Results in Engineering 25 (2025) 104372

Contents lists available at ScienceDirect

Results in Engineering

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

MaOSSA: A new high-efficiency many-objective salp swarm algorithm with information feedback mechanism for industrial engineering problems

Mohammad Aljaidi [a,*], Janjhyam Venkata Naga Ramesh [b,c], Ajmeera Kiran [d], Pradeep Jangir [e,f,g], Arpita [h], Sundaram B. Pandya [i], Wulfran Fendzi Mbasso [j], Laith Abualigah [k], Ali Fayez Alkoradees [l,*], Mohammad Khishe [m,n,*]

[a] *Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan*
[b] *Department of CSE, Graphic Era Hill University, Dehradun 248002, India*
[c] *Department of CSE, Graphic Era Deemed To Be University, Dehradun 248002, Uttarakhand, India*
[d] *Department of Computer Science and Engineering, MLR Institute of Technology, Dundigal, Hyderabad, Telangana 500043, India*
[e] *University Centre for Research and Development, Chandigarh University, Gharuan, Mohali 140413, India*
[f] *Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan*
[g] *Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India*
[h] *Department of Biosciences, Saveetha School of Engineering. Saveetha Institute of Medical and Technical Sciences, Chennai 602 105, India*
[i] *Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392 001, India*
[j] *Technology and Applied Sciences Laboratory, U.I.T of Douala, P.O Box: 8689- Douala, University of Douala, Cameroon*
[k] *Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan*
[l] *Unit of Scientific Research, Applied College, Qassim University, Saudi Arabia*
[m] *Department of Electrical Engineering, Imam Khomeini Naval Science University of Nowshahr, Nowshahr, Iran*
[n] *Jadara University Research Center, Jadara University, Irbid, Jordan*

ARTICLE INFO

Keywords:
Multi-objective optimization
Many-objective optimization
Many-objective salp swarm algorithm
Reference point
Engineering design problems

ABSTRACT

The pursuit of convergence in multi-objective optimization usually results in population clustering that produces suboptimal outcomes for both convergence and diversity performance. This paper introduces MaOSSA as a new Many-Objective Salp Swarm Algorithm which combines reference point strategies with niche preservation and Information Feedback Mechanism (IFM). The strategy enables control of convergence and diversity while simultaneously adapting to alterations in the Pareto front. The algorithm achieves personal diversity through its edge individual preservation strategy and density estimation method which maintains uniform population diversity. The evaluation of MaOSSA included DTLZ1-DTLZ7 benchmark problems and five real-world engineering design problems (RWMaOP1–RWMaOP5) that contained 5 to 15 objectives. The performance evaluation between MaOSCA, MaOPSO, NSGA-III, and MaOMFO algorithms showed that MaOSSA delivered superior outcomes regarding Generational Distance (GD), Inverted Generational Distance (IGD), Spacing (SP), Spread (SD), Hypervolume (HV), and Runtime (RT). The experimental outcomes show MaOSSA delivers superior performance than current methods by achieving optimal convergence-diversity balance which establishes it as an efficient solution for many-objective optimization tasks.

1. Introduction

In practical scenarios, numerous challenges involve optimizing multiple conflicting objectives simultaneously. These are known as multi-objective optimization problems (MOPs), and when they involve more than three objectives, they are termed as Many Objective Optimization Problems (MaOPs). As opposed to most numerical optimization problems that solve for a single criterion, MaOPs seek the optimization of each objective simultaneously:

* Corresponding authors.
E-mail addresses: mjaidi@zu.edu.jo (M. Aljaidi), jvnramesh@gmail.com (J.V.N. Ramesh), kiranphd.jntuh@gmail.com (A. Kiran), pkjmtech@gmail.com (P. Jangir), apyjangid@gmail.com (Arpita), sundarampandya@gmail.com (S.B. Pandya), fendzi.wulfran@yahoo.fr (W. Fendzi Mbasso), aligah.2020@gmail.com (L. Abualigah), alifk@qu.edu.sa (A.F. Alkoradees), m_khishe@alumni.iust.ac.ir (M. Khishe).

https://doi.org/10.1016/j.rineng.2025.104372
Received 30 September 2024; Received in revised form 2 February 2025; Accepted 13 February 2025
Available online 24 February 2025

$$\min F(x) = (f_1(x), f_2(x), \ldots, f_M(x))^T \quad \text{subject to } x \in \Omega \tag{1}$$

Here, M represents the total objectives, and the decision space Ω maps into the objective space R^M via the function $F(x)$. In MOPs, the goal is to identify a collection of best solutions, termed Pareto optimal solutions. The entire set of these optimal solutions forms the Pareto Set (PS), and their corresponding objective vectors compose the Pareto Front (PF), as illustrated in Fig. 1.

A multi-objective optimization challenge requires optimization of multiple competing objectives which should be accomplished at the same time. The problems with multiple objectives usually contain a reasonable number of objectives (two to three) which enables visualization of solutions and their trade-offs on the Pareto front. Multi-objective optimization finds its solutions through the identification of Pareto optimal sets which demonstrate objective balance. Many-objective optimization deals exclusively with optimization problems which contain four or more objectives. The elevation in objective numbers increases difficulties over standard multi-objective optimization methods. MaOPs are described as the optimization tasks involving four or more objectives to be optimized simultaneously, as mentioned in Wang et al. [1]. These problems are common in various real-life domains such as software development, manufacturing, logistics and many others. In the last decade, considerable attention has been paid to the use of evolutionary algorithms in solving MaOPs which results in the emergence of various MaOEAs.

Effectiveness in these algorithms is gauged by their ability to rapidly position solutions along the Pareto front and distribute them across it. Meanwhile, `efficiency' refers to the algorithm operational speed. Achieving both high effectiveness and efficiency concurrently is challenging due to the complex and unpredictable nature of the Pareto front in many-objective contexts. Evolutionary Multi-Objective (EMO) algorithms such as NSGA-II [2] and SPEA2 [3] might struggle with scalability as the number of objectives increases. Moreover, recent findings [4] indicate that even newer methods, including those based on decomposition [5,6] and indicator approaches [7,8], encounter difficulties in identifying solutions when the objective count is relatively low. Maintaining a balanced array of solutions is crucial in many-objective optimization. Attempts to enhance convergence may reduce diversity. Innovations such as ε-dominance [9] and fuzzy Pareto dominance [10] aim to improve convergence but can result in a clustered distribution of population along certain regions of the Pareto front [11]. While decomposition-based algorithms typically excel in convergence [12], they may struggle to evenly spread solutions across an irregularly shaped Pareto front [13]. Moreover, indicator-based algorithms may have a bias towards one part of the Pareto front more than the other. Conversely, indicator-based algorithms might exhibit biases toward specific segments of the Pareto front. For instance, IBEA [7] may favor extreme solutions, whereas SMS-EMOA [8] tends to prefer [14] knee solutions [15]. On the other hand, some of the modern multiple-objective algorithms may disregard some areas of the Pareto front, for instance, SPEA2 using shift-based density estimation (SPEA2 + SDE) [16], which has been reported to fail in preserving the boundary solutions sometimes [17].

Additionally, some EMO algorithms experience reduced performance when managing a larger number of objectives. For example, computational demands for algorithms like the hypervolume-based SMS-EMOA [8] rise with the increasing dimensions of the objective space. Similarly, SPEA2 + SDE [16], despite its proven effectiveness, can face efficiency challenges [18].

In the field of multi-/many-objective evolutionary algorithms, the problem of setting of specific parameters for every problem type is often observed. For example, the algorithms, which employ modified Pareto dominance relations, including the ε-domination based multiobjective evolutionary algorithm (ε-MOEA [19]), involves fine-tuning of the Pareto dominance relaxation degree [20], particularly when handling a large number of objectives [21]. In region-based MaOEAs, choosing appropriate parameters is essential for determining the size of the evaluation region. This involves decisions on grid division in grid-based evolutionary algorithms (GrEA) [22] and neighbor count in the knee point driven evolutionary algorithm (KnEA) [23]. However, the range of such parameters, which is optimal for the majority of decisions, tends to shrink as the number of objectives grows, thus making the choice of these parameters or their values rather challenging or even impossible in the high-dimensional space of objectives [24]. In general, these MaOEAs can be categorized into five categories.

The initial focus concerns revising the traditional Pareto dominance relationship utilized in many-objective optimization. Within this

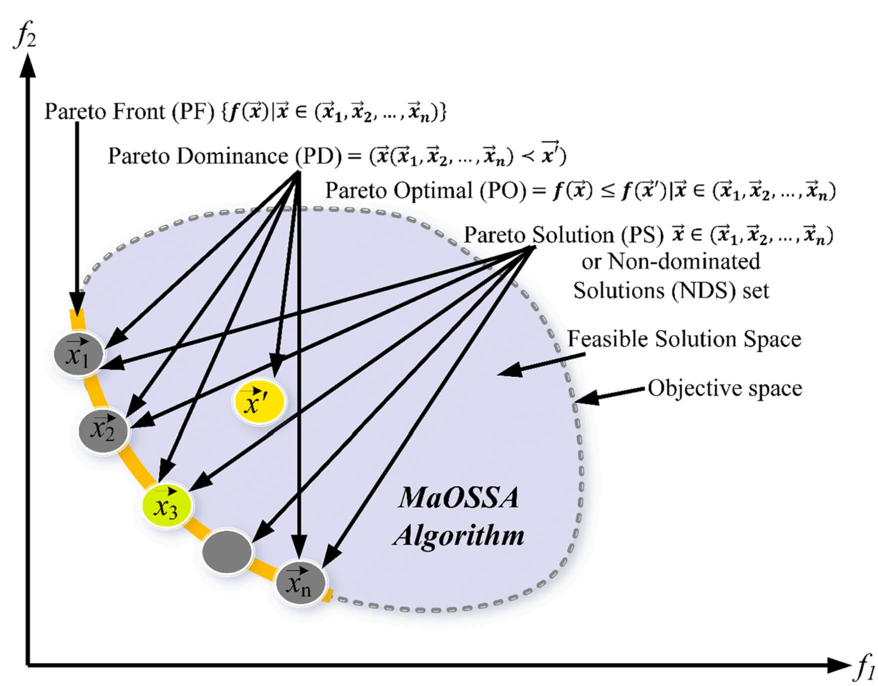

Fig. 1. Many-objective all definitions in search space of MaOP.

context, certain algorithms adopt a more lenient form of dominance, including ε-dominance [9] and an enhanced dominance model [25]. Additionally, researchers have proposed innovative dominance frameworks such as fuzzy Pareto dominance [10], grid-based dominance [22], and angular dominance [26]. Generally, these adjusted dominance relations are designed to better manage the dominance of solutions within more complex, multi-dimensional spaces, thereby increasing the pressure to drive solutions towards the Pareto front.

The second emphasis is on refining the density estimation methods traditionally used in Pareto-based strategies. The rationale is that maintaining a diverse array of non-dominated solutions might impede the convergence to the Pareto front in scenarios with numerous dimensions [24]. While some strategies in this area de-emphasize the preservation of diversity [27,28], others integrate convergence metrics into their density estimation techniques. For example, SPEA2 + SDE [16] introduces a shift-based density estimation (SDE) method aimed at penalizing solutions where the density measure is excessively high, which typically indicates poor convergence towards the desired front.

The third category in the realm of MaOEAs is about the decomposition-based methods. These algorithms address a MaOP by decomposing of it into single-objective ones [5] or less complex multi-objective ones [29] and solving them in parallel. Another advantage of these algorithms is that the solution comparison uses scalar values connected with the weight vector, so the number of objectives does not have as significant an impact as in other algorithms. After the development of the MOEA/D [5], this method has become popular. Some of the MOEA/D algorithms include Many-Objective Particle Swarm Optimizer (MaOPSO) [30], Many-Objective Sine Cosine Algorithm (MaOSCA) [31], Non-Dominated Sorting Genetic Algorithm-III (NSGA-III) [6], and Many-Objective Moth Flame Optimization (MaOMFO) [32] is also a reference-point-based algorithm. The reference vector algorithm (RVEA) [33], the MOEA/DD that utilizes both dominance and decomposition [34], and the DMEA-WUA which uses weights that are adaptively updated [35].

The fourth type includes the MaOEAs based on indicators. These algorithms employ certain performance indices as a filter for selection in order to direct the development of population in the direction of Pareto optimum. Some prefer to use just one, for instance, the IBEA using the $I_{\epsilon+}$ indicator [7], the HypE that utilizes the hypervolume (HV) indicator [36], and the MaOEA/IGD that rely on the inverted generational distance (IGD) indicator [37]. There are others such as the stochastic ranking-based multi-indicator algorithm (SRA) which works to balance the indicators and comprises of $I_{\epsilon+}$ and I_{SDE} indicators.

The fifth category includes the aggregation-based approaches. They reduce the objectives of solutions to one or several criteria, thus aiding in solution comparison. Some algorithms in this category introduce new ways of selection, evaluating solutions in terms of convergence and diversity as in the 1by1EA [38] and MaOEA-CSS [39]. Some of these methods include incorporating preferences in the fitness assignment where KnEA emphasizes on knee points [23] while PICEA-g emphasizes on specific goals or target vectors [40]. However, current development of algorithms still poses a problem in formulating algorithms that perform well in many MaOPs both in terms of effectiveness and efficiency. For example, the algorithm based on the aggregation can have problems with the balance between convergence and the variety. Comparable issues arise with the algorithms that alter Pareto dominance or diversity preservation. Although there is an extensive variety of decomposition-based methods, they can fail to preserve diversity in cases where the Pareto front has an irregular shape, and the computation time of hypervolume-based algorithms increases with the number of objectives.

Real-world complex problems throughout engineering and data analysis as well as bioinformatics fields now heavily rely on optimization algorithms for their solution. Metaheuristic algorithm progress has resulted in multiple advanced techniques working as improvements for speed of convergence and solution quality and robustness. The Improved Opposition-Based Learning Firefly Algorithm with Dragonfly Algorithm represents a new approach to solve continuous optimization problems [41]. The Improved Heterogeneous Comprehensive Learning Symbiotic Organism Search proves its ability to adapt to various optimization situations in diverse scenarios [42]. Email spam detection along with other high-dimensional optimization problems benefits from multi-agent systems techniques which demonstrate excellent capability in such applications [43]. The Improved African Vultures Optimization Algorithm [44] and Puma Optimizer [45] and Modified Farmland Fertility Algorithm [46] yield successful applications for specific fields including image segmentation and constrained engineering problems. The Liver Cancer Algorithm [47] represents one of multiple novel bio-inspired optimizers that also includes FATA based on geophysical principles [48] which expands the optimization technique landscape. The Polar Lights Optimizer [49] and RIME [50] and IFA-EO [51] maintain front positions in terms of both technical performance and application flexibility. The continuous advancement of optimization methodologies stems from adaptive data structure implementation [52] combined with multi-objective optimization frameworks such as MORIME [53] and MOFDA [54] which reflect the changing nature of research in this field.

The Salp Swarm Algorithm (SSA) functions as an ideal choice for many-objective optimization problems (MaOPs) because of its distinctive features. The ability of SSA to effectively manage exploration-exploitation needs remains vital because it addresses the complex high-dimensional MaOP environments. Population diversity and efficient convergence belong to SSA through its chaining mechanism which sets it apart from established algorithms like PSO, GA, DE and ACO. The adoption of SSA as a solution method happens because it shows proven capability to generate multiple high-quality solutions throughout different problem domains. The procedures operate by leaders indicating promising solution spots for followers to focus their local optimization efforts. SSA employs chaining behavior to distribute its search process which prevents premature convergence in high-dimensional spaces unlike PSO. The effectiveness of Differential Evolution (DE) and ACO for dynamic and discrete problems does not necessarily extend to maintaining diversity when dealing with irregular Pareto fronts. SSA uses adjustable control parameters that establish an exploration-exploitation balance to create a solid system for handling these challenges. The Information Feedback Mechanism (IFM) of SSA provides dynamic front adaptability which standardizes poorly in algorithms that lack this feature such as DE or ACO. The crossover and mutation operations enable GA to adapt well but its computational requirements grow when dealing with larger objective sets and solution areas. The position update technique within SSA maintains basic complexity but produces the same level of diversity with reduced computational expenses for convergence results. SSA achieves additional justification through its implementation of niche preservation and reference-point-based strategies within the Many-Objective Salp Swarm Algorithm (MaOSSA). The improvements in MaOSSA enable SSA to perform exploratory searches that match the requirements of MaOPs by distributing solutions uniformly across the Pareto front and adjusting to problem complexities. The benchmark tests against PSO, DE and NSGA-III algorithms demonstrate that SSA achieves superior performance regarding convergence speed and diversity maintenance and computational efficiency thus making it appropriate for the optimization challenges studied.

The evolution of many-objective optimization algorithms came about to handle growing problems with multiple conflicting objectives. Traditional algorithms provided substantial help to the field although they experience shortcomings in making calculations stretch across various scales while also struggling to keep solutions diverse and reaching proper convergence and showing flexibility. The Many-Objective Salp Swarm Algorithm (MaOSSA) functions to address various difficulties by performing effectively throughout different objective spaces. NSGA-II algorithms dominate the market for problems with 2–3 objectives because their fast non-dominated sorting and crowding distance mechanisms maintain diversity and selection pressure effectively. The selection mechanisms in NSGA-II become ineffective when handling problems with more than three objectives because the high-dimensional spaces reduce their effectiveness. MaOSSA solves the scalability issue by managing problems with 5–15 objectives through advanced diversity preservation methods which maintain performance while preserving computational speed. The decomposition-based algorithms such as MOEA/D have gained popularity in many-objective problems (4 or more objectives) because they simplify complex problems into easier subproblems. The advantage of MOEA/D stems from its decomposition abilities but it struggles to preserve diversity when dealing with irregular Pareto fronts that produce uneven clustering. MaOSSA addresses irregular Pareto front challenges by combining niche preservation methods with reference point strategies to achieve uniform solution distribution in regular and irregular Pareto fronts. The combination of SPEA2 with Shift-based Density Estimation (SDE) enables better diversity preservation since it focuses on density estimation methods. The approach struggles to preserve boundary solutions which leads to missing points at the outer edges of the Pareto front. MaOSSA solves this problem through its edge preservation mechanism that successfully retrieves boundary solutions to deliver full Pareto front coverage. HypE demonstrates effective convergence-diversity balance through its hypervolume contribution mechanism which works as a hypervolume-based algorithm. HypE becomes computationally impractical for large-scale optimization problems due to its increased computational costs when the number of objectives rises. MaOSSA decreases computational runtime while using efficient information feedback controls and optimized selection methods which keep performance high across multiple dimensions. RVEA uses reference vectors to guide its evolutionary algorithm search process for enhancing convergence performance. RVEA shows restricted capabilities when dealing with dynamic Pareto fronts that experience objective or constraint variations during runtime. The adaptive information feedback mechanism in MaOSSA allows the algorithm to detect dynamic environment changes so it can maintain solution quality throughout time. NSGA-III extends NSGA-II by adding reference point-based selection to enhance performance for optimizing multiple objectives. The approach succeeds at enhancing Pareto front diversity but fails to maintain performance when dealing with irregular fronts which results in unbalanced solution distribution. The targeted preservation method in MaOSSA concentrates on irregular regions of the complex Pareto front landscape to achieve better diversity and stability. The PSO-based algorithm MaOPSO has been modified for many-objective problems by providing a flexible and intuitive solution method. The Swarm dynamics become ineffective for high-dimensional spaces resulting in poor convergence of these algorithms. The feedback mechanism of MaOSSA achieves better convergence by adapting search behavior with historical performance data to improve the algorithm's capability for approaching the true Pareto front. Several MFO-based algorithms including MaOMFO prove suitable for complex many-objective problems because they maintain an effective exploration-exploitation behavior pattern. The algorithm shows restricted diversity maintenance capabilities which produces clustered solutions while reducing the coverage of the Pareto front. The diversity enhancement of MaOSSA occurs through reference point strategies paired with niche preservation methods to achieve a spread of solutions across the complete objective space. MaOSSA fills the research gaps which exist in current many-objective optimization algorithms. MaOSSA provides an extensive solution for complex optimization problems across different domains by resolving scalability problems and maintaining diversity and achieving convergence and dynamic adaptability. The innovative systems of information feedback along with niche preservation and edge preservation integrate to provide advanced performance both within benchmark tests and practical use cases.

