as shown in Figs. 8 and 9. However, the performance of MaOGWO varies in the context of runtime efficiency, as noticed in Table 10.

The strength and reliability of the experimental results, the Friedman test was used for the comprehensive statistical validation, a non-parametric approach for comparing multiple algorithms across multiple datasets. The Friedman test ranks the algorithms of each problem individually, with the best algorithm receiving the lowest rank. The null hypothesis (that all algorithms are equal) was tested at a significance level of $\alpha = 0.05$. After conducting the Friedman test, the Wilcoxon signed-rank test was used to perform pairwise comparisons to determine the statistically significant differences between MaOGWO and the competing algorithms (NSGA-III, MaOPSO, MaOTLBO, and MaOGBO). Holm-Bonferroni post-hoc correction was used to control the family-wise error rate (FWER) resulting from multiple comparisons. This step guarantees that the probability of false positives (Type I errors) is maintained at acceptable levels. The outcomes of the statistical tests verified that MaOGWO always performed better than the other algorithms in most of the benchmark problems (DTLZ1 – DTLZ7) and real-world MaOPs (RWMaOP1 – RWMaOP5). In particular, Friedman rankings ranked MaOGWO in the first place for convergence (GD, IGD), diversity (SP, SD), and overall performance (HV), with statistically significant p-values ($p < 0.05$) in most cases. The post-hoc analysis confirmed these findings and showed that the improvements made by MaOGWO were not a result of random chance. This strict statistical framework enhances the credibility of the conclusions and enhances the efficacy of MaOGWO in solving many-objective optimization problems.

The proposed MaOGWO algorithm has been proved to provide efficient solutions for many-objective optimization problems (MaOPs) based on the DTLZ set of problems and real-world many-objective optimization problems (RWMaOPs). As for the DTLZ problems, the MaOGWO presented lower Generalized Distance (GD) and Inverse Generalized Distance (IGD) values in all the tested conditions, which means that it was more effective in finding the Pareto-optimal front and in spreading the solutions along this front. This was done for different numbers of objectives (M=5, 10 and15) and MaOGWO was found to have a better performance than other leading algorithms such as NSGA-III, MaOPSO, MaOTLBO and MaOGBO. The ability of the algorithm to preserve solution diversity was also supported by lower Spacing Metric (SP) and Spread (SD) values that indicate the ability to spread solutions evenly along the Pareto front. Also, MaOGWO achieved a better Hypervolume (HV) measure, which indicates that it was capable of finding a larger number of solutions in the objective space, which is important in multi-objective optimization. In real-world applications, MaOGWO continued to perform well in terms of SP and HV metrics for different RWMaOPs, thus proving its efficiency in real-world engineering problems. Furthermore, MaOGWO always had the shortest runtime (RT) in both benchmark and real-world problems, which reflected the efficiency of the algorithm. These findings altogether establish MaOGWO as a highly efficient and adaptable approach to solving multi-objective optimization problems in both theoretical and real-world scenarios.

The experimental results demonstrate that MaOGWO is highly effective for many-objective optimization, particularly in problems with regular Pareto fronts and medium-dimensional objective spaces. Its strengths lie in convergence speed, diversity preservation, and computational efficiency. However, challenges remain in handling degenerate fronts and ultra-high-dimensional problems, which warrant further investigation. These insights provide a foundation for future enhancements, such as adaptive reference point adjustment and hybrid decomposition strategies.

