Decision Science Letters 12 (2023) 571–590

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

MaOMFO: Many-objective moth flame optimizer using reference-point based non-dominated sorting mechanism for global optimization problems

M. Premkumar[a*], Pradeep Jangir[b], R. Sowmya[c], and Laith Abualigah[d,e,f,g]

[a]*Department of Electrical and Electronics Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka 560078, India*
[b]*Rajasthan Rajya Vidyut Prasaran Nigam Ltd., Sikar, 332025, Rajasthan, India*
[c]*Department of Electrical and Electronics Engineering, National Institute of Technology, Tiruchirapalli, Tamil Nadu 620015, India*
[d]*Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al al-Bayt University, Mafraq 25113, Jordan*
[e]*Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan*
[f]*MEU Research Unit, Middle East University, Amman, Jordan*
[g]*School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia*

C H R O N I C L E

Article history:
Received: November 21, 2022
Received in revised format: December 28, 2022
Accepted: April 16, 2023
Available online: April 16, 2023

Keywords:
Evolutionary
Many-objective algorithm
Moth flame optimizer
Non-dominated sorting
Optimization algorithm

A B S T R A C T

Many-objective optimization (MaO) deals with a large number of conflicting objectives in optimization problems to acquire a reliable set of appropriate non-dominated solutions near the true Pareto front, and for the same, a unique mechanism is essential. Numerous papers have reported multi-objective evolutionary algorithms to explain the absence of convergence and diversity variety in many-objective optimization problems. One of the most encouraging methodologies utilizes many reference points to segregate the solutions and guide the search procedure. The above-said methodology is integrated into the basic version of the Moth Flame Optimization (MFO) algorithm for the first time in this paper. The proposed Many-Objective Moth Flame Optimization (MaOMFO) utilizes a set of reference points progressively decided by the hunt procedure of the moth flame. It permits the calculation to combine with the Pareto front yet synchronize the decent variety of the Pareto front. MaOMFO is employed to solve a wide range of unconstrained and constrained benchmark functions and compared with other competitive algorithms, such as non-dominated sorting genetic algorithm, multi-objective evolutionary algorithm based on dominance and decomposition, and novel multi-objective particle swarm optimization using different performance metrics. The results demonstrate the superiority of the algorithm as a new many-objective algorithm for complex many-objective optimization problems.

1. Introduction

Optimization problems can be found in a wide range of fields, in which the objective is to find optimal values for the unknowns of a given problem to minimize or maximize a set of objectives. This expectation is scientifically an advancement problem. There are many streamlining issues with different goals, and regular logical inconsistencies exist among these objectives (Abbasi et al., 2021; Brest et al., 2017). It is frequently challenging to locate the ideal arrangement that fulfils all the objectives simultaneously. In most real-time applications, multi-objective problems have many solutions as opposed to a single solution, and multi-objective enhancement calculations have pulled in an ever-increasing number of specialists' consideration. Generally, a problem with two to four objectives is referred to as a multi-objective problem (MOP), while problems with more than four objectives are called many-objective problems (MaOPs) (Behmanesh et al., 2021; Liu et al.,

* Corresponding author.
E-mail address: mprem.me@gmail.com (M. Premkumar)

doi: 10.5267/dsl.2023.4.006

2017; Premkumar, Jangir, Sowmya, et al., 2021). Currently, strategies for tackling improvement issues are generally separated into two classes. One is the traditional enhancement calculations, including the angle plummet technique, Newton strategy and semi-Newton strategy, and conjugate inclination technique. And the other one is heuristic calculations driven by an individual's involvement in caring for certain problems or the actions of certain living things on Earth. Traditional improvement calculations, as a rule, need to compute subsidiaries or differentials, so it is hard to apply to many complicated problems as a general rule (Guo & Wang, 2019; M. Li et al., 2013).