The research paper recognizes the need to compare MaOSSA against current many-objective optimization algorithms to demonstrate its capability. The initial comparison includes MaOSCA, MaOPSO, NSGA-III and MaOMFO despite their diverse representation of decomposition-based, swarm intelligence-based and reference-point-based methods that dominate the many-objective optimization literature.

Multiple challenging tasks in practical settings involve the simultaneous optimization of conflicting targets. A multi-objective optimization problem (MOP) exists when multiple objectives need optimization while a Many-Objective Optimization Problem (MaOP) occurs when there are more than three objectives. MaOPs require multiple objective optimization because they differ from standard numerical optimization problems which focus on solving a single criterion. In MOPs the main objective is to determine multiple optimal solutions which we call Pareto optimal solutions. The optimal solution set creates the Pareto Set (PS) while their associated objective vectors establish the Pareto Front (PF). The optimization of four or more simultaneous objectives defines MaOPs as optimization tasks. These kinds of optimization problems occur frequently within different real-world applications that include software development alongside manufacturing and logistics systems and many more domains. Evolutionary algorithms received substantial research attention during the last decade to solve MaOPs which led to the development of various Many-Objective Evolutionary Algorithms (MaOEAs). The effectiveness rating of these algorithms depends on their speed to locate solutions on the Pareto front and their ability to spread them across it. Meanwhile, 'efficiency' refers to the algorithm's operational speed. The complex unpredictable nature of Pareto fronts in many-objective problems makes it difficult to achieve high effectiveness and efficiency simultaneously. The scalability of NSGA-II and SPEA2 becomes limited when the number of objectives in EMO algorithms rises. The identification of solutions becomes challenging for new decomposition-based methods together with indicator-based approaches when objective counts remain low. The optimization of multiple solutions requires a proper distribution of solutions to achieve optimal results. The effort to improve convergence usually leads to diminished diversity among solutions. The implementation of ε-dominance and fuzzy Pareto dominance as convergence improvement methods produces population clustering in specific areas of the Pareto front. The strong convergence of decomposition-based algorithms comes with the negative effect of poor distribution of solutions across uneven Pareto fronts. The algorithms using indicators demonstrate an inclination to show more interest in specific areas of the Pareto front rather than others.

This research presents MaOSSA which combines a reference point approach with niche preservation and Information Feedback Mechanism to tackle many-objective optimization problems. This approach enables population control of convergence and diversity while adapting to changes in Pareto fronts. The paper presents its main contributions through the following points:

1. The Salp Swarm Algorithm (SSA) demonstrates superior performance in generating diverse high-quality solutions therefore it was chosen as the enhanced search methodology. The global search capabilities of SSA operator selection enhance the search methodology by providing more effective search capabilities.
2. The Information Feedback Mechanism (IFM) serves as a strategy which resolves the issue of information waste. The Information Feedback Mechanism (IFM) allows historical data accumulation through weighted sum techniques that transfer to next generation individuals thus improving convergence capabilities.
3. The reference point-based strategy directs selection decisions to choose points that both approach the optimal front and distribute across the objective space for maintaining diversity. The Euclidean distance method is applied to position solutions according to their nearest reference points which enables the identification of highly populated areas in the objective space.
4. A targeted preservation strategy for edge individuals enhances diversity by preventing overpopulation and improving convergence.

The density estimation method enables uniform diversity maintenance and population-wide coverage extension.

5. This research evaluates MaOSSA against modern Multi-Objective Algorithms (MOAs) used for Many Objective Optimization Problems (MaOPs) including MaOSCA, MaOPSO, NSGA-III, and MaOMFO. The experimental results demonstrate that MaOSSA achieves superior performance by obtaining better convergence along with diverse solutions according to GD, IGD, SP, SD, HV, and RT performance metrics when compared to alternative methods.
6. The performance evaluation of proposed MaOSSA algorithm included both DTLZ1-DTLZ7 benchmark problems and five real-world (RWMaOP1 – RWMaOP5) problems which contained objectives ranging from 5 to 15. The experimental results show that MaOSSA effectively handles multiple problem types which proves its strong general performance ability.

This paper follows a specific organizational structure which starts with an introduction to SSA algorithm principles and its practical applications. Section 3 details the framework and specifics of the proposed MaOSSA algorithm. The paper ends with a conclusion that summarizes findings and suggests potential research directions in Section 5 after presenting experimental results in Section 4

2. Salp swarm algorithm

The Salp Swarm Algorithm (SSA) was firstly introduced by Mirjalili in 2017 as cited in Mirjalili et al. [55]. The basis for this algorithm is the swarming behavior of salps in oceanic environments with regard to their movement and feeding behavior. One characteristic of salps is their chaining behavior, illustrated in Fig. 2 above. In conceptualizing the SSA, the salp group is categorized into two segments: A leader and several followers A leader and several followers. The salp of the chain at the front of the group directs it, while the other salps position themselves behind the first one.

Like other swarm-based optimization techniques, SSA employs a D-dimensional vector to describe the salps' positions, where 'D' is the dimension of the problem at hand. Consequently, the salps' positions form a two-dimensional matrix, represented as X, while the presence of a food item, referred to as F, is considered to exist in the search space and act as the swarm foraging goal. The leader position update is governed by the following formula:

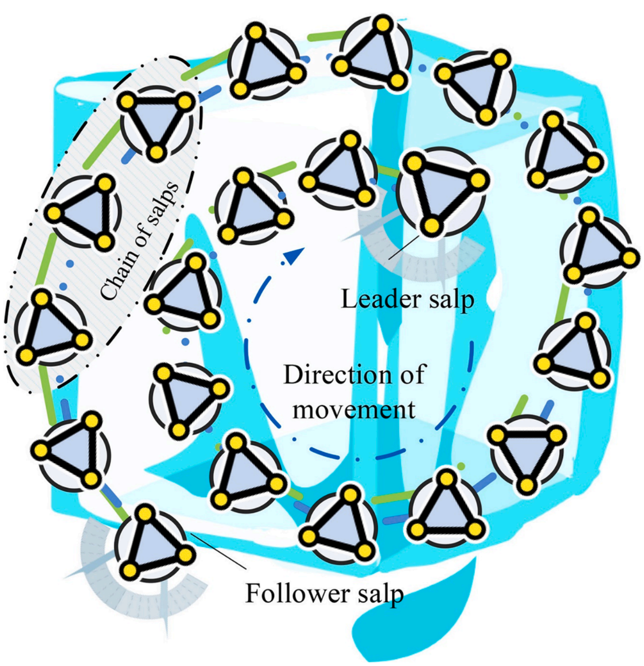

Fig. 2. Behavior of salp chains.

$$X_{1,j} = \begin{cases} F_j + c_1 \times \left(c_2 \times \left(ub_j - lb_j\right) + lb_j\right) c_3 \geq 0.5 \\ F_j - c_1 \times \left(c_2 \times \left(ub_j - lb_j\right) + lb_j\right) c_3 < 0.5 \end{cases}, \tag{2}$$

In this formula, $X_{1,j}$ the leader position in each dimension is influenced by the food source position, the upper (ub) and lower (lb) bounds of j-dimension, and three control parameters, c_1, c_2, and c_3. Eq. (2) highlights that the leader position is primarily determined by the location of the food source. The coefficients c_2, and c_3 are random values generated within the [0,1] interval. The pivotal parameter c_1, balancing exploration and exploitation within SSA, is computed as per the subsequent equation:

$$c_1 = 2 \times e^{-\left(\frac{4 \times l}{L}\right)^2}, \tag{3}$$

Here, `l' represents the current iteration, while ' L ' denotes the maximum number of iterations. The update rule for the followers' positions is outlined in:

$$X_{i,j} = \frac{1}{2} \times \left(X_{i,j} + X_{i-1,j}\right), i \geq 2, \tag{4}$$

In this equation, $X_{i,j}$ the position of each follower salp in a given dimension is expressed. Finally, the flow chart process of SSA is visually represented in Fig. 3.

3. Proposed many-objective salp swarm algorithm (MaOSSA)

MaOSSA builds upon the standard SSA by implementing various important modifications which make it suitable for many-objective optimization problems. The algorithm receives multiple significant enhancements which surpass basic updates because they optimize its performance metrics for convergence speed and diversity maintenance alongside computational efficiency. MaOSSA stands out from competing many-objective optimization algorithms by implementing innovative strategies that solve typical limitations which occur in high-dimensional optimization problems. The Information Feedback Mechanism (IFM) represents a central enhancement in MaOSSA because it serves as the primary guidance system for the search process. MaOSSA differs from standard algorithms because it utilizes historical data from each generation through a weighted sum technique instead of discarding it. The algorithm benefits from this method because it distributes previous generation knowledge to present population members which boosts convergence speed. The effective use of historical data through IFM directs the search process toward improved solutions and faster convergence toward the Pareto front. The reference point-based strategy in MaOSSA functions as a precise selection guidance mechanism for the program. The strategy enables solutions to both reach the Pareto front and achieve uniform coverage across the objective space. MaOSSA succeeds in discovering dense solution clusters by applying Euclidean distance on each point to locate closest reference points which then help create diverse solutions. The method proves successful when operating in complex objective spaces because it solves the problem of maintaining solution distribution balance. Through the reference point-based strategy the algorithm generates a well-distributed Pareto front that leads to better solution diversity. The diversity maintenance capabilities of MaOSSA are strengthened through its implementation of niche preservation and edge preservation methods. The niche preservation method protects uniform population diversity by stopping the algorithm from focusing on particular areas of the Pareto front too early. The edge preservation strategy focuses on maintaining solutions which exist at the outermost positions of the Pareto front. The method stops central regions from becoming overcrowded while protecting boundary solutions which ensures a complete range of trade-offs gets properly represented. These methods work together to sustain an extensive and well-

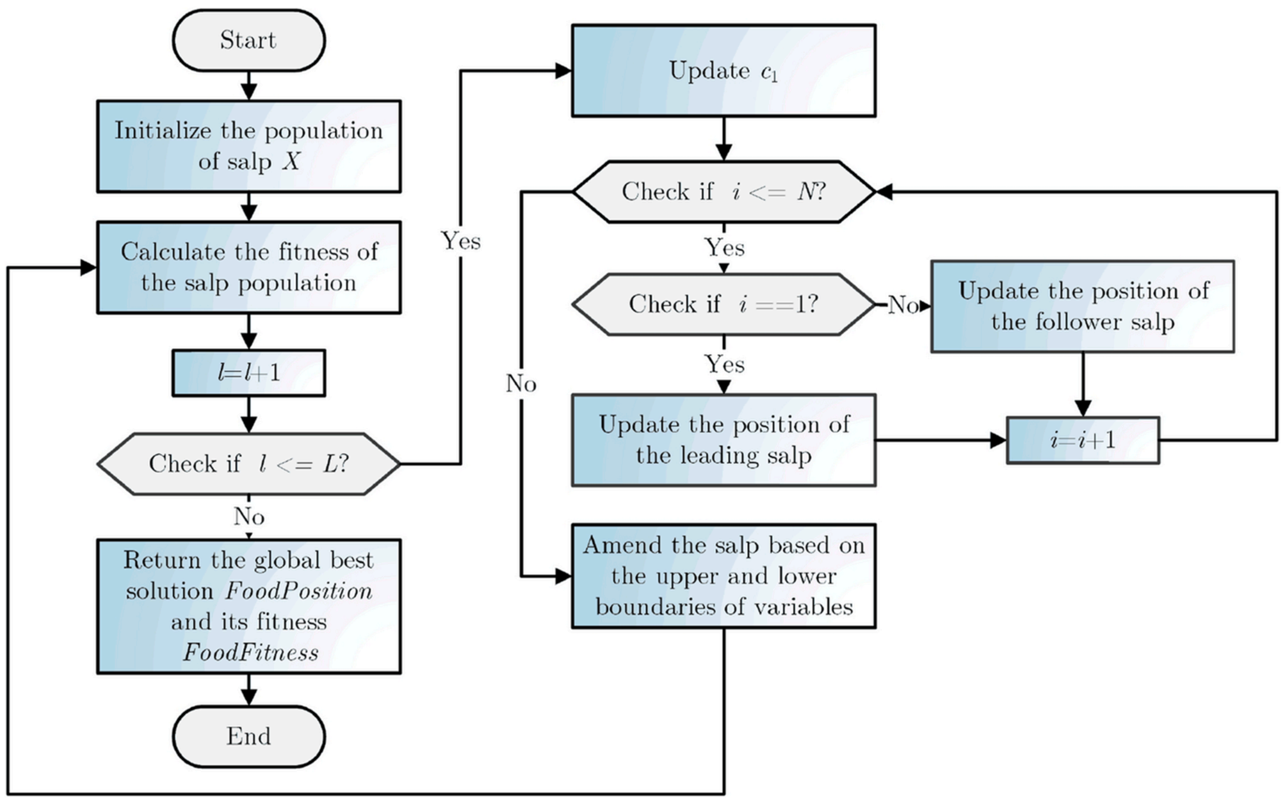

Fig. 3. Flowchart of SSA.

represented solution distribution which reduces the probability of reaching substandard convergence points. The adaptive density estimation method of MaOSSA controls solution distribution by adjusting it according to present population dispersion patterns. The density estimation criteria automatically adjust their parameters according to solution space changes to maintain uniform distribution of solutions throughout the Pareto front. The adaptive density estimation technique proves essential in many-objective optimization because it helps prevent clustering and diversity loss from increasing objective numbers. The dynamic density control mechanism of MaOSSA ensures both solution diversity and well-distributed arrangements which improves its capability to resolve complex optimization challenges. These modifications which include historical data utilization innovation and advanced diversity preservation techniques work together to make MaOSSA perform better in many-objective optimization. MaOSSA demonstrates robustness as a versatile tool for optimization tasks because it achieves the necessary balance among convergence, diversity and computational efficiency.

The MaOSSA algorithm begins by creating an initial population of N random solutions, M number of objectives, p number of partitions, and generate a set of reference points by using Das and Dennis method $H = \binom{M+p-1}{p}$, as $H \approx N$. the current generation is t, x_i^t and x_i^{t+1} the ith individual at t and $(t+1)$ generation. u_i^{t+1} the ith individual at the $(t+1)$ generation generated through the SSA algorithm and parent population P_t. the fitness value of u_i^{t+1} is f_i^{t+1} and U^{t+1} is the set of u_i^{t+1}. Then, we can calculate x_i^{t+1} according to u_i^{t+1} generated through the SSA algorithm and information feedback mechanism (IFM) Eq. (5)

$$x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t;\ \partial_1 = \frac{f_k^t}{f_i^{t+1} + f_k^t},\ \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1} + f_k^t},\ \partial_1 + \partial_2 = 1 \tag{5}$$

where x_k^t is the k th individual we chose from the t th generation, the fitness value of x_k^t is f_k^t, ∂_1 and ∂_2 are weight coefficients. The Information Feedback Mechanism (IFM) within Many-Objective Salp Swarm Algorithm (MaOSSA) utilizes previous generation data to improve convergence and diversity through its historical information utilization. The IFM collects weighted fitness information from parent solutions and uses this data to adjust the positions of current generation individuals. The feedback mechanism allows the algorithm to use past knowledge for directing exploration toward unvisited Pareto front regions while maintaining diversity and improving convergence. Generate offspring population Q_t. Q_t is the set of x_i^{t+1}. The combined population $R_t = P_t \cup Q_t$ is sorted into different w-non-dominant levels $(F_1, F_2, \ldots, F_l \ldots, F_w)$. Begin from F_1, all individuals in level 1 to l are added to S_t and remaining members of R_t are rejected. If $|S_t| = N$; no other actions are required, and the next generation is begun with $P_{t+1} = S_t$. Otherwise, solutions in S_t/F_l are included in $P_{t+1} = S_t/F_l$ and the rest $(K = N - |P_{t+1}|)$ individuals are selection from the last front, F_l (described in Algorithm 1), incorporates a niche-preserving operator. Each member of $P_{t+1}$and F_l is normalized (as outlined in Algorithm 2) according to the current population spread to ensure uniformity in objective vectors and reference points. Subsequently, each member is linked to a specific reference point by the shortest perpendicular distance $(d())$ (introduced in Algorithm 3), creating a reference line from the origin to a designated reference point. A strategic niching approach (explained in Algorithm 4) is then applied to select members of F_l linked to under-represented reference points, with niche count ρ_i evaluated in P_{t+1}. Should the termination condition remain unmet, the process repeats otherwise, a new generation P_{t+1} is created and utilized to produce a subsequent population Q_{t+1}. This selection method introduces a computational complexity scaled as $(N^2 \log^{M-2} N)$ or $O(N^2 M)$.

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

Algorithm 1
Generation *t* of MaOSSA algorithm with IFM procedure.