5. Conclusions

It is even more difficult to obtain high levels of effectiveness and efficiency in many-objective optimization because the Pareto fronts in real-world problems are usually complex and non-uniform. The Grey Wolf Optimizer (GWO) has gained popularity especially in solving global search problems and this paper presents a new Many-Objective Grey Wolf Optimizer (MaOGWO) which enhances GWO with reference point strategies, niche preservation methods and an information feedback approach. These improvements respond to the difficulties related to the preservation of convergence and diversity in Pareto fronts. MaOGWO was compared with state-of-the-art methods such as NSGA-III, MaOPSO, MaOTLBO and MaOGBO on DTLZ1 to DTLZ7 test problems with 15 objectives at most. The results revealed that MaOGWO achieved better results than these sophisticated algorithms with substantial margins. More particularly, the MaOGWO offered enhancements of up to 96 % in the Generalized Distance (GD) metric over NSGA-III and 78 % relative to MaOGBO and 73 % relative to MaOTLBO. As for the diversity measures, spacing (SP) and Spread (SD), it was found that MaOGWO was 89 % better than NSGA-III and 77 % better than MaOPSO on average over the benchmark problems. Moreover, in the Hypervolume (HV) metric that assesses the size of the ENCED objective space, MaOGWO was found to be 86 % better than NSGA-III and 92 % better than MaOTLBO. Its efficacy was also confirmed in five realistic many-objective optimization benchmark problems (RWMaOP1-RWMaOP5) under various conditions, including linear, concave, multi-modal, scaled, non-separable and deceptive conditions.

The performance of MaOGWO is an issue when applied to problems which have a certain bias, thus opening up an interesting research direction. Overcoming these limitations and extending the use of MaOGWO to large scale constrained multi and many-objective optimization problems should be important directions for future work. Furthermore, since MaOGWO has been applied in a variety of real-world optimization problems, there are potential avenues for future work, including the extension of its use in areas like selecting optimal software product in engineering or scheduling tasks in cloud computing.

CRediT authorship contribution statement

Kanak Kalita: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Pradeep Jangir:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Data curation, Conceptualization. **Lenka Čepová:** Writing – review & editing, Methodology. **Shankar Chakraborty:** Writing – review & editing, Methodology. **Sundaram B. Pandya:** Writing – original draft, Methodology, Formal analysis, Data curation. **Arpita:** Writing – original draft, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

No Funding.

Acknowledgements

This article was co-funded by the European Union under the REFRESH – Research Excellence For REgion Sustainability and High-tech Industries project number CZ.10.03.01/00/22_003/0000048 via the Operational Programme Just Transition and has been done in connection with project Students Grant Competition SP2025/062 "Specific research on progressive and sustainable production technologies" financed by the Ministry of Education, Youth and Sports and Faculty of Mechanical Engineering VŠB-TUO.

Appendix

Appendix A: Real-world many-objective engineering design optimization problems

A.1 RWMaOP1 [36]

$$f_1(x) = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 0.00001x_6 + 2.73x_7$$

$$f_2(x) = \max\{g_1(x), 0\}$$

$$f_3(x) = \max\{g_2(x), 0\}$$

$$f_4(x) = \max\{g_3(x), 0\}$$

$$f_5(x) = \max\{g_4(x), 0\}$$

$$f_6(x) = \max\{g_5(x), 0\}$$

$$f_7(x) = \max\{g_6(x), 0\}$$

$$f_8(x) = \max\{g_7(x), 0\}$$

$$f_9(x) = \max\{g_8(x), 0\}$$

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

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

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

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

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

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

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

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

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

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

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

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

A.2 RWMaOP2 [37]

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

$$F_2(X) = \delta^T \times F$$

$$F_3(X) = 1000000 \times \left(\frac{1}{f_1}\right)$$

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

$$g_1(X): = \frac{\max(|\sigma_j|) - \sigma_{\text{allowable}}}{\sigma_{\text{allowable}}} \leq 0$$

$$g_2(X) = max\left(\frac{|\sigma_j^{comp}| - \sigma_j^{cr}}{\sigma_j^{cr}}\right) \leq 0, \text{ where } \sigma_j^{cr} = \frac{kA_jE}{L_j^2}$$

A.3 RWMaOP3 [38]

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

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

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

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

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

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

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

A.4 RWMaOP4 [39]

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

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

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

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

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

A.5 RWMaOP5 [40]

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

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

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

$$f_4(\mathbf{x}) = 0.153 - 0.322 \times a + 0.396 \times \Delta HA + 0.424 \times \Delta OA + 0.0226 \times OPTT + 0.175 \times a^2 + 0.0185 \times \Delta HA \times a - 0.0701 \times \Delta HA^2 - 0.251 \times \Delta OA \times a + 0.179 \times \Delta OA \times \Delta HA + 0.0150 \times \Delta OA^2 + 0.0134 \times OPTT \times a + 0.0296 \times OPTT \times \Delta HA + 0.0752 \times OPTT \times \Delta OA + 0.0192 \times OPTT^2$$

Data availability

No data was used for the research described in the article.

