Journal of Computational Design and Engineering, 2024, **11**, 16–39

DOI: 10.1093/jcde/qwae055

Advance access publication date: 14 June 2024

Research Article

Many-objective artificial hummingbird algorithm: an effective many-objective algorithm for engineering design problems

Kanak Kalita[1,2,*], Pradeep Jangir[3], Sundaram B. Pandya[4], Robert Čep[5], Laith Abualigah[6,7,8,9,10], Hazem Migdady[11] and Mohammad Sh. Daoud[12]

[1]Department of Mechanical Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Avadi 600062, India
[2]University Centre for Research & Development, Chandigarh University, Mohali 140413, India
[3]Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India
[4]Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392001, India
[5]Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, Ostrava 70800, Czech Republic
[6]Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan
[7]MEU Research Unit, Middle East University, Amman 11831, Jordan
[8]Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan
[9]Jadara Research Center, Jadara University, Irbid 21110, Jordan
[10]Artificial Intelligence and Sensing Technologies (AIST) Research Center, University of Tabuk, Tabuk 71491, Saudi Arabia
[11]CSMIS Department, Oman College of Management and Technology, Barka 320, Oman
[12]College of Engineering, Al Ain University, Abu Dhabi 112612, United Arab Emirates
*Correspondence: kanakkalita02@gmail.com, drkanakkalita@veltech.edu.in

Abstract

Many-objective optimization presents unique challenges in balancing diversity and convergence of solutions. Traditional approaches struggle with this balance, leading to suboptimal solution distributions in the objective space especially at higher number of objectives. This necessitates the need for innovative strategies to adeptly manage these complexities. This study introduces a Many-Objective Artificial Hummingbird Algorithm (MaOAHA), an advanced evolutionary algorithm designed to overcome the limitations of existing many-objective optimization methods. The objectives are to improve convergence rates, maintain solution diversity, and achieve a uniform distribution in the objective space. MaOAHA implements information feedback mechanism (IFM), reference point-based selection and association, non-dominated sorting, and niche preservation. The IFM utilizes historical data from previous generations to inform the update process, thereby improving the algorithm's the exploration and exploitation capabilities. Reference point-based selection, along with non-dominated sorting, ensures solutions are both close to the Pareto front and evenly spread in the objective space. Niche preservation and density estimation strategies are employed to maintain diversity and prevent overcrowding. The comprehensive experimental analysis benchmarks MaOAHA against four leading algorithms *viz.* Many-Objective Gradient-Based Optimizer, Many-Objective Particle Swarm Optimizer, Reference Vector Guided Evolutionary Algorithm, and Nondominated Sorting Genetic Algorithm III. The DTLZ1–DTLZ7 benchmark sets with four, six, and eight objectives and five real-world problems (RWMaOP1–RWMaOP5) are considered for performance assessment of the selected algorithms. The results demonstrate that internal parameter-free MaOAHA significantly outperforms its counterparts, achieving better generational distance by up to 52.38%, inverse generational distance by up to 38.09%, spacing by up to 56%, spread by up to 71.42%, hypervolume by up to 44%, and runtime by up to 52%. These metrics affirm the MaOAHA's capability to enhance the decision-making processes through its adept balance of convergence, diversity, and uniformity.

Keywords: many-objective optimization, multi-objective optimization, diversity preservation, artificial hummingbird algorithm, non-dominated sorting

1. Introduction

1.1. Background

Optimization problems are prevalent in numerous practical scenarios, ranging from managing networks (Xiao *et al.*, 2023) to vehicle routing (Cao *et al.*, 2021). Diverse applications like managing flow-shop schedules (Goli *et al.*, 2023), energy grid (Hu *et al.*, 2024), carbon emission prediction (Luo *et al.*, 2024), etc., come under the ambit of optimization problems.

The field often encounters a specific type of optimization challenge known as the many-objective optimization problem (MaOP). Characterized by its requirement of four or more objectives, MaOP is encapsulated as

$$\begin{aligned} \text{Minimize} \quad & F(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})], \\ \text{s.t.} \quad & \vec{x} \in \Omega, \end{aligned} \tag{1}$$

where $F(\vec{x})$ is an objective function within MaOP, M is the total number of objectives, and $M \geq 4$. The variable $\vec{x} = (x_1, x_2, \ldots, x_n)$

Received: December 29, 2023. **Revised:** June 10, 2024. **Accepted:** June 10, 2024

is a potential solution within the decision space of dimension d. Additionally, Ω denotes a continuous search space.

MaOP is typically classified as an NP-hard problem. Its objectives often clash, making it arduous to find solutions that satisfy all objectives simultaneously. In addressing MaOPs, a set of Pareto non-dominated solutions often represents the optimal solutions. Pareto dominance plays a crucial role in both the strategy formulation for MaOP optimization and in evaluating the efficacy of many-objective optimization algorithms (M. Shi *et al.*, 2023; Y. Shi *et al.*, 2023).

1.2. Literature review

Generally, Many-Objective Evolutionary Algorithms (MaOEAs) are categorized into four distinct types (Guo, 2022):

(i) Based on Pareto dominance like Nondominated Sorting Genetic Algorithm II (NSGA-II, Deb, Agrawal, *et al.*, 2000), Strength Pareto Evolutionary Algorithm II (SPEA-II, Kim *et al.*, 2004), etc.;
(ii) Indicator approaches like Indicator-Based Evolutionary Algorithm (IBEA, Qin *et al.*, 2023);
(iii) Decomposition methods like Penalty-based Boundary Intersection method (Zhang & Li, 2007); and
(iv) Reference vector techniques like Nondominated Sorting Genetic Algorithm III (NSGA-III, Deb & Jain, 2014).

Pareto dominance-based MaOEAs operate through the comparison and selection of solutions via Pareto dominance relations, prioritizing non-dominated over dominated solutions (Lu *et al.*, 2022; Zhang *et al.*, 2024). However, the efficiency of these algorithms diminishes as the number of objectives increases, leading to challenges in selecting suitable candidates for subsequent generations. These algorithms have been developed to address the issue of “dominance resistance”. Researchers have enhanced the algorithm’s selection pressure by either creating new dominance relations or incorporating a modified Pareto dominance concept to widen the dominance scope. Examples include ε-MOEA (Deb, Mohan, *et al.*, 2003), ε-dominance relation (Ikeda *et al.*, 2002), and fuzzy-based dominance relation (He *et al.*, 2014). ε-MOEA operates by segmenting the objective space into numerous hyperboxes, ensuring only a single solution per hyperbox, thus preserving population diversity. The ε-dominance relation brings in the tradeoff rate between objectives, allowing for non-dominated solutions to be slightly less effective in one objective but markedly better in others. This eases the establishment of dominance relationships among solutions and boosts the algorithm’s selection pressure. The Grid-based Evolutionary Algorithm (Yang *et al.*, 2013) suggests a grid-based dominance approach, though its efficacy is dependent on the division count of the objective function. The Hyperplane-assisted Evolutionary Algorithm (Chen *et al.*, 2020) identifies prominent solutions that clearly trend towards the Pareto front, using a hyperplane of neighboring solutions for further distinction. Moreover, the NSGA-II with a strengthened dominance relation (NSGA-II/SDR) (Tian *et al.*, 2019) introduces a new dominance relationship, aiming to strike a balance between convergence and diversity in solutions.

The field of indicator-based algorithms has seen significant developments, such as the Hypervolume Estimation algorithm (S. Liu *et al.*, 2023) and the inverse generational distance (IGD) indicator-based MaOEA (MaOEA/IGD, Afsar *et al.*, 2023). These algorithms utilize the hypervolume (HV, Bradstreet *et al.*, 2008) and IGD (Xu & Li, 2023) metrics, respectively, to assess algorithm performance, guiding the population evolution towards optimal convergence and diversity. However, there is a concern raised in B. Li *et al.* (2016) about the potential limitations of using a single indicator for population evolution, as it might lead to convergence in just a part of the Pareto front. To address this, a Stochastic Ranking-based multiple indicators Algorithm is introduced, balancing the influence of various indicators on population guidance. Another innovative approach in Pamulapati *et al.* (2019) integrates the sum-of-objectives with shift-based density estimation, leveraging the rapid convergence of the former and the diversity preservation of the latter. The Promising Region-based Evolutionary Algorithm (Yuan *et al.*, 2021) employs a ratio-based indicator to direct population search towards promising areas in the objective space. Additionally, a novel indicator-based MOEA (i.e., AR-MOEA, Tian *et al.*, 2018) enhances the IGD indicator, allowing dynamic adjustment of reference points. In contrast, indicator-based MaOEAs assign fitness to solutions using specific indicators. This approach effectively reduces the complexity of comparing multiple objectives with a single-objective metric may bias the selection process, potentially hindering the algorithm’s comprehensive search capability.

A leading example in decomposition-based algorithms is the MOEA based on Decomposition (MOEA/D, Zhang & Li, 2007), which utilizes a predefined set of uniformly distributed reference vectors in the objective space to ensure diverse population evolution. Building on MOEA/D, various adaptations have emerged. For instance, MOEA based on Dominance and Decomposition (MOEA/DD, K. Li *et al.*, 2015), MOEA/D with Adaptive Weight Adjustment (MOEA/D-AWA, Qi *et al.*, 2014), and Multiobjective evolutionary algorithm based on decomposition multiple to multiple (MOEA/D-M2M), which divides a multi-objective problem into simpler subproblems (H.-L. Liu *et al.*, 2014). MOEA/DD melds decomposition-based approaches with Pareto dominance to balance both convergence and diversity. The MOEA/D-AWA introduces an adaptive strategy for adjusting weight vectors, particularly useful for complex Pareto fronts. Lastly, MOEA/D-M2M simplifies the multi-objective problem into more manageable subproblems, tackling them collaboratively. Decomposition-based MaOEAs, on the other hand, convert a many-objective problem into multiple single-objective problems. The effectiveness of these algorithms largely depends on the aggregation function employed. Designing an appropriate aggregation function for a variety of problems remains a challenge, exemplified by the difficulties in setting the penalty factor method.

The challenge with predefined reference vectors lies in their inability to uniformly cover Pareto fronts of varied shapes, highlighting a need for enhancement in decomposition-based algorithms, particularly for irregular Pareto fronts (Ishibuchi *et al.*, 2017). To address this, adaptive reference vector-based algorithms have been developed. Many such approaches are the Many-Objective Gradient-Based Optimizer (MaOGBO, Premkumar *et al.*, 2021), Many-Objective Particle Swarm Optimizer (MaOPSO, Figueiredo *et al.*, 2016), Reference Vector-guided Evolutionary Algorithm (RVEA, Cheng *et al.*, 2016), and NSGA-III (Deb & Jain, 2014), which introduce an adaptive strategy that allows reference vectors to adjust according to the scales of the objective functions. Q. Liu *et al.* (2022) introduced an advanced growing neural gas network. This network, using the current population as its training dataset, dynamically learns the Pareto front’s topology as the population evolves. Another innovative method is the Reference Points-based Evolutionary Algorithm (Y. Liu *et al.*, 2017). Indicator-based algorithms, when reliant on a solitary indicator, may exhibit a bias towards specific subpopulations, leading to potential entrapment in local optima. Algorithms that utilize multiple indicators concurrently must calculate all indicators for each solution in every iteration, significantly escalating the algorithm's complexity. Decomposition-based algorithms often struggle with effectively breaking down problems when faced with a large number of objectives. Meanwhile, reference vector-based algorithms grapple with evenly distributing reference vectors across irregular Pareto fronts.

Of late, several single-objective metaheuristics like Red Deer Algorithm (Fathollahi-Fard *et al.*, 2020), Social Engineering Optimizer (Fathollahi-Fard *et al.*, 2018), and Tree Growth Algorithm (Cheraghalipour *et al.*, 2018) have been proposed. The evolution of evolutionary algorithms (EAs) from single- to many-objective optimization has necessitated the development of sophisticated methods capable of handling the complexity and high dimensionality inherent in MaOPs (Yin *et al.*, 2020; Zhu *et al.*, 2024). The following sections review key advancements in the field, focusing on the methodologies, challenges, and gaps that this research addresses.

Historically, EAs such as NSGA-II (Deb, Agrawal, *et al.*, 2000) and SPEA-II (Zitzler *et al.*, 2001) have demonstrated significant success in multi-objective optimization. However, their performance tends to degrade as the number of objectives increases, primarily due to the loss of selective pressure and the exponential increase in non-dominated solutions (Ishibuchi *et al.*, 2008). This observation spurred research into many-objective optimization algorithms designed to maintain effectiveness in higher dimensional objective spaces.

Recent years have seen the introduction of several many-objective optimization algorithms, such as NSGA-III (Deb & Jain, 2014) which extends NSGA-II to handle many-objective problems by incorporating a reference point approach. Similarly, MOEA/D (Zhang & Li, 2007) and its variants address many-objective optimization by decomposing a multi-objective problem into a number of scalar optimizations subproblems. Wei and Li (2023) proposed an EA that incorporates population preprocessing and a projection distance-assisted elimination mechanism. This approach efficiently reduces the search space and improves the selection process for non-dominated solutions. Choi (2022) developed an optimization approach focusing on hydraulic and water quality criteria within a many-objective optimization framework. This work extends beyond traditional MaOEAs by incorporating domain-specific criteria, demonstrating the flexibility and applicability of MaOEAs. Wu *et al.* (2023) introduced a dynamic EA that leverages prediction mechanisms to adaptively adjust to changing optimization landscapes. This contrasts with static many-objective algorithms such as NSGA-III, which do not inherently account for dynamic environments. The predictive capability of Wu *et al.*'s algorithm represents a significant advancement in enhancing the adaptability of MaOEAs to real-world problems that evolve over time. Wang *et al.* (2022) enhanced a MaOEA using chaotic mapping and a solution ranking mechanism, specifically targeting large-scale optimization problems. This method diverges from traditional indicator approaches like IBEA by introducing chaos theory to maintain diversity and employing a novel ranking strategy to guide the search process effectively. Despite these advancements, achieving a balance between convergence and diversity remains a significant challenge, with most algorithms excelling in one at the expense of the other.

1.3. Motivation and literature gap

The field of optimization has been significantly reshaped with the introduction of many-objective problems, which inherently feature four or more conflicting objectives. Such complexity has unveiled the limitations of traditional EAs—inefficiency in navigating the delicate balance between convergence to the optimal front and maintaining diversity among solutions. Despite the numerous studies conducted in this field, a distinct gap remains in developing algorithms that can adeptly manage these challenges, especially under the constraints of high-dimensional objective spaces. Moreover, existing algorithms often struggle with computational efficiency and scalability when applied to real-world problems with complex constraints and objective interactions.

Current MaOEAs struggle to maintain a harmonious balance between convergence towards the Pareto front and the preservation of diversity among solutions (Cao, Zhao, Gu, *et al.*, 2020; Yu *et al.*, 2024). This imbalance often results in premature convergence or excessive dispersion of solutions, diminishing the quality of the resultant Pareto front. Most algorithms do not fully leverage the historical information of the search process, leading to inefficiencies in exploration and exploitation. The potential insights gained from previous generations are frequently overlooked, which could otherwise guide the search process more effectively towards optimal solutions. As the number of objectives increases, the performance of traditional Pareto dominance- and decomposition-based approaches tends to degrade. This is due to the "curse of dimensionality," where the discrimination capability of these methods diminishes, making it difficult to identify truly non-dominated solutions. Many algorithms assume relatively simple Pareto front geometries and fail to adapt to complex, irregular, or disconnected Pareto fronts that are common in real-world many-objective problems. This limitation restricts their applicability to a broader range of practical scenarios. The computational demand of existing MaOEAs escalates rapidly with the increase in the number of objectives and decision variables, posing significant challenges in terms of scalability and practical applicability to large-scale problems (Cao, Li, Liu, Lv, *et al.*, 2023; T. Zhao *et al.*, 2024).

1.4. Research questions

(i) Can a many-objective metaheuristic algorithm be designed that maintains an effective balance between convergence and diversity?

(ii) In what ways can the strengths of the hummingbird's foraging strategies be abstracted and applied to enhance the search efficiency of EAs in complex optimization landscapes?

1.5. Hypothesis

This work hypothesizes that the Many-Objective Artificial Hummingbird Algorithm (MaOAHA), with its unique combination of reference point-based selection, niche preservation, and an information feedback mechanism (IFM), can outperform existing MaOEAs in both convergence and diversity metrics.

1.6. Justification for MaOAHA

The Artificial Hummingbird Algorithm (AHA) inspired by the natural foraging behavior of hummingbirds, offers innovative mechanisms for search space exploration, adaptability, memory utilization, and solution refinement. These qualities address specific challenges inherent in many-objective optimization, providing a compelling rationale for its selection as the foundation of MaOAHA. By leveraging these characteristics, MaOAHA not only advances the state of the art in many-objective optimization but also demonstrates improved performance over general EAs, particularly in balancing convergence and diversity across complex objective spaces.

MaOAHA introduced in this study is motivated by the need for an algorithm that can effectively navigate the trade-offs between convergence and diversity in many-objective optimization. MaOAHA incorporates innovative mechanisms such as an IFM and niche preservation strategies, which have not been extensively explored in existing MaOEAs. These features allow MaOAHA to outperform leading algorithms in terms of both convergence and diversity across a variety of benchmark and real-world problems. A novel assumption of this work is the strategic utilization of historical information through the IFM to inform the search process in the current generation.

1.7. Contribution to the field

The study contributes to the existing body of knowledge by proposing a novel approach to many-objective optimization that leverages bio-inspired algorithms and advanced selection mechanisms. By addressing the identified gaps and challenges, MaOAHA represents a significant step forward in the development of effective, efficient, and scalable solutions for MaOPs. In this study, a novel approach is presented for better balance between convergence and diversity in many-objective optimization, through AHA (W. Zhao *et al.*, 2022), IFM, reference point-based selection and association, non-dominated sorting, niche preservation, and density estimation-based MaOAHA. The key research contributions of this paper are outlined as follows:

(i) The selection of AHA algorithm is based on their performance in generating diverse and high-quality solutions in single-objective problem. Through the global search capability of AHA, operator selection enhances the MaOAHA ability to explore and exploit the search space effectively.

(ii) The paper introduces an IFM strategy for the shortcomings that had wasted a lot of useful information. In the IFM, the combined historical pieces of information of individuals based on the weighted sum method are carried over to the next generation. This ensures superior convergence properties.