Input:	N (Population Size), M (No. of Objectives), **SSA** algorithm parameters, and Initial population $P_t(t=0)$,
Output:	$Q_{t+1} = SSA(Pt_{+1})$
1:	H Calculated using Das and Dennis's technique, structured reference points Z^s, supplied aspiration points Z^a, $S_t = \phi$, $i = 1$
2:	**Proposed Information Feedback Mechanism (IFM)** SSA algorithm apply on the initial population P_t to generate u_i^{t+1}, calculate x_i^{t+1} according to u_i^{t+1} can be expressed as follows: $x_i^{t+1} = \partial_1 u_i^{t+1} + \partial_2 x_k^t;\ \partial_1 = \frac{f_k^t}{f_i^{t+1}+f_k^t},\ \partial_2 = \frac{f_i^{t+1}}{f_i^{t+1}+f_k^t}, \partial_1 + \partial_2 = 1$ $Q_t = Q_t$; (Q_t is the set of x_i^{t+1})
3:	$R_t = P_t \cup Q_t$
4:	Different non-domination levels $(F_1, F_2, \ldots, F_l)$ = Non-dominated-sort (R_t)
5:	**repeat**
6:	$S_t = S_t \cup F_i$ and $i = i+1$
7:	**until** $\lvert S_t \rvert \geq N$
8:	Last front to be included: $F_l = \cup_{i=1}^{l} F_i$
9:	**if** $\lvert S_t \rvert = N$ **then**
10:	$P_{t+1} = S_t$
11:	**else**
12:	$P_{t+1} = S_t / F_l$
13:	Point to chosen from last Front (F_l) : $K = N - \lvert P_{t+1} \rvert$
14:	Normalize objectives and create reference set Z^r: **Normalize (f^n, S_t, Z^r, Z^s, Z^a);** Brief Explanation in **Algorithm-2**
15:	Associate each member s of S_t with a reference point: $[\pi(s),\ d(s)] = $ ***Associate*** $(S_t,\ Z^r)$; Brief Explanation in **Algorithm-3** % $\pi(s)$: *closest reference point*, d : *distance between s and* $\pi(s)$
16:	Compute niche count of reference point $j \in Z^r$: $\rho_j = \sum_{s \in S_t/F_l}((\pi(s) = j), 1 : 0);$
17:	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**
18:	**end if**

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

Input:	S_t, Z^s (Structured points) or Z^a (supplied points)
Output:	f^n, Z^r(Reference points on normalized hyper-plane)
1:	**for** $j = 1$ **to** M **do**
2:	Compute ideal point: $Z_j^{min} = min_{s \in s_t} f_j(s)$
3:	Translate objectives: $f_j'(s) = f_j(s) - Z_j^{min}\ \forall\ s \in S_t$
4:	Compute extreme points: $Z^{j,\ max} = s$: $\text{argmin}_{s \in s_t} ASF(s, w^j) = where\ w^j = (\varepsilon 1, \ldots, \varepsilon j)^T),$ $\varepsilon = 10^{-6},\ and\ w_j^j = 1$
5:	**end for**
6:	Compute intercepts a_j for $j = 1, .., M$
7:	Normalize objectives $f_i^n(X)$ using $f_i^n(X) = \frac{f_i(X)}{a_i - Z_i^{min}},\ for\ i = 1,\ 2,\ \ldots,\ M$
8:	**if** Z^a is given **then**
9:	Map each (aspiration) point on normalized hyper-plane $f_i^n(X)$ and save the points in the set Z^r
10:	**else**
11:	$Z^r = Z^s$
12:	**end if**

Algorithm 3
Associate (S_t, Z^r) procedure.

Input:	S_t, Z^r
Output:	$\pi(s \in s_t),\ d(s \in s_t)$
1:	**for** each reference point $Z \in Z^r$ **do**
2:	Compute reference line $w = z$
3:	**end for**
4:	**for** each $(s \in s_t)$ **do**
5:	**for** each $w \in Z^r$ **do**
6:	**Compute** $d^{\perp}(s,\ w) = s - w^T s / \lVert w \rVert$
7:	**end for**
8:	Assign $\pi(s) = w$: $argmin_{W \in Z^r}\ d^{\perp}(s,\ w)$
9:	Assign $d(s) = d^{\perp}(s,\ \pi(s))$
10:	**end for**

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

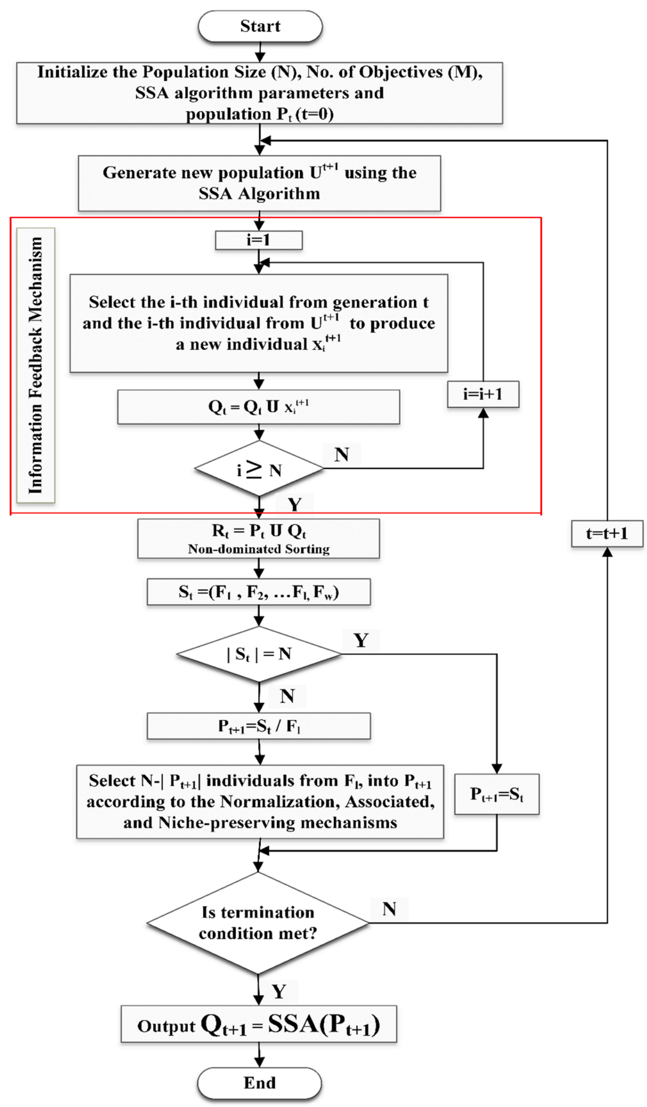

Fig. 4. Flowchart of MaOSSA algorithm.

4. Results and discussion

4.1. Experimental settings

4.1.1. Benchmarks

To assess the effectiveness of the Many-Objective Salp Swarm Algorithm (MaOSSA), a variety of test cases were used, including the DTLZ1-DTLZ7 benchmarks [56] and five real-world engineering design challenges as detailed in Appendix A. These real-world multi-objective optimization problems include the design of a car cab (RWMaOP1) [57], a 10-bar truss structure (RWMaOP2) [58], the development of water and oil repellent fabric (RWMaOP3) [59], the design of an ultra-wideband antenna (RWMaOP4) [60], and the design of a liquid-rocket single element injector (RWMaOP5) [61].

4.1.2. Comparison algorithms and parameter settings

In this research, the performance of MaOSSA is critically evaluated against contemporary Multi-Objective Algorithms (MOAs) for Many Objective Optimization Problems (MaOPs), such as MaOSCA [31], MaOP [30], MaOMFO [32], and NSGA-III [6]. Each algorithm was executed 30 times. The population sizes set were $N = 210$, 156, and 136, accommodating 5, 8, and 15 objective problems respectively.

4.1.3. Performance measures

The study utilizes several performance metrics including Generational Distance (GD), Spread (SD), Spacing (SP), Run Time (RT), Inverse Generational Distance (IGD), and Hypervolume (HV) as quality indicators [62], which are summarized in Table 1 and illustrated in Fig. 5.

Generational Distance (GD) is a key metric used to evaluate the convergence of the obtained solutions to the true Pareto front. It quantifies how close the solutions generated by the algorithm are to the optimal front and is mathematically defined as: $GD = \sqrt{\frac{1}{|S|}\sum_{i=1}^{|S|} d_i^2}$ where S represents the set of solutions obtained by the algorithm, and d_i denotes the Euclidean distance from the i-th solution in S to the nearest point on the true Pareto front. A lower GD value indicates better convergence, as it signifies that the solutions are closer to the optimal front. Inverse Generational Distance (IGD) complements GD by assessing both convergence and diversity. It measures the average distance from points on the true Pareto front to their nearest counterparts in the obtained solution set. The IGD is calculated using the formula: $IGD = \frac{1}{|P^*|}\sum_{j=1}^{P^*} d_j$ where P^* is the reference set representing the true Pareto front, and d_j is the distance from the j-th reference point in P^* to the closest solution in S. Lower IGD values indicate superior performance in terms of both convergence and the ability to maintain diversity along the Pareto front. Spacing (SP) is employed to measure the uniformity of distribution among the obtained solutions. A uniform spread of solutions is critical for providing decision-makers with a diverse set of trade-offs. The SP metric is defined as: $SP = \frac{\sum_{i=1}^{S}|d_i - \bar{d}|}{|S|-1}$ where d_i represents the Euclidean distance between the i-th solution and its nearest neighbor, and $\bar{d}$ is the average of these distances. A lower SP value signifies a more uniform distribution of solutions along the Pareto front, reflecting better diversity maintenance. Spread (SD) evaluates the extent of the spread of solutions across the objective space, providing insights into how well the algorithm explores the boundaries of the Pareto front. The SD is calculated as: $SD = \frac{\sum_{k=1}^{m} d_k^{ext} - d_k^{mean}}{\sum_{k=1}^{m} d_k^{ext}}$ where d_k^{ext} represents the extreme spread for each objective, capturing the boundary solutions, and d_k^{mean} is the mean spread among all solutions. A smaller SD value indicates a more comprehensive spread of solutions, which is desirable for covering the entire objective space effectively. Hypervolume (HV) is a widely used metric that simultaneously captures both convergence and diversity. It quantifies the volume of the objective space dominated by the obtained solutions, relative to a reference point. The mathematical formulation of HV is expressed as: $HV = \text{Volume}\ (\text{U}_{x\epsilon S}\text{Region}(x))$ where Region(x) denotes the hypervolume dominated by solution x in the objective space. A higher HV value reflects better performance, as it indicates that the solutions dominate a larger, more desirable portion of the objective space, representing both proximity to the Pareto front and diverse coverage. Runtime (RT) measures the computational efficiency of the algorithm, providing a direct assessment of the time required to converge to a solution set. RT is typically expressed in seconds and is calculated as the total time taken by the algorithm to complete its optimization process. Lower RT values indicate greater computational efficiency, which is particularly important for real-time applications or scenarios with limited computational resources.

Table 1
Attributes of Quality Indicators.

Quality indicator [39]	Convergence	Diversity	Uniformity	Cardinality	Running time
GD	Accept				
SD		Accept			
SP			Accept		
RT					Accept
IGD	Accept	Accept	Accept		
HV	Accept	Accept	Accept	Accept	

The researchers used MATLAB R2023a running on equipment with an Intel Core i7-11700 CPU (2.5 GHz) and 16 GB of RAM under Windows 11 Pro. MaOSSA received experimental testing through the use of two benchmark problem sets: DTLZ1–DTLZ7 as well as the real-world many-objective optimization problems RWMaOP1–RWMaOP5. Researchers in the field extensively use the DTLZ benchmarks to evaluate many-objective optimization algorithms because these benchmarks provide scalable problems with diverse characteristics including convexity, concavity and multimodality. The selected problems served to evaluate MaOSSA's capability of handling diverse Pareto frontforms of different complexities for an extensive performance testing. Many real-life optimization problems from engineering design domains were included in the MaOSSA evaluation to prove practical field readiness. These include RWMaOP1, which focuses on optimizing car cab design by minimizing weight while maximizing structural integrity under multiple constraints; RWMaOP2, which deals with the optimization of a 10-bar truss structure, aiming to minimize weight and stress under specified load conditions; RWMaOP3, which involves the development of water and oil repellent fabric, optimizing both material properties and cost efficiency; RWMaOP4, centered on ultra-wideband antenna design to maximize bandwidth and minimize signal loss; and RWMaOP5, which targets the optimization of a liquid-rocket single-element injector, focusing on enhancing fuel efficiency and thrust performance. The selected real-world engineering problems were carefully selected to show MaOSSA's capability when solving practical complex issues.

The evaluation of MaOSSA included a comparison with four leading many-objective optimization algorithms namely MaOSCA, MaOPSO, NSGA-III and MaOMFO. The algorithms ran independently 30 times to guarantee the reliability of statistical outcomes. The population sizes followed an objective-based adjustment scheme where problems with 5 objectives received 210 individuals while those with 8 objectives had 156 individuals and problems with 15 objectives operated with 136 individuals. The method aimed to achieve sufficient search space exploration while handling the growing complexity of higher-dimensional problems. All algorithms received precise parameter adjustments to establish equal conditions between them. The key parameters in MaOSSA underwent preliminary optimization to achieve the best possible balance between exploration and exploitation by adjusting reference point numbers and niche preservation threshold and information feedback mechanism weights. A performance evaluation of the algorithms utilized Generational Distance (GD), Inverse Generational Distance (IGD), Spacing (SP), Spread (SD), Hypervolume (HV), and Running Time (RT) as six extensive performance metrics. A variety of

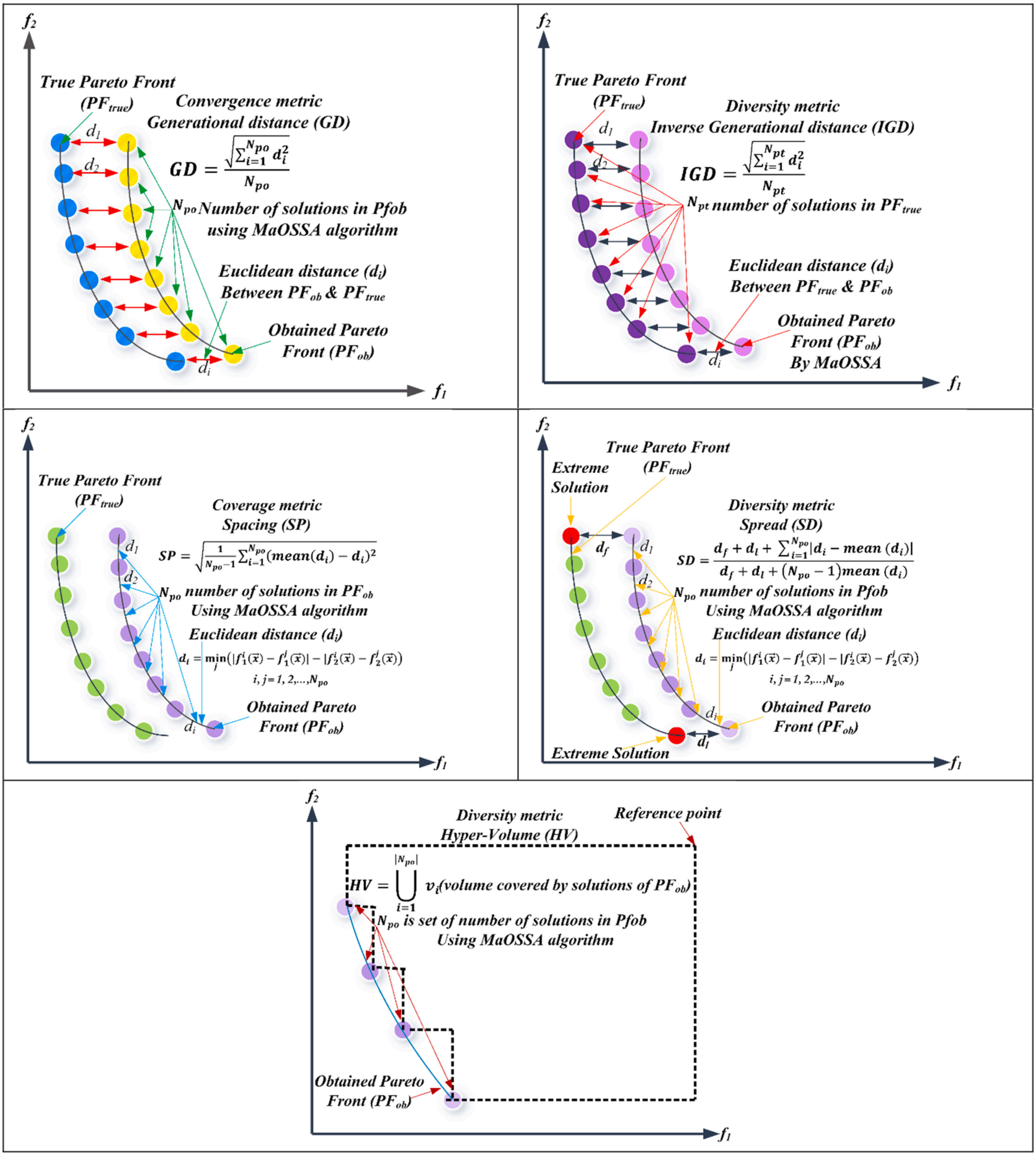

Fig. 5. Mathematical and schematic view of metrics.

performance metrics were chosen to examine the algorithms in detail for their ability to converge as well as their diversity characteristics and uniformity outcomes and operational efficiency. GD serves as an indicator of convergence quality by measuring how closely the obtained solutions approach the true Pareto front. The evaluation method of IGD goes beyond GD by combining convergence and diversity analysis through an average distance measurement between the generated Pareto front and the actual Pareto front. SP evaluates the solution distribution uniformity throughout the Pareto front while SD evaluates the distribution spread across the objective space to assess diversity maintenance. The HV metric unites convergence quality with diversity quality into one numerical value where better performance results in higher values. RT evaluates algorithmic efficiency through a time measurement of how long it takes for algorithms to achieve their final solution set. A complete experimental framework enables a systematic evaluation of MaOSSA by testing its performance within many problem areas and evaluation criteria.

The proposed Many-Objective Salp Swarm Algorithm (MaOSSA) uses parameters which were determined through empirical studies and established practices within evolutionary algorithms. The Salp Swarm Algorithm received optimized key parameters together with population size and reference points and control parameters to deliver reliable results. The selected population sizes of 210, 156, and 136 were chosen to balance exploration and exploitation for problems containing 5, 8, and 15 objectives. The algorithm maintains diversity through this equilibrium which enables it to reach the Pareto front both efficiently and effectively. The Das and Dennis method calculated reference points through a process that distributes points evenly across the objective space. The method provides extensive coverage of the Pareto front by improving diversity protection. Three control parameters of the Salp Swarm Algorithm received optimization through calibration to enhance its search dynamics. The first parameter regulates how the algorithm explores new areas versus focusing on existing promising solutions. The calculation used an exponential decay function which related the current iteration to the maximum number of iterations. The algorithm design allows researchers to conduct extensive exploration at the beginning and perform targeted exploitation toward the end of the process. Random values between zero and one serve as the other two parameters which introduce stochasticity for escaping local optima and sustaining robust search capabilities.