References

[1] W. Qiu, J. Zhu, G. Wu, M. Fan, P.N. Suganthan, Evolutionary many-objective algorithm based on fractional dominance relation and improved objective space decomposition strategy, Swarm. Evol. Comput. 60 (2021) 100776.
[2] N.H. Luong, H. La Poutŕe, P.A.N Bosman, Multi-objective gene-pool optimal mixing evolutionary algorithm with the interleaved multi-start scheme, Swarm. Evol. Comput. 40 (2018) 238–254.
[3] D. Adalja, K. Kalita, L. Cepova, P. Patel, N. Mashru, P. Jangir, Arpita, Advancing truss structure optimization— A multi-objective weighted average algorithm with enhanced convergence and diversity, Results. Eng. 25 (March) (2025) 104241, https://doi.org/10.1016/j.rineng.2025.104241.
[4] Z. Wang, Y.S. Ong, H. Ishibuchi, On scalable multiobjective test problems with hardly dominated boundaries, IEEE Trans. Evol. Comput. 23 (2) (2019) 217–231.
[5] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.
[6] I. Hisao, T. Noritaka, N. Yusuke, Evolutionary many-objective optimization: A short review, in: Proc. of the IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 2419–2426.
[7] Z. He, G.G. Yen, J. Zhang, Fuzzy-based pareto optimality for many-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 18 (2) (2014) 269–285.
[8] M. Elarbi, S. Bechikh, A. Gupta, L.B. Ben Said, Y. Ong, A new decomposition-based NSGA-II for many-objective optimization, IEEE Trans. Syst., Man Cybernet.: Syst. 48 (7) (2018) 1191–1210.
[9] Y. Liu, N. Zhu, K. Li, M. Li, J. Zheng, K. Li, An angle dominance criterion for evolutionary many-objective optimization, Inf Sci (Ny) 509 (2020) 376–399.
[10] Y. Tian, R. Cheng, X. Zhang, Y. Su, Y. Jin, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, IEEE Trans. Evol. Comput. 23 (2) (2019) 331–345.
[11] J. Shen, P. Wang, X. Wang, A controlled strengthened dominance relation for evolutionary many-objective optimization, IEEe Trans. Cybern. 52 (5) (2022 May) 3645–3657, https://doi.org/10.1109/TCYB.2020.3015998.
[12] W. Zhang, J. Liu, J. Liu, Y. Liu, S. Tan, A dual distance dominance based evolutionary algorithm with selection-replacement operator for many-objective optimization, Expert. Syst. Appl. 237 (2024) 121244.
[13] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[14] X. He, Y. Zhou, Z. Chen, Q. Zhang, Evolutionary many-objective optimization based on dynamical decomposition, IEEE Trans. Evol. Comput. 23 (3) (2019) 361–375, https://doi.org/10.1109/TEVC.2018.2865590. June.
[15] D. Han, W. Du, W. Du, Y. Jin, C. Wu, An adaptive decomposition-based evolutionary algorithm for many-objective optimization, Inf. Sci. (Ny) 491 (2019) 204–222.
[16] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.
[17] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791.
[18] M. Premkumar, P. Jangir, R. Sowmya, R.M. Elavarasan, Many-objective gradient-based optimizer to solve optimal power flow problems: analysis and validations, Eng. Appl. Artif. Intell. 106 (2021) 104479.
[19] E.M.N. Figueiredo, T.B. Ludermir, C.J.A. Bastos-Filho, Many objective particle swarm optimization, Inf. Sci. (Ny) 374 (2016) 115–134.
[20] P. Jangir, P. Manoharan, S. Pandya, R. Sowmya, MaOTLBO: many-objective teaching-learning-based optimizer for control and monitoring the optimal power flow of modern power systems, Int. J. Indust. Eng. Comput. 14 (2) (2023) 293–308.
[21] H. Ishibuchi, Y. Setoguchi, H. Masuda, Y. Nojima, Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes, IEEE Trans. Evol. Comput. 21 (2) (2017) 169–190.
[22] S. Liu, Q. Yu, Q. Lin, K.C. Tan, An adaptive clustering-based evolutionary algorithm for many-objective optimization problems, Inf. Sci. (Ny) 537 (2020) 261–283.
[23] X. Zhou, X. Wang, X. Gu, A decomposition-based multiobjective evolutionary algorithm with weight vector adaptation, Swarm. Evol. Comput. 61 (2021) 100825.
[24] L.R.C. de Farias, A.F.R. Araújo, A decomposition-based many-objective evolutionary algorithm updating weights when required, Swarm. Evol. Comput. 68 (2022) 100980.
[25] K. Shang, H. Ishibuchi, A new hypervolume-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 24 (5) (2020) 839–852.
[26] Y. Sun, G.G. Yen, Z. Yi, IGD indicator-based evolutionary algorithm for many-objective optimization problems, IEEE Trans. Evol. Comput. 23 (2) (2019) 173–187.
[27] F. Li, R. Cheng, J. Liu, Y. Jin, A two-stage R^2 indicator based evolutionary algorithm for many-objective optimization, Appl. Soft. Comput. 67 (2018) 245–260.