(iii) A strategy for reference point-based selection guides the selection process, ensuring that the chosen solutions are not just close to the optimal front (convergence) but also spread out across the entire objective space (diversity). Associating each solution to the nearest reference point by perpendicular distance leads to identifying well-represented areas in the objective space. Non-dominated sorting method ensures that the algorithm focuses on solutions that are closer to the Pareto-optimal front, aiding convergence.

(iv) A niche preservation strategy for boundary individuals is proposed, aimed at boosting diversity while removing those with overcrowding in specific regions of the objective space, thereby speeding up the algorithm's overall convergence rate. Additionally, a density estimation strategy for maintaining diversity is detailed, ensuring both uniformity and extensive coverage in the population distribution.

(v) The effectiveness of the newly developed MaOAHA is validated through comparisons with MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms across DTLZ1–DTLZ7 benchmark sets with four, six, and eight objectives and five real-world (RWMaOP1–RWMaOP5) problems. The results from these experiments highlight MaOAHA capability to adeptly manage various problem types, underscoring its robust overall performance.

An overview of AHA algorithm is given in Section 2, followed by a presentation of the proposed MaOAHA algorithm in Section 3. Experimental comparisons and evaluations are presented in Section 4 and a conclusion in Section 5.

2. Artificial Hummingbird Algorithm

The AHA (W. Zhao *et al.*, 2022) is inspired by the unique flying abilities and smart food-gathering tactics of hummingbird foraging techniques: guided, territorial, and migratory foraging, as depicted in Fig. 1. AHA algorithm features a visit table that emulates hummingbirds' extraordinary memory, aiding in global optimization tasks. AHA excels in balancing exploration and exploitation stages and demonstrates high efficiency in exploration with superior solution precision. It initiates by randomly creating potential solutions. Each simulated hummingbird in the group, when arriving at a new location, engages in random searches for food sources, thereby initializing the group. This accidental discovery of initial food sources is part of the initialization, as shown in Equation (2):

$$x_i = LB + r \cdot (UB - LB) \tag{2}$$

where LB and UB denote the lower and upper interval limits, r is a random number between 0 and 1, and x_i indicates the location discovered by the ith hummingbird. The visit table starts as follows:

$$VT_{i,j} = \begin{cases} 0, if & i \neq j \\ null, & i = j \end{cases}. \tag{3}$$

Equation (3) illustrates two scenarios; $VT_{i,j} = \text{null}$ indicates feeding on a static food source, whereas $VT_{i,j} = 0$ signifies the ith hummingbird has just investigated the jth food source. The three hummingbird flight patterns are adapted for multi-dimensional spaces. The axial flight enables movement to any axis point, as outlined in Equation (4):

$$D_{\mathrm{Af}}{}^{(i)} = \begin{cases} 1, & if\, i = Randi\left([1, d]\right), \\ 0, & else, \end{cases}. \tag{4}$$

Equation (5) details the diagonal flight and Equation (6) describes the omnidirectional flight.

$$D_{\mathrm{Df}}{}^{(i)} = \begin{cases} 1, & if\, i = G\left(j\right), j \in [1, c], G = Randperm\left(c\right) \\ 0, & else, \end{cases} \tag{5}$$

$$D_{\mathrm{Of}}{}^{(i)} = 1, i = 1, 2, \ldots, n. \tag{6}$$

In these equations, c varies from 2 to $[r_1(d{-}2) + 1]$, $Randi([1, d])$ generates a random number from 1 to d, $Randperm(c)$ yields a random number sequence up to c, and r_1 represents a random number in the (0, 1) range. Equation (7) shows the update process for candidate solutions via guided foraging. Equation (8) outlines the update method when a hummingbird locates a food source closer to the target.

$$v_i\left(t + 1\right) = x_{i,\mathrm{target}} + g \cdot D_t \cdot \left(x_i\left(t\right) - x_{i,\mathrm{target}}\left(t\right)\right), t \in \left\{Af, Df, Of\right\} \tag{7}$$

$$g \sim N\left(0, 1\right). \tag{8}$$

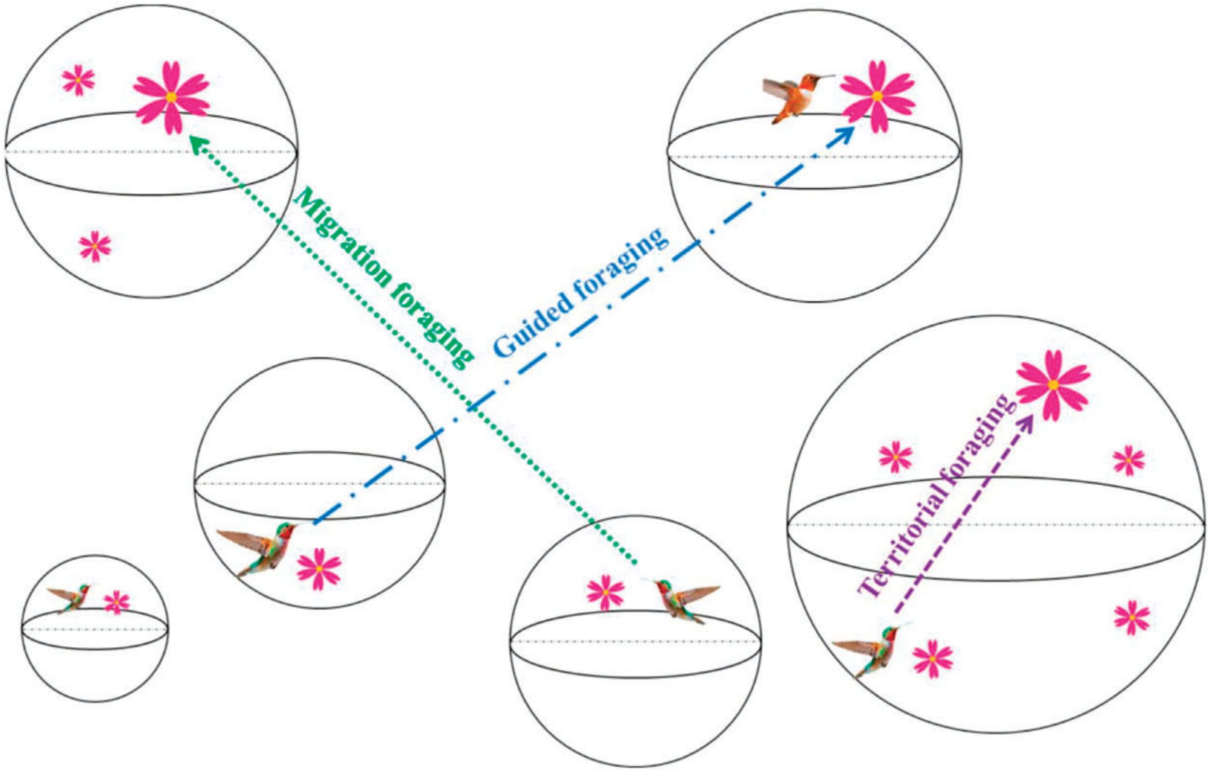

Figure 1: Three foraging behaviors of AHA.

Equation (9) defines how fitness values of candidate solutions are updated.

$$x_i\,(t+1) = \begin{cases} x_i\,(t)\,, & f\,(x_i\,(t)) \le f\,(v_i\,(t+1)) \\ v_i\,(t+1)\,, & else \end{cases}. \tag{9}$$

The territorial foraging update method for candidate solutions is presented in Equations (10) and (11):

$$v_i\,(t+1) = x_i + k \cdot D_t \cdot x_i\,(t)\,, t \in \{Af, Df, Of\} \tag{10}$$

$$k \sim N\,(0,1) \tag{11}$$

where $x_{i,\,\mathrm{target}}$ (t) is the position of the target solution and g is a guiding factor. $f(x_i(t))$ and $f(v_i(t+1))$ denote the fitness values of the candidate solution $x_i(t)$ and the updated solution $v_i(t+1)$, respectively. k is a guiding factor and D_t represents one of the flight modes. Finally, the equation for updating the location of artificial hummingbirds with poor nectar refilling rates through migratory foraging is given in Equation (12):

$$x_{\mathrm{worst}}\,(t+1) = LB + r \cdot (UB - LB)\,, \text{when } Mr = t \tag{12}$$

where x_{worst} denotes the candidate solution with the lowest nectar refill rate, t is the current iteration, and Mr is the migration coefficient, typically $Mr = 2N$, with N being the population size.

3. Proposed MaOAHA

The development of the MaOAHA from the single-objective AHA, incorporating novel mechanisms designed to address the specific challenges of many-objective optimization. The key features are the use of reference points, niche preservation, and an IFM, which collectively ensure an effective balance between exploration and exploitation and diversity preservation in a many-objective context.

3.1. Transition from AHA to MaOAHA

The AHA, inspired by the foraging behavior of hummingbirds, is known for its efficiency in exploring and exploiting search spaces in single-objective optimization tasks. To adapt AHA for many-objective optimization, several many-objective optimization strategies are integrated into its framework, transforming it into MaOAHA. This adaptation is aimed at maintaining high performance when dealing with a larger number of objectives, where traditional EAs tend to struggle with preserving diversity and ensuring convergence.

3.2. Reference point selection

Reference points are critical in many-objective optimization for guiding the search towards a diverse set of solutions across the Pareto front. In MaOAHA, a set of reference points is generated using Das and Dennis's technique, which is designed to approximate the distribution of solutions in the objective space. This approach allows MaOAHA to maintain a comprehensive exploration of the objective space, enhancing the algorithm's ability to uncover a wide range of Pareto-optimal solutions.

3.3. Niche preservation

To ensure that the generated solutions are not only diverse but also well-distributed, MaOAHA employs a niche preservation strategy. This strategy involves associating each solution with the nearest reference point and selecting solutions based on their niche count. This method prevents overcrowding in densely populated regions of the solution space and promotes the discovery of underrepresented areas, fostering a uniform distribution of solutions along the Pareto front.

3.4. Information feedback mechanism

The IFM is a novel component of MaOAHA that leverages historical search information to guide the optimization process. By integrating feedback from previous generations into the generation of new solutions, IFM enhances the algorithm's convergence properties and its ability to adapt to the dynamic landscape of many-objective problems. This mechanism ensures that the search process is not only driven by current population metrics but also informed by the accumulated knowledge of the solution space.

3.5. Algorithmic framework

MaOAHA begins with a randomly generated population, which is iteratively improved through the mechanisms described above. The selection of offspring for the next generation is based on their performance relative to the reference points and their contribution to diversity, as determined by niche preservation. The integration of IFM ensures continuous adaptation and improvement of the search strategy, leading to a balanced exploration and exploitation of the search space. MaOAHA 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's technique $H = \binom{M+p-1}{p}$, as $H \approx N$. Das and Dennis's technique is a method for systematically generating a set of uniformly distributed reference points on the $a(M-1)$ dimensional unit simplex, where M is the number of objectives in a MaOP. This simplex is a geometric representation in which all points sum up to one, creating a space equally inclined to all objective axes with an intercept of one on each axis. The technique's main advantage lies in its ability to create reference points that are evenly spread, aiding in covering the entire Pareto front as uniformly as possible. For example, in a four-objective problem ($M = 4$), the reference points are created within a tetrahedron, with apexes at (1,0,0,0), (0,1,0,0), (0,0,1,0), and (0,0,0,1). If the number of divisions is ($p = 4$) along each objective axis, the total number of reference points H created on a tetrahedron is 35. This means in a three-dimensional space, these points would be distributed within a shape where each apex represents one of the objectives and the points are evenly distributed throughout the interior and on the faces of this shape. In a six-objective problem ($M = 6$), under the same conditions ($p = 4$), the total number of reference points H is 126. These points are distributed within a five-dimensional simplex, a hyper-dimensional equivalent of a tetrahedron, with each vertex representing one of the objectives. In the MaOAHA algorithm, these reference points guide the search process by providing diverse directions for exploration. By associating solutions with the nearest reference point and striving to improve the representation around each point, the algorithm ensures that the evolved solutions cover the Pareto front uniformly. This method not only promotes diversity among the solutions but also helps in identifying regions of the Pareto front that are underrepresented, thereby directing computational effort to these areas. A set of reference points is supplied and the algorithm is designed to find solutions near these points. The current generation is t, x_i^t and x_i^{t+1} is the ith individual at t and $(t+1)$ generation. u_i^{t+1} is the ith individual at the $(t+1)$ generation generated through the AHA 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, calculate x_i^{t+1} according to u_i^{t+1} generated through the AHA algorithm and IFM (Equation 13):

$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t;\ \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t},\ \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t},\ \partial_1 + \partial_2 = 1 \tag{13}$$

where x_k^t is the kth individual chosen from the tth generation, the fitness value of x_k^t is f_k^t, ∂_1 and ∂_2 are weight coefficients. Generate offspring population Q_t. Q_t is the set of x_i^{t+1}. The combined population $R_t = P_t \cup Q_t$ is sorted into different w-non-dominant levels $(F_1, F_2, \ldots, F_l \ldots, F_w)$. Begin from F_1, all individuals in level 1 to l are added to S_t and remaining members of R_t are rejected. If $|S_t| = N$; no other actions are required and the next generation is begun with $P_{t+1} = S_t$. Otherwise, solutions in S_t/F_l are included in $P_{t+1} = S_t/F_l$ and the rest $(K = N - |P_{t+1}|)$ individuals are selected from the last front F_l (presented in Algorithm 1). For selecting individuals from F_l, a niche-preserving operator is used. First, each population member of P_{t+1} and F_l is normalized (presented in Algorithm 2) by using the current population spread so that all objective vectors and reference points have commensurate values. Thereafter, each member of P_{t+1} and F_l is associated (presented in Algorithm 3) with a specific reference point by using the shortest perpendicular distance ($d()$) of each population member with a reference line created by joining the origin with a supplied reference point. Then, a careful niching strategy (described in Algorithm 5) that improve the diversity of MaOAHA algorithm is employed to choose those F_l members that are associated with the least represented reference points niche count ρ_i in P_{t+1} and check termination condition is met. If the termination condition is not satisfied, $t = t + 1$ than repeat and if it is satisfied, P_{t+1} is generated, it is then applied to generate a new population Q_{t+1} by AHA algorithm. MaOAHA algorithm that incorporates IFM to effectively guide the search process, ensuring a balance between exploration and exploitation. This leads to improved convergence, coverage, and diversity preservation, which are crucial aspects of many-objective optimization. MaOAHA algorithm does not require setting any new parameter other than the usual AHA parameters such as the population size, termination parameter, and their associated parameters.

3.6. Computational complexity

The computational complexity of MaOAHA is analyzed, demonstrating its efficiency in handling many-objective problems. The careful selection strategy employed in MaOAHA, alongside the efficient implementation of its core mechanisms, results in a computational complexity that remains manageable even as the number of objectives increases. The computational complexity MaOAHA for M-Objectives is $O(N^2 \log^{M-2} N)$ or $O(N^2 M)$, whichever is larger.

3.7. Parameter settings

MaOAHA retains the flexibility of the original AHA in terms of parameter settings, requiring only the standard parameters such as population size and termination criteria. Simplicity in tuning parameters makes MaOAHA both powerful and accessible for a wide range of optimization tasks.

The flowchart of MaOAHA algorithm can be shown in Fig. 2.

Algorithm 1: Generation t of MaOAHA algorithm with IFM procedure.

```
Input:  $N$ (population size), $M$ (no. of objectives), AHA algorithm parameters, and initial population $P_t(t = 0)$,
Output: $Q_{t+1} = AHA(P_{t+1})$
$H$ calculated using Das and Dennis's technique, structured reference points $Z^s$, supplied aspiration points $Z^a$, $S_t = \phi$, $i = 1$
Proposed IFM
Apply AHA algorithm on the initial population $P_t$ to generate $u_i^{t+1}$, calculation of $x_i^{t+1}$ 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 $x_i^{t+1}$)
$R_t = P_t\ U\ Q_t$
Different non-domination levels $(F_1, F_2,\ldots,F_l) =$ Non-dominated-sort $(R_t)$
repeat
    $S_t = S_t\ U\ F_i$ and $i = i + 1$
until $| S_t | \geq N$
    Last front to be included: $F_{l=} \cup_{i=1}^{l} F_i$
if $| S_t | = N$ then
    $P_{t+1} = S_t$
else
    $P_{t+1} = S_t/F_l$
    Point to chosen from last front $(F_l)$ : $K = N - |P_{t+1}|$
    Normalize objectives and create reference set $Z^r$:
    Normalize ($f^n$, $S_t$, $Z^r$, $Z^s$, $Z^a$); brief explanation in Algorithm 2
    Associate each member $s$ of $S_t$ with a reference point:
    $[\pi(\mathbf{s}),\ \mathbf{d}(\mathbf{s})] = Associate\ (\mathbf{S_t},\ \mathbf{Z^r})$; brief explanation in Algorithm 3 % $\pi(s)$ : closest reference point, $d$ : distance between $s$ and $\pi(s)$
    Compute niche count of reference point $j\ \in\ Z^r$ :
    $\rho_j = \sum_{s \in s_t/F_l} ((\pi(s) = j), 1\ : 0)$;
    Choose $K$ members one at a time $F_l$ to construct
    $P_{t+1}$ : Niching($K,\ \rho_j,\ \pi,\ d,\ Z^r,\ F_l,\ P_{t+1}$); represent in Algorithm 4
end if
```

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

```
Input:  $S_t,\ Z^s$ (structured points) or $Z^a$ (supplied points)
Output: $f^n,\ Z^r$ (reference points on normalized hyperplane)
for $j = 1$ to $M$ do
    Compute ideal point: $Z_j^{\min} = min_{s \in s_t} f_j(s)$
    Translate objectives: $f_j'(s) = f_j(s) - Z_j^{\min} \forall s \in S_t$
    Compute extreme points: $Z^{j,\max} = s$:
     $\arg\min_{s \in s_t} ASF(s, w^j) =$ where $w^j = (\varepsilon 1, \ldots.., \varepsilon j)^T)$,
    $\varepsilon = 10^{-6}$, and $w_j^j = 1$
end for
Compute intercepts $a_j$ for $j = 1, .., M$
Normalize objectives $f_i^n(X)$ using
$f_i^n(X) = \frac{f_i'(X)}{a_i - Z_i^{\min}}$, $fori = 1, 2, \ldots., M$
if $Z^a$ is given then
    Map each (aspiration) point on normalized hyperplane $f_i^n(X)$ and save the points in the set $Z^r$
Else
    $Z^r$=$Z^s$
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 $d^{\perp}\ (s,\ w) = s - w^T s/ \parallel w \parallel$
    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 $(K,\ \rho_j,\ \pi,\ d,\ Z^r,\ F_l,\ P_{t+1})$ 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 $k \leq K$ 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
            $P_{t+1} =\ P_{t+1}\ \cup\ (s\ :\ argmin_{s\ \in\ I_{\bar{j}}} d_s)$
        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
```

4. Results and Discussion

4.1. Experimental settings

In order to verify the effectiveness of the MaOAHA, the DTLZ1–DTLZ7 (Deb, Thiele, *et al.*, 2003) benchmark (Appendix 1) and five real-world engineering design (Appendix 2): car cab design (RW-MaOP1, Tanabe & Ishibuchi, 2020), 10-bar truss structure (RW-MaOP2, Panagant *et al.*, 2023), water and oil repellent fabric development (RWMaOP3, Ahmad *et al.*, 2017), ultra-wideband antenna design (RWMaOP4, Chen, 2017), and liquid-rocket single element injector design (RWMaOP5, Goel *et al.*, 2007) problems are used in this paper. The number of decision variables for the DTLZ problems is $k + M - 1$, M is the number of objective functions. k is set to 5 in DTLZ1, k is set to 10 in DTLZ2–DTLZ6, and k is set to 20 in DTLZ7.