The algorithms, such as Genetic Algorithm (GA) (Fonseca & Fleming, 1993; Yang, 2021), Differential Evolutionary (DE) (Elsayed & Sarker, 2016; B. V. Kumar et al., 2022; Stanovov et al., 2020), Ant Colony Optimization (Dorigo et al., 1996; J. Zhou et al., 2017), Particle Swarm Optimizer (PSO) (Fan et al., 2017; Mousa et al., 2012; Sierra & Coello, 2005), simulated annealing (Kirkpatrick et al., 1983), Firefly optimizer (Johari et al., 2013), grey wolf optimizer (Mirjalili et al., 2014; Premkumar et al., 2022), dragonfly algorithm (Mirjalili, 2016), salp swarm optimizer (Premkumar, Kumar, Sowmya, & Pradeep, 2021), whale optimizer (Premkumar & Sumithira, 2018), slime mould algorithm (Premkumar, Sowmya, Jangir, Haes Alhelou, et al., 2021), arithmetic optimization algorithm (Premkumar, Jangir, Santhosh Kumar, et al., 2021), gradient-based optimizer (Premkumar, Jangir, & Sowmya, 2021), teaching-learning optimization algorithm (P. Jangir et al., 2023), heat transfer algorithm (S. Kumar et al., 2022), plasma generation algorithm (S. Kumar et al., 2021), Jaya algorithm (Venkata Rao, 2016), Rao algorithm (Rao, 2020), political optimizer (Premkumar et al., 2020), etc. can find the global solution for a given problem in view of one objective and few algorithm's multi-objective versions can solve the multi-objective problems (up to three objectives). Nevertheless, conflicting multiple objectives in real-world design problems must be optimized simultaneously (Deb, Pratap, et al., 2002; Deb & Jain, 2014). Evolutionary Multi-Objective Optimization (EMOO) is a branch of evolutionary algorithms that deals with such challenges, which handles the theory and applications of evolutionary multi-objective optimization algorithms (Duan et al., 2012; Jain & Deb, 2014; Ke et al., 2013). In any case, the performance of multi-objective algorithms significantly degrades when applied to many-objective problems. These calculations do not just acquire the first calculation structures and favourable circumstances yet additionally include new procedures, including data from the reference point, neighbour, or outstanding individual, to comprehend the inadequacies of the first calculations. The outcomes additionally demonstrate that these new techniques are powerful, so many researchers have started to utilize various systems to join the exploration of calculations. For instance, nondominated sorting genetic algorithm-Version III (NSGA-III) (Ibrahim et al., 2016; Vesikar et al., 2019), Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) (A. Zhou et al., 2012), and improved versions of NSGA-III (Bi & Wang, 2018; Wang et al., 2018; J. Zhang et al., 2019) are reported to solve different many-objective problems.

Though few algorithms are reported in the literature to solve many-objective problems, as stated in the no-free-lunch theorem, no single algorithm is suitable for all types of problems (Wolpert & Macready, 1997). Therefore, proposing a new or improvising the existing algorithm is necessary. This work proposes a many-objective version of the original Moth Flame Optimizer (MFO) algorithm (Mirjalili, 2015) called the Many-Objective MFO (MaOMFO) algorithm, and this algorithm is formulated using non-dominated sorting and reference point mechanisms.
The remainder of the paper is structured as follows. Section 2 describes the basic terminologies of many-objective optimization and related works. Section 3 introduces the basic version of the MFO optimizer and recommends the MaOMFO algorithm. Section 4 discusses the outcomes of various many-objective benchmark optimization problems. Lastly, concluding remarks are provided in Section 5. Also, the future direction for further investigation is discussed in section 5.

2. Related Works

This section introduces elementary descriptions of many-objective optimization problems, such as Pareto optimality, Pareto dominance, Pareto optimality set, and Pareto optimality front. In addition, numerous pasts and recently proposed many-objective algorithms are reviewed and commented on.