[28] B. Li, K. Tang, J. Li, X. Yao, Stochastic ranking algorithm for many-objective optimization based on multiple indicators, IEEE Trans. Evol. Comput. 20 (6) (2016) 924–938.
[29] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artif. Intell. 228 (2015) 45–65.
[30] Z.-Z. Liu, Y. Wang, P.-Q. Huang, AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation, Inf. Sci. (Ny) 509 (2020) 400–419.
[31] J. Shen, P. Wang, H. Dong, J. Li, W. Wang, A multistage evolutionary algorithm for many-objective optimization, Inf. Sci. (Ny) 589 (2022) 531–549.
[32] W. Zhang, J. Liu, Y. Liu, J. Liu, S. Tan, A many-objective evolutionary algorithm under diversity-first selection based framework, Expert. Syst. Appl. 250 (2024) 123949.
[33] F. Ming, W. Gong, L. Wang, A two-stage evolutionary algorithm with balanced convergence and diversity for many-objective optimization, IEEE Trans. Syst., Man, and Cybernetics: Syst. 52 (10) (2022) 6222–6234, https://doi.org/10.1109/TSMC.2022.3143657. Oct.
[34] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.
[35] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proc. of the 2002 Congress on Evolutionary Computation 1, Honolulu, USA, 2002, pp. 825–830.
[36] R. Tanabe, H. Ishibuchi, An easy-to-use real-world multi-objective optimization problem suite, Appl. Soft. Comput. 89 (2020) 106078.
[37] N. Panagant, S. Kumar, G.G. Tejani, N. Pholdee, S. Bureerat, Many objective meta-heuristic methods for solving constrained truss optimisation problems: A comparative analysis, MethodsX. 10 (2023) 102181.
[38] 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. (Basel) Res. Express 4 (3) (2017) 035302.
[39] Y.-S. Chen, Performance enhancement of multiband antennas through a two-stage optimization technique, Int. J. RF Microwave Comput.-Aided Eng. 27 (2) (2017) e21064.
[40] 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. (Long Beach Calif)Comput. Methods Appl. Mech. Eng. 196 (4-6) (2007) 879–893.
[41] C.A. Coello Coello, G.B. Lamont, D.A. Van Veldhuizen, Evolutionary algorithms for solving multi-objective problems. Genetic and Evolutionary Computation Series, Springer, New York, 2007.