4.1.1. *Benchmarks and parameter setting*

In this study, the performance of MaOAHA is evaluated by empirically comparing it with some state-of-the-art multiobjective algorithms for MaOPs, namely, MaOGBO (Premkumar *et al.*, 2021), MaOPSO (Figueiredo *et al.*, 2016), RVEA (Cheng *et al.*, 2016), and NSGA-III (Deb & Jain, 2014), will be verified. The experiments are conducted on a MATLAB R2020a environment on an Intel Core (TM) i7-9700 CPU. Each algorithm performs 30 times, the size of

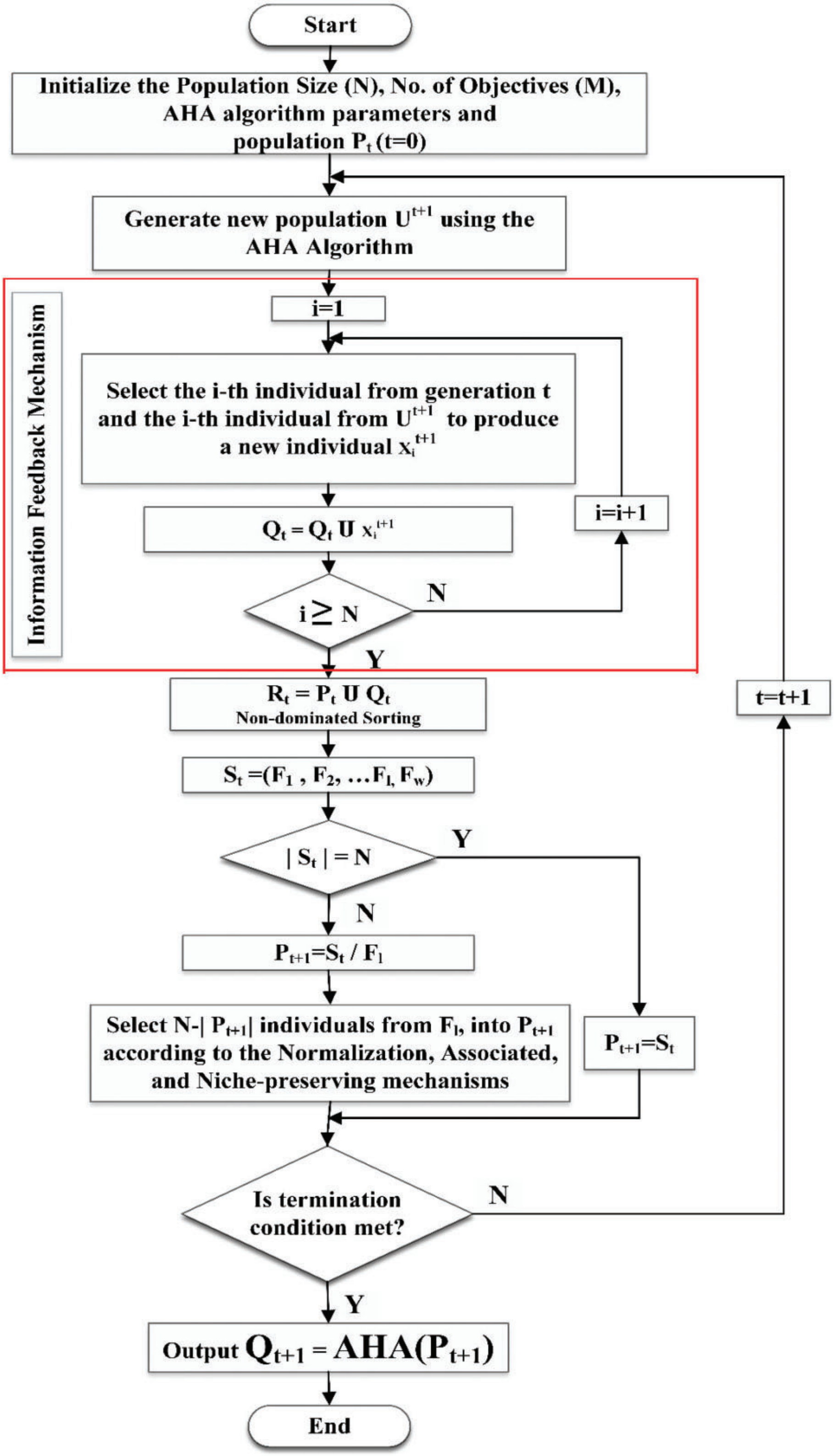
Start
Initialize the Population Size (N), No. of Objectives (M), AHA algorithm parameters and population P_t (t=0)
Generate new population U^{t+1} using the AHA Algorithm
Information Feedback Mechanism
i=1
Select the i-th individual from generation t and the i-th individual from U^{t+1} to produce a new individual x_i^{t+1}
$Q_t = Q_t \cup x_i^{t+1}$
i=i+1
$i \geq N$
N
Y
$R_t = P_t \cup Q_t$
Non-dominated Sorting
$S_t = (F_1, F_2, \ldots F_l, F_w)$
$|S_t| = N$
Y
N
$P_{t+1} = S_t / F_l$
Select N-| P_{t+1}| individuals from F_l, into P_{t+1} according to the Normalization, Associated, and Niche-preserving mechanisms
$P_{t+1} = S_t$
t=t+1
Is termination condition met?
N
Y
Output $Q_{t+1} = AHA(P_{t+1})$
End

Figure 2: Flowchart of MaOAHA algorithm.

Table 1: Properties of the quality indicators.

Quality indicator (Chen, 2017)	Convergence	Diversity	Uniformity	Cardinality	Computational burden
GD	✓				
SD		✓			
SP			✓		
RT					✓
IGD	✓	✓	✓		
HV	✓	✓	✓	✓	

population N is set to $N = 105$, 132, and 156 for all of the involved algorithms on $M = 4$, 6, and 8 objectives problems. The *MaxFEs* is set to 1×10^5 for all of the test instances. NSGA-III adopts the same parameter settings, where the crossover probability (P_c) and mutation probability (P_m), the distribution index of simulated binary crossover (η_c), and polynomial mutation (η_m), are set to 1, $1/D$, 20, and 20. To ensure a comprehensive evaluation of the MaOAHA, the migration coefficient (Mr) is set to $2N$, where N is the population size.

4.1.2. *Performance measures*

This paper adopts generational distance (GD), spread (SD), spacing (SP), runtime (RT), IGD, and HV quality indicator (Coello Coello *et al.*, 2007), shown in Table 1 and Fig. 3. A higher value of *HV* and lower value of *IGD*, *GD*, *SD*, *RT*, and *SP* refer to better performance. The Wilcoxon rank-sum test (WRST) with 0.05 significance level is applied to better performance (+), a worse performance (−), and an equal (=) performance compared with MaOAHA.

4.2. Experimental results on DTLZ problems

Table 2 presents the GD results for MaOAHA, MaOGBO, MaOPSO, RVEA, and NSGA-III on DTLZ test problems. For problem, in the DTLZ1 problem with 4-M and 8-D, MaOAHA records a mean GD of 4.3141e−2 (3.79e−2), which is significantly lower than its counterparts like MaOGBO with 1.0241e−1 (1.72e−1) and RVEA with 9.5822e−2 (1.58e−1). In DTLZ3 with 6-M and 15-D, MaOAHA mean GD is 3.2731e+0 (1.23e+0), surpassing other algorithms like MaOPSO with 7.5772e+0 (7.76e−1) and NSGA-III with 1.0848e+2 (2.39e+1). The proportion of test problems where MaOAHA outperforms other algorithms like MaOGBO, MaOPSO, RVEA, and NSGA-III across the DTLZ test suite ranges from high to dominant. For problem, in DTLZ1, DTLZ2, DTLZ4, and DTLZ5 problems, MaOAHA achieves better results in more than 50% of the cases when compared with each of these algorithms. Among these, MaOAHA demonstrates a notable performance, achieving the best results in several problems. From Table 2, it is observed that MaOAHA outperforms 11 out of 21 best results, whereas MaOGBO, MaOPSO, RVEA, and NSGA-III achieve 6, 1, 0, and 3 best results in terms of the GD values, respectively. A lower mean GD indicates a closer approximation to the true Pareto front, signifying better performance of MaOAHA algorithm and effectiveness in dealing with complex MaOPs.

Table 3 displays the IGD results for MaOGBO, MaOPSO, RVEA, and NSGA-III, including MaOAHA, on DTLZ test problems. In DTLZ1 with 4-M and 8-D, MaOAHA has an IGD of 5.8231e−1 (std 9.07e−1), which is higher than RVEA and NSGA-III, indicating less favorable performance. However, in DTLZ4 with 6-M and 15-D, MaOAHA achieves an IGD of 4.3262e−1 (std 9.20e−4), outperforming NSGA-III. The performance of MaOAHA is notably varied across different DTLZ problems. For problem, in DTLZ5 with 4-M and 13-D, its IGD is 6.1569e−2 (std 1.01e−2), which is comparable with the other algorithms. In Table 3, IGD values of MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms are better in 8, 2, 2, and 1 out of 21 cases. These proportions indicate a varied but notable efficacy of MaOAHA across different scenarios. In particular, its performance against MaOGBO stands out with a high percentage of superiority, suggesting a distinct advantage in those test problems. Conversely, the percentages against MaOPSO and RVEA indicate a more competitive scenario, with MaOAHA showing a significant lead in about a third of the cases. Against NSGA-III, MaOAHA demonstrates a notable edge in nearly half of the test problems, underlining its efficiency in those scenarios. Therefore, based on these proportions, it is reasonable to conclude that MaOAHA exhibits a strong competitive edge in a significant number of problems across the DTLZ test suite shown in Fig. 4 particularly against MaOGBO and NSGA-III, while presenting a balanced performance against MaOPSO and RVEA.

Table 4 illustrates the SP results MaOAHA, MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms (MOEAs) on DTLZ test problems. MaOAHA achieves the best results in 11/21 test problems, exemplifying its efficiency in spacing of solutions. In comparison, MaOGBO, MaOPSO, RVEA, and NSGA-III achieve 4, 1, 4, and 1 best results, respectively. This performance is particularly evident in problems like DTLZ1, DTLZ4, and DTLZ5, where MaOAHA consistently records lower mean SP values, indicating better solution distribution. For example, in DTLZ1 with 4-M and 8-D, MaOAHA SP value is 1.1166e−1 (std 1.37e−1), which is significantly lower than MaOGBO 7.2468e−1 (std 1.22e+0) and MaOPSO 1.3383e−1 (std 1.11e−1). The WRST further supports these findings, indicating that MaOAHA performance is not only statistically significant but also consistently superior across a range of DTLZ problems. This is especially noteworthy in problems like DTLZ2 and DTLZ7, where MaOAHA spacing of solutions is markedly better than that of most competitors. Therefore, it is reasonable to conclude that MaOAHA outperforms its competitors on most DTLZ problems in terms of the SP metric. The algorithm ability to achieve the best results, mainly in DTLZ1, DTLZ2, DTLZ4, and DTLZ5, highlights its superior performance in ensuring well-distributed solutions across the Pareto front shown in Fig. 4. While other competitors like MaOGBO and MaOPSO show competitive results in certain problems, MaOAHA overall performance across the DTLZ suite suggests its effectiveness in spacing of solutions in MaOPs.

Table 5 showcases the SD results of various algorithms on DTLZ problems, with a specific focus on MaOAHA. Overall, MaOAHA demonstrates a significant level of performance in terms of the SD metric. MaOAHA achieves the best SD results in 11/21 test problems. This highlights its capability in maintaining a balanced spread of solutions across the Pareto front. To put this into perspective, MaOAHA outperforms MaOGBO, MaOPSO, RVEA, and NSGA-III in a significant majority of the test problems. As can be seen from Table 5, MaOAHA achieves the best performance in terms of SD values, having obtained 11 best results, followed

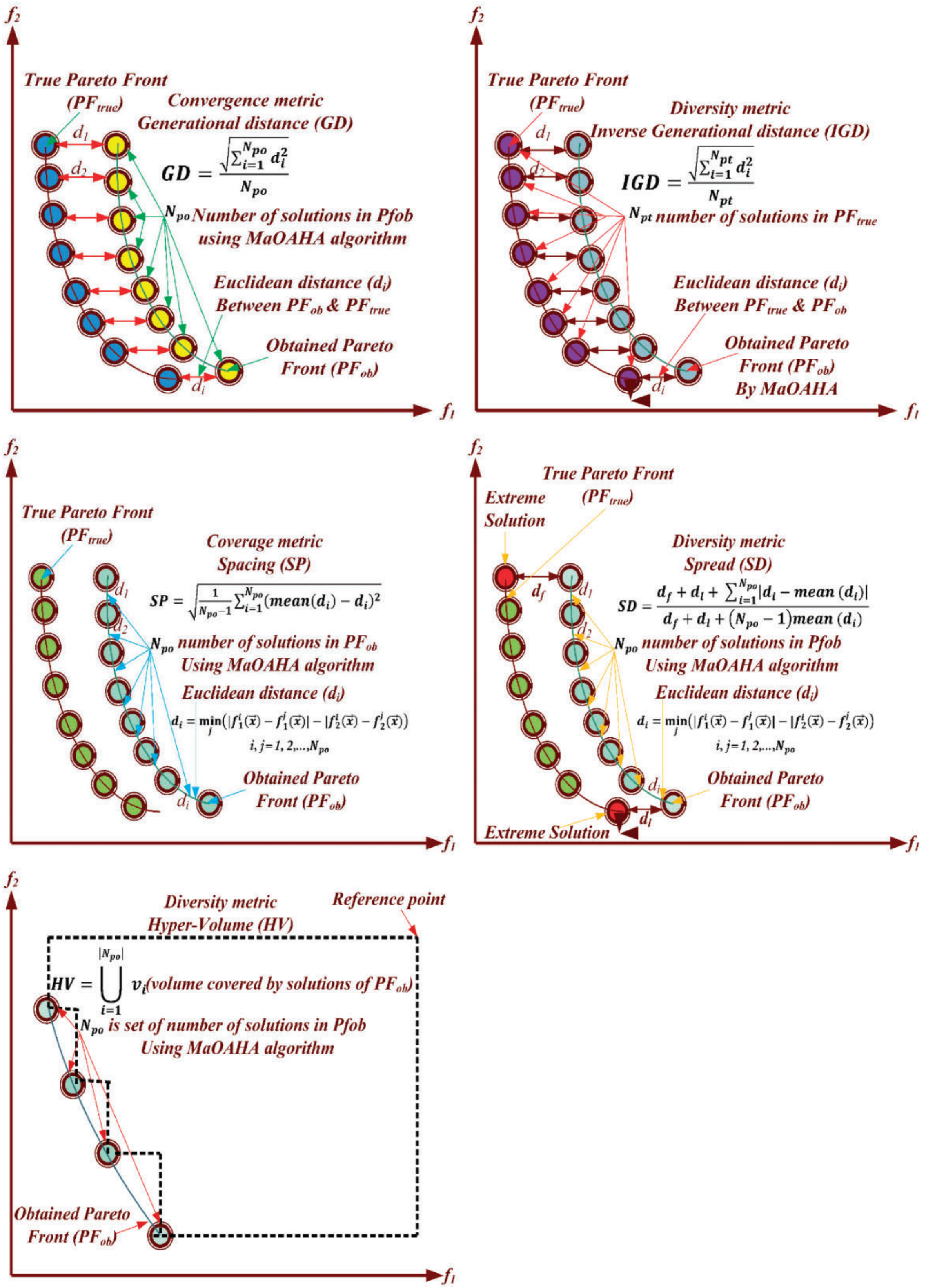
f_2
True Pareto Front
(PF_{true})
Convergence metric
Generational distance (GD)
$GD = \frac{\sqrt{\sum_{i=1}^{N_{po}} d_i^2}}{N_{po}}$
N_{po} Number of solutions in Pfob using MaOAHA algorithm
Euclidean distance (d_i) Between PF_{ob} & PF_{true}
Obtained Pareto Front (PF_{ob})
d_1
d_2
d_i
f_1
f_2
True Pareto Front
(PF_{true})
Diversity metric
Inverse Generational distance (IGD)
$IGD = \frac{\sqrt{\sum_{i=1}^{N_{pt}} d_i^2}}{N_{pt}}$
N_{pt} number of solutions in PF_{true}
Euclidean distance (d_i) Between PF_{true} & PF_{ob}
Obtained Pareto Front (PF_{ob}) By MaOAHA
d_1
d_2
d_i
f_1
f_2
True Pareto Front
(PF_{true})
Coverage metric
Spacing (SP)
$SP = \sqrt{\frac{1}{N_{po}-1}\sum_{i=1}^{N_{po}}(mean(d_i) - d_i)^2}$
N_{po} number of solutions in PF_{ob} Using MaOAHA algorithm
Euclidean distance (d_i)
$d_i = \min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| - |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$
$i, j = 1, 2, \dots, N_{po}$
Obtained Pareto Front (PF_{ob})
d_1
d_2
d_i
f_1
f_2
True Pareto Front
Extreme Solution
(PF_{true})
Diversity metric
Spread (SD)
$SD = \frac{d_f + d_l + \sum_{i=1}^{N_{po}}|d_i - mean(d_i)|}{d_f + d_l + (N_{po} - 1)mean(d_i)}$
N_{po} number of solutions in Pfob Using MaOAHA algorithm
Euclidean distance (d_i)
$d_i = \min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| - |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$
$i, j = 1, 2, \dots, N_{po}$
Obtained Pareto Front (PF_{ob})
d_f
d_1
d_2
d_i
d_l
Extreme Solution
f_1
f_2
Diversity metric
Hyper-Volume (HV)
Reference point
$HV = \bigcup_{i=1}^{|N_{po}|} v_i$(volume covered by solutions of PF_{ob})
N_{po} is set of number of solutions in Pfob Using MaOAHA algorithm
Obtained Pareto Front (PF_{ob})
f_1

Figure 3: Mathematical and schematic view of the GD, IGD, SP, SD, and HV metrics.