2.1. Many-Objective Optimization Definitions

In general, many-objective optimization problems are defined as a maximization/minimization of a given problem, and the problem is formulated as follows (Premkumar, Jangir, & Sowmya, 2021; Premkumar, Sowmya, Jangir, Haes Alhelou, et al., 2021):

$$\begin{aligned} &\frac{MinF}{Max}\ F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \ldots, f_o(\vec{x})\} \\ &\text{Subject to: } g_i(\vec{x}) \geq 0,\ i = 1,2,\ldots,m \\ &\qquad h_i(\vec{x}) = 0,\ i = 1,2,\ldots,p \\ &\qquad Lb_i \leq x_i \leq Ub_i,\ i = 1,2,\ldots,n \end{aligned} \tag{1}$$

where $[Lb_i, Ub_i]$ denotes the lower and upper boundaries of an i^{th} variable, h_i shows the i^{th} equality constraints, g_i denotes i^{th} inequality constraints, p signifies the number of constraint limits, m signifies the number of unconstraint limits, o signifies objective function counts, and n signifies the number of design parameters. Comparing two solutions when considering more than one objective is no longer possible using relational operators. A new operator called Pareto optimality is applied in this circumstance. The following are the key principles in this regard:

Def. 1. Pareto Optimality (Branke et al., 2001):

$$\nexists\, \vec{y} \in X \mid F(\vec{y}) \prec F(\vec{x}) \quad (2)$$

Def. 2. Pareto Dominance (Branke et al., 2001):

$$\forall i \in \{1,2,\dots,k\}: f_i(\vec{x}) \le f_i(\vec{y}) \quad \wedge \quad \exists i \in \{1,2,\dots,k\}: f_i(\vec{x}) < f_i(\vec{y}) \quad (3)$$

Def. 3. Pareto optimal set (Branke et al., 2001):

$$P_s \coloneqq \{x, y \in X \mid \exists F(\vec{y}) \succ F(\vec{x})\} \quad (4)$$

Def. 4. Pareto optimal front (Branke et al., 2001):

$$P_f \coloneqq \{F(\vec{x}) | \vec{x} \in P_s\} \quad (5)$$

The parametric space and objective space are illustrated in Fig. 1. In Fig. 1, both search spaces are compared, in which the circle denotes the best solution than the rectangle as it dominates the rectangle considering all objectives.

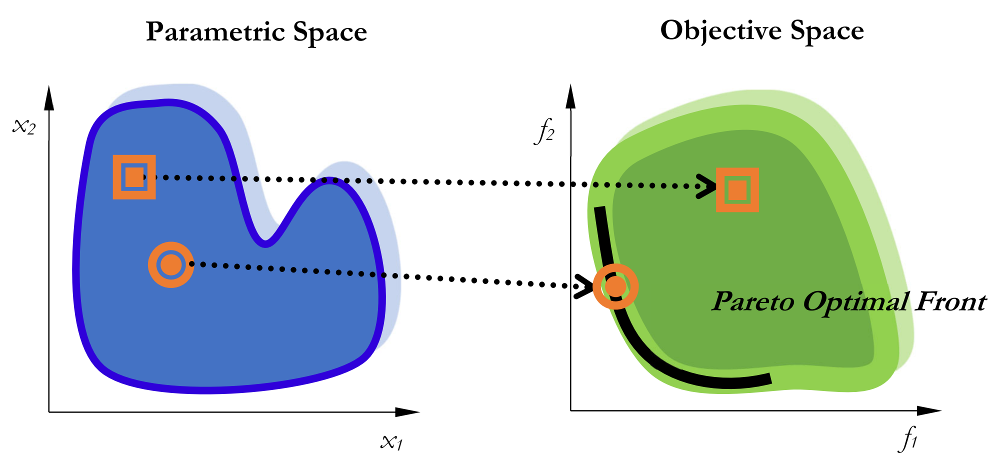


Fig. 1. Search space (both objective and parameter) in multiple-objective optimization

2.2. Related Works

In recent years, multi-objective evolutionary algorithms (MOEAs) have improved significantly since Schaffer first effectively utilized evolutionary algorithms to solve MOPs (Coello Coello et al., 2019; A. Zhou et al., 2011). Researchers are unhappy with the in-depth analysis of Multi-Objective Optimization (MOO) but pay even more importance to MaOPs. More many-objective evolutionary algorithms (MaOEAs) have been investigated in recent years to address practical many-objective problems (He & Yen, 2014; Liu et al., 2017). However, almost all of these methods only concentrate on multi/many-objective small-scale problems, but few concentrate on large-scale optimisation problems. In order to solve MOPs, Schaffer suggested a vector-assessed genetic algorithm (VEGA) that integrated GA with MOO for the first time. Since more and more researchers have introduced GAs to address MOPs (B. Li et al., 2015). Goldberg integrates the Pareto dominance method with EA to address MOPs for the first time. The principle of Goldberg and Schaffer influences many classical MOEAs, such as Niched Pareto Genetic Algorithm (NPGA) (Horn et al., 1994), NPGA2 (Erickson et al., 2001), NSGA (H. Li & Zhang, 2009), and NSGA-II (H. Li & Zhang, 2009).