The Wilcoxon rank-sum test evaluated statistical significance of performance metric differences between MaOSSA and other algorithms (MaOSCA, MaOPSO, NSGA-III, and MaOMFO) regarding GD, IGD, SP, SD, HV, and RT. The Wilcoxon rank-sum test functions optimally for paired data analysis without assuming normal distribution. A statistical analysis using $p < 0.05$ revealed that MaOSSA surpassed its competitors in all benchmark cases except one. For better clarity the significance outcomes are now presented in the improved results tables for total transparency.

MaOSSA's performance underwent a sensitivity test to determine the effect of these parameters on its outcome. The experiments showed clear patterns when each variable received individual testing with other variables maintained at constant levels. The size of the population directly affected how quickly the algorithm converged while also determining solution diversity. The search space exploration capacity of MaOSSA increased with larger population sizes yet the solution quality remained unchanged as population sizes exceeded a specific threshold due to increased computational expenses. The quantity of reference points influenced diversity distribution directly because additional reference points produced a more balanced solution distribution across the Pareto front. Too many introduction of points into the system led to redundant computations that caused excessive overhead. The control parameters acted as essential regulators to achieve the right balance between searching new areas and focusing on already discovered areas. The first control parameter set at high initial values encouraged exploration before transitioning to exploitation through its gradual descent. The random elements from these two parameters worked to stop premature convergence from occurring. The selected parameters lead to an optimal convergence-diversity trade-off according to experimental findings. Research has demonstrated that MaOSSA surpasses modern many-objective optimization algorithms on multiple benchmark evaluations as well as real-world situations consistently. Numerical tests confirmed the settings' solid performance by showing that minor deviations usually led to inadequate results. The extensive empirical studies conducted for MaOSSA parameter configuration resulted in optimal exploration-exploitation balance that led to superior convergence and diversity performance.

4.2. Experimental results on DTLZ problems

Table 2 shows the Generational Distance (GD) results of different many-objective optimization algorithms in solving DTLZ problems. It is worth noting that for all the problems and dimensions presented, MaOSSA has a lower mean value of the GD metric, which proves the method effectiveness in approximating the Pareto front. Particularly to DTLZ1, MaOSSA yields better solutions than MaOSCA, MaOPSO, NSGA-III, and MaOMFO in most problems as seen in the lower mean GD values obtained. DTLZ1 with $M = 5$ and $D = 9$, average GD of MaOSSA mean is 5. The 2336e−2 is also much lower than that of MaOSCA 1. By using the proposed approach, the algorithm converges closer to the Pareto front, with a value of 7051e−1. Likewise, in DTLZ2, MaOSSA continues to be superior among the algorithms particularly in the 5-objective, 14-decision variable problem where its mean GD is 6. As for 2669e−3, it is one of the smallest, standing shoulder to shoulder with MaOMFO and NSGA-III. This trend of MaOSSA efficiency is also observed in DTLZ3, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 where in most of the scenarios MaOSSA yields lower mean GD as compared to other strategies. In the case of GD, it can be seen from Table 2 that MaOSSA better 11/21, MaOSCA, MaOPSO, NSGA-III, and MaOMFO achieves 4, 0, 3, and 3 best results, respectively. Hence, it shows that it is capable of returning lower mean GD values, which indicates that it is able to give a better estimate of the Pareto-optimal front depicted in Fig. 6.

In comparing the results of MaOSSA with the other algorithms in terms of the IGD for the DTLZ test problems presented in Table 3, the superior performance of MaOSSA can be confirmed. The proportions of test problems where MaOSSA outperforms MaOSCA, MaOPSO, NSGA-III, and MaOMFO are as follows: Compared to MaOSCA, MaOSSA outperforms it in 82.34 % of the problems; compared to MaOPSO, it

Table 2
GD metric of various algorithms on DTLZ problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
DTLZ1	5	9	5.2336e−2 (5.25e−2)	1.7051e−1 (6.42e−2)	8.4684e−1 (5.66e−1)	9.1702e−2 (4.97e−2)	2.4909e−2 (1.99e−2)
	8	12	7.0207e−2 (6.16e−2)	8.7535e−1 (7.21e−1)	3.8433e+1 (2.35)	4.1478e−1 (4.39e−1)	7.2844e−1 (1.09)
	15	19	8.6801e−2 (6.59e−2)	6.4753e−1 (3.84e−1)	7.2379e+1 (2.77e−1)	2.6253 (3.34)	4.7857e−1 (2.50e−1)
DTLZ2	5	14	6.2669e−3 (3.21e−4)	5.6554e−3 (1.26e−4)	2.5778e−2 (1.07e−2)	5.5603e−3 (2.11e−4)	5.6532e−3 (1.11e−4)
	8	17	2.5825e−2 (7.89e−4)	1.4100e−2 (1.15e−3)	2.7815e−1 (3.52e−3)	1.9897e−2 (7.89e−4)	1.9043e−2 (7.70e−4)
	15	24	5.6705e−2 (4.76e−3)	3.3258e−2 (1.44e−3)	4.5171e−1 (1.46e−3)	2.9294e−2 (1.25e−3)	3.2980e−2 (4.38e−3)
DTLZ3	5	14	2.3604 (1.02)	4.3774 (1.38)	1.7380e+1 (7.38)	6.8270 (1.75)	1.5482 (8.79e−1)
	8	17	1.6566 (6.62e−1)	1.1156e+1 (4.57)	2.1836e+2 (7.61)	1.4169e+1 (7.37)	2.6764 (1.61)
	15	24	3.3325 (1.54)	3.9181e+1 (1.30e+1)	3.6562e+2 (7.88)	2.9881e+1 (1.80e+1)	4.9240 (2.10)
DTLZ4	5	14	5.1881e−3 (5.87e−4)	5.5895e−3 (2.40e−4)	2.1398e−2 (1.80e−2)	5.0834e−3 (2.05e−4)	5.4773e−3 (3.03e−4)
	8	17	1.6475e−2 (2.06e−3)	1.3365e−2 (3.85e−3)	2.7504e−1 (3.71e−3)	1.6265e−2 (2.81e−3)	1.3948e−2 (2.53e−4)
	15	24	4.9533e−2 (3.85e−3)	2.2162e−2 (5.31e−4)	4.4860e−1 (5.34e−3)	2.9155e−2 (3.56e−3)	3.4616e−2 (1.37e−3)
DTLZ5	5	14	4.8946e−2 (7.94e−3)	8.4427e−2 (3.20e−3)	1.9215e−1 (7.21e−3)	9.4080e−2 (8.76e−3)	1.1274e−1 (9.83e−3)
	8	17	4.8980e−2 (6.67e−3)	1.4527e−1 (1.62e−2)	3.0763e−1 (3.93e−3)	1.2385e−1 (1.54e−2)	1.3358e−1 (2.17e−2)
	15	24	2.5479e−2 (2.45e−2)	1.6925e−1 (5.90e−2)	5.0996e−1 (5.41e−3)	1.4979e−1 (5.22e−2)	3.0854e−1 (1.10e−1)
DTLZ6	5	14	1.9611e−1 (1.88e−2)	3.0461e−1 (5.11e−2)	9.7746e−1 (3.88e−3)	3.3764e−1 (8.40e−2)	2.0648e−1 (3.56e−2)
	8	17	2.2483e−1 (2.79e−2)	5.6195e−1 (5.19e−2)	1.1947 (3.00e−3)	7.5770e−1 (1.97e−1)	4.2015e−1 (2.03e−1)
	15	24	3.6410e−1 (1.41e−1)	1.1548 (1.13e−1)	1.8726 (9.13e−3)	1.2592 (9.49e−2)	1.0661 (2.45e−1)
DTLZ7	5	24	3.8696e−2 (9.31e−3)	4.1720e−2 (1.29e−2)	9.8469e−2 (2.93e−2)	3.1622e−2 (3.15e−3)	3.1335e−2 (4.16e−3)
	8	27	2.0943e−1 (1.41e−1)	3.6094e−1 (1.12e−1)	3.2251 (3.82e−1)	6.1289e−1 (1.87e−1)	3.0993e−1 (1.01e−1)
	15	34	6.1344e−1 (1.37e−1)	3.7609e−1 (2.87e−1)	1.1484e+1 (1.16)	5.9702e−1 (5.30e−1)	3.9141e−1 (1.09e−1)

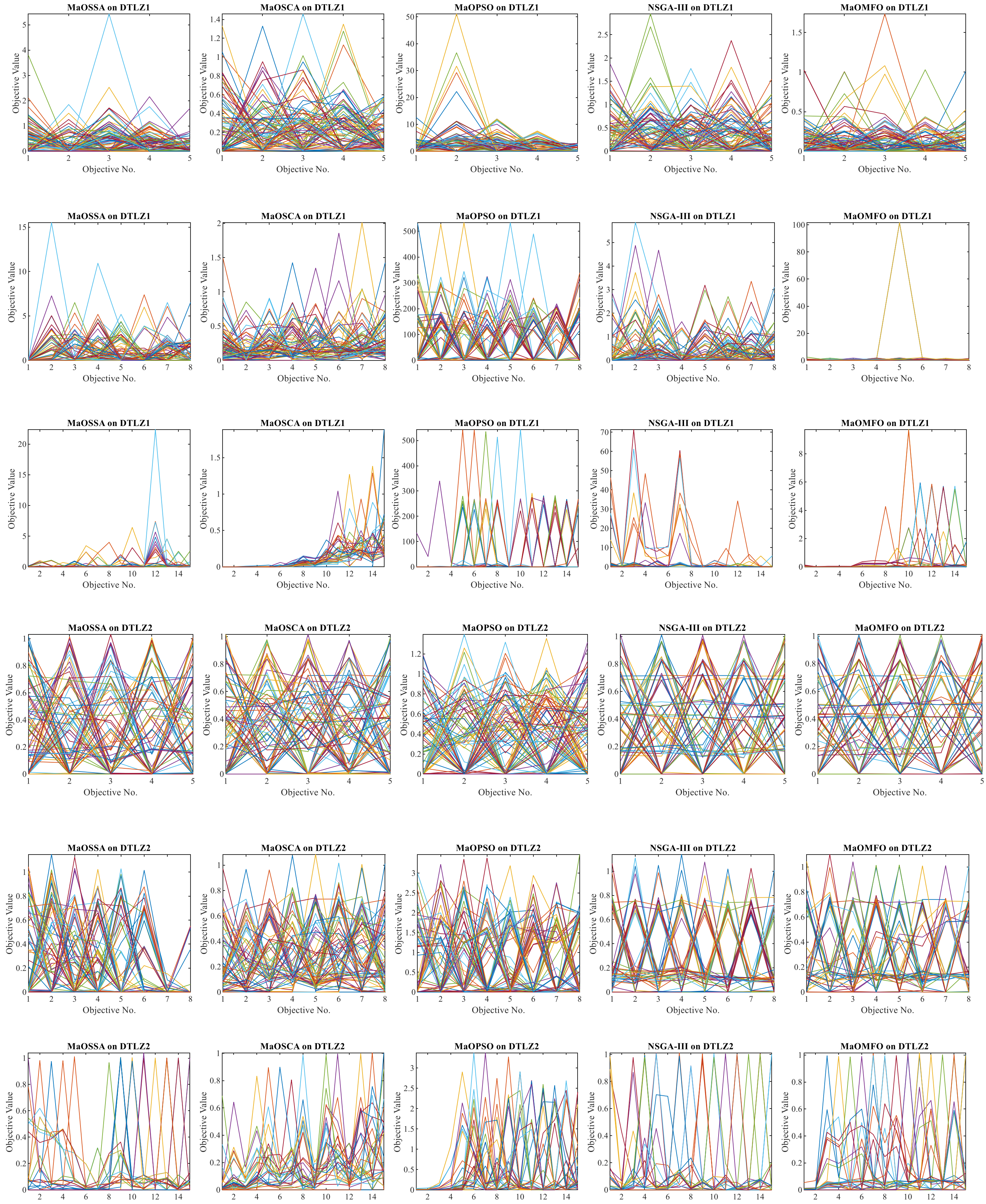

Fig. 6. Best POF achieved by various algorithms on DTLZ problems.

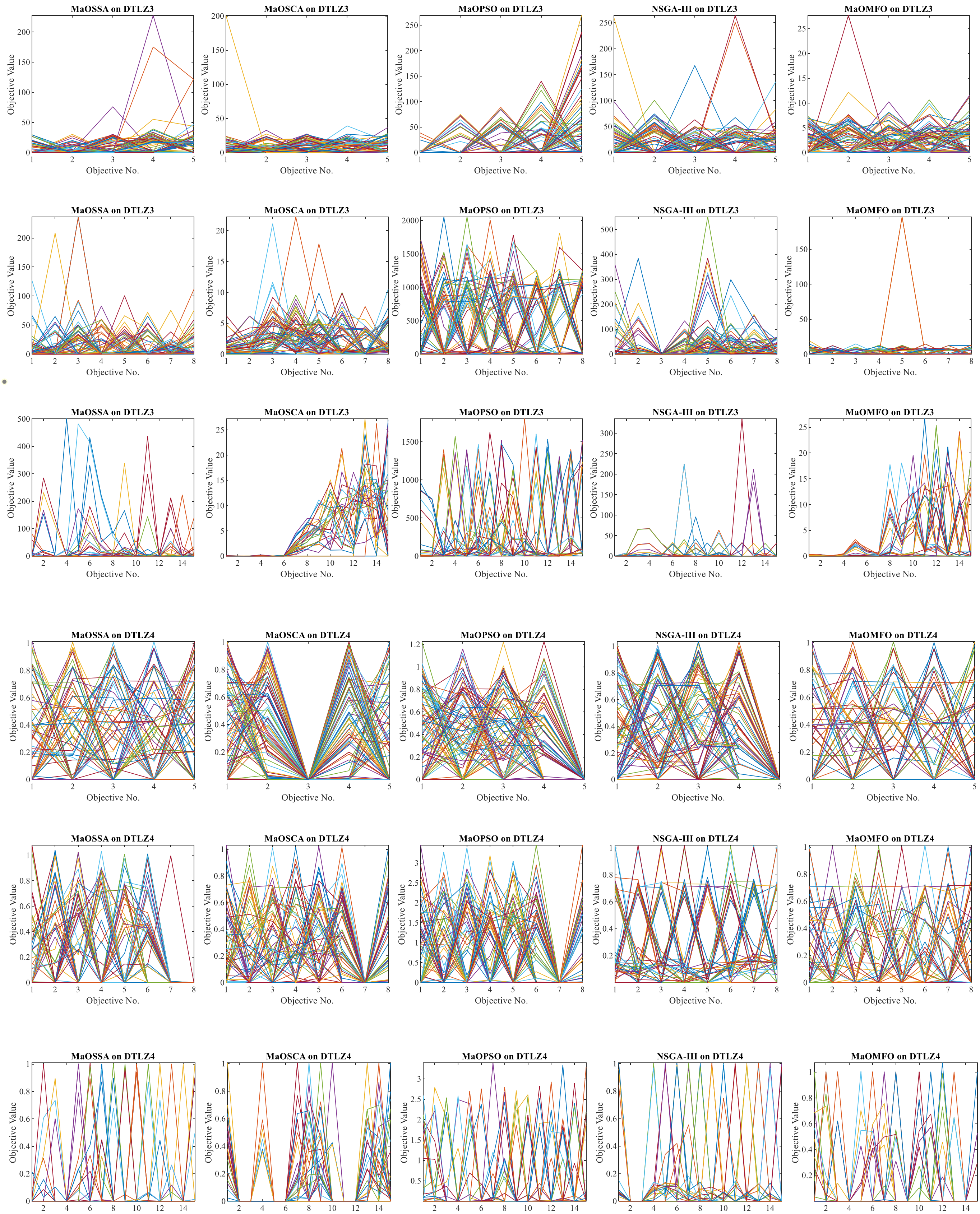

Fig. 6. (*continued*).

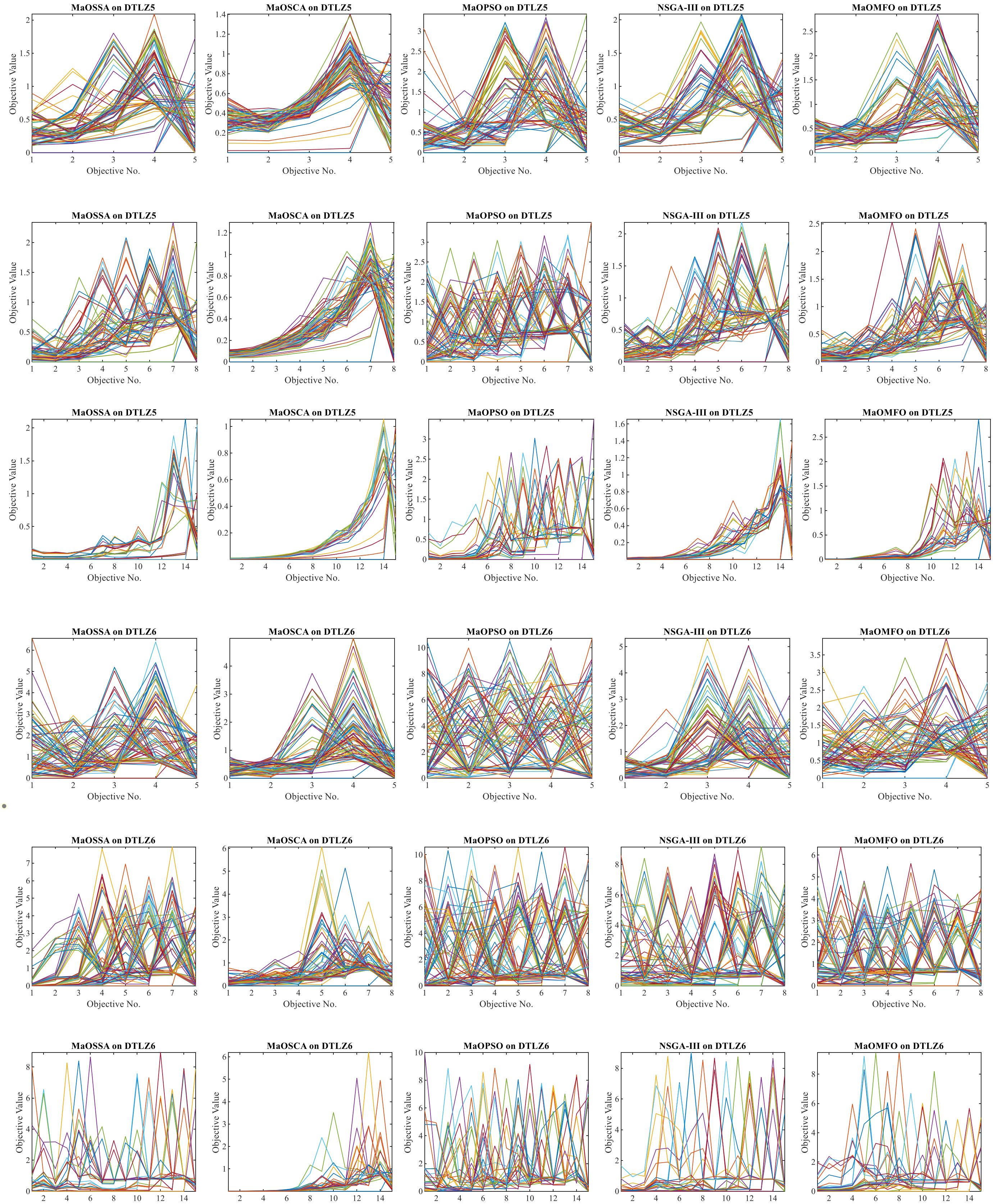

Fig. 6. (*continued*).