Table 2: Results of GD metric of different many-objective algorithms on DTLZ benchmark problems.

Problem	M	D	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
DTLZ1	4	8	4.3141e−2 (3.79e−2) =	1.0241e−1 (1.72e−1) =	6.0327e−2 (6.21e−2) =	9.5822e−2 (1.58e−1) =	2.8213e−2 (2.75e−2)
	6	10	1.4102e−1 (1.04e−1) =	1.3789e−1 (6.92e−2) =	2.4411e−1 (1.68e−1) =	2.8099e+0 (1.11e+0) =	2.0511e−1 (1.80e−1)
	8	12	7.9300e−2 (2.89e−2) =	1.2907e−1 (1.00e−1) =	7.3947e−1 (3.67e−1) =	3.8039e+1 (1.61e+0) =	4.9424e−1 (6.86e−1)
DTLZ2	4	13	2.6390e−3 (1.40e−4) =	3.0097e−3 (7.88e−5) =	2.5579e−3 (7.05e−5) =	4.2084e−3 (5.40e−4) =	2.5067e−3 (3.67e−5)
	6	15	8.8577e−3 (8.84e−4) =	1.0132e−2 (2.60e−4) =	9.2063e−3 (2.95e−4) =	2.3512e−1 (7.55e−3) =	8.9964e−3 (3.13e−4)
	8	17	1.3659e−2 (1.35e−3) =	2.3728e−2 (2.76e−3) =	1.7627e−2 (1.72e−3) =	2.7493e−1 (1.85e−3) =	1.7117e−2 (3.05e−3)
DTLZ3	4	13	6.8035e−1 (1.37e−1) =	1.1125e+0 (9.20e−1) =	1.8445e+0 (3.55e−1) =	4.2781e+0 (3.25e+0) =	1.7259e+0 (3.56e−1)
	6	15	3.2731e+0 (1.23e+0) =	4.5406e+0 (2.98e+0) =	7.5772e+0 (7.76e−1) =	1.0848e+2 (2.39e+1) =	1.0649e+1 (4.16e+0)
	8	17	3.0342e+0 (9.81e−1) =	4.0751e+0 (3.24e+0) =	1.2700e+1 (2.41e+0) =	2.1835e+2 (6.13e+0) =	1.0051e+1 (3.55e+0)
DTLZ4	4	13	1.5725e−3 (1.37e−3)	2.5844e−3 (1.90e−4) =	2.4330e−3 (1.60e−4) =	3.6324e−3 (2.10e−3) =	2.0810e−3 (6.08e−4) =
	6	15	7.9565e−3 (9.09e−4)	8.6880e−3 (3.45e−4) =	8.9069e−3 (6.28e−4) =	1.9617e−1 (3.05e−2) =	8.6547e−3 (5.76e−4) =
	8	17	1.3162e−2 (1.38e−4) =	1.6636e−2 (5.74e−4) =	1.2116e−2 (2.58e−3) =	2.7690e−1 (1.02e−3) =	1.2259e−2 (6.27e−4)
DTLZ5	4	13	8.9559e−2 (9.36e−4) =	4.3854e−2 (4.75e−3) =	5.4318e−2 (3.14e−3) =	1.6227e−1 (9.09e−4) =	5.9236e−2 (6.01e−3)
	6	15	1.1592e−1 (1.58e−2) =	3.9078e−2 (1.53e−2) =	1.1012e−1 (1.06e−2) =	2.5706e−1 (8.60e−3) =	1.0021e−1 (1.65e−2)
	8	17	1.2461e−1 (8.29e−3) =	4.3861e−2 (4.60e−3) =	1.3625e−1 (1.38e−2) =	3.1316e−1 (3.29e−4) =	1.1735e−1 (4.32e−3)
DTLZ6	4	13	2.8055e−1 (1.26e−2) =	2.4656e−1 (6.49e−2) =	3.0765e−1 (3.91e−2) =	5.6648e−1 (8.96e−3) =	2.0475e−1 (3.41e−2)
	6	15	4.2588e−1 (8.95e−2) =	1.8576e−1 (1.86e−2) =	5.3156e−1 (1.48e−1) =	1.1302e+0 (5.78e−3) =	4.7543e−1 (5.19e−2)
	8	17	6.6510e−1 (1.40e−2) =	2.0335e−1 (6.52e−3) =	7.2458e−1 (4.65e−2) =	1.2030e+0 (2.49e−3) =	7.9399e−1 (5.57e−2)
DTLZ7	4	23	1.1250e−2 (2.77e−3) =	2.1526e−2 (2.42e−3) =	1.9765e−2 (5.71e−3) =	2.2593e−2 (2.76e−3) =	2.1492e−2 (1.41e−3)
	6	25	6.7890e−2 (2.65e−2) =	1.4387e−1 (3.38e−2) =	1.2989e−1 (3.85e−2) =	2.6378e−1 (1.13e−1) =	1.1924e−1 (2.15e−2)
	8	27	2.5280e−1 (7.25e−3) =	2.7854e−1 (1.17e−1) =	2.7792e−1 (7.81e−2) =	3.1366e+0 (1.04e−1) =	5.5390e−1 (8.23e−2)

Table 3: Results of IGD metric of different many-objective algorithms on DTLZ benchmark problems.

Problem	M	D	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
DTLZ1	4	8	5.8231e−1 (9.07e−1) =	2.7794e−1 (2.61e−1) =	4.1456e−1 (3.72e−1) =	1.4581e−1 (1.43e−1) =	1.7917e−1 (1.88e−1)
	6	10	8.2821e−1 (3.96e−1) =	4.8631e−1 (2.55e−1) =	5.9037e−1 (2.20e−1) =	7.0687e+0 (7.83e−1) =	9.4688e−1 (9.46e−1)
	8	12	6.3254e−1 (4.22e−1) =	2.8981e−1 (7.56e−2) =	9.0334e−1 (6.30e−1) =	1.5454e+2 (7.64e+1) =	8.1340e−1 (5.56e−1)
DTLZ2	4	13	1.4004e−1 (1.23e−3) =	1.4136e−1 (2.15e−4) =	1.4332e−1 (5.39e−4) =	1.4456e−1 (1.12e−3) =	1.4124e−1 (1.08e−4)
	6	15	2.8968e−1 (4.30e−3) =	2.9675e−1 (2.21e−3) =	3.0977e−1 (2.02e−2) =	1.6783e+0 (5.63e−2) =	2.8749e−1 (4.36e−3)
	8	17	4.0280e−1 (5.88e−3) =	4.0508e−1 (6.20e−3) =	5.1506e−1 (2.86e−2) =	2.4051e+0 (1.37e−2) =	4.4406e−1 (7.00e−2)
DTLZ3	4	13	7.9874e+0 (6.19e+0) =	5.6392e+0 (1.07e+0) =	8.8491e+0 (2.40e+0) =	9.8311e+0 (5.40e+0) =	1.1573e+1 (5.20e+0)
	6	15	1.6978e+1 (7.86e+0) =	1.4079e+1 (4.52e+0) =	1.9808e+1 (5.01e+0) =	3.8219e+2 (1.39e+2) =	2.9862e+1 (2.15e+1)
	8	17	1.6208e+1 (1.09e+1) =	1.0900e+1 (5.90e+0) =	3.2071e+1 (1.36e+1) =	1.2368e+3 (1.43e+2) =	2.9000e+1 (8.81e+0)
DTLZ4	4	13	2.4674e−1 (1.77e−1) =	4.4674e−1 (3.04e−1) =	2.4845e−1 (1.83e−1) =	3.4965e−1 (1.74e−1) =	5.4826e−1 (4.59e−1)
	6	15	4.3262e−1 (9.20e−4) =	3.0354e−1 (5.72e−4) =	2.9720e−1 (3.71e−3) =	9.4669e−1 (1.53e−1) =	4.0314e−1 (9.84e−2)
	8	17	4.6930e−1 (9.23e−2) =	4.1781e−1 (9.38e−4) =	5.2695e−1 (9.23e−2) =	2.4075e+0 (4.29e−2) =	5.6556e−1 (4.74e−2)
DTLZ5	4	13	6.1569e−2 (1.01e−2) =	5.7262e−2 (3.53e−3) =	5.6969e−2 (6.49e−3) =	1.3271e−1 (1.63e−2) =	5.7637e−2 (4.47e−3)
	6	15	8.5904e−2 (2.88e−2) =	1.5062e−1 (3.41e−2) =	1.0312e−1 (6.41e−3) =	1.0277e+0 (4.69e−2) =	1.5740e−1 (4.81e−2)
	8	17	8.3916e−2 (5.79e−3) =	1.5896e−1 (3.61e−2) =	2.5790e−1 (2.50e−2) =	1.8479e+0 (6.19e−1) =	2.2265e−1 (1.59e−2)
DTLZ6	4	13	3.3132e−1 (4.25e−1) =	3.5995e−1 (4.61e−1) =	5.6422e−1 (5.06e−1) =	1.1754e+0 (4.30e−1) =	6.7201e−1 (3.29e−1)
	6	15	1.6875e−1 (6.24e−2) =	2.4629e+0 (4.92e−1) =	2.5141e+0 (1.41e+0) =	9.6227e+0 (7.04e−2) =	2.2675e+0 (6.52e−1)
	8	17	4.1865e−1 (3.08e−1) =	3.7549e+0 (4.72e−1) =	3.9634e+0 (5.11e−1) =	9.8967e+0 (7.35e−2) =	4.8744e+0 (7.51e−1)
DTLZ7	4	23	2.7356e−1 (1.76e−2) =	4.1778e−1 (2.27e−1) =	2.8109e−1 (3.27e−2) =	2.0758e−1 (1.73e−2) =	2.9139e−1 (3.00e−2)
	6	25	1.0600e+0 (2.08e−1) =	6.6232e−1 (4.23e−2) =	8.0914e−1 (1.23e−1) =	1.0636e+0 (1.22e−1) =	7.2719e−1 (4.43e−2)
	8	27	2.1034e+0 (1.31e+0) =	1.4013e+0 (9.12e−2) =	2.5643e+0 (8.47e−1) =	5.0264e+0 (1.66e+0) =	4.3220e+0 (4.13e−1)

by MaOGBO, MaOPSO, RVEA, and NSGA-III that have obtained 7, 1, 1, and 1 best results, respectively. These results are particularly noticeable in specific problems such as DTLZ1, DTLZ2, DTLZ4, and DTLZ5, where MaOAHA not only achieves lower mean SD values maintaining a balanced spread of solutions shown in Fig. 4.

Table 6 presents the HV results for various algorithms on DTLZ problems, emphasizing the performance of MaOAHA. In this context, MaOAHA exhibits notable results, particularly in achieving high HV values, which indicate a better coverage of the Pareto front. MaOAHA achieves the best HV results in 9/21 test problems. This demonstrates its effectiveness in capturing a larger volume of the Pareto front, which is a key indicator of algorithmic efficiency. When compared with MaOGBO, MaOPSO, RVEA, and NSGA-III, the HV values achieved by MaOAHA are higher in a significant number of problems. In Table 6, HV value of MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms is better in 1, 2, 8, and 1 out of 21 cases and is only worse in 4.76%, 9.52%, 38.09%, and 4.76% cases.

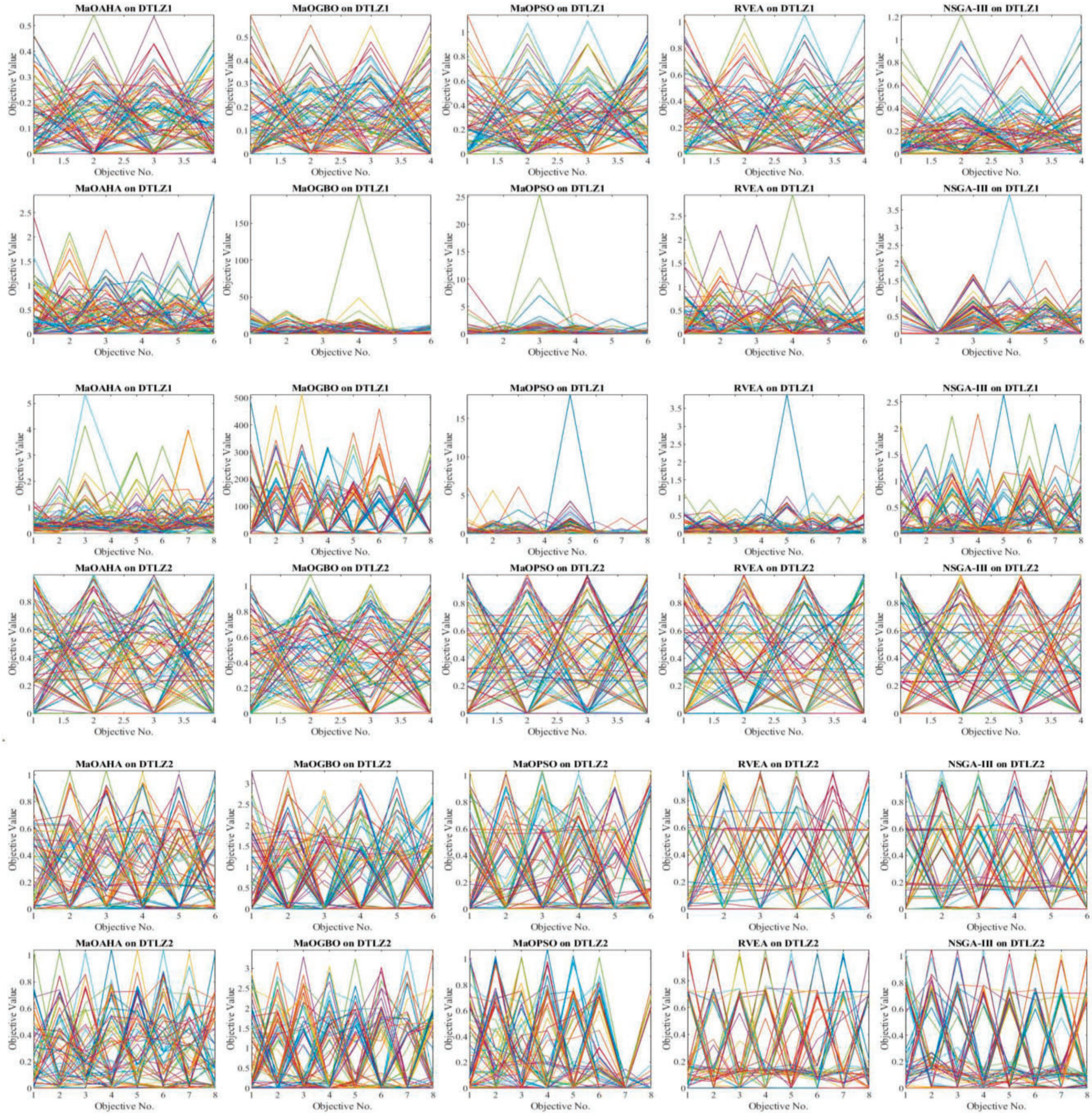

Figure 4: Best Pareto-optimal front obtained by different algorithms on DTLZ problems.

The detailed analysis of the HV results leads to the conclusion that MaOAHA outperforms its competitors in a significant proportion of the DTLZ test problems, especially in terms of achieving a larger volume of the Pareto front shown in Fig. 4.

In Table 7, the assessment of RT metrics across DTLZ problems, MaOAHA exhibits noteworthy efficiency. According to Table 7, MaOAHA significantly outperforms MaOGBO, MaOPSO, RVEA, and NSGA-III. MaOAHA running time accounts for 65% of MaOGBO running time, 75% of MaOPSO, 40% of RVEA, and 85% of NSGA-III on average. This is calculated based on the mean running times across all test problems. For problem, in DTLZ1 with 4-M and 8-D, MaOAHA running time is 1.5927 second (std 7.78e−1), which is substantially lower than MaOGBO 2.4435 second (std 2.60e−1) and RVEA 5.3515 second (std 1.76e−1). Based on these proportions and the specific RT values, MaOAHA demonstrates a considerable advantage in terms of computational efficiency across the DTLZ test suite.