Multi/Many objective optimizations first effectively utilized transformative calculation to multiple-objective problems by Schaffer; however, they give increasingly more consideration to many-objective problems. Lately, several MaOEAs have been developed and used to tackle many-objective problems and common-sense issues. In any case, the vast majority of these calculations centred around little scope multi/many-objective issues, and not many concentrated on huge scope improvement issues. Here we will quickly present some related EAs: NSGA-III (Vesikar et al., 2019), MOEA/D (Q. Zhang & Li, 2007), and Vector Genetic Algorithm (VEGA) (Schaffer, 1984).

NSGA-II is the most well-regarded algorithm for tackling MOPs, and numerous MOEAs depend on the concept of non-dominated sorting integrated into this algorithm. This algorithm sort individual in each population based on their domination level. Like GA, the solutions are then selected using inversely proportional to the domination level to undergo crossover and then mutation. Molina et al. (Luo et al., 2019) used data from the reference point, and afterwards, they joined the g-predominance with NSGA-II calculation to manage a few MOPs with muddled PFs better. To improve the decent variety of NSGA-II, Vachhani et al. (Vachhani et al., 2016) introduced an enhanced version of NSGA-II, in which another assorted variety of strategies followed agglomerative various levelled grouping techniques and outrageous arrangements protection to supplant the swarming separation technique. The test results indicated that the proposed technique improved the decent variety of unique NSGA-II on two-objective test cases. Dissimilar to the decay technique, a MOP can be broken into problems with multiple objectives and solved by MOEA/D. Qi et al. (Zheng et al., 2018) set forward a versatile weight vector modification way to deal with improving the presentation of MOEA/D just utilized another approach to introducing weight vector and a procedure which can adaptively re-arrange sub-problems and use an outer best population data to help include new sub-problems into the genuine inadequate area of the best PF. These calculations referenced above are extremely powerful in settling MOPs; however, a considerable number of them have terrible showing in managing MaOPs. For instance, NSGA-II is serious about illuminating MOPs, but it does not perform well when managing MaOPs. To solve MaOPs and handle having a large number of non-dominated solutions in each iteration, Deb et al. recommended an evolutionary algorithm (EA) in light of the reference point and NSGA-II system (NSGA-III) (Vesikar et al., 2019). He utilized the reference point to choose parents for crossover. This method substantially improved the decent variety of populations and the capacity to illuminate MaOPs. In this work, MaoMFO proposes a similar manner in NSGA-III.

3. Many-Objective Moth-Flame Optimizer (MaOMFO)

This section briefly presents the basic notions of the original moth flame optimizer and comprehensively discusses the formulation procedure of the proposed Many-Objective Moth Flame Optimizer (MaOMFO) algorithm.

3.1. Moth-Flame Optimizer

The Moth-Flame Optimizer (Mirjalili, 2015) (MFO) was proposed by Mirjalili in 2015. This algorithm mimics the photo-tactic phenomenon in moths and other insects, in which they move towards a light source. Moths use traverse orientation by keeping the moon as their main light source at night for navigation. These insects maintain a fixed angle with the moon, which allows them to travel in a straight line due to the long distance to the moon. When replacing the moon with artificial light, moths get trapped in a spiral movement, which disrupts their navigation but converges them towards a single point. This behaviour has been mathematically modelled in the MFO algorithm as follows:

$$S(M_i,F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \tag{6}$$

$$D_i = |F_j - M_i| \tag{7}$$

$$flameno = round\left(N - l * \frac{N-1}{T}\right) \tag{8}$$

where M_i represent the i^{th} moth, F_j represents the j^{th} flame, and where D_i expresses the path length of the i^{th} moth for the j^{th} flame. The authors are encouraged to read the base paper for more details about the MFO algorithm.