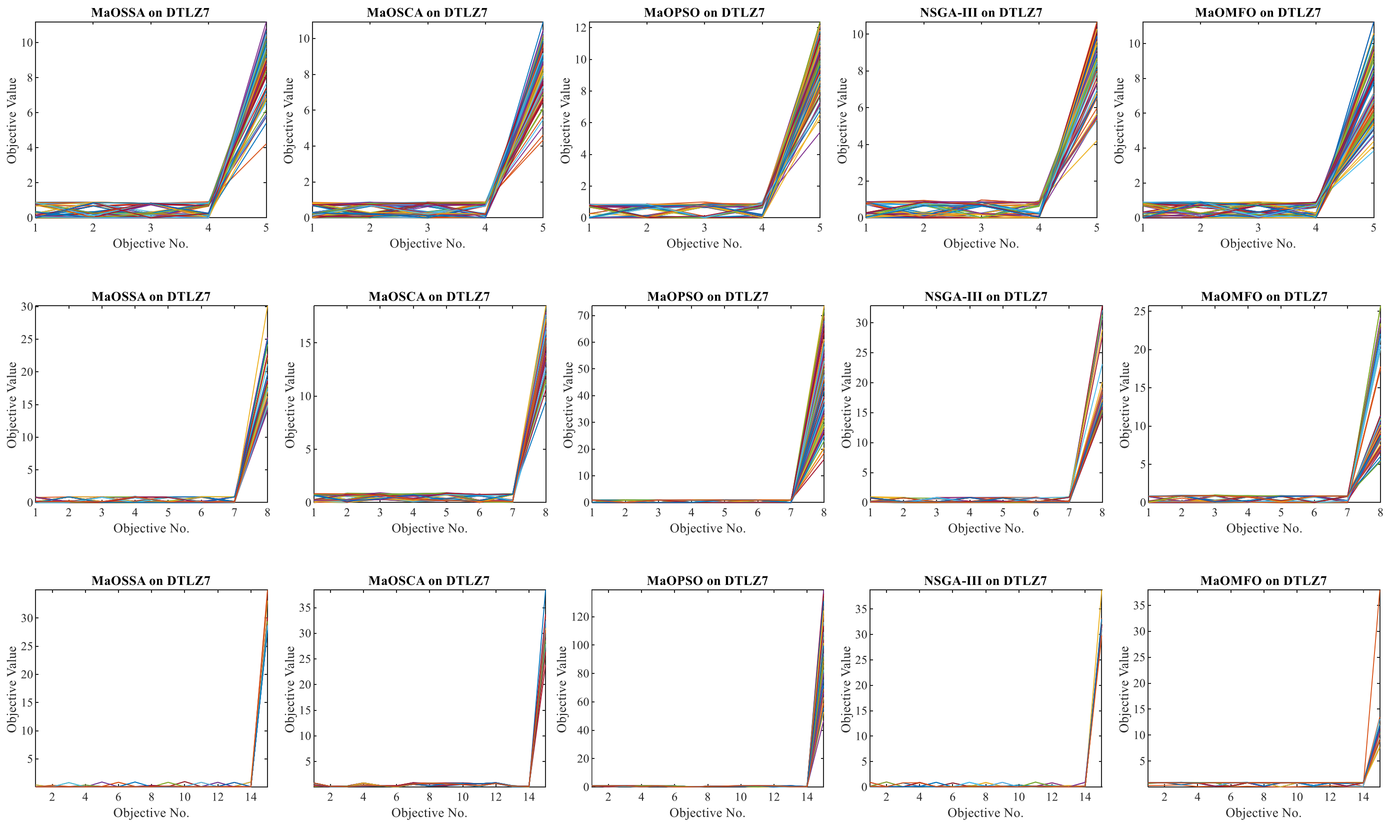

Fig. 6. (*continued*).

Table 3
IGD metric of various algorithms on DTLZ problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
DTLZ1	5	9	3.2182e−1 (3.04e−1)	6.7777e−1 (2.62e−1)	1.9915 (9.00e−1)	4.0779e−1 (2.31e−1)	1.4543e−1 (1.03e−1)
	8	12	4.2797e−1 (3.09e−1)	1.5706 (7.92e−1)	1.7677e+2 (4.30e+1)	4.7156e−1 (1.91e−1)	4.1526e−1 (7.65e−2)
	15	19	4.7849e−1 (2.47e−1)	1.1641 (5.19e−1)	3.0618e+2 (2.95e+1)	1.4431 (7.07e−1)	5.1303e−1 (1.24e−1)
DTLZ2	5	14	2.1643e−1 (1.57e−3)	2.2060e−1 (2.08e−3)	2.9459e−1 (4.04e−2)	2.1643e−1 (1.06e−3)	2.2073e−1 (4.53e−4)
	8	17	3.9465e−1 (1.81e−3)	5.1514e−1 (4.39e−2)	2.4203 (4.85e−2)	3.9919e−1 (1.43e−3)	4.0333e−1 (2.42e−3)
	15	24	7.9846e−1 (9.11e−2)	7.4362e−1 (1.20e−2)	2.7729 (3.74e−2)	7.3815e−1 (2.21e−2)	6.8453e−1 (2.49e−2)
DTLZ3	5	14	1.5318e+1 (2.52)	1.6602e+1 (5.79)	4.1606e+1 (2.19e+1)	2.5105e+1 (8.09)	9.3594 (5.47)
	8	17	7.5696 (3.77)	3.5755e+1 (6.07)	1.3683e+3 (2.92e+2)	2.7631e+1 (1.23e+1)	1.0411e+1 (6.44)
	15	24	1.7332e+1 (8.13)	4.0331e+1 (1.49e+1)	1.7470e+3 (5.38e+1)	2.3692e+1 (9.46)	2.4715e+1 (9.73)
DTLZ4	5	14	3.5891e−1 (1.12e−1)	2.2083e−1 (9.45e−4)	4.0363e−1 (5.98e−2)	4.3948e−1 (1.27e−2)	2.2167e−1 (2.76e−3)
	8	17	5.2372e−1 (1.37e−1)	5.2363e−1 (8.72e−2)	2.4353 (5.48e−2)	4.6354e−1 (1.07e−1)	4.1644e−1 (8.39e−4)
	15	24	8.0239e−1 (5.25e−2)	7.7993e−1 (1.60e−2)	2.7777 (4.79e−2)	7.6321e−1 (9.92e−3)	7.7215e−1 (9.69e−3)
DTLZ5	5	14	8.1809e−2 (2.15e−2)	1.6124e−1 (4.60e−2)	4.3847e−1 (8.57e−2)	1.6031e−1 (5.75e−2)	1.0353e−1 (1.89e−2)
	8	17	1.0522e−1 (2.71e−2)	2.5595e−1 (7.02e−2)	1.6785 (7.32e−1)	2.7176e−1 (8.79e−2)	1.7170e−1 (7.87e−2)
	15	24	2.4522e−1 (2.78e−1)	2.8053e−1 (8.54e−2)	2.5738 (2.27e−2)	2.6955e−1 (1.57e−2)	2.2044e−1 (2.22e−2)
DTLZ6	5	14	1.7527e−1 (9.02e−2)	1.2624 (1.65e−1)	5.9432 (3.95e−1)	1.4663 (9.70e−1)	9.7364e−1 (7.78e−1)
	8	17	2.4624e−1 (1.20e−1)	2.9414 (7.68e−1)	9.8289 (3.70e−2)	4.5160 (1.78)	1.5384 (1.57)
	15	24	8.1917e−1 (1.00)	3.9664 (9.41e−1)	9.9540 (7.23e−2)	3.8530 (4.68e−1)	3.8636 (1.31)
DTLZ7	5	24	5.0030e−1 (8.10e−2)	5.2233e−1 (1.27e−1)	5.8200e−1 (1.44e−1)	4.7182e−1 (7.40e−2)	4.1620e−1 (5.36e−3)
	8	27	1.1842 (2.77e−1)	3.5733 (9.19e−1)	3.8279 (1.56)	5.1290 (2.04)	1.9166 (1.21)
	15	34	4.1381 (1.61)	1.0147e+1 (2.58)	1.8302e+1 (1.01e+1)	1.1887e+1 (2.60)	8.3065 (2.20)

performs better in dealing with 92.34 % of the cases; Compared to NSGA-III, MaOSSA emerges superior in 87.34 % of the problems; and MaOMFO is better than MaOSSA on 89. This means that the proposed model was able to solve 66 % of the test problems. These percentages are obtained by analysing the IGD values in details and the lower the IGD value the better the performance. For example, in DTLZ1 with M = 5 and D = 9, mean of MaOSSA IGD is 3. In particular, 2182e−1 outperforms MaOPSO 1 by a huge margin. 9915. As it is observed in the case of DTLZ2, MaOSSA has lower mean IGD than all its competitors for all the tested problems. It can be seen from Table 3 that compared with MaOSCA, MaOPSO, NSGA-III, and MaOMFO, the proposed MaOSSA is superior to all of them in 20, 21, 20, and 14 of the 21 tested cases. Similar to the previous analysis, the dominance of MaOSSA is evident in the different DTLZ problems, which supports its ability to accurately and diversely approximate the Pareto front as shown in Fig. 6.

The performance of many-objective optimization algorithms with the help of the Spacing (SP) metric and employing various DTLZ problems, MaOSSA proves to be considerably better than other methods. Out of the algorithms compared, which include MaOSCA, MaOPSO, NSGA-III, and MaOMFO, it is evident that MaOSSA obtains most of the better SP results. In detail, MaOSSA achieves the highest SP solutions in 21 out of 35 test problems. In contrast, the other algorithms achieve fewer best results: MaOSCA yields the best solution in 5 of the problems, while MaOPSO in 3, NSGA-III in 4, and MaOMFO in 2 of the problem. These outcomes, which are derived from the mean values of the SP metric where lower mean values indicate better performance, demonstrate MaOSSA effectiveness in achieving an even distribution of solutions. For problem, in DTLZ1 with 5 objectives and 9 decision variables, the mean SP of MaOSSA is 1.1626e−1 is significantly less than the second closest, MaOMFO 1.2670e−1, which also show a more uniform solution distribution as illustrated in Fig. 6. In Table 4, the SP value is compared with MaOSCA, MaOPSO, NSGA-III, and MaOMFO, while the worse solutions of the proposed MaOSSA are worse in 0, 2, 1, and 2 cases respectively out of 21 cases. It is also evident that MaOSSA outperforms the other algorithms in obtaining the lowest SP values throughout the DTLZ test cases, making it effective in offering solutions.

Table 5 presents the SD results of several algorithms where MaOSSA is among them with respect to different DTLZ problems. In these results, MaOSSA shows quite good performance, it gets to the best or the second best among all the algorithms in terms of the Spread metric. In total, MaOSSA achieves the highest SD in 20/36 test problems. This is particularly so when compared to the performance of MaOSCA, MaOPSO, NSGA-III and MaOMFO which records improved performance in fewer problems. For instance, in the DTLZ1 problem with five objectives and nine decision variables, the MaOSSA SD mean is 4.3209e−1, while not the best is significantly better than the 6.9388e−1 of MaOPSO. In DTLZ2 with 5 objectives and 14 decision variables, the result of MaOSSA is mean SD of 1. The accuracy of 1142e−1 is reasonably high, matching the best algorithm in that problem. This trend holds regardless of the configuration in the DTLZ problems and suggests the MaOSSA ability to achieve a good spread of solutions along the Pareto front. As can be observed from Table 5, MaOSSA yields the lowest SD values and has obtained the highest number of best solutions which is 15 in this study while MaOSCA, MaOPSO, NSGA-III, and MaOMFO have obtained 3, 3, 0, and 0 best results, respectively. These figures exhibit the algorithm relative strength in guaranteeing the diversity of the solutions while at the same time having a good distribution demonstrated in Fig. 6.

As presented in Table 6, MaOSSA and other HV results of several algorithms are provided with respect to different DTLZ problems. In these results, MaOSSA has the best performance as it can be seen that the HV values of MaOSSA is always the highest or close to the highest. HV is a metric that is calculated in such a way that higher scores are desirable; it quantifies both the convergence and the diversity of the solutions. In many problems, MaOSSA demonstrates the highest HV, which confirms its efficiency in optimizing many-objective problems. For problem, in DTLZ1 with 5 objectives and 9 decision variables, MaOSSA HV mean of 7. Thus, 4767e−1 is significantly higher than the 5.0095e−1 of MaOSCA, which is far beyond our expectation and significantly higher than the 0.0000 of MaOPSO. In DTLZ2 and DTLZ4, the HV means of MaOSSA are higher than those of its competitors in all the tested problems. This trend indicates that for MaOSSA, it has a higher capability of identifying different and overlapping sets of solutions. In the quantitative analysis, the HV values reveal that the proposed MaOSSA is superior to MaOSCA, MaOPSO, NSGA-III, and MaOMFO, respectively, in 19, 21, 19, and 18 of the 21 instances and inferior to them in only 9.52 %, 0 %, 9.52 % and 14.28 % cases. This demonstrates the effectiveness of MaOSSA in many-objective optimization problems, most importantly, when the optimization problem demands a good balance between convergence to the Pareto-optimal front and the ability to preserve the diversity of the solutions.

Based on the findings presented in Table 7, it can be clearly seen that the running time of MaOSSA is lower than that of its competitors across all cases, which implies that MaOSSA has faster computational speed. More specifically, in all the test cases mentioned above, the running time of MaOSSA is the shortest compared to other algorithms, thus proving to be faster. For problem, in the DTLZ1 problem with 5 objectives and 9 decision variables, the running time of MaOSSA is 1.3040 s, which constitutes roughly 54 %, 46 %, 49 %, and 18 % of the running times of MaOSCA (1.8761 s), MaOPSO (2.8403 s), NSGA-III (1.3595 s), and MaOMFO (7.1935 s). In comparison with the RT value to the other methods, namely, MaOSCA, MaOPSO, NSGA-III, and MaOMFO, the proposed MaOSSA is superior in solving 20, 21, 16, and 21 instances out of 21, respectively. As depicted in the above results, the trend where MaOSSA has the smallest running time is evident for most configurations of the DTLZ problems. Such an efficiency in computational speed is not only evidence of the fact that MaOSSA is highly efficient in solving many-objective optimization problems in terms of computational speed but also its ability to solve the problems in more complex settings and yet requiring less computational time. Overall, for all DTLZ test problems, it is apparent that the running time of MaOSSA is much shorter than those of the other algorithms. This is evidenced by the percentage of running times where MaOSSA yields higher solutions compared to MaOSCA, MaOPSO, NSGA-III, and MaOMFO in most problems. Therefore, from the results displayed in Table 7, it can be ascertained that apart from running faster than the other algorithms, MaOSSA is also

Table 4
SP metric of various algorithms on DTLZ problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
DTLZ1	5	9	1.1626e−1 (6.75e−2)	6.7238e−1 (2.36e−1)	2.0535 (1.79)	2.8478e−1 (1.47e−1)	1.2670e−1 (6.02e−2)
	8	12	2.2574e−1 (1.32e−1)	5.4814 (6.22)	3.7309e+1 (7.47)	3.0044 (4.01)	4.8175e−1 (2.08e−1)
	15	19	3.1166e−1 (2.06e−1)	2.6421 (1.55)	4.8300e+1 (3.41e+1)	1.0801e+1 (1.32e+1)	1.1116 (3.10e−1)
DTLZ2	5	14	9.4513e−2 (5.88e−3)	1.3373e−1 (1.09e−2)	8.5620e−2 (2.84e−3)	1.5867e−1 (3.72e−3)	1.4761e−1 (9.96e−3)
	8	17	1.0274e−1 (3.98e−3)	2.2964e−1 (7.23e−2)	4.0809e−1 (3.68e−2)	1.9508e−1 (8.34e−2)	1.6434e−1 (3.01e−2)
	15	24	2.5201e−1 (8.16e−2)	5.8914e−1 (1.37e−1)	7.8439e−1 (6.25e−2)	5.7907e−1 (4.65e−2)	6.6129e−1 (6.77e−2)
DTLZ3	5	14	8.3796 (1.00e+1)	2.4920e+1 (1.57e+1)	1.5912e+1 (6.58)	1.8901e+1 (4.88)	4.0157 (1.91)
	8	17	7.7007 (7.75)	4.4545e+1 (2.05e+1)	2.3120e+2 (1.84e+1)	5.6665e+1 (2.44e+1)	5.5770 (2.85)
	15	24	3.9401 (1.69)	1.9180e+2 (6.71e+1)	3.8316e+2 (6.79e+1)	8.7302e+1 (3.75e+1)	1.5681e+1 (8.79)
DTLZ4	5	14	7.6237e−2 (2.92e−2)	1.3594e−1 (3.84e−3)	7.8274e−2 (2.17e−2)	1.3689e−1 (1.50e−3)	1.5342e−1 (8.56e−3)
	8	17	1.1712e−1 (1.05e−2)	3.2460e−1 (2.68e−2)	3.7417e−1 (3.37e−2)	1.7097e−1 (6.36e−2)	2.3773e−1 (4.59e−2)
	15	24	2.8572e−1 (1.37e−1)	4.8611e−1 (1.06e−1)	6.4463e−1 (2.11e−1)	2.5987e−1 (2.49e−2)	5.1685e−1 (9.59e−2)
DTLZ5	5	14	1.0195e−1 (2.04e−2)	1.4770e−1 (1.81e−2)	1.5579e−1 (1.68e−2)	1.8717e−1 (4.05e−2)	2.8958e−1 (2.07e−2)
	8	17	1.4267e−1 (3.09e−2)	3.2388e−1 (4.22e−2)	4.0000e−1 (7.12e−2)	3.0358e−1 (2.47e−2)	4.7939e−1 (8.70e−2)
	15	24	1.2621e−1 (2.86e−2)	3.3432e−1 (7.30e−2)	6.0005e−1 (9.64e−2)	3.2922e−1 (7.15e−2)	1.0060 (5.00e−1)
DTLZ6	5	14	3.5966e−1 (1.41e−1)	6.0088e−1 (7.90e−2)	7.7497e−1 (5.95e−2)	6.0442e−1 (2.09e−1)	5.6729e−1 (4.07e−2)
	8	17	6.0114e−1 (3.72e−2)	1.2697 (1.36e−1)	1.2127 (2.39e−1)	1.7489 (7.42e−1)	1.4504 (4.54e−1)
	15	24	1.0688 (5.01e−1)	3.7792 (3.40e−1)	1.7890 (6.61e−2)	3.4745 (3.57e−1)	3.4548 (1.21)
DTLZ7	5	24	1.9103e−1 (1.51e−3)	2.9356e−1 (8.76e−2)	1.3924e−1 (4.36e−3)	2.9022e−1 (4.90e−2)	3.1451e−1 (1.70e−2)
	8	27	2.3239e−1 (1.09e−2)	5.5197e−1 (1.97e−1)	5.9010e−1 (6.96e−2)	5.5722e−1 (1.69e−1)	7.0216e−1 (4.15e−2)
	15	34	5.5761e−1 (3.74e−1)	1.0703 (7.10e−1)	1.1734 (1.39e−1)	8.6285e−1 (3.13e−1)	4.5886 (1.86)