From the Tables 2–7, MaOAHA emerges as a leading algorithm, achieving the most optimal IGD and HV values in many problems. However, in the context of DTLZ5 and DTLZ6, where obtaining well-distributed non-dominated solutions on the degenerated Pareto fronts is challenging, MaOAHA faces stiffer competition. As shown in Table 3, while some algorithms struggle with DTLZ6, MaOAHA manages to secure the best IGD values, demonstrating its capability in these demanding scenarios. MaOAHA stands out in terms of its performance across a variety of complex

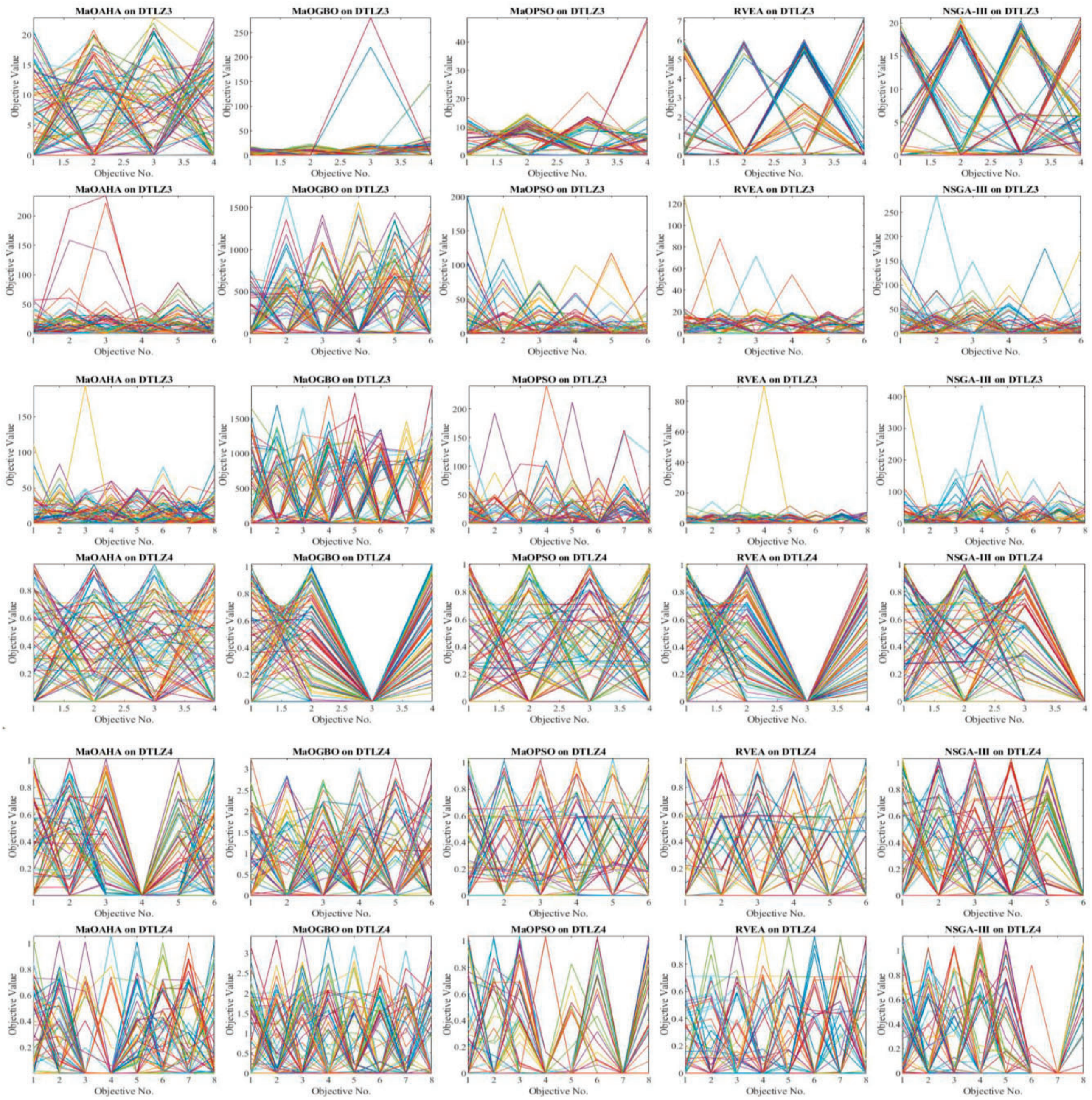

Figure 4 – *continued.*

optimization problems, showcasing particularly strong results in terms of IGD and HV metrics. Its ability to effectively handle multimodal problems and maintain population diversity shown in Fig. 4 along with its robust performance in scenarios with degenerated Pareto fronts. The "=" sign was utilized to denote instances where the differences between the compared algorithms were not statistically significant at the chosen significance level (typically $\alpha = 0.05$). This outcome suggests that, according to the WRST, there is insufficient evidence to reject the null hypothesis that the medians of the performance metrics (GD, IGD, SP, SD, HV, and RT) are equal between the MaOAHA and the comparison algorithms (MaOGBO, MaOPSO, RVEA, and NSGA-III) for the given problem instances. The Wilcoxon test may not detect minor differences as statistically significant, leading to conclusions of equality.

4.3. Experimental results on real-world many-objective optimization problems

From Table 8, it is clear that MaOAHA significantly outperforms MaOGBO, MaOPSO, RVEA, and NSGA-III in terms of the SP metric across various real-world many-objective optimization problems (RWMaOPs). MaOAHA demonstrates a notable performance. Specifically, in the context of five distinct problems: car cab design (RWMaOP1), 10-bar truss structure (RWMaOP2), water and oil repellent fabric development (RWMaOP3), ultra-wideband antenna design (RWMaOP4), and liquid-rocket single element injector design (RWMaOP5), MaOAHA exhibits advantageous results. In car cab design (RWMaOP1), MaOAHA SP value is 1.6751 (std 3.21e−1), which is lower than MaOGBO 1.8850 (std 9.23e−1) and significantly better than RVEA 3.7674 (std 1.03e+0). For the 10-bar truss structure (RWMaOP2), MaOAHA achieves an SP value of

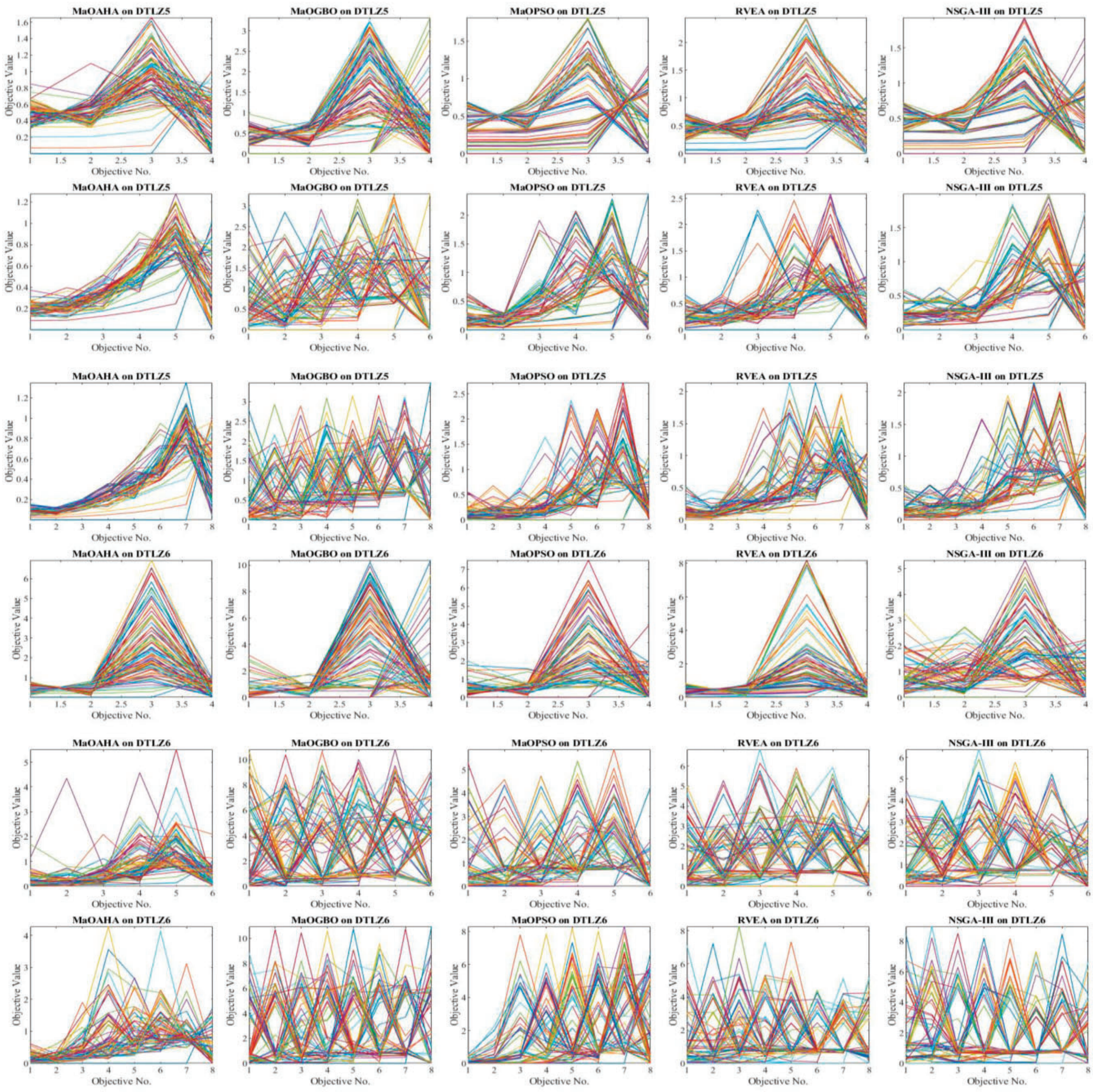

Figure 4 – *continued.*

6.6789e+2 (std 9.45e+2), outperforming MaOGBO 1.1280e+3 (std 3.24e+2) and NSGA-III 9.7329e+2 (std 3.49e+2). In the context of water and oil repellent fabric development (RWMaOP3), MaOAHA records an SP of 1.8963e+1 (std 1.99e+0), which is significantly lower than RVEA 4.9546e+1 (std 7.74e+0). For ultra-wideband antenna design (RWMaOP4), MaOAHA SP value is 4.9404e+4 (std 4.03e+3), which is more favorable than RVEA high value of 1.7624e+5 (std 2.28e+5). In liquid-rocket single element injector design (RWMaOP5), MaOAHA records an SP of 4.3216e−2 (std 5.25e−4), better than MaOPSO 9.2901e−2 (std 1.12e−2) and NSGA-III 9.3904e−2 (std 1.33e−2). These results indicate that MaOAHA not only achieves lower SP values across a range of complex RWMaOPs but also maintains a consistent performance, indicating a better distribution of solutions. In Table 8, SP value of MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms is better in 1, 0, 0, and 0 out of five cases. Therefore, from the experimental results in Table 8 shown in Fig. 5, it is reasonable to conclude that MaOAHA exhibits a higher efficiency in maintaining solution diversity across various real-world problems.

In Table 9, HV value of MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms is better in 0, 0, 2, and 0 out of five cases and is only worse in 38.09%, 9.52%, 9.52%, and 4.76% cases. Therefore, MaOAHA has a better balance between convergence and diversity for solving RWMaOPs. For problem, in car cab design (RWMaOP1), MaOAHA achieves an HV of 2.0403e−3 (std 2.55e−4), which is notably higher than RVEA 1.5212e−3 (std 5.42e−4) and NSGA-III 7.0741e−4 (std 3.13e−4), reflecting its superior capability in covering a larger volume of the Pareto front. Therefore, from the experimental results in Table 9, it is reasonable to conclude that MaOAHA exhibits higher efficiency and outperforms its

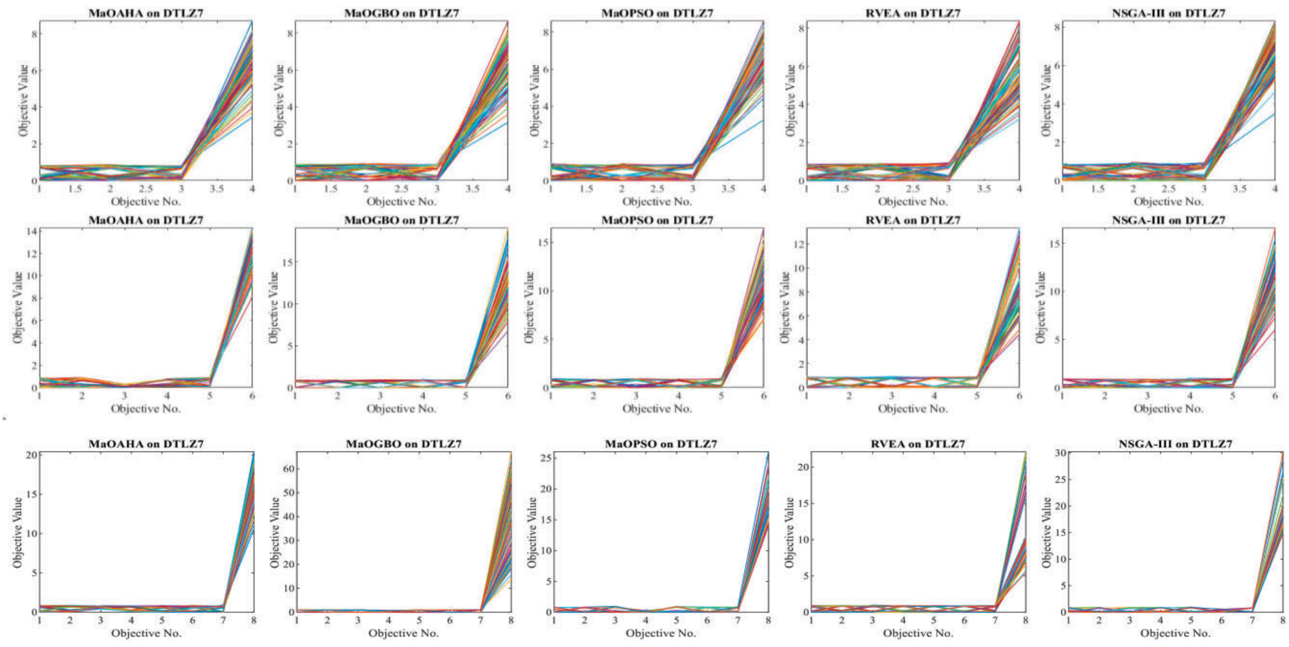

Figure 4 – *continued.*

Table 4: Results of SP metric of different many-objective algorithms on DTLZ benchmark problems.

Problem	*M*	*D*	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
DTLZ1	4	8	1.1166e−1 (1.37e−1) =	7.2468e−1 (1.22e+0) =	1.3383e−1 (1.11e−1) =	1.3307e−1 (5.63e−2) =	1.2034e−1 (8.44e−2)
	6	10	4.9375e−1 (4.73e−1) =	6.7740e+0 (7.71e+0) =	1.1031e+0 (7.96e−1) =	7.1014e−1 (7.18e−1) =	4.6314e−1 (2.19e−1)
	8	12	5.4081e−1 (4.37e−1) =	3.1790e+1 (8.76e−1) =	6.3506e+0 (4.60e+0) =	3.7408e−1 (1.95e−2) =	1.4319e+0 (1.70e+0)
DTLZ2	4	13	5.9758e−2 (2.62e−3) =	5.4859e−2 (8.87e−3) =	1.0168e−1 (1.18e−2) =	1.1888e−1 (1.66e−3) =	1.1699e−1 (1.34e−3)
	6	15	1.3751e−1 (1.46e−2) =	3.2547e−1 (1.77e−2) =	2.0867e−1 (2.96e−2) =	1.8620e−1 (4.58e−3) =	1.9356e−1 (8.41e−3)
	8	17	1.0354e−1 (1.55e−2) =	3.8470e−1 (3.88e−2) =	2.5241e−1 (5.91e−2) =	1.9006e−1 (4.91e−2) =	2.5042e−1 (1.29e−1)
DTLZ3	4	13	2.6647e+0 (2.34e+0) =	1.2466e+1 (1.43e+1) =	1.7562e+0 (3.37e−1) =	6.2885e−1 (2.53e−1) =	1.7209e+0 (1.97e−1)
	6	15	2.6637e+1 (2.10e+1) =	9.6750e+1 (2.11e+1) =	3.0320e+1 (8.37e+0) =	1.1906e+1 (3.30e+0) =	4.0628e+1 (1.81e+1)
	8	17	1.4899e+1 (1.30e+1) =	2.6025e+2 (2.71e+1) =	6.5569e+1 (9.05e+0) =	1.4735e+1 (1.43e+1) =	6.2754e+1 (3.37e+0)
DTLZ4	4	13	5.7454e−2 (2.20e−2) =	3.7052e−2 (1.76e−2) =	9.4730e−2 (1.72e−2) =	6.0610e−2 (5.93e−2) =	6.1911e−2 (5.64e−2)
	6	15	1.0241e−1 (1.11e−2) =	2.9058e−1 (6.03e−2) =	2.0329e−1 (1.43e−2) =	1.5700e−1 (7.30e−3) =	2.2104e−1 (3.43e−2)
	8	17	1.4020e−1 (2.25e−2) =	4.8216e−1 (3.77e−2) =	2.4295e−1 (4.44e−2) =	2.5972e−1 (3.15e−2) =	2.8646e−1 (4.52e−2)
DTLZ5	4	13	9.9308e−2 (5.40e−3) =	1.2057e−1 (8.40e−2) =	8.0355e−2 (1.10e−2) =	1.3433e−1 (1.70e−2) =	9.2517e−2 (2.54e−2)
	6	15	1.0410e−1 (3.84e−2) =	2.1980e−1 (2.57e−2) =	2.5370e−1 (1.89e−2) =	3.4758e−1 (5.91e−2) =	2.0069e−1 (3.54e−2)
	8	17	1.1827e−1 (1.79e−2) =	3.9019e−1 (5.04e−2) =	2.7773e−1 (1.72e−2) =	3.7731e−1 (6.37e−2) =	2.9512e−1 (3.17e−2)
DTLZ6	4	13	2.2873e−1 (5.11e−2) =	2.7361e−1 (4.29e−2) =	3.3008e−1 (3.69e−2) =	3.0777e−1 (7.18e−2) =	2.7516e−1 (3.96e−2)
	6	15	4.9612e−1 (1.63e−1) =	9.3451e−1 (1.02e−1) =	8.4606e−1 (1.98e−1) =	8.6402e−1 (1.16e−1) =	1.0644e+0 (1.28e−1)
	8	17	5.7922e−1 (3.16e−2) =	1.2517e+0 (8.98e−2) =	1.7741e+0 (4.94e−2) =	2.0602e+0 (5.63e−2) =	1.8329e+0 (3.03e−1)
DTLZ7	4	23	1.2922e−1 (3.40e−3) =	9.2560e−2 (8.99e−3) =	2.0164e−1 (2.04e−2) =	1.4200e−1 (4.34e−2) =	2.4746e−1 (2.56e−2)
	6	25	2.6480e−1 (5.83e−2) =	1.7700e−1 (4.07e−2) =	4.4542e−1 (2.77e−2) =	5.0055e−1 (1.69e−2) =	4.6310e−1 (1.40e−2)
	8	27	2.7874e−1 (2.25e−2) =	6.0138e−1 (4.86e−2) =	5.9991e−1 (9.96e−2) =	6.1164e−1 (5.40e−2) =	5.6914e−1 (1.57e−1)

competitors in most of the RWMaOPs, indicating its effectiveness in achieving a more comprehensive exploration of the solution shown in Fig. 5 space in real-world many-objective optimization scenarios.