3.2. Many-Objective Moth-Flame Optimizer (MaOMFO)

The proposed MaoMFO algorithm uses diversity preservation and an elitist non-dominated sorting with a well-distributed Pareto front reference point mechanism (Premkumar, Jangir, & Sowmya, 2021; Premkumar, Jangir, Sowmya, et al., 2021). The following measures are included in non-dominated sorting:

- Determine the non-dominated solution
- Apply non-dominated sorting (NDS) mechanism
- Find the non-dominated ranking (NDR) of all non-dominated solutions
- Apply reference point mechanism

The NDR procedure, with two fronts presented, is shown in Fig. 2. Because any other solutions do not dominate them, the solutions in the first front have an index of 0, but at minimum, one of the solutions in the second front dominates the solutions in the first front. The number of solutions that exceed such solutions is equal to their NDR.

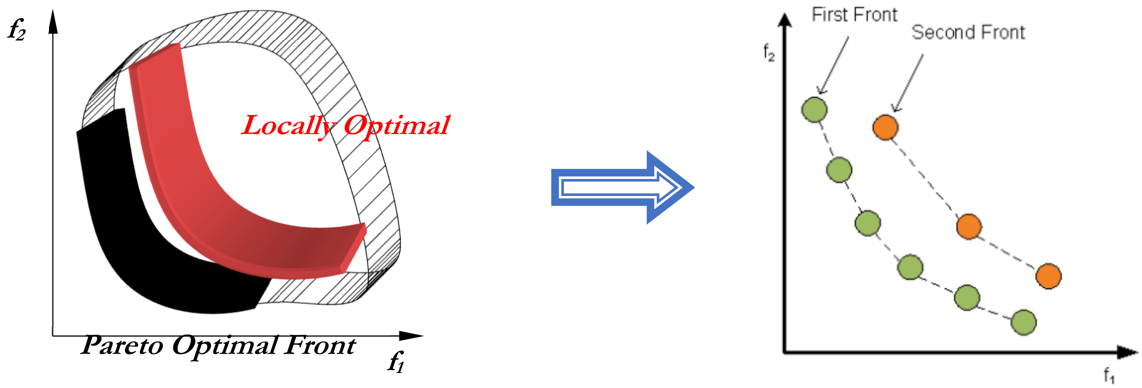


Fig. 2. Diagram of non-dominated sorting

The various steps of the proposed MaOMFO algorithm are represented as pseudocodes. $Algorithm\ 1$, $Algorithm\ 2$, $Algorithm\ 3$, $Algorithm\ 4$, and $Algorithm\ 5$ discussed the pseudocodes of the proposed MaoMFO method detailed explanation. The pseudocode of MFO for many-objective optimization is shown in Algorithm 1.

$Algorithm\ 1\ -\ MFO\ (Mt, Ft, Mo)$

Input:

M_t: current moth population

F_t: current flame population

D: number of decision variables

N: size of the population

Output:

Q_t: offspring population

$Q_t = \varnothing$, $i = 1$

while ($|Q_t| < N$)

  current flame individual: $F_{t,i}$

  current moth individual: $M_{t,i}$

  for $j = 1: D$

    dis = $F_{t,ij}$ - $M_{t,ij}$

    t = random (0,2), $b = 1$

    if $t <= 1$

      $P_{ij} = Dis_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j$

    else random (0,1) < 0.2

      random choose an individual from M_t

      $P_{ij} = Dis_i \cdot e^{bt} \cdot \cos(2\pi t) + F_{flame}$

    end if

  end for

  Polynomial-mutation(P_i)

  update F_i. If single-objective: compare the objective value of P_i and F_i to take a better solution as F_i; if multi-objective: if F_i is dominated by P_i, F_i is replaced by P_i; if P_i is dominated by F_i, F_i is not changed; if F_i and P_i do not dominate each other, randomly choose an individual M_{or} from moon population to replace F_i.

  $Q_t = Q_t \cup P_i$

  $i = i + 1$

end while

$Mt' = Mt \cup Q_t$

update Mt' as M_t. If single-objective, M_t is the best solution for M_t'; if multi-objective, M_t is maintained from M_t' by a specific strategy. In this paper, the specific strategy is the Reference-domination sorting method.