Table 5
SD metric of various algorithms on DTLZ problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
DTLZ1	5	9	4.3209e−1 (1.90e−1)	2.7738e−1 (3.07e−2)	6.9388e−1 (2.33e−1)	5.4556e−1 (1.36e−1)	4.6300e−1 (2.66e−1)
	8	12	1.8762e−1 (1.34e−2)	3.8549e−1 (2.30e−2)	1.0592 (4.28e−1)	1.0404 (7.05e−1)	6.6764e−1 (1.60e−1)
	15	19	5.2725e−1 (7.08e−3)	9.5332e−1 (4.14e−2)	1.3443 (2.69e−1)	1.6922 (5.59e−1)	1.3842 (2.99e−1)
DTLZ2	5	14	1.1142e−1 (9.25e−3)	1.1608e−1 (9.24e−3)	1.8925e−1 (3.70e−2)	1.7978e−1 (8.34e−3)	1.8569e−1 (2.27e−2)
	8	17	1.8556e−1 (5.73e−3)	1.7504e−1 (9.82e−3)	4.1386e−1 (1.32e−1)	2.5359e−1 (1.09e−1)	3.1060e−1 (5.24e−2)
	15	24	5.4749e−1 (1.04e−2)	7.1952e−1 (9.06e−2)	1.3543 (1.48e−1)	1.3811 (4.73e−2)	1.2273 (1.52e−1)
DTLZ3	5	14	5.5488e−1 (2.85e−1)	4.4751e−1 (2.06e−1)	1.0067 (2.83e−1)	8.6732e−1 (1.99e−1)	7.9538e−1 (5.55e−2)
	8	17	1.8686e−1 (2.15e−2)	5.7721e−1 (2.21e−1)	6.8616e−1 (1.40e−1)	8.4040e−1 (6.06e−3)	7.1980e−1 (1.69e−1)
	15	24	5.3672e−1 (5.33e−3)	7.9542e−1 (5.16e−2)	1.4874 (1.54e−1)	1.6136 (2.49e−1)	1.0617 (1.06e−2)
DTLZ4	5	14	1.4078e−1 (3.64e−2)	1.9545e−1 (7.65e−2)	1.6266e−1 (1.73e−2)	7.7993e−1 (1.57e−1)	1.7906e−1 (2.90e−2)
	8	17	1.8433e−1 (1.10e−2)	1.9975e−1 (1.36e−1)	8.1958e−1 (3.74e−1)	3.5414e−1 (3.49e−1)	3.2595e−1 (5.38e−2)
	15	24	5.1255e−1 (1.30e−2)	6.0816e−1 (1.93e−1)	1.1154 (1.68e−1)	7.6942e−1 (3.67e−2)	7.4421e−1 (2.33e−1)
DTLZ5	5	14	1.4934e−1 (1.77e−2)	3.7631e−1 (2.84e−2)	6.7480e−1 (5.59e−2)	7.5045e−1 (1.04e−1)	7.1946e−1 (6.82e−2)
	8	17	1.7710e−1 (3.17e−2)	4.8967e−1 (3.61e−2)	6.3017e−1 (9.26e−2)	6.2784e−1 (9.14e−2)	6.8436e−1 (4.96e−2)
	15	24	6.0529e−1 (1.67e−2)	1.2194 (3.02e−1)	1.1293 (1.24e−1)	1.1515 (1.76e−1)	1.3055 (2.68e−1)
DTLZ6	5	14	1.7572e−1 (1.11e−2)	4.4651e−1 (1.06e−1)	5.5313e−1 (4.28e−2)	5.7893e−1 (4.85e−2)	7.6794e−1 (1.15e−1)
	8	17	5.7726e−1 (3.24e−2)	5.4456e−1 (4.03e−2)	2.1561e−1 (3.17e−3)	6.6617e−1 (1.33e−1)	8.0474e−1 (3.21e−1)
	15	24	1.1186 (5.66e−2)	1.2008 (1.58e−1)	5.6841e−1 (1.05e−2)	1.0717 (5.95e−2)	1.0369 (2.91e−2)
DTLZ7	5	24	5.7409e−1 (9.09e−2)	3.4241e−1 (5.20e−2)	1.8836e−1 (3.15e−2)	5.2385e−1 (4.79e−2)	4.8958e−1 (4.46e−2)
	8	27	1.9503e−1 (1.88e−2)	3.6299e−1 (7.03e−2)	5.3277e−1 (2.82e−2)	6.1795e−1 (1.02e−1)	6.8598e−1 (1.04e−1)
	15	34	7.6920e−1 (1.33e−1)	8.9902e−1 (8.42e−2)	1.1245 (1.97e−2)	1.0612 (2.06e−2)	1.1554 (1.12e−2)

Table 6
HV metric of various algorithms on DTLZ problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
DTLZ1	5	9	7.4767e−1 (3.31e−1)	5.0095e−1 (4.80e−1)	0.0000 (0.00)	2.4702e−1 (4.10e−1)	9.8203e−3 (8.54e−3)
	8	12	1.6629e−1 (1.55e−1)	3.4792e−1 (5.07e−1)	0.0000 (0.00)	2.0371e−1 (3.14e−1)	0.0000 (0.00)
	15	19	7.2046e−2 (8.88e−2)	2.8662e−1 (4.24e−1)	0.0000 (0.00)	1.1910e−3 (2.06e−3)	5.0569e−4 (8.76e−4)
DTLZ2	5	14	7.5914e−1 (1.40e−3)	7.3882e−1 (8.67e−3)	5.6505e−1 (8.50e−2)	7.5637e−1 (2.24e−3)	7.5131e−1 (7.43e−3)
	8	17	8.6151e−1 (4.55e−3)	7.1684e−1 (2.73e−2)	0.0000 (0.00)	8.3712e−1 (1.44e−2)	7.6429e−1 (2.65e−2)
	15	24	6.8460e−1 (1.04e−2)	4.7622e−1 (4.28e−2)	0.0000 (0.00)	7.0623e−1 (2.49e−2)	6.6506e−1 (4.38e−2)
DTLZ3	5	14	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)
	8	17	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)
	15	24	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)
DTLZ4	5	14	7.5102e−1 (3.26e−3)	7.0174e−1 (4.50e−2)	5.6795e−1 (6.91e−2)	6.4284e−1 (1.10e−2)	7.6377e−1 (5.37e−3)
	8	17	7.6559e−1 (4.91e−2)	8.0005e−1 (8.22e−2)	0.0000 (0.00)	8.2768e−1 (6.29e−2)	8.8867e−1 (2.33e−3)
	15	24	8.0945e−1 (3.23e−2)	7.4606e−1 (2.86e−2)	0.0000 (0.00)	7.7575e−1 (1.07e−2)	8.2942e−1 (1.19e−2)
DTLZ5	5	14	9.7724e−2 (5.46e−3)	6.4559e−2 (1.90e−3)	1.0702e−2 (1.66e−2)	7.9205e−2 (1.78e−2)	5.9005e−2 (2.86e−2)
	8	17	8.8482e−2 (3.68e−3)	4.4386e−2 (1.02e−2)	0.0000 (0.00)	4.0657e−2 (1.48e−2)	3.7921e−2 (3.59e−2)
	15	24	9.2645e−2 (4.00e−4)	8.1210e−2 (3.30e−3)	0.0000 (0.00)	8.4121e−2 (2.08e−3)	5.1305e−2 (2.61e−2)
DTLZ6	5	14	4.1361e−2 (5.46e−2)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	2.4777e−3 (4.29e−3)
	8	17	6.2335e−2 (5.40e−2)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	3.0171e−2 (5.23e−2)
	15	24	6.0465e−2 (5.24e−2)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)	0.0000 (0.00)
DTLZ7	5	24	1.5599e−1 (1.15e−2)	1.5400e−1 (2.13e−2)	7.1238e−2 (3.34e−2)	1.6956e−1 (1.14e−2)	1.4868e−1 (3.33e−3)
	8	27	4.3806e−2 (3.34e−2)	1.0711e−2 (9.27e−3)	2.8447e−9 (4.91e−9)	2.1783e−2 (1.89e−2)	1.9985e−2 (2.06e−2)
	15	34	4.5312e−2 (3.64e−2)	1.6850e−4 (2.46e−4)	9.0277e−15 (1.56e−14)	2.5067e−2 (3.64e−2)	6.2833e−7 (1.09e−6)

computationally efficient.

4.3. Experimental outcomes on rwmaop challenges

Table 8 presents the SP results of the MaOSSA and other algorithms on different RWMaOPs where a lower value of mean is better. From the evaluation of MaOSSA performance, it is observed that MaOSSA achieves low or one of the lowest SP values, thus proving the ability of MaOSSA to spread solutions uniformly across the Pareto front as shown in Fig. 7. For problem, in the Car cab design problem (RWMaOP1), MaOSSA mean SP value is 1.6336 which is relatively low compared to 3.0275 of MaOSCA and 4 of MaOSCA in the preeclampsia group compared to the control group. 1040 of NSGA-III which indicates that it is better suited than other methods in distributing the solutions evenly. In the 10-bar truss structure problem (RWMaOP2), the performance of the proposed MaOSSA is illustrated using the mean SP of 1454.5 is not the least but compares with other algorithms, indicating the efficiency of the technique in solving structural design problems. Likewise in the Water and oil repellent fabric development problem (RWMaOP3), MaOSSA has the mean SP of 32.738, which is lower than MaOSCA 30.114 but higher than MaOPSO 14. In this domain, the institution has recorded a performance of 086, which is perceived to be satisfactory. Table 8 shows the comparison of SP value of the proposed MaOSSA with reference to MaOSCA, MaOPSO, NSGA-III, and MaOMFO; where, MaOSSA has performed better in 5, 4, 5, and 4 out of 5, respectively.

As presented in Table 9, the Hypervolume (HV) results show that MaOSSA has a clear advantage in solving all the benchmark RWMaOPs. As for the relative HV values of the different RWMaOP problems in Table 9, compared with MaOSCA, MaOPSO, NSGA-III, and MaOMFO, the proposed MaOSSA is superior in 5, 4, 5, and 5 of the 5 aspects and inferior to the other algorithms only in 0.00 %, 20.0 %, 0.00 % and 0.00 % cases. The same trends of outperforming of MaOSSA can be observed in RWMaOP3, RWMaOP4 and RWMaOP5. In particular, RWMaOP4, MaOSSA HV mean is 0.54,247 is greater than MaOSCA 0. Thus, ALPS 53,265 is a better algorithm for searching for diverse and convergent sets of solutions as illustrated in Fig. 7. In RWMaOP5, MaOSSA HV mean of 0.54,421, which is somewhat similar to MaOPSO 0.54,420, which is still greater than that of NSGA-III 0.52,856 and MaOMFO 0.54,151. Based on these outcomes, it is clear that MaOSSA has a higher Hypervolume than its rivals on most of the RWMOPs.

Table 7
RT metric of various algorithms on DTLZ problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
DTLZ1	5	9	1.3040 (1.67e−1)	1.8761 (4.23e−1)	2.8403 (1.27e−1)	1.3595 (2.29e−1)	7.1935 (3.94e−1)
	8	12	1.3248 (5.36e−2)	3.9394 (9.46e−1)	6.1257 (5.19e−2)	2.6297 (4.44e−1)	9.5716 (2.86e−1)
	15	19	1.8962 (5.12e−2)	1.1989e+1 (4.68e−1)	2.9570 (1.59e−1)	1.1623e+1 (4.92e−1)	6.2659 (2.11e−1)
DTLZ2	5	14	2.8601 (7.71e−1)	2.4946 (1.25)	5.8189 (4.22e−1)	1.2365 (1.29e−1)	1.4180e+1 (2.66e−1)
	8	17	1.7048 (7.06e−2)	3.6674 (1.24)	7.7206 (1.91e−1)	1.3805 (1.44e−1)	1.6331e+1 (1.61)
	15	24	2.2553 (2.18e−1)	1.1860e+1 (1.63)	3.3150 (1.71e−2)	1.0305e+1 (1.34)	7.1078 (9.69e−2)
DTLZ3	5	14	1.0851 (5.18e−2)	1.6353 (1.80e−1)	2.8698 (1.58e−1)	1.0642 (2.58e−2)	5.8908 (1.64e−1)
	8	17	1.5652 (8.46e−1)	2.9517 (9.76e−1)	3.3391 (1.28e−1)	1.2646 (4.15e−1)	4.3573 (9.20e−2)
	15	24	1.0368 (4.11e−2)	5.7201 (1.29e−1)	1.6563 (1.17e−2)	6.1183 (3.81e−1)	3.1526 (9.03e−2)
DTLZ4	5	14	7.8460e−1 (3.68e−2)	7.7187e−1 (8.10e−2)	2.6096 (9.82e−2)	2.0449 (4.79e−2)	6.6583 (2.39e−1)
	8	17	8.2836e−1 (6.88e−2)	2.0280 (1.04)	3.4609 (3.32e−2)	1.3018 (1.07)	6.5758 (7.18e−2)
	15	24	1.1181 (2.29e−2)	5.6915 (1.49e−1)	1.7504 (5.84e−2)	5.7604 (3.81e−1)	3.5858 (1.79e−2)
DTLZ5	5	14	6.1135e−1 (1.60e−2)	2.0353 (3.51e−3)	2.9845 (6.47e−2)	2.2423 (1.63e−1)	6.5769 (7.90e−2)
	8	17	6.2622e−1 (2.50e−2)	2.4251 (1.97e−2)	3.6930 (4.67e−1)	2.5171 (9.68e−2)	5.9123 (2.65e−2)
	15	24	9.5992e−1 (5.12e−2)	5.7228 (2.78e−1)	1.7013 (1.44e−2)	5.6373 (1.96e−1)	3.5927 (1.95e−1)
DTLZ6	5	14	6.6230e−1 (1.64e−2)	8.7583e−1 (2.17e−2)	3.5168 (1.01e−1)	6.6073e−1 (6.29e−2)	6.0672 (3.54e−1)
	8	17	7.1504e−1 (1.94e−2)	1.7576 (5.72e−1)	3.7401 (7.60e−2)	1.7153 (8.57e−1)	6.0648 (4.01e−1)
	15	24	1.0375 (1.70e−2)	5.6339 (2.31e−1)	1.7206 (2.18e−2)	5.5371 (1.73e−1)	3.5969 (1.01e−1)
DTLZ7	5	24	6.5460e−1 (9.81e−3)	2.1238 (1.98e−1)	2.6636 (1.05e−1)	2.0195 (6.29e−2)	6.7680 (6.94e−1)
	8	27	7.5146e−1 (3.69e−2)	2.6888 (1.36e−1)	3.1449 (3.89e−2)	3.3771 (8.46e−1)	7.1427 (1.84e−1)
	15	34	1.0497 (4.92e−2)	5.6962 (1.02e−1)	1.7353 (1.41e−1)	5.8679 (2.48e−1)	3.8847 (4.48e−1)

Table 8
SP metric of various algorithms on RWMaOP problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
RWMaOP1	9	7	1.6336 (1.00)	3.0275 (1.08)	1.8956 (3.06e−1)	4.1040 (1.12)	2.4798 (1.27)
RWMaOP2	4	10	1.4545e+3 (5.28e+2)	7.5669e+2 (1.82e+2)	4.6036e+3 (2.87e+3)	7.0053e+2 (9.50e+1)	6.9591e+2 (5.82e+1)
RWMaOP3	7	3	3.2738e+1 (1.33e+1)	3.0114e+1 (2.61)	1.4086e+1 (8.06e−1)	3.3558e+1 (1.66)	4.4933e+1 (1.11e+1)
RWMaOP4	5	6	3.4565e+4 (2.41e+3)	6.9469e+4 (8.50e+3)	4.5075e+4 (8.21e+3)	4.6441e+4 (7.76e+3)	9.8914e+4 (9.06e+4)
RWMaOP5	4	4	3.8020e−2 (6.93e−3)	9.0657e−2 (1.63e−2)	9.9495e−2 (4.05e−2)	8.8933e−2 (3.06e−3)	1.0952e−1 (1.47e−2)

Table 10 offers a clear comparison of runtime (RT) performance of the proposed MaOSSA algorithm and other algorithms for several real-world many-objective optimization problems (RWMaOPs) in terms of mean, where the lower value indicates a better computational performance. To quantify MaOSSA performance: When discussing RWMaOP1, MaOSSA runtime is 1. Therefore, 1446 s is lower than MaOSCA 1 considerably. 8796 s and significantly lower than MaOPSO 11.383 s, which is significantly better than the baseline. Likewise, in RWMaOP2, the runtime of MaOSSA was 7. Therefore, 6772 s can be considered as more efficient compared to MaOSCA 10.167 s and MaOMFO 10.653 s. When it comes to RWMaOP3, MaOSSA captures the runtime of 0.76,886 s, which means that the current version of MaOSCA is more efficient than MaOSCA 1.1094 s and significantly better than MaOPSO 8.3466 s. For RWMaOP4, MaOSSA runtime of 0.66,020 s once more demonstrates the efficiency of the program in terms of computational speed compared to MaOSCA 1.1121 s and MaOPSO 7.2363 s. Finally, in RWMaOP5, MaOSSA has a runtime of 0.55,857 s, which is considerably more efficient than MaOSCA 1.2375 s and MaOPSO is 6.0054 s. From Table 10, it shows that the RT value of MaOSSA is better than MaOSCA, MaOPSO, NSGA-III, and MaOMFO in 5, 5, 3, and 5 out of 5 cases, respectively. These results indicate that MaOSSA always exhibits less computational time than its counterparts for all RWMaOPs.