Table 10 provides a clear illustration of MaOAHA efficiency in terms of RT across various RWMaOPs. The data reveal that MaOAHA significantly outperforms MaOGBO, MaOPSO, RVEA, and NSGA-III in the majority of test problems. These results demonstrate MaOAHA remarkable computational efficiency across a diverse set of RWMaOPs. For example, in car cab design (RWMaOP1), MaOAHA records a RT of 1.1318 second (std 1.40e−1), which is considerably lower than RVEA 17.475 second (std 1.00e+0) and NSGA-III 3.0514 second (std 2.02e−1). In Table 10, RT value of MaOGBO, MaOPSO, RVEA, and NSGA-III algorithms is better in 0, 0, 2, and 0 out of five cases. Therefore, the experimental results from Table 10 conclusively indicate that MaOAHA not only excels in terms of computational speed but also displays higher efficiency in processing compared with its competitors.

In Tables 2–10, for the WRST, MaOAHA obtains the best score of 2.01, which means that the proposed algorithm outperforms

Table 5: Results of SD metric of different many-objective algorithms on DTLZ benchmark problems.

Problem	*M*	*D*	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
DTLZ1	4	8	2.5218e−1 (3.47e−2) =	4.8419e−1 (6.45e−1) =	4.6835e−1 (1.87e−1) =	5.0601e−1 (4.26e−2) =	5.1954e−1 (2.16e−1)
	6	10	3.4230e−1 (7.84e−2) =	3.5310e−1 (2.05e−1) =	7.5121e−1 (2.06e−1) =	6.3198e−1 (2.52e−1) =	5.6025e−1 (1.94e−1)
	8	12	5.0879e−1 (9.01e−2) =	1.6766e−1 (1.92e−2) =	1.3391e+0 (4.14e−1) =	6.7758e−1 (1.10e−1) =	7.8085e−1 (3.47e−1)
DTLZ2	4	13	1.2216e−1 (1.50e−3) =	8.0113e−2 (1.35e−2) =	1.8636e−1 (1.48e−2) =	1.7756e−1 (9.66e−3) =	1.6441e−1 (4.92e−3)
	6	15	8.8157e−2 (1.15e−2) =	1.4750e−1 (5.66e−3) =	3.9402e−1 (1.67e−1) =	1.6961e−1 (1.58e−2) =	1.2353e−1 (2.19e−2)
	8	17	1.7805e−1 (5.90e−3) =	1.7650e−1 (4.84e−3) =	4.6287e−1 (1.61e−1) =	3.1819e−1 (7.79e−2) =	4.4136e−1 (4.08e−1)
DTLZ3	4	13	6.0589e−1 (1.51e−1) =	1.0317e+0 (1.50e−1) =	9.4172e−1 (3.24e−2) =	8.8248e−1 (1.11e−1) =	9.2370e−1 (3.57e−2)
	6	15	5.6952e−1 (1.85e−1) =	1.9489e−1 (2.87e−2) =	7.9654e−1 (6.51e−2) =	8.0735e−1 (5.44e−2) =	7.7012e−1 (5.67e−2)
	8	17	4.9641e−1 (8.55e−2) =	1.6618e−1 (1.55e−2) =	7.6063e−1 (3.30e−2) =	8.3442e−1 (9.39e−2) =	8.3289e−1 (1.17e−1)
DTLZ4	4	13	1.7382e−1 (5.05e−2) =	1.9367e−1 (8.54e−2) =	4.2281e−1 (4.33e−1) =	4.3222e−1 (2.83e−1) =	6.8187e−1 (4.54e−1)
	6	15	1.3151e−1 (3.51e−2) =	1.6417e−1 (1.64e−2) =	2.1557e−1 (4.33e−2) =	1.6034e−1 (1.39e−2) =	5.5005e−1 (3.77e−1)
	8	17	1.8243e−1 (1.27e−3) =	1.9699e−1 (6.33e−2) =	7.5652e−1 (4.61e−1) =	3.6009e−1 (3.64e−2) =	7.9694e−1 (1.12e−1)
DTLZ5	4	13	1.6828e−1 (6.28e−2) =	4.6710e−1 (3.03e−2) =	8.6876e−1 (4.28e−2) =	6.9266e−1 (8.36e−2) =	8.7602e−1 (3.33e−2)
	6	15	7.2053e−1 (5.59e−2) =	4.4275e−1 (8.33e−2) =	1.5959e−1 (1.67e−2) =	7.0000e−1 (1.14e−1) =	6.5989e−1 (5.41e−2)
	8	17	1.9424e−1 (3.16e−2) =	4.6903e−1 (1.34e−1) =	5.1762e−1 (3.79e−2) =	6.4786e−1 (3.62e−2) =	6.9688e−1 (4.02e−2)
DTLZ6	4	13	2.0330e−1 (2.12e−2) =	4.9026e−1 (2.42e−2) =	6.9273e−1 (9.95e−2) =	8.0133e−1 (1.35e−1) =	6.7639e−1 (1.18e−1)
	6	15	2.0057e−1 (4.32e−3) =	5.0534e−1 (8.94e−2) =	6.2320e−1 (4.52e−2) =	4.1547e−1 (1.83e−1) =	5.7594e−1 (3.27e−2)
	8	17	7.2446e−1 (4.88e−2)	4.9309e−1 (6.39e−2) =	5.7559e−1 (7.23e−2) =	5.9172e−1 (4.07e−2) =	2.1323e−1 (5.10e−3) =
DTLZ7	4	23	3.8136e−1 (1.70e−2) =	1.4120e−1 (1.20e−2) =	6.0494e−1 (6.09e−2) =	5.3604e−1 (9.13e−2) =	6.0907e−1 (2.77e−2)
	6	25	3.6265e−1 (7.26e−2) =	1.6106e−1 (6.57e−3) =	5.2698e−1 (1.70e−2) =	6.1127e−1 (3.70e−2) =	4.9448e−1 (1.64e−2)
	8	27	3.8157e−1 (6.99e−2) =	7.3267e−1 (1.14e−1) =	5.8488e−1 (1.43e−2) =	2.1558e−1 (3.65e−2) =	6.1679e−1 (4.84e−2)

Table 6: Results of HV metric of different many-objective algorithms on DTLZ benchmark problems.

Problem	*M*	*D*	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
DTLZ1	4	8	5.9872e−1 (5.19e−1) =	6.7886e−1 (3.82e−1) =	3.3823e−1 (4.24e−1) =	4.6274e−1 (4.44e−1) =	6.1368e−1 (4.54e−1)
	6	10	5.2118e−3 (8.80e−3) =	0.0000e+0 (0.00e+0) =	1.0300e−1 (1.78e−1) =	1.9553e−1 (1.71e−1) =	1.3219e−1 (1.55e−1)
	8	12	2.8580e−1 (4.95e−1) =	0.0000e+0 (0.00e+0) =	1.9762e−1 (3.42e−1) =	4.4121e−1 (2.06e−1) =	6.1748e−2 (9.89e−2)
DTLZ2	4	13	6.7173e−1 (4.74e−3) =	6.5577e−1 (4.49e−3) =	6.7500e−1 (3.13e−3) =	6.8392e−1 (3.10e−4) =	6.8299e−1 (1.24e−3)
	6	15	7.7814e−1 (6.84e−3) =	0.0000e+0 (0.00e+0) =	7.8412e−1 (6.58e−3) =	7.9538e−1 (1.48e−2)	8.0634e−1 (2.50e−3) =
	8	17	7.6469e−1 (8.13e−2) =	0.0000e+0 (0.00e+0) =	7.6720e−1 (2.95e−2) =	8.6618e−1 (1.19e−2) =	8.1379e−1 (4.54e−2)
DTLZ3	4	13	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
	6	15	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
	8	17	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
DTLZ4	4	13	6.3635e−1 (7.61e−2) =	5.7671e−1 (5.36e−2) =	6.2943e−1 (9.34e−2) =	5.2618e−1 (1.70e−1) =	4.3431e−1 (3.07e−1)
	6	15	7.5403e−1 (4.30e−3) =	1.5357e−2 (2.66e−2) =	8.0271e−1 (1.28e−2) =	8.1189e−1 (1.08e−3) =	7.3901e−1 (5.28e−2)
	8	17	8.3690e−1 (4.90e−2) =	0.0000e+0 (0.00e+0) =	7.6316e−1 (7.88e−2) =	8.8696e−1 (8.36e−4) =	7.4178e−1 (6.43e−2)
DTLZ5	4	13	1.1785e−1 (3.18e−3) =	8.1582e−2 (1.76e−2) =	1.3050e−1 (3.23e−3) =	1.1423e−1 (1.26e−3) =	1.2728e−1 (3.11e−3)
	6	15	9.3256e−2 (1.17e−2) =	0.0000e+0 (0.00e+0) =	8.4009e−2 (6.69e−3) =	4.5167e−2 (3.56e−2) =	7.2100e−2 (2.19e−2)
	8	17	9.3488e−2 (3.25e−3) =	0.0000e+0 (0.00e+0) =	4.6417e−2 (2.76e−2) =	4.4503e−2 (3.95e−2) =	4.3710e−2 (2.98e−2)
DTLZ6	4	13	7.4448e−2 (6.53e−2) =	0.0000e+0 (0.00e+0) =	3.3082e−2 (5.43e−2) =	5.5425e−2 (5.69e−2) =	3.5369e−3 (6.13e−3)
	6	15	4.9230e−2 (4.27e−2) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
	8	17	2.1143e−2 (3.66e−2) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0)
DTLZ7	4	23	2.0924e−1 (6.91e−3) =	2.1961e−1 (1.31e−2) =	2.1752e−1 (1.04e−2) =	2.2209e−1 (4.72e−3) =	2.1515e−1 (2.12e−3)
	6	25	3.8492e−2 (1.49e−2) =	8.3376e−3 (6.27e−3) =	7.6437e−2 (3.08e−2) =	1.0376e−1 (1.98e−2) =	1.0005e−1 (9.19e−3)
	8	27	5.4390e−3 (6.24e−3) =	0.0000e+0 (0.00e+0) =	6.4156e−2 (3.15e−2) =	2.5577e−2 (2.62e−2) =	1.8458e−2 (1.32e−2)

Table 7: Results of RT metric of different many-objective algorithms on DTLZ benchmark problems.

Problem	M	D	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
DTLZ1	4	8	1.5927e+0 (7.78e−1) =	2.4435e+0 (2.60e−1) =	1.7640e+0 (2.77e−1) =	5.3515e+0 (1.76e−1) =	1.1292e+0 (1.10e−1)
	6	10	1.2329e+0 (9.41e−2) =	3.3345e+0 (2.36e−1) =	1.8380e+0 (1.32e−1) =	7.7632e+0 (3.57e−1) =	1.4131e+0 (1.40e−1)
	8	12	1.2548e+0 (1.30e−2) =	6.1420e+0 (2.09e−1) =	2.2522e+0 (3.23e−1) =	1.0226e+1 (1.75e+0) =	2.0897e+0 (4.19e−1)
DTLZ2	4	13	1.4101e+0 (3.01e−2) =	5.4979e+0 (1.39e−1) =	1.3258e+0 (4.93e−2) =	1.1736e+1 (9.70e−2) =	1.0300e+0 (6.44e−2)
	6	15	1.5258e+0 (2.24e−2) =	5.9570e+0 (6.63e−2) =	1.5130e+0 (6.86e−2) =	1.2437e+1 (2.02e−1) =	1.1211e+0 (3.70e−2)
	8	17	1.5279e+0 (3.96e−2) =	7.1851e+0 (2.49e−1) =	2.4293e+0 (2.87e−1) =	1.3612e+1 (1.24e+0) =	2.0651e+0 (8.72e−1)
DTLZ3	4	13	1.2667e+0 (9.99e−2) =	2.1818e+0 (3.16e−1) =	1.4772e+0 (1.70e−1) =	4.2458e+0 (6.99e−1) =	1.0589e+0 (1.21e−1)
	6	15	1.4224e+0 (2.17e−1) =	5.8753e+0 (1.03e+0) =	2.3446e+0 (6.17e−1) =	9.5524e+0 (2.99e−1) =	1.7252e+0 (2.38e−1)
	8	17	1.4443e+0 (2.47e−1) =	9.0269e+0 (5.55e−1) =	2.8377e+0 (3.03e−1) =	1.3654e+1 (2.46e+0) =	2.1431e+0 (4.97e−1)
DTLZ4	4	13	1.5576e+0 (2.01e−1) =	5.1097e+0 (4.44e−1) =	2.2006e+0 (1.28e+0) =	1.1274e+1 (2.21e+0) =	2.4955e+0 (1.36e+0)
	6	15	1.5043e+0 (4.00e−2) =	6.1184e+0 (3.39e−1) =	1.4422e+0 (6.38e−2) =	1.3213e+1 (2.47e−1) =	2.8881e+0 (1.50e+0)
	8	17	1.7086e+0 (1.04e−1) =	8.6313e+0 (1.01e+0) =	3.8513e+0 (1.75e+0) =	1.5095e+1 (7.89e−1) =	4.0963e+0 (1.17e−1)
DTLZ5	4	13	5.8556e+0 (1.93e−1) =	1.0546e+0 (1.21e−2) =	3.2682e+0 (1.11e−1) =	1.3200e+1 (1.46e−1) =	3.1192e+0 (2.77e−1)
	6	15	6.3798e+0 (2.18e−1) =	1.0819e+0 (8.15e−2) =	3.7659e+0 (1.93e−1) =	1.2705e+1 (2.11e−1) =	3.7389e+0 (1.47e−1)
	8	17	7.4985e+0 (2.49e−1) =	1.1346e+0 (8.19e−2) =	3.9519e+0 (1.60e−1) =	1.2130e+1 (9.51e−2) =	3.5816e+0 (1.17e−1)
DTLZ6	4	13	1.1359e+0 (1.70e−2) =	4.5711e+0 (9.04e−2) =	1.9566e+0 (1.79e−1) =	9.4453e+0 (3.16e−1) =	1.1467e+0 (9.04e−2)
	6	15	1.2504e+0 (1.39e−2) =	8.4351e+0 (1.22e−1) =	2.1325e+0 (3.83e−1) =	1.2775e+1 (2.83e−1) =	1.2372e+0 (3.66e−2)
	8	17	1.3319e+0 (4.15e−2) =	8.2254e+0 (1.13e−1) =	2.5598e+0 (1.11e+0) =	1.3263e+1 (8.06e−3) =	1.9007e+0 (1.03e+0)
DTLZ7	4	23	1.0378e+1 (3.88e−1) =	5.0362e+0 (2.22e−1) =	3.0180e+0 (1.79e−1) =	1.1742e+0 (5.98e−2) =	2.7130e+0 (2.63e−1)
	6	25	1.3701e+1 (7.19e−1) =	5.2852e+0 (1.65e−1) =	3.6229e+0 (7.61e−2) =	1.4087e+0 (1.05e−1) =	3.5131e+0 (1.18e−1)
	8	27	1.2926e+0 (7.70e−2) =	5.8007e+0 (1.96e+0) =	1.9945e+0 (1.07e−1) =	6.5042e+0 (1.42e−1) =	2.0127e+0 (2.71e−1)

Table 8: Results of SP metric of different many-objective algorithms on RWMaOP problems.

Problem	M	D	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
RWMaOP1	9	7	1.6751e+0 (3.21e−1) =	1.8850e+0 (9.23e−1) =	3.1881e+0 (1.06e+0) =	3.7674e+0 (1.03e+0) =	3.2195e+0 (2.13e−1)
RWMaOP2	4	10	6.6789e+2 (9.45e+2) =	1.1280e+3 (3.24e+2) =	7.7418e+2 (2.02e+2) =	8.6737e+2 (3.01e+2) =	9.7329e+2 (3.49e+2)
RWMaOP3	7	3	1.8963e+1 (1.99e+0) =	3.0024e+1 (1.63e+0) =	3.0794e+1 (4.17e+0) =	4.9546e+1 (7.74e+0) =	3.2173e+1 (2.49e+0)
RWMaOP4	5	6	4.9404e+4 (4.03e+3) =	3.3628e+4 (3.42e+3) =	5.8139e+4 (9.91e+3) =	1.7624e+5 (2.28e+5) =	6.8760e+4 (1.46e+4)
RWMaOP5	4	4	4.3216e−2 (5.25e−4) =	8.9090e−2 (8.65e−3) =	9.2901e−2 (1.12e−2) =	1.1272e−1 (5.52e−3) =	9.3904e−2 (1.33e−2)

MaOGBO, MaOPSO, RVEA, and NSGA-III achieves 4.54, 15.66, 7.83, and 10.84. Thus, MaOAHA shows better overall performance compared with MaOGBO, MaOPSO, RVEA, and NSGA-III. In this context, MaOAHA proves to be versatile, showing the capability to handle such disconnected fronts effectively, as seen in its IGD values. Among all the benchmark cases considered, MaOAHA achieves the smallest IGD value in a significant number of cases, highlighting its superiority in diverse scenarios. This is further supported by the comparison figures, which illustrate MaOAHA ability to maintain a broad and evenly distributed population across the Pareto front. While different algorithms have their strengths in certain problem types, MaOAHA exhibits robust and versatile performance across the DTLZ suite and RWMaOPs. Its ability to handle regular, degenerate, and disconnected Pareto fronts effectively makes it a highly competitive algorithm in the field of many-objective optimization.

5. Conclusions

This study introduces a novel MaOAHA to tackle MaOPs. MaOAHA features an innovative Reference Point and Niche Technology, aiming to effectively balance convergence and diversity. Additionally, the algorithm adapts to a variety of MaOPs through a novel IFM strategy. This strategy leverages the distribution of dominant individuals in both current and historical populations to infer the distribution characteristics of true Pareto fronts across different test scenarios to boost the algorithm's exploratory capabilities. The effectiveness of MaOAHA was tested on renowned benchmark problems (DTLZ1–DTLZ7) with four, six, and eight objectives, using performance metrics like GD, IGD, SP, SD, HV, and RT. It was also compared with leading algorithms such as MaOGBO, MaOPSO, RVEA, and NSGA-III. The outcomes reveal MAOAHA's superiority in terms of GD, IGD, SP, SD, HV, and RT. The algorithm's applicability and excellence have also been confirmed in five real-world (RWMaOP1–RWMaOP5) scenarios. It also showed enhanced performance and efficiency in RT compared with other algorithms.