$Algorithm\ 2$ is the overall framework of the MaoMFO algorithm. There are several important functions in $Algorithm\ 2$, including MFO(), Normalize(), Associate(), Reference point mechanism(). The respective pseudocodes are shown in $Algorithm\ 1$, 3, 4, and 5, respectively.

$Algorithm\ 3 - Normalize(St)$

Input:

S_t: to be normalized population

O: number of objective functions

1: **for** $i = 1{:}O$

2: compute the ideal point: $f_i^{min} = min f_i(s)$, $s \in S_t$

3: translate objectives: $f_i'(s) = f_i(s) - f_i^{min}$, $s \in S_t$

4: compute extreme points $Y^{i,\max}$ of S_t

5: **end for**

6: **for** $i = 1{:}O$

7: compute intercepts a_i according to $Y^{i,\max}$

8: normalize objectives $\overline{f_i}(s) = f_i'(s) - f_i^{min}/a_i$, $s \in S_t$

9: **end for**

$Algorithm\ 4 - Associate\ (St, Zr)$

Input:

S_t: to be associated with the population

Z^r: reference points

1: **for each** $s \in S_t$

2: **for each** $w \in Z^r$

3: compute vertical distance $d(s,w)=||(s-w^T sw/||w||^2)||$

4: **end for**

5: find the reference point $Zr\ m$ for the minimum d

6: associate s with $Zr\ m$

7: **end for**

$Algorithm\ 5 - Reference - value\ (St, Zr)$

Input:

S_t: population

Z^r: reference points

1: **for** $i = 1$: length(Z^r)

2: get all individuals associated with Z_i^r: P_i

3: sort P_i by ascending order of vertical distance from individual to Z_i^r

4: **for** j = 1: length(P_i)

5: set P_{ij}'s Reference-value $= j + d_j$

6: **end for**

7: **end for**

The complete procedure of the proposed MaOMFO algorithm is presented in $Algorithm\ 6$.

$Algorithm\ 6 - Pseudocode\ of\ proposed\ MaoMFO$

Initialize the parameters population size, D, M, FEs;

[Z, N] = Uniform-Point (N, M); // **Step 1**

Initialize the positions of Moth $X_i (i = 1,2, \ldots, n)$; // **Step 2**

Zmin = min (Moth Position, objects); // **Step 3**

While ($t \leq$ Max_iteraition)// **Step 4**

Mating Pool= Tournament-Selection (N, constraint violation); // **Step 5**

Update moth position by **Eq.** $(6 - 8)$;

Offspring = MFO (Moth Position (Mating Pool)); // **Step 6**

Z'min = min (Offspring, objects); // **Step 7**

[Front No, Max. Front No] = Non-Dominated Sorting (objects, constraint, N); // **Step 8**

Moth Position-Next = Normalization (Next-objects, Last-objects, N-sum(Next), Z, Zmin); // **Step 9 to Step 12**

End While

Return Final Moth Position;

Step by Step presentation of the Many-objective moth flame optimizer (MaoMFO) is given below:

Step 1: Generating reference points using the uniform point function, a utility function for generating about N uniformly distributed points with M objectives on the unit hyperplane. Z is the set of reference points, and the Moth Position Size N is reset to the same as the number of reference points in Z.

Step 2: An initial random population is generated using the initialization function.

Step 3: Find the minimum objective value using a random Moth Position.
Step 4: After that, the termination criteria are invoked to check whether the number of evaluated fitness exceeds the maximum number of function evaluations, and Moth Position is passed to the function to be the final output.
Step 5: Afterwards, the mating pool selection using Tournament Selection. Returns the indices of N solutions by two-tournament selection based on their fitness values. In each selection, the candidate having the minimum value can be selected.
Step 6: Then, the Moth Flame Optimizer function generates offspring using Eq. 6-8.
Step 7: Find the minimum and maximum objective value generated via Moth Flame Optimizer; then combine both values using the union operator.
Step 8: Then apply the non-dominated sorting approach using objective value, constraint violation, and Moth Position size until it cannot be reached at the maximum front number or Select part of the solutions in the last front.
Step 9: After applying the normalization approach, detect the extreme points and calculate the intercepts of the hyperplane constructed by the extreme points on the axes.
Step 10: After that, calculate the distance of each solution to each reference vector that associates each solution with its nearest reference point and calculates the number of associated solutions except for the last front of each reference point.
Step 11: Afterwards, select K-remaining solutions one by one and find the least crowded reference point. Then, select one solution associated with this reference point
Step 12: Afterwards, get Moth Position for the next generation