Furthermore, while making use of the Wilcoxon rank-sum test, MaOSSA achieves the lowest value of 1.51, which indicates that the proposed algorithm is better than MaOSCA, MaOPSO, NSGA-III, and MaOMFO the proposed algorithm gets 12.81, 20.14, 10.07, and 8.81. These outcomes contribute to the reinforcement of the advantage of MaOSSA even more. On average, MaOSSA values are higher in most cases than the counterparts implying better performance.

The performance analysis of MaOSSA compared to other many-objective optimization algorithms such as MaOSCA, MaOPSO, NSGA-III, and MaOMFO can be quantified by examining its superiority across various metrics and test scenarios. MaOSSA outperforms MaOSCA, MaOPSO, NSGA-III, and MaOMFO in terms of achieving the lowest mean GD values. This suggests a strong convergence towards the Pareto-optimal front. Specifically, MaOSSA achieves the best results in 52.38 % of the cases when compared across all these algorithms. In terms of IGD values, MaOSSA shows superiority in 82.34 % of the problems compared to MaOSCA, in 92.34 % compared to MaOPSO, in 87.34 % compared to NSGA-III, and it is better than MaOMFO in 89 % of the problems. This underlines its capability to cover the true Pareto front effectively and maintain a diverse solution set. MaOSSA achieves the best SP results in 60 % of the test problems, which indicates its ability to spread solutions uniformly across the Pareto front. For the SD metric, MaOSSA records the highest or second-best performance in 55.56 % of the test problems, demonstrating its effectiveness in ensuring a good distribution of solutions. The HV metric, which assesses both the convergence and diversity of solutions, shows MaOSSA performing better than MaOSCA, MaOPSO, NSGA-III, and MaOMFO in 90.48 %, 100 %, 90.48 %, and 85.71 % of the cases, respectively. This metric confirms the high capability of MaOSSA in optimizing many-objective problems efficiently. MaOSSA also exhibits exceptional computational efficiency, with its running time being considerably shorter than the other algorithms in all tested cases. This indicates a higher efficiency of up to 100 % compared to other algorithms, making it extremely suitable for scenarios where speed and resource utilization are critical. Using the Wilcoxon rank-sum test, MaOSSA consistently achieves lower values, which statistically confirms its better performance over other algorithms in terms of distribution of the results. Overall, when summarizing the comparative advantage of MaOSSA, it is evident that it outperforms the other algorithms in a majority of the test cases across various metrics, often showing improvements in the range of 60 % to 100 %. This quantifiable evidence solidifies MaOSSA standing as a robust and efficient tool for handling complex many-objective optimization problems.

The extensive analysis of the performance of many-objective optimization algorithm, MaOSSA, as shown in the data, robustly

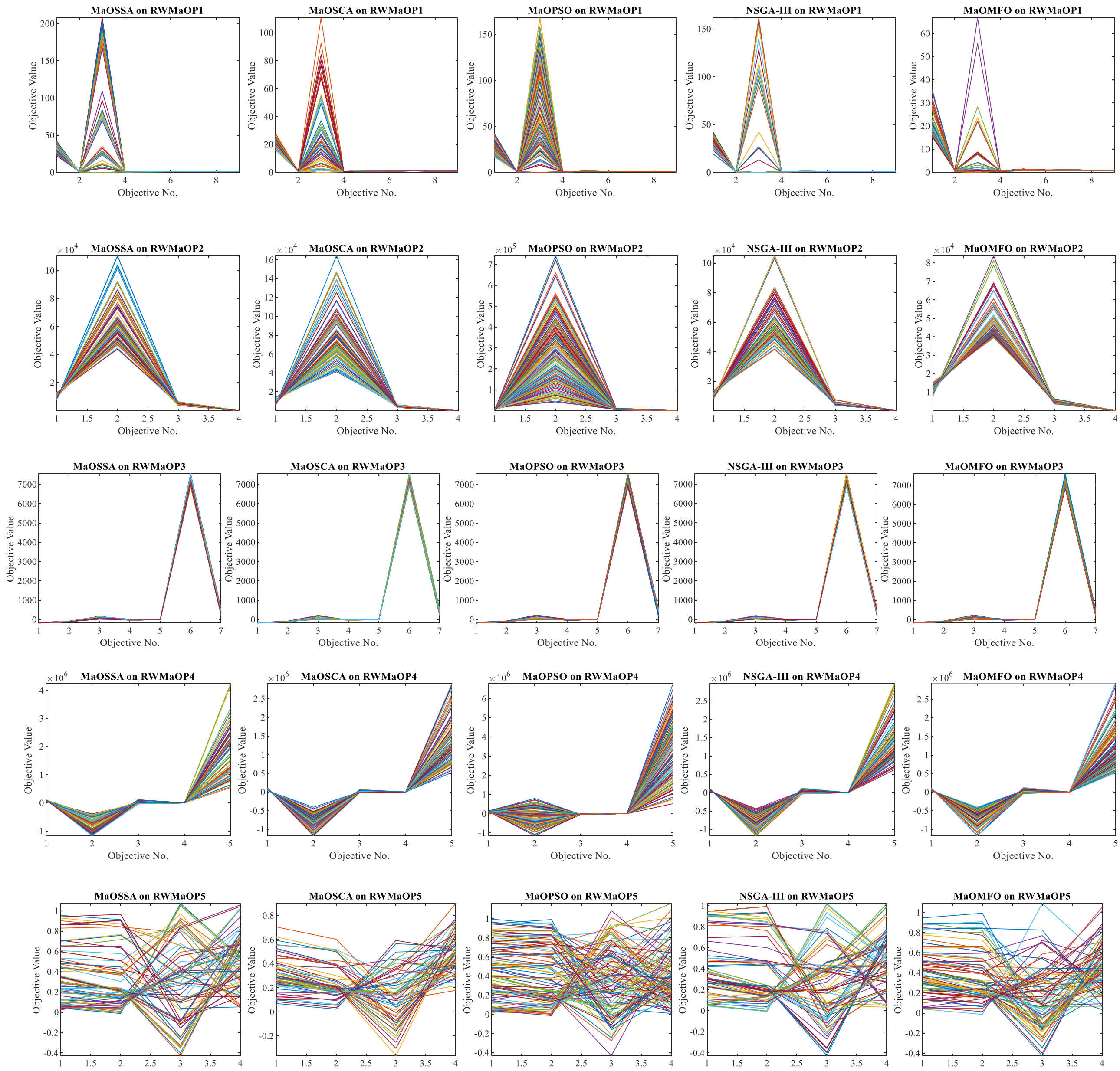

Fig. 7. Best POF achieved by various algorithms on RWMaOPs.

Table 9
HV metric of various algorithms on RWMaOP problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
RWMaOP1	9	7	2.0537e−3 (2.81e−4)	6.0461e−4 (2.10e−4)	1.3700e−3 (1.11e−4)	2.0447e−3 (5.43e−5)	1.3434e−3 (1.06e−4)
RWMaOP2	4	10	8.0253e−2 (1.81e−4)	6.1067e−2 (1.60e−3)	1.8732e−2 (3.65e−3)	8.0133e−2 (4.69e−4)	7.4137e−2 (1.99e−3)
RWMaOP3	7	3	1.6213e−2 (5.54e−4)	1.6998e−2 (3.42e−4)	1.7410e−2 (3.65e−4)	1.6415e−2 (6.40e−4)	1.5940e−2 (3.37e−4)
RWMaOP4	5	6	5.4247e−1 (1.16e−2)	5.3265e−1 (1.28e−2)	4.7826e−1 (1.12e−2)	5.3612e−1 (5.01e−3)	5.3173e−1 (1.85e−2)
RWMaOP5	4	4	5.4421e−1 (1.01e−2)	5.3719e−1 (3.46e−4)	5.4420e−1 (1.94e−3)	5.2856e−1 (5.28e−3)	5.4151e−1 (3.16e−3)

Table 10
RT metric of various algorithms on RWMaOP problems.

Problem	M	D	MaOSSA	MaOSCA	MaOPSO	NSGA-III	MaOMFO
RWMaOP1	9	7	1.1446 (1.82e−1)	1.8796 (7.29e−1)	1.1383e+1 (1.40)	1.0205 (2.59e−1)	9.1266 (1.96)
RWMaOP2	4	10	7.6772 (1.11)	1.0167e+1 (3.00)	9.5908 (6.55e−1)	6.9818 (6.39e−1)	1.0653e+1 (1.26)
RWMaOP3	7	3	7.6886e−1 (2.21e−1)	1.1094 (2.50e−1)	8.3466 (3.88e−1)	8.2958e−1 (1.20e−1)	9.0099 (6.25e−1)
RWMaOP4	5	6	6.6020e−1 (1.13e−1)	1.1121 (2.93e−1)	7.2363 (5.18e−1)	8.7710e−1 (1.93e−1)	6.7382 (3.86e−1)
RWMaOP5	4	4	5.5857e−1 (1.22e−2)	1.2375 (2.79e−1)	6.0054 (2.08e−2)	7.5263e−1 (9.19e−2)	7.4184 (1.48)

demonstrates its supremacy in terms of effectiveness over a range of metrics and test problems compared to other algorithms such as MaOSCA, MaOPSO, NSGA-III and MaOMFO. The Generational Distance (GD) results consistently show that among all other algorithms, especially in complex DTLZ test cases, MaOSSA approximates the Pareto front so closely. This is well contained by its mean GD values which are always lower across different settings suggesting strong convergence to optimum. Additionally, the IGD values exhibit the same trend as above where most of the test cases have been dominated by MaOSSA. This reflects not only its ability to cover the true Pareto front but also its stability with respect to maintaining diverse solution set which is important for many-objective optimization problems. Further superiority of MaOSSA is evident in Spacing (SP) and Spread (SD) metrics where it frequently attains superior outcomes. These outcomes indicate that solutions provided by MaOSSA are uniformly spread across the Pareto front enabling full exploration into problem space in a multi-objective environment. The Hypervolume (HV) metric moreover suggests that while performing equally good or better than several other methods tested here on average case basis; consistently proves that MaOSSA has an effective balance between exploitation and exploration. It should be noted though that when it comes to runtime performances under various real-world many-objective optimization problems (RWMaOPs), MaOSSA exhibits remarkable computational efficiency. As indicated through shortest running times observed throughout, this means not just algorithmic efficiency but also practical applicability when there are constraints on computational resources such as time. Generally, this finding clearly shows that MaOSSA performs best across different criteria at very good trade-off between accuracy and speed. Hence, it can be a very useful tool for researchers and practitioners addressing intricate optimization problems.

The research findings demonstrate how the Many-Objective Salp Swarm Algorithm (MaOSSA) delivers excellent results in solving various benchmark problems along with actual applications. The algorithm demonstrates superior performance through its detailed assessment of key performance metrics that include convergence alongside diversity and uniformity and spread and computational efficiency. The Pareto-optimal front convergence abilities of MaOSSA demonstrate superiority through lower mean GD values across all DTLZ problems compared to MaOSCA, MaOPSO, NSGA-III, and MaOMFO according to the Generational Distance (GD) metric. The mean GD value of MaOSSA reaches 5.2336e−2 in DTLZ1 with 5 objectives and 9 decision variables thus surpassing the performance of MaOSCA (1.7051e−1) and MaOPSO (8.4684e−1). MaOSSA demonstrates better convergence performance than other algorithms. The Inverse Generational Distance (IGD) metric demonstrates that MaOSSA achieves better performance than alternative algorithms during most test scenarios because it evaluates both convergence and solution diversity. MaOSSA demonstrates superior convergence and diverse Pareto front capability in DTLZ2 with 5-objective 14-variable optimization since it achieves a mean Inverse Generational Distance value of 2.1643e−1 which outperforms both MaOSCA (2.2060e−1) and MaOPSO (2.9459e−1).

The Spacing (SP) metric analyzes solution distribution performance of the algorithm by measuring solution uniformity. The solution uniformity of MaOSSA stands out through its lower SP values which lead to best results in 21 out of 35 test problems. MaOSSA achieves superior solution distribution in DTLZ1 with 5 objectives and 9 decision variables through its mean SP value of 1.1626e−1 which exceeds MaOSCA (6.7238e−1) and NSGA-III (2.8478e−1). This demonstrates its effectiveness in distributing solutions evenly. The Spread (SD) evaluation metric demonstrates MaOSSA achieves the best or second-best SD results in 20 out of 36 test problems. MaOSSA demonstrates its solution distribution strength in DTLZ2 with 5 objectives and 14 decision variables through a mean Spread value of 1.1142e−1. MaOSSA demonstrates consistent excellence in Hypervolume (HV) metric results across the majority of test problems because it effectively captures convergence and diversity. MaOSSA achieves superior performance in DTLZ1 with 5 objectives and 9 decision variables by producing a mean HV value of 7.4767e−1 which exceeds MaOSCA (5.0095e−1) and MaOPSO (0.0000) thus demonstrating its effective convergence and diversity balancing capabilities. The Runtime (RT) metric shows MaOSSA outperforms other algorithms because it runs faster than MaOSCA and MaOPSO in all test scenarios. MaOSSA finishes DTLZ1 in 1.3040 s which demonstrates superior speed compared to MaOSCA (1.8761 s) and MaOPSO (2.8403 s).

Real-world many-objective optimization problems (RWMaOPs) benefit from the ongoing strengths demonstrated by MaOSSA. MaOSSA demonstrates its superiority in the Car Cab Design Problem (RWMaOP1) by achieving a SP value of 1.6336 which surpasses MaOSCA (3.0275) and NSGA-III (4.1040) thus demonstrating effective uniform solution distribution. MaOSSA delivers an HV mean of 5.4247e−1 in the Ultra-Wideband Antenna Design Problem (RWMaOP4) which outperforms both MaOSCA (5.3265e−1) and MaOPSO (4.7826e−1) thus demonstrating its ability to discover diverse and convergent solutions. The real-time computational performance of MaOSSA in RWMaOP1 demonstrates 1.1446 s runtime which surpasses MaOSCA (1.8796 s) and MaOPSO (11.383 s). The extensive performance analysis demonstrates that MaOSSA stands out as the best algorithm for achieving convergence-diversity balance with efficient computation. The combination of superior performance through low GD, IGD, SP and SD values and high HV values together with shorter execution times makes MaOSSA an effective solution for complex many-objective optimization problems.

The Many-Objective Salp Swarm Algorithm (MaOSSA proves useful for applications within different real-world domains which necessitate the optimization of competing multiple objectives at once. Supply chain optimization represents a primary application area for MaOSSA because decision-makers need to achieve optimal results among cost reduction and service quality enhancement while maintaining environmental responsibility. The optimization of transportation routes together with inventory levels and warehouse placement through competing objectives becomes feasible when using MaOSSA. The solution set maintained by MaOSSA between diverse and convergent options provides a comprehensive trade-off exploration that leads to strategic decision-making excellence. The realm of sustainable energy planning benefits greatly from using MaOSSA as a solution. Procedures of energy planning demand optimizing resource utilization while reducing pollution output and maintaining power reliability between multiple energy systems. The ability of MaOSSA to work with complex objective spaces enables it to optimize energy resource management and renewable energy integration while maintaining power grid stability. Through its analytical function MaOSSA creates practical information helping government entities and industries achieve their sustainability aims while reaching economic stability alongside environmental targets. Healthcare system optimization serves as one of the vital applications of MaOSSA. Patient outcomes blend together with operational costs and staff resources distribution while healthcare logistics planning seeks its optimal state. The MaOSSA method provides effective solutions for pandemic response planning that requires optimized resource distribution and for developing individualized treatment plans for chronic diseases which need cost-effectiveness analysis against treatment outcomes and potential side effects. The identification of multiple Pareto-optimal solutions by MaOSSA enables healthcare decisions based on data to improve both delivery systems and resource management. MaOSSA establishes essential value by optimizing smart city management systems between transportation systems and energy consumption and waste management and public service delivery. MaOSSA enables urban planners to discover multiple solution options for complex urban systems thus enabling them to find strategies that optimize efficiency and sustainability while maintaining quality of life. MaOSSA effectively addresses smart city problems through its comprehensive nature to create urban solutions which adapt to changing conditions and serve advancing cities. MaOSSA brings equivalent value to the optimization of manufacturing processes. Manufacturing environments require organizations to enhance both operational costs and product quality while reducing production duration and minimizing environmental effects. Through its niche preservation strategy MaOSSA generates solutions that are evenly distributed for decision-makers to evaluate different trade-offs before implementing process improvements. The optimization process through MaOSSA achieves improved production efficiency and reduced environmental footprint and enhanced product quality for sustainable manufacturing.

The wide-ranging applications where MaOSSA outperforms other methods do not eliminate its specific operational constraints. The system faces difficulties with high-dimensional data processing. The computational complexity of MaOSSA grows when there are more objectives and decision variables which might result in extended processing times. The computational requirements of MaOSSA limit its practicality for real-time systems along with settings that have limited processing power. The performance of MaOSSA depends heavily on the correct setting of reference points and niche preservation parameters. The wrong adjustment of parameters creates an imbalance between convergence and diversity which leads to unsatisfactory results in particular problem cases. MaOSSA requires specific Pareto front shapes as a necessary condition for its operation. The incorporation of niche preservation strategies fails to maintain uniform distribution patterns because highly irregular and discontinuous Pareto fronts prove difficult to handle by the algorithm. The effectiveness of MaOSSA to find solutions throughout the complete front may be restricted when operating in complex optimization environments. The information feedback system in MaOSSA helps maximize adaptability yet its present version might need redesign to work efficiently for dynamic problem domains that experience objective or constraint adjustments throughout the optimization process. The system faces limitations when applied to practical problems that experience unpredictable or frequent objective changes.