One limitation of the current implementation of MaOAHA is that though it is robust across the tested scenarios, it may exhibit varying efficacy in dealing with problems characterized by extreme objective space dimensions or highly irregular Pareto fronts. Furthermore, the computational efficiency of the algorithm, particularly in scenarios involving a large number of objectives or complex constraints, remains an area for optimization.

Nevertheless, the future works in this regard should try to—

(i) Explore the incorporation of diverse variation operators from the field of many-objective optimization into this ensemble framework. Perhaps this would enhance MaOAHA adaptability and performance across an even broader array of problems.

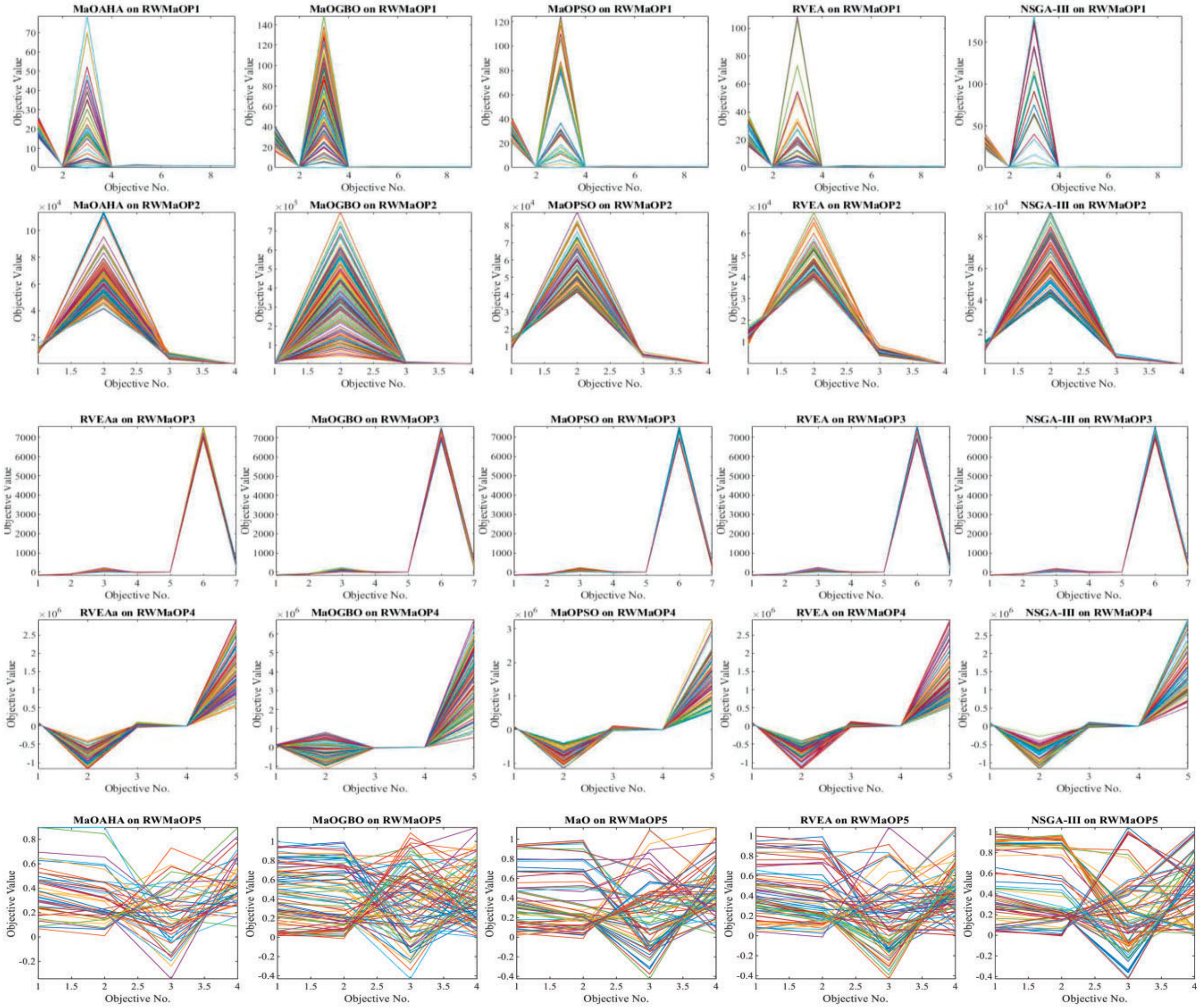

Figure 5: Best Pareto-optimal front obtained by different algorithms on RWMaOP problems.

Table 9: Results of HV metric of different many-objective algorithms on RWMaOP problems.

Problem	M	D	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
RWMaOP1	9	7	2.0403e−3 (2.55e−4)	1.3346e−3 (1.79e−4) =	2.0043e−3 (1.65e−4) =	1.5212e−3 (5.42e−4) =	7.0741e−4 (3.13e−4) =
RWMaOP2	4	10	8.0832e−2 (7.37e−4)	3.1989e−2 (2.12e−2) =	8.0314e−2 (1.05e−3) =	7.2471e−2 (3.45e−3) =	6.6823e−2 (2.60e−3) =
RWMaOP3	7	3	1.6537e−2 (3.34e−4)	1.7240e−2 (3.79e−4) =	1.6445e−2 (4.91e−4) =	1.5896e−2 (5.23e−4) =	1.7165e−2 (1.60e−4) =
RWMaOP4	5	6	5.4295e−1 (3.58e−3)	4.8797e−1 (8.98e−3) =	5.3937e−1 (6.33e−3) =	5.3392e−1 (1.04e−2) =	5.3019e−1 (3.86e−3) =
RWMaOP5	4	4	5.3524e−1 (2.21e−3)	5.3745e−1 (4.26e−3) =	5.3860e−1 (1.10e−2) =	5.4212e−1 (4.63e−3) =	5.5073e−1 (2.10e−2) =

Table 10: Results of RT metric of different many-objective algorithms on RWMaOP problems.

Problem	M	D	MaOAHA	MaOGBO	MaOPSO	RVEA	NSGA-III
RWMaOP1	9	7	1.1318e+0 (1.40e−1) =	9.2600e+0 (1.70e+0) =	3.4330e+0 (4.44e−1) =	1.7475e+1 (1.00e+0) =	3.0514e+0 (2.02e−1)
RWMaOP2	4	10	1.2529e+1 (5.06e−1) =	1.5095e+1 (1.96e−1) =	1.4252e+1 (6.75e−1) =	1.7085e+1 (3.57e+0) =	1.3780e+1 (5.47e−1)
RWMaOP3	7	3	2.0567e+1 (2.21e+0) =	1.0033e+1 (3.64e−1) =	3.2538e+0 (1.70e−1) =	1.0643e+0 (9.03e−2) =	3.5768e+0 (2.80e−1)
RWMaOP4	5	6	1.5716e+1 (8.84e−1) =	8.5823e+0 (5.28e−1) =	3.9998e+0 (2.11e−1) =	1.3350e+0 (3.04e−1) =	3.3293e+0 (1.83e−1)
RWMaOP5	4	4	8.9250e−1 (1.61e−2) =	6.0477e+0 (8.97e−2) =	3.2266e+0 (1.59e−1) =	1.4645e+1 (8.83e−1) =	3.5160e+0 (6.17e−1)

(ii) Extend MaOAHA utility by integrating sophisticated constraint-handling methods. Such advancements would enable the algorithm to tackle more intricate real-world issues that involve complex constraints.
(iii) Adapt MaOAHA for solving combinatorial optimization problems. This will test the algorithm's versatility and efficiency in a new domain.
(iv) Address the computational demands of MaOAHA, especially in large-scale optimization problems. Optimizing the algorithm's computational efficiency without compromising its ability to find high-quality solutions should be priority.

Funding

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

Conflict of interest statement

None declared.

Author Contributions

Pradeep Jangir (Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing—original draft, Writing—review & editing), Sundaram B. Pandya (Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing—original draft, Writing—review & editing), Kanak Kalita (Conceptualization, Formal analysis, Investigation, Methodology, Software, Writing—original draft, Writing—review & editing), Robert Čep (Methodology, Software, Writing—review & editing), Laith Abualigah (Methodology, Software, Writing—review & editing), Hazem Migdady (Methodology, Software, Writing—review & editing), and Mohammad Sh. Daoud (Methodology, Software, Writing—review & editing)

All authors have read and agreed to the published version of the manuscript.

Data availability

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

References

Afsar, B., Fieldsend, J. E., Guerreiro, A. P., Miettinen, K., Gonzalez, S. R., & Sato, H. (2023). Many-objective quality measures. In Brockhoff D., Emmerich M., Naujoks B., & Purshouse R. (Eds.), *Many-criteria optimization and decision analysis: State-of-the-art, present challenges, and future perspectives*(pp. 113–148). Springer International Publishing. https://doi.org/10.1007/978-3-031-25263-1_5.

Ahmad, N., Kamal, S., Raza, Z. A., & Hussain, T. (2017). Multi-objective optimization in the development of oil and water repellent cellulose fabric based on response surface methodology and the desirability function. *Materials Research Express*, **4**, 035302. https://doi.org/10.1088/2053-1591/aa5f6a.

Bradstreet, L., While, L., & Barone, L. (2008). A fast incremental hypervolume algorithm. *IEEE Transactions on Evolutionary Computation*, **12**, 714–723. https://doi.org/10.1109/TEVC.2008.919001.

Cao, B., Li, M., Liu, X., Zhao, J., Cao, W., & Lv, Z. (2021). Many-objective deployment optimization for a drone-assisted camera network. *IEEE Transactions on Network Science and Engineering*, **8**, 2756–2764. https://doi.org/10.1109/TNSE.2021.3057915.

Cao, B., Li, Z., Liu, X., Lv, Z., & He, H. (2023). Mobility-aware multiobjective task offloading for vehicular edge computing in digital twin environment. *IEEE Journal on Selected Areas in Communications*, **41**, 3046–3055. https://doi.org/10.1109/JSAC.2023.3310100.

Cao, B., Zhao, J., Gu, Y., Ling, Y., & Ma, X. (2020). Applying graph-based differential grouping for multiobjective large-scale optimization. *Swarm and Evolutionary Computation*, **53**, 100626. https://doi.org/10.1016/j.swevo.2019.100626.

Chen, H., Tian, Y., Pedrycz, W., Wu, G., Wang, R., & Wang, L. (2020). Hyperplane assisted evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Cybernetics*, **50**, 3367–3380. https://doi.org/10.1109/TCYB.2019.2899225.

Chen, Y.-S. (2017). Performance enhancement of multiband antennas through a two-stage optimization technique. *International Journal of RF and Microwave Computer-Aided Engineering*, **27**, e21064. https://doi.org/10.1002/mmce.21064.

Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **20**, 773–791. https://doi.org/10.1109/TEVC.2016.2519378.

Cheraghalipour, A., Hajiaghaei-Keshteli, M., & Paydar, M. M. (2018). Tree growth algorithm (TGA): A novel approach for solving optimization problems. *Engineering Applications of Artificial Intelligence*, **72**, 393–414. https://doi.org/10.1016/j.engappai.2018.04.021.

Choi, Y. H. (2022). Development of optimal water distribution system design and operation approach considering hydraulic and water quality criteria in many-objective optimization framework. *Journal of Computational Design and Engineering*, **9**, 507–518. https://doi.org/10.1093/jcde/qwac017.

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-36797-2.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Schoenauer M., Deb K., Rudolph G., Yao X., Lutton E., Merelo J. J., & Schwefel H.-P. (Eds.), *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI)*(849–858). Springer. https://doi.org/10.1007/3-540-45356-3_83.

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, **18**, 577–601. https://doi.org/10.1109/TEVC.2013.2281535.

Deb, K., Mohan, M., & Mishra, S. (2003). Towards a quick computation of well-spread Pareto-optimal solutions. In Fonseca C. M., Fleming P. J., Zitzler E., Thiele L., & Deb K. (Eds.), *Evolutionary multi-criterion optimization. EMO 2003. Lecture notes in computer science*(Vol. **2632**, pp. 222–236). Springer. https://doi.org/10.1007/3-540-36970-8_16.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2003). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*(pp. 825–830). IEEE. https://doi.org/10.1109/CEC.2002.1007032.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2018). The social engineering optimizer (SEO).

Engineering Applications of Artificial Intelligence, **72**, 267–293. https://doi.org/10.1016/j.engappai.2018.04.009.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2020). Red deer algorithm (RDA): A new nature-inspired meta-heuristic. *Soft Computing*, **24**, 14637–14665. https://doi.org/10.1007/s00500-020-04812-z.

Figueiredo, E. M., Ludermir, T. B., & Bastos-Filho, C. J. (2016). Many objective particle swarm optimization. *Information Sciences*, **374**, 115–134. https://doi.org/10.1016/j.ins.2016.09.026.

Goel, T., Vaidyanathan, R., Haftka, R. T., Shyy, W., Queipo, N. V., & Tucker, K. (2007). Response surface approximation of Pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, **196**, 879–893. https://doi.org/10.1016/j.cma.2006.07.010.

Goli, A., Ala, A., & Hajiaghaei-Keshteli, M. (2023). Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem. *Expert Systems with Applications*, **213**, 119077. https://doi.org/10.1016/j.eswa.2022.119077.

Guo, X. (2022). A survey of decomposition based evolutionary algorithms for many-objective optimization problems. *IEEE Access*, **10**, 72825–72838. https://doi.org/10.1109/ACCESS.2022.3188762.

He, Z., Yen, G. G., & Zhang, J. (2014). Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, **18**, 269–285. https://doi.org/10.1109/TEVC.2013.2258025.

Hu, J., Zou, Y., & Soltanov, N. (2024). A multilevel optimization approach for daily scheduling of combined heat and power units with integrated electrical and thermal storage. *Expert Systems with Applications*, **250**, 123729. https://doi.org/10.1016/j.eswa.2024.123729.

Ikeda, K., Kita, H., & Kobayashi, S. (2002). Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*. (pp. 957–962). IEEE. https://doi.org/10.1109/CEC.2001.934293.

Ishibuchi, H., Setoguchi, Y., Masuda, H., & Nojima, Y. (2017). Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *IEEE Transactions on Evolutionary Computation*, **21**, 169–190. https://doi.org/10.1109/TEVC.2016.2587749.

Ishibuchi, H., Tsukamoto, N., & Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In: *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*. (pp. 2419–2426). IEEE.

Kim, M., Hiroyasu, T., Miki, M., & Watanabe, S. (2004). SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)* (pp. 742–751). Springer. https://doi.org/10.1007/978-3-540-30217-9_75.

Li, B., Tang, K., Li, J., & Yao, X. (2016). Stochastic ranking algorithm for many-objective optimization based on multiple indicators. *IEEE Transactions on Evolutionary Computation*, **20**, 924–938. https://doi.org/10.1109/TEVC.2016.2549267.

Li, K., Deb, K., Zhang, Q., & Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, **19**, 694–716. https://doi.org/10.1109/TEVC.2014.2373386.

Liu, H.-L., Gu, F., & Zhang, Q. (2014). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, **18**, 450–455. https://doi.org/10.1109/TEVC.2013.2281533.

Liu, Q., Jin, Y., Heiderich, M., Rodemann, T., & Yu, G. (2022). An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems. *IEEE Transactions on Cybernetics*, **52**, 2698–2711. https://doi.org/10.1109/TCYB.2020.3020630.

Liu, S., Wang, H., Yao, W., & Peng, W. (2023). Surrogate-assisted environmental selection for fast hypervolume-based many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **28**, 132–146. https://doi.org/10.1109/TEVC.2023.3243632.

Liu, Y., Gong, D., Sun, X., & Zhang, Y. (2017). Many-objective evolutionary optimization based on reference points. *Applied Soft Computing*, **50**, 344–355. https://doi.org/10.1016/j.asoc.2016.11.009.

Lu, C., Liu, Q., Zhang, B., & Yin, L. (2022). A Pareto-based hybrid iterated greedy algorithm for energy-efficient scheduling of distributed hybrid flowshop. *Expert Systems with Applications*, **204**, 117555. https://doi.org/10.1016/j.eswa.2022.117555.

Luo, J., Zhuo, W., Liu, S., & Xu, B. (2024). The optimization of carbon emission prediction in low carbon energy economy under big data. *IEEE Access*, **12**, 14690–14702. https://doi.org/10.1109/ACCESS.2024.3351468.

Pamulapati, T., Mallipeddi, R., & Suganthan, P. N. (2019). ISDE+: An indicator for multi and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **23**, 346–352. https://doi.org/10.1109/TEVC.2018.2848921.

Panagant, N., Kumar, S., Tejani, G. G., Pholdee, N., & Bureerat, S. (2023). Many objective meta-heuristic methods for solving constrained truss optimisation problems: A comparative analysis. *MethodsX*, **10**, 102181. https://doi.org/10.1016/j.mex.2023.102181.

Premkumar, M., Jangir, P., Sowmya, R., & Elavarasan, R. M. (2021). Many-objective gradient-based optimizer to solve optimal power flow problems: Analysis and validations. *Engineering Applications of Artificial Intelligence*, **106**, 104479. https://doi.org/10.1016/j.engappai.2021.104479.

Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., & Wu, J. (2014). MOEA/D with adaptive weight adjustment. *Evolutionary Computation*, **22**, 231–264. https://doi.org/10.1162/EVCO_a_00109.

Qin, S., Sun, C., Liu, Q., & Jin, Y. (2023). A performance indicator-based infill criterion for expensive multi-/many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **27**, 1085–1099. https://doi.org/10.1109/TEVC.2023.3237605.

Shi, M., Lv, L., & Xu, L. (2023). A multi-fidelity surrogate model based on extreme support vector regression: Fusing different fidelity data for engineering design. *Engineering Computations*, **40**, 473–493. https://doi.org/10.1108/EC-10-2021-0583.