4. Results of Various Test Suites

To test the optimization efficiency of the proposed Many-objective Moth Flame Optimizer (MaOMFO) algorithm, the following multi-objective and many-objective benchmark functions are used:

- Unconstrained many-objective test functions with 10 objectives (DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7, and DTLZ8) (Deb et al., 2005; Deb, Thiele, et al., 2002)
- Unconstrained multi-objective test functions with 3-objectives (DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7, DTLZ8, IMOP6, IMOP7, and IMPO4) (Deb et al., 2005; Deb, Thiele, et al., 2002)
- Constrained multi-objective test functions with 3-objectives (C1-DTLZ1 and C3-DTLZ4)
- Unconstrained multi-objective test functions with 2-objectives (ZDT1, ZDT2, ZDT3, and ZDT4) (Deb et al., 2005; Deb, Thiele, et al., 2002)

To quantify the performance of the MaOMFO algorithm and contrast the results with other selected algorithms, the following performance metrics have been used:

$$\text{Generational Distance (GD)} = \frac{\sqrt{\sum_{i=1}^{no} d_i^2}}{n} \tag{9}$$

$$\text{Inverted Generational Distance (IGD)} = \frac{\sqrt{\sum_{i=1}^{nt} (d_i')^2}}{n} \tag{10}$$

$$\text{HV}(\text{z}^{\dagger}, A)[32,33] = L\left\{\bigcup_{\text{a}\in A}\{\text{b} \in \Lambda \middle| \text{a} \prec \text{b} \prec \text{z}^{\dagger}\}\right) \tag{11}$$

$$\text{Maximum Spread (MS)}[34] = \sqrt{\sum_{i=1}^{o} \max\big(d(a_i, b_i)\big)} \tag{12}$$

where *nt* shows the size of the true Pareto optimal solutions set, *no* denotes the number of True PS and d_i and d_i' indicates the Euclidean distance (ED), $\overline{d}$ signifies the average of all d_i, *n* represents the number of obtained PS, and $d_i = \min_j\left(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})\right)$ for all *i,j=1,2,…,n*., a_i and b_i is the maximum and minimum value in the i^{th} objective, and *o* denotes the total objectives. The first two performance metrics measure convergence, and the last two measure the coverage of Pareto optimal solutions obtained by the algorithms. Note that for IGD and GD, lower esteems show better outcomes. MS and HV are high for a superior calculation and show higher diversity. The best Pareto optimal front is shown in the following subsections to analyze the outcomes and observe the convergence performance. To observe the results qualitatively, PlatEMO (Tian et al., 2017) is used to visualize the best Pareto optimal fronts obtained by the MaOMFO algorithm on the case studies, as seen in Figures 3-5. Each algorithm was run 30 times, function evaluation=100000, population size (N)=100, and reference points for 3 & 10 objectives, respectively, used 91 & 275. For results verification, MaOMFO is compared with MOEA/DD (Castro et al., 2017), NSGA-III, and NMPSO (Lin et al., 2018). The results of MaOMFO calculations on numerous and many-objective test suites are introduced in Table 1, Table 2, Table 3, and Table 4. The best results are highlighted with boldfaces in all tables. As discussed in all tables, it is clear that MaOMFO outperforms other algorithms based on different metrics in most test suites. The metrics, such as GD, IGD, MS, and HV, are used to evaluate the convergence and coverage of numerous and many-objective optimization problems. The outcomes show that MaOMFO outperforms all selected algorithms due to the inclusion of non-sorting and reference point mechanisms. The best Pareto fronts computed by the proposed algorithm are illustrated in Figs. 3-5. It is seen that Pareto's optimal solutions assessed are of high coverage overall objectives.