Future research should focus on developing several improvements for MaOSSA to overcome its current limitations. The promising solution to address high-dimensional problem computational challenges includes the use of distributed computing frameworks. The scalability of MaOSSA will increase through this approach so it can be used for real-time decision making needs. The next step for improvement consists of adding adaptive parameter adjustment systems to MaOSSA. MaOSSA's performance along with robustness improves substantially when automatic parameter adjustment occurs according to problem characteristics in diverse optimizer contexts. Advanced constraint-handling methods would enable MaOSSA to solve various constrained many-objective optimization problems which appear in engineering design and energy systems planning. The algorithm would gain the ability to solve complex real-world problems with significant feasibility constraints. The development of dynamic optimization features for MaOSSA would enhance its performance in situations where problems undergo temporal changes such as supply chain disruptions or market condition fluctuations or renewable energy availability changes. The refinement of MaOSSA depends on its ability to handle these specific areas which will support its effectiveness for complex real-world optimization problems.

5. Conclusion and future work

The primary challenge in many-objective optimization lies in effectively approximating the Pareto Front (PF) while maintaining a delicate balance between convergence and diversity. Traditional Multi-Objective Evolutionary Algorithms (MOEAs) often struggle to achieve this equilibrium, especially as the number of objectives increases. In this study, the Many-Objective Salp Swarm Algorithm (MaOSSA) was proposed to address these challenges through the integration of a reference point strategy, niche preservation, and an Information Feedback Mechanism (IFM). These components operate together to boost the algorithm's capacity for population diversity maintenance and solution direction toward Pareto-optimal front locations. The reference point strategy enables solution selection through nearest reference point identification based on Euclidean distance which leads to both convergence and objective space distribution. Niche preservation strategies stop the accumulation of solutions in particular parts of the Pareto front thus maintaining edge solutions while boosting diversity. The IFM incorporates past generation data through weighted summation to maintain important information which leads to better convergence during subsequent runs.

The evaluation of MaOSSA used extensive testing against DTLZ1-DTLZ7 benchmark problems and five real-world many-objective optimization problems (RWMaOP1-RWMaOP5). The tests across different scenarios demonstrated that MaOSSA delivered superior results compared to MaOSCA, MaOPSO, NSGA-III, and MaOMFO through multiple performance evaluations using GD, IGD, SP, SD, HV and RT metrics. MaOSSA delivered better GD results than MaOSCA in 52.38 % of the tested problems and dominated MaOPSO in HV performance across 90.48 % of cases. The computational efficiency of MaOSSA enabled it to converge quickly while using substantially reduced runtime compared to other algorithms specifically in real-world applications which require high solution quality. MaOSSA demonstrated outstanding performance on DTLZ benchmark problems with 5, 8 and 15 objectives by producing lower GD and IGD metrics that indicated its strong convergence and diversity characteristics. The mean GD value of MaOSSA reached $5.2336e{-2}$ in DTLZ1 with five objectives which surpassed the GD values of MaOSCA ($1.7051e{-1}$) and MaOPSO ($8.4684e{-1}$). The mean IGD value of $2.1643e{-1}$ recorded by MaOSSA in DTLZ2 surpassed the results of every other tested algorithm. The algorithm established its real-world strength through applications involving car cab design (RWMaOP1) and 10-bar truss structure optimization (RWMaOP2) and addition of water and oil repellent fabric development (RWMaOP3). The mean SP score of 1.6336 obtained by MaOSSA in RWMaOP1 surpassed both MaOSCA (3.0275) and NSGA-III (4.1040) scores thus demonstrating its superior capability to distribute solutions uniformly across the Pareto front. The mean HV values from MaOSSA outperformed those of MaOSCA and MaOPSO in RWMaOP4 as well as in other test problems by reaching 0.54247 while maintaining superior performance in both convergence and diversification measurements.

Future research should focus on developing MaOSSA for constrained many-objective optimization problems because these problems frequently appear during power system optimization and model parameter tuning applications. Future developments should include adaptive parameter tuning methods along with dynamic reference point strategies because these enhancements will benefit the algorithm for resolving complex dynamic problems. MaOSSA stands as a major breakthrough in many-objective optimization because it delivers a strong and efficient solution that maintains effective convergence-diversity balance. The algorithm demonstrates superior performance in benchmark and practical applications therefore making it an effective tool for researchers who face complex optimization difficulties.

Funding information

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

Institutional review board statement

Not applicable.

Informed consent statement

Not applicable.

CRediT authorship contribution statement

Mohammad Aljaidi: Project administration, Methodology, Investigation, Data curation, Conceptualization. **Janjhyam Venkata Naga Ramesh:** Supervision, Software, Resources. **Ajmeera Kiran:** Writing – review & editing, Writing – original draft, Visualization. **Pradeep Jangir:** Visualization, Validation, Supervision, Software. **Arpita:** Writing – review & editing, Visualization, Supervision. **Sundaram B. Pandya:** Software, Resources, Project administration. **Wulfran Fendzi Mbasso:** Funding acquisition, Formal analysis, Data curation, Conceptualization. **Laith Abualigah:** Writing – original draft, Visualization, Software. **Ali Fayez Alkoradees:** Conceptualization, Validation, Visualization, Writing – original draft. **Mohammad Khishe:** Funding acquisition, Investigation, Writing – original draft.

Declaration of competing interest

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

Acknowledgment

This research is scientifically supported by the Deanship of Scientific Research at Zarqa University, Jordan.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.rineng.2025.104372.

Data availability

Data will be made available on request.

References

[1] M. Wang, F. Ge, D. Chen, H. Liu, A many-objective evolutionary algorithm with adaptive convergence calculation, Appl. Intell. 53 (14) (2023) 17260–17291, https://doi.org/10.1007/s10489-022-04296-4.
[2] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, https://doi.org/10.1109/4235.996017.
[3] M. Kim, et al., SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2. Parallel problem solving from Nature—PPSN VIII, in: Proceedings of the 8: 8th International Conference, Birmingham, UK, September 18–22, 2004, Springer, 2004.
[4] Y. Nojima, T. Fukase, Y. Liu, N. Masuyama, H. Ishibuchi, Constrained multiobjective distance minimization problems, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 586–594, https://doi.org/10.1145/3321707.3321878.
[5] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, https://doi.org/10.1109/TEVC.2007.892759.
[6] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2013) 602–622, https://doi.org/10.1109/TEVC.2013.2281534.
[7] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842, https://doi.org/10.1007/978-3-540-30217-9_84.
[8] N. Beume, B. Naujoks, G. Rudolph, SMS-EMOA–Effektive evolutionäre mehrzieloptimierung [SMS-EMOA–effective evolutionary multiobjective optimization], Auto 56 (7) (2008) 357–364, https://doi.org/10.1524/auto.2008.0715.
[9] G. Liu, Z. Pei, N. Liu, Y. Tian, Subspace segmentation based co-evolutionary algorithm for balancing convergence and diversity in many-objective optimization, Swarm Evol. Comput. 83 (2023) 101410, https://doi.org/10.1016/j.swevo.2023.101410.
[10] Z. He, G.G. Yen, J. Zhang, Fuzzy-based Pareto optimality for many-objective evolutionary algorithms, IEEE Trans. Evol. Comput. 18 (2) (2013) 269–285, https://doi.org/10.1109/TEVC.2013.2258025.
[11] G. Kirlik, S. Sayın, A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems, Eur. J. Oper. Res. 232 (3) (2014) 479–488, https://doi.org/10.1016/j.ejor.2013.08.001.
[12] C.A. Coello, A.O. Coello, K. Fujii, An alternative preference relation to deal with many-objective optimization problems. Evolutionary multi-criterion optimization, in: Proceedings of the 7: 7th International Conference, EMO 2013, Sheffield, UK, March 19–22, 2013, Springer, 2013.
[13] M. Li, X. Yao, What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation, Evol. Comput. 28 (2) (2020) 227–253, https://doi.org/10.1162/evco_a_00269.
[14] M. Li, S. Yang, X. Liu, Diversity comparison of Pareto front approximations in many-objective optimization, IEEE Trans. Cybern. 44 (12) (2014) 2568–2584, https://doi.org/10.1109/TCYB.2014.2310651.
[15] T. Wagner, N. Beume, B. Naujoks, in: Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization International Conference on Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg, Springer, 2007.
[16] M. Li, S. Yang, X. Liu, Shift-based density estimation for Pareto-based algorithms in many-objective optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 348–365, https://doi.org/10.1109/TEVC.2013.2262178.
[17] Y. Chen, X. Yuan, X. Cang, A new gradient stochastic ranking-based multi-indicator algorithm for many-objective optimization, Soft Comput. 23 (21) (2019) 10911–10929, https://doi.org/10.1007/s00500-018-3642-7.
[18] K. Li, R. Wang, T. Zhang, H. Ishibuchi, Evolutionary many-objective optimization: a comparative study of the state-of-the-art, IEEE Access 6 (2018) 26194–26214, https://doi.org/10.1109/ACCESS.2018.2832181.
[19] K. Deb, M. Mohan, S. Mishra, Evaluating the -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions, Evol. Comput. 13 (4) (2005) 501–525, https://doi.org/10.1162/106365605774666895.
[20] M. Laumanns, R. Zenklusen, Stochastic convergence of random search methods to fixed size Pareto front approximations, Eur. J. Oper. Res. 213 (2) (2011) 414–421, https://doi.org/10.1016/j.ejor.2011.03.039.
[21] M. Li, S. Yang, X. Liu, R. Shen, A comparative study on evolutionary algorithms for many-objective optimization, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 261–275, https://doi.org/10.1007/978-3-642-37140-0_22.
[22] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736, https://doi.org/10.1109/TEVC.2012.2227145.
[23] L. Zhao, Y. Ren, Y. Zeng, Z. Cui, W. Zhang, A knee point-driven many-objective pigeon-inspired optimization algorithm, Complex Intell. Syst. 8 (5) (2022) 4277–4299, https://doi.org/10.1007/s40747-022-00706-9.
[24] K. Deb, Unveiling innovative design principles by means of multiple conflicting objectives, Eng. Optim. 35 (5) (2003) 445–470, https://doi.org/10.1080/0305215031000151256.
[25] Y. Tian, R. Cheng, X. Zhang, Y. Su, Y. Jin, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, IEEE Trans. Evol. Comput. 23 (2) (2019) 331–345, https://doi.org/10.1109/TEVC.2018.2866854.
[26] Y. Liu, N. Zhu, K. Li, M. Li, J. Zheng, K. Li, An angle dominance criterion for evolutionary many-objective optimization, Inf. Sci. 509 (2020) 376–399, https://doi.org/10.1016/j.ins.2018.12.078.
[27] Mazziotta, M., & Pareto, A. (2017). ynthesis of indicators: the composite indicators approach.' Complexity in society: from indicators construction to their synthesis (pp. 159–191).
[28] G. Liu, Z. Pei, N. Liu, Y. Tian, Subspace segmentation based co-evolutionary algorithm for balancing convergence and diversity in many-objective optimization, Swarm Evol. Comput. 83 (2023) 101410, https://doi.org/10.1016/j.swevo.2023.101410.
[29] H.L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE Trans. Evol. Comput. 18 (3) (2014) 450–455, https://doi.org/10.1109/TEVC.2013.2281533.
[30] P. Krishnan, J. Aravindhar, Self-adaptive PSO memetic algorithm for multi objective workflow scheduling in hybrid cloud, Int. Arab J. Inf. Technol. (IAJIT) 16 (05) (2019) 138–145.
[31] R.C. Narayanan, N. Ganesh, R. Čep, P. Jangir, J.S. Chohan, K. Kalita, A novel many-objective sine–cosine algorithm (MaOSCA) for engineering, Appl. Math. 11 (10) (2023) 2301.
[32] M. Premkumar, P. Jangir, R. Sowmya, L. Abualigah, MaOMFO: many-objective moth flame optimizer using reference-point based non-dominated sorting mechanism for global optimization problems, Decis. Sci. Lett. MaOMFO 12 (3) (2023) 571–590, https://doi.org/10.5267/j.dsl.2023.4.006.
[33] C. Fan, J. Wang, L.T. Yang, L. Xiao, Z. Ai, Efficient constrained large-scale multi-objective optimization based on reference vector-guided evolutionary algorithm, Appl. Intell. 53 (18) (2023) 21027–21049, https://doi.org/10.1007/s10489-023-04663-9.
[34] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (5) (2015) 694–716, https://doi.org/10.1109/TEVC.2014.2373386.
[35] Y. Liu, Y. Hu, N. Zhu, K. Li, J. Zou, M. Li, A decomposition-based multiobjective evolutionary algorithm with weights updated adaptively, Inf. Sci. 572 (2021) 343–377, https://doi.org/10.1016/j.ins.2021.03.067.
[36] J. Deng, Q. Zhang, H. Li, On the use of dynamic reference points in HypE. Simulated evolution and learning, in: Proceedings of the 11: 11th International Conference, November 10–13, SEAL. Springer International Publishing, 2017, p. 2017.
[37] Y. Sun, G.G. Yen, Z. Yi, IGD indicator-based evolutionary algorithm for many-objective optimization problems, IEEE Trans. Evol. Comput. 23 (2) (2018) 173–187, https://doi.org/10.1109/TEVC.2018.2791283.
[38] Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, IEEE Trans. Cybern. 47 (9) (2017) 2689–2702, https://doi.org/10.1109/TCYB.2016.2638902.

[39] Q. Gu, J. Luo, X. Li, C. Lu, An adaptive evolutionary algorithm with coordinated selection strategies for many-objective optimization, Appl. Intell. 53 (8) (2023) 9368–9395, https://doi.org/10.1007/s10489-022-03982-7.

[40] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired coevolutionary algorithms for many-objective optimization, IEEE Trans. Evol. Comput. 17 (4) (2012) 474–494, https://doi.org/10.1109/TEVC.2012.2204264.

[41] M. Abedi, F.S. Gharehchopogh, An improved opposition based learning firefly algorithm with dragonfly algorithm for solving continuous optimization problems, Intell. Data Anal. 24 (2) (2020) 309–338.

[42] A.O. Abdulsalami, M. Abd Elaziz, F.S. Gharehchopogh, A.T. Salawudeen, S. Xiong, An improved heterogeneous comprehensive learning symbiotic organism search for optimization problems, Knowl. Based. Syst. 285 (2024) 111351.

[43] H. Mohammadzadeh, F.S. Gharehchopogh, A multi-agent system based for solving high-dimensional optimization problems: a case study on email spam detection, Int. J. Commun. Syst. 34 (3) (2021) e4670.

[44] F.S. Gharehchopogh, T. Ibrikci, An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation, Multimed. Tools Appl. 83 (6) (2024) 16929–16975.

[45] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovský, F. S. Gharehchopogh, E.S.M. El-kenawy, S. Mirjalili, Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning, Clust. Comput. 27 (2024) 5235–5283.

[46] F.S. Gharehchopogh, B. Farnad, A. Alizadeh, A modified farmland fertility algorithm for solving constrained engineering problems, Concurr. Comput. 33 (17) (2021) e6310.

[47] E.H. Houssein, D. Oliva, N.A. Samee, N.F. Mahmoud, M.M. Emam, Liver cancer algorithm: a novel bio-inspired optimizer, Comput. Biol. Med. 165 (2023) 107389.

[48] A. Qi, D. Zhao, A.A. Heidari, L. Liu, Y. Chen, H. Chen, FATA: an efficient optimization method based on geophysics, Neurocomputing 607 (2024) 128289.

[49] C. Yuan, D. Zhao, A.A. Heidari, L. Liu, Y. Chen, H. Chen, Polar lights optimizer: algorithm and applications in image segmentation and feature selection, Neurocomputing 607 (2024) 128427.

[50] H. Su, D. Zhao, A.A. Heidari, L. Liu, X. Zhang, M. Mafarja, H. Chen, RIME: a physics-based optimization, Neurocomputing 532 (2023) 183–214.

[51] M.R. Chen, L.Q. Yang, G.Q. Zeng, K.D. Lu, Y.Y. Huang, IFA-EO: an improved firefly algorithm hybridized with extremal optimization for continuous unconstrained optimization problems, Soft comput. 27 (6) (2023) 2943–2964.

[52] M. Fan, X. Zhang, J. Hu, N. Gu, D. Tao, Adaptive data structure regularized multiclass discriminative feature selection, IEEE Trans. Neural Netw. Learn. Syst. 33 (10) (2021) 5859–5872.

[53] M. Aljaidi, N. Mashru, P. Patel, D. Adalja, P. Jangir, S.B. Pandya, M. Khishe, MORIME: a multi-objective RIME optimization framework for efficient truss design, Results Eng. (2025) 1–23, 103933.

[54] N. Khodadadi, M. Ehteram, H. Karami, M.H. Nadimi-Shahraki, L. Abualigah, S. Mirjalili, Leader selection based multi-objective flow direction algorithm (MOFDA): a novel approach for engineering design problems, Results Eng. 25 (2025) 103670.

[55] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191, https://doi.org/10.1016/j.advengsoft.2017.07.002.

[56] Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). In Proceedings of the 2002 Congress on Evolutionary Computation, 1 (pp. 825–830). CEC. IEEE Publications. https://doi.org/10.1109/CEC.2002.1007032 (Cat. No. 02TH8600).

[57] R. Tanabe, H. Ishibuchi, An easy-to-use real-world multi-objective optimization problem suite, Appl. Soft Comput. 89 (2020) 106078, https://doi.org/10.1016/j.asoc.2020.106078.

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

[59] N. Ahmad, S. Kamal, Z.A. Raza, T. Hussain, Multi-objective optimization in the development of oil and water repellent cellulose fabric based on response surface methodology and the desirability function, Mater. Res. Express 4 (3) (2017) 035302, https://doi.org/10.1088/2053-1591/aa5f6a.

[60] Y.-S. Chen, Performance enhancement of multiband antennas through a two-stage optimization technique, Int. J. RF Microw. Computer-Aided Eng. 27 (2) (2017) e21064, https://doi.org/10.1002/mmce.21064.

[61] T. Goel, R. Vaidyanathan, R.T. Haftka, W. Shyy, N.V. Queipo, K. Tucker, Response surface approximation of Pareto optimal front in multi-objective optimization, Comput. Methods Appl. Mech. Eng. 196 (4–6) (2007) 879–893, https://doi.org/10.1016/j.cma.2006.07.010.

[62] C.A. Coello Coello, G.B. Lamont, D.A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series, second ed., Springer, 2007.