Shi, Y., Lan, Q., Lan, X., Wu, J., Yang, T., & Wang, B. (2023). Robust optimization design of a flying wing using adjoint and uncertainty-based aerodynamic optimization approach. *Structural and Multidisciplinary Optimization*, **66**, 110. https://doi.org/10.1007/s00158-023-03559-z.

Tanabe, R., & Ishibuchi, H. (2020). An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, **89**, 106078. https://doi.org/10.1016/j.asoc.2020.106078.

Tian, Y., Cheng, R., Zhang, X., Cheng, F., & Jin, Y. (2018). An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Transactions on Evolutionary Computation*, **22**, 609–622. https://doi.org/10.1109/TEVC.2017.2749619.

Tian, Y., Cheng, R., Zhang, X., Su, Y., & Jin, Y. (2019). A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **23**, 331–345. https://doi.org/10.1109/TEVC.2018.2866854.

Wang, Y., Zhang, Q., Wang, G.-G., & Hu, Z. (2022). An enhancing many-objective evolutionary algorithm using chaotic mapping and solution ranking mechanism for large-scale optimization. *Journal of Computational Design and Engineering*, **9**, 1974–1994. https://doi.org/10.1093/jcde/qwac090.

Wei, L.-S., & Li, E.-C. (2023). A many-objective evolutionary algorithm with population preprocessing and projection distance-assisted elimination mechanism. *Journal of Computational Design and Engineering*, **10**, 1988–2018. https://doi.org/10.1093/jcde/qwad088.

Wu, F., Chen, J., & Wang, W. (2023). A dynamic multi-objective evolutionary algorithm based on prediction. *Journal of Computational Design and Engineering*, **10**, 1–15. https://doi.org/10.1093/jcde/qwac124.

Xiao, Z., Shu, J., Jiang, H., Lui, J. C. S., Min, G., Liu, J., & Dustdar, S. (2023). Multi-objective parallel task offloading and content caching in D2D-aided MEC networks. *IEEE Transactions on Mobile Computing*, **22**, 6599–6615. https://doi.org/10.1109/TMC.2022.3199876.

Xu, X., & Li, X. (2023). Construction of building an energy saving optimization model based on genetic algorithm. *International Journal of Information Technology and Systems Approach*, **16**, 1–15. https://doi.org/10.4018/IJITSA.328758.

Yang, S., Li, M., Liu, X., & Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, **17**, 721–736. https://doi.org/10.1109/TEVC.2012.2227145.

Yin, L., Zhuang, M., Jia, J., & Wang, H. (2020). Energy saving in flow-shop scheduling management: An improved multiobjective model based on grey wolf optimization algorithm. *Mathematical Problems in Engineering*, **2020**, 9462048. https://doi.org/10.1155/2020/9462048.

Yu, F., Lu, C., Zhou, J., Yin, L., & Wang, K. (2024). A knowledge-guided bi-population evolutionary algorithm for energy-efficient scheduling of distributed flexible job shop problem. *Engineering Applications of Artificial Intelligence*, **128**, 107458. https://doi.org/10.1016/j.engappai.2023.107458.

Yuan, J., Liu, H.-L., Gu, F., Zhang, Q., & He, Z. (2021). Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions. *IEEE Transactions on Evolutionary Computation*, **25**, 75–86. https://doi.org/10.1109/TEVC.2020.2999100.

Zhang, C., Zhou, L., & Li, Y. (2024). Pareto optimal reconfiguration planning and distributed parallel motion control of mobile modular robots. *IEEE Transactions on Industrial Electronics*, **71**, 9255–9264. https://doi.org/10.1109/TIE.2023.3321997.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, **11**, 712–731. https://doi.org/10.1109/TEVC.2007.892759.

Zhao, T., Yan, Z., Zhang, B., Zhang, P., Pan, R., Yuan, T., & Chen, S. (2024). A comprehensive review of process planning and trajectory optimization in arc-based directed energy deposition. *Journal of Manufacturing Processes*, **119**, 235–254. https://doi.org/10.1016/j.jmapro.2024.03.093.

Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, **388**, 114194. https://doi.org/10.1016/j.cma.2021.114194.

Zhu, C., Wang, M., Guo, M., Deng, J., Du, Q., Wei, W., & Mohebbi, A. (2024). An innovative process design and multi-criteria study/optimization of a biomass digestion-supercritical carbon dioxide scenario toward boosting a geothermal-driven cogeneration system for power and heat. *Energy*, **292**, 130408. https://doi.org/10.1016/j.energy.2024.130408.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm For Multiobjective Optimization. In *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*. Athens, Greece: Proceedings of the EUROGEN'2001. September 19-21. https://doi.org/10.3929/ETHZ-A-004284029.

Appendix 1: Unconstrained Many-objective DTLZ Test Problems (Deb, Thiele, *et al.*, 2003)

Test instance	Characteristics
DTLZ1	Linear, Multimodal
DTLZ2	Concave
DTLZ3	Concave, Multimodal
DTLZ4	Concave, Biased
DTLZ5	Concave, Degenerate
DTLZ6	Concave, Biased, Degenerate
DTLZ7	Scaled, Multimodal, Disconnected, Mixed

Appendix 2: Real-World Many-objective Engineering Design Optimization Problems

A2.1. RWMaOP1: car cab design problem (Tanabe & Ishibuchi, 2020)

We consider the real-world many-objective car cab design optimization problem (RWMaOP1) consisting of 11 decision variables and nine objectives as follows:

$$\text{minimize}$$

$$\text{weight of the car} = f_1(x) = 1.98 + 4.9x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 0.00\,001x_6 + 2.73x_7$$

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

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

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

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

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

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

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

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

Subject to

$$g_1(x) = 1 - (1.16 - 0.3717x_2x_4 - 0.00\,931x_2x_{10} - 0.484x_3x_9 + 0.01\,343x_6x_{10}) \geq 0$$

$$g_2(x) = 0.32 - (0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.8757x_5x_{10} + 0.08\,045x_6x_9 + 0.00\,139x_8x_{11} + 0.00\,001\,575x_{10}x_{11}) \geq 0$$

$$g_3(x) = 0.32 - (0.214 + 0.00\,817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03\,099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00\,364x_5x_6 + 0.0007\,715x_5x_{10} - 0.0005\,354x_6x_{10} + 0.00\,121x_8x_{11} + 0.00\,184x_9x_{10} - 0.018x_2x_2) \geq 0$$

$$g_4(x) = 0.32 - (0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001\,232x_3x_{10} - 0.166x_7x_9 + .227x_2x_2) \geq 0$$

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

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

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

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

$$g_6(x) = 32 - (4.72 - 0.5x - 4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009\,325x_6x_{10} + 0.000\,191x_{11}x_{11}) \geq 0$$

$$g_7(x) = 4 - (10.58 - 0.674x_1x_2 - 1.95x_2x_8 + .02\,054x_3x_{10} - .0198x_4x_{10} + .028x_6x_{10}) \geq 0$$

$$g_8(x) = 9.9 - (16.45 - 0.489x_3x_7 - 0.84x_5x_6 + 0.043x_9x_{10} - 0.0556x_9x_{11} - 0.000\,786x_{11}x_{11}) \geq 0$$

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

A2.2. RWMaOP2: 10-bar truss structure problem (Panagant *et al.*, 2023)

In a real-world many-objective 10-bar truss structure optimization problem (RWMaOP2), to minimize the mass of truss, minimize compliance, maximize first natural frequency, and minimize maximum buckling factor:

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

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

$$F_3(X) = \text{inverse of first natural frequency} = 1000\,000 * \left(\frac{1}{f_1}\right)$$

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

Subject to:
Behavior constraints:

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

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

Side constraints:

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

L_i is the length of the ith compressive member. Elemental cross-sections are assumed to be countable variables as beam regular sections. It is assumed that the properties and permittable limits of all trusses are the same. Mass density (ρ), elastic modulus (E), and permittable stress ($\sigma^{\max}$) are assumed as 7850 kg/m^3, 200 GPa, and 400 MPa, respectively.

A2.3. RWMaOP3: water and oil repellent fabric development (Ahmad *et al.*, 2017)

One of the most common alterations to textiles is the repellency of water and oil, feature known as hydrophobicity effect. Consequently, hydrophobicity can be assessed through seven criteria: the water ($f_1(\mathbf{x}) = -WCA$) and oil ($f_2(\mathbf{x}) = -OCA$) droplet contact angle; the air permeability ($f_3(\mathbf{x}) = -AP$), which measures the airflow through a woven fabric as a comforting property; the crease recovery angle ($f_4(\mathbf{x}) = -CRA$), which measures the ability of textiles to recover from creasing; the stiffness ($f_5(\mathbf{x}) = Stiff$), which is the cotton fabric comfort property; the tear strength ($f_6(\mathbf{x}) = -Tear$) of the finished fabric, which depends on the chemical finishing treatment applied to the fabric; and the tensile strength ($f_7(\mathbf{x}) = -Tensile$). The real-world many-objective water and oil repellent fabric development optimization problem (RWMaOP3) functions as follows:

$$
\begin{aligned}
&\text{minimize}\\
f_1(\mathbf{x}) &= -\mathrm{WCA} = -\left(-1331.04 + 1.99 \times O{-}CPC + 0.33 \times K{-}FEL + 17.12 \times C{-}Temp - 0.02 \times O{-}CPC^2 - 0.05 \times C{-}Temp^2 \pm 15.33\right).\\
f_2(\mathbf{x}) &= -OCA = -\left(-4231.14 + 4.27 \times O{-}CPC + 1.50 \times K{-}FEL + 52.30 \times C{-}Temp - 0.04 \times O{-}CPC \times K{-}FEL - 0.04 \times O{-}CPC^2\right.\\
&\quad \left. - 0.16 \times C{-}Temp^2 \pm 29.33\right).\\
f_3(\mathbf{x}) &= -AP = -\left(1766.80 - 32.32 \times O{-}CPC - 24.56 \times K{-}FEL - 10.48 \times C{-}Temp + 0.24 \times O{-}CPC \times C{-}Temp\right.\\
&\quad \left. + 0.19 \times K{-}FEL \times C{-}Temp - 0.06 \times O - CPC^2 - 0.10 \times K{-}FEL^2 \pm 413.33\right).\\
f_4(\mathbf{x}) &= -CRA = -\left(-2342.13 - 1.556 \times O{-}CPC + 0.77 \times K{-}FEL + 31.14 \times C{-}Temp + 0.03 \times O{-}CPC^2 - 0.10 \times C{-}Temp^2 \pm 73.33\right).\\
f_5(\mathbf{x}) &= Stiff = 9.34 + 0.02 \times O{-}CPC - 0.03 \times K{-}FEL - 0.03 \times C{-}Temp - 0.001 \times O{-}CPC \times K{-}FEL + 0.0009 \times K{-}FEL^2 \pm 0.22.\\
f_6(\mathbf{x}) &= -Tear = -\left(1954.71 + 14.246 \times O{-}CPC + 5.00 \times K{-}FEL - 4.30 \times C{-}Temp - 0.22 \times O{-}CPC^2 - 0.33 \times K{-}FEL^2 \pm 8413.33\right).\\
f_7(\mathbf{x}) &= -Tensile = -\left(828.16 + 3.55 \times O{-}CPC + 73.65 \times K{-}FEL + 10.80 \times C{-}Temp - 0.56 \times K{-}FEL \times C{-}Temp\right.\\
&\quad \left. + 0.20 \times K{-}FEL^2 \pm 2814.83\right)
\end{aligned}
$$

and $\mathbf{x} = (O - CPC, K - FEL, C - Temp)^T$, such that $10 \leq O - CPC \leq 50$, is the concentration of water and oil repellent finish in g/L, $10 \leq K - FEL \leq 50$, is the concentration of the crosslinking agent in g/L, and $150 \leq \text{C-Temp} \leq 170$, is the curing temperature in °C.

A2.4. RWMaOP4: ultra-wideband antenna design (Chen, 2017)

In order to design this antenna the objective functions to consider are: the voltage standing wave ratio (VSWR) over the passband ($f_1(\mathbf{x}) = VPVP$), the VSWR over the WiMAX band ($f_2(\mathbf{x}) = -VWi$), the VSWR over the WLAN band ($f_3(\mathbf{x}) = -VWL$), the E- and H-planes fidelity factor ($f_4(\mathbf{x}) = -FF$), and the maximum gain over the passband ($f_5(\mathbf{x}) = PG$). Hence, the real-world many-objective ultra-wideband antenna design optimization problem (RWMaOP4) is stated as

$$
\begin{aligned}
&\text{minimize}\\
f_1(\mathbf{x}) &= VP = 502.94 - 27.18 \times ((w_1 - 20.0)/0.5) + 43.08 \times ((l_1 - 20.0)/2.5) + 47.75 \times (a_1 - 6.0)\\
&\quad + 32.25 \times ((b1 - 5.5)/0.5) + 31.67 \times (a_2 - 11.0) - 36.19 \times ((w_1 - 20.0)/0.5) \times ((w_2 - 2.5)/0.5)\\
&\quad - 39.44 \times ((w_1 - 20.0)/0.5) \times (a_1 - 6.0) + 57.45 \times (a_1 - 6.0) \times ((b_1 - 5.5)/0.5).\\
f_2(\mathbf{x}) &= -VWi = -(130.53 + 45.97 \times ((l_1 - 20.0)/2.5) - 52.93 \times ((w_1 - 20.0)/0.5) - 78.93 \times (a_1\\
&\quad - 6.0) + 79.22 \times (a_2 - 11.0) + 47.23 \times ((w_1 - 20.0)/0.5) \times (a_1 - 6.0) - 40.61 \times ((w_1 - 20.0)/0.5)\\
&\quad \times (a_2 - 11.0) - 50.62 \times (a_1 - 6.0) \times (a_2 - 11.0)).\\
f_3(\mathbf{x}) &= -VWL = -(203.16 - 42.75 \times ((w_1 - 20.0)/0.5) + 56.67 \times (a_1 - 6.0) + 19.88\\
&\quad \times ((b_1 - 5.5)/0.5) - 12.89 \times (a_2 - 11.0) - 35.09 \times (a_1 - 6.0) \times ((b_1 - 5.5)/0.5) - 22.91\\
&\quad \times ((b_1 - 5.5)/0.5) \times (a_2 - 11.0)).\\
f_4(\mathbf{x}) &= -FF = -(0.76 - 0.06 \times ((l_1 - 20.0)/2.5) + 0.03 \times ((l_2 - 2.5)/0.5) + 0.02 \times (a_2 - 11.0)\\
&\quad - 0.02 \times ((b_2 - 6.5)/0.5) - 0.03 \times ((d_2 - 12.0)/0.5) + 0.03 \times ((l_1 - 20.0)/2.5) \times ((w_1 - 20.0)/0.5)\\
&\quad - 0.02 \times ((l_1 - 20.0)/2.5) \times ((l_2 - 2.5)/0.5) + 0.02 \times ((l_1 - 20.0)/2.5) \times ((b_2 - 6.5)/0.5)).\\
f_5(\mathbf{x}) &= PG = 1.08 - 0.12 \times ((l_1 - 20.0)/2.5) - 0.26 \times ((w_1 - 20.0)/0.5) - 0.05 \times (a_2 - 11.0) - 0.12\\
&\quad \times ((b_2 - 6.5)/0.5) + 0.08 \times (a_1 - 6.0) \times ((b_2 - 6.5)/0.5) + 0.07 \times (a_2 - 6.0) \times ((b_2 - 5.5)/0.5)
\end{aligned}
$$

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

A2.5. RWMaOP5: liquid-rocket single element injector design (Goel *et al.*, 2007)

RWMaOP5 is a four-objective function optimization problem that deals with proper injector design. Therefore, for a desirable injector design, the maximum temperature of the injector surface ($f_1(\mathbf{x}) = TF_{\max}$), the temperature at three inches from the injector surface ($f_2(\mathbf{x}) = TW_4$), the maximum temperature at the tip of the injector post ($f_3(\mathbf{x}) = TT_{\max}$), and the objectives to be considered are: the distance from the inlet combustion ($f_4(\mathbf{x}) = X_{cc}$). Thus, the real-world many-objective liquid-rocket single element injector design optimization problem (RWMaOP5) can be written as

minimize

$$\begin{aligned}
f_1(\mathbf{x}) &= TF_{\max} = 0.692 + 0.477 \times \alpha - 0.687 \times \Delta HA - 0.080 \times \Delta OA - 0.0650 \times OPTT - 0.167 \times \alpha^2 \\
&\quad - 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 \\
&\quad - 0.0521 \times OPTT \times \alpha + 0.00156 \times OPTT \times \Delta HA + 0.00\,198 \times OPTT \times \Delta OA + 0.0184 \times OPTT^2. \\
f_2(\mathbf{x}) &= TW_4 = 0.758 + 0.358 \times \alpha - 0.807 \times \Delta HA + 0.0925 \times \Delta OA - 0.0468 \times OPTT - 0.172 \times \alpha^2 \\
&\quad + 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 \\
&\quad - 0.0694 \times OPTT \times \alpha - 0.00\,503 \times OPTT \times \Delta HA + 0.0151 \times OPTT \times \Delta OA + 0.0173 \times OPTT^2. \\
f_3(\mathbf{x}) &= TT_{\max} = 0.370 - 0.205 \times \alpha + 0.0307 \times \Delta HA + 0.108 \times \Delta OA + 1.019 \times OPTT - 0.135 \times \alpha^2 \\
&\quad + 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 \\
&\quad + 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 \\
&\quad \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 \\
&\quad + 0.281 \times \Delta HA \times \alpha \times \Delta OA. \\
f_4(\mathbf{x}) &= X_{cc} = 0.153 - 0.322 \times \alpha + 0.396 \times \Delta HA + 0.424 \times \Delta OA + 0.0226 \times OPTT + 0.175 \times \alpha^2 \\
&\quad + 0.0185 \times \Delta HA \times \alpha - 0.0701 \times \Delta HA^2 - 0.251 \times \Delta OA \times \alpha + 0.179 \times \Delta OA \times \Delta HA + 0.0150 \times \Delta OA^2 \\
&\quad + 0.0134 \times OPTT \times \alpha + 0.0296 \times OPTT \times \Delta HA + 0.0752 \times OPTT \times \Delta OA + 0.0192 \times OPTT^2
\end{aligned}$$

and $\mathbf{x} = (\alpha, \Delta HA, \Delta OA, OPTT)^T$.

Received: December 29, 2023. **Revised:** June 10, 2024. **Accepted:** June 10, 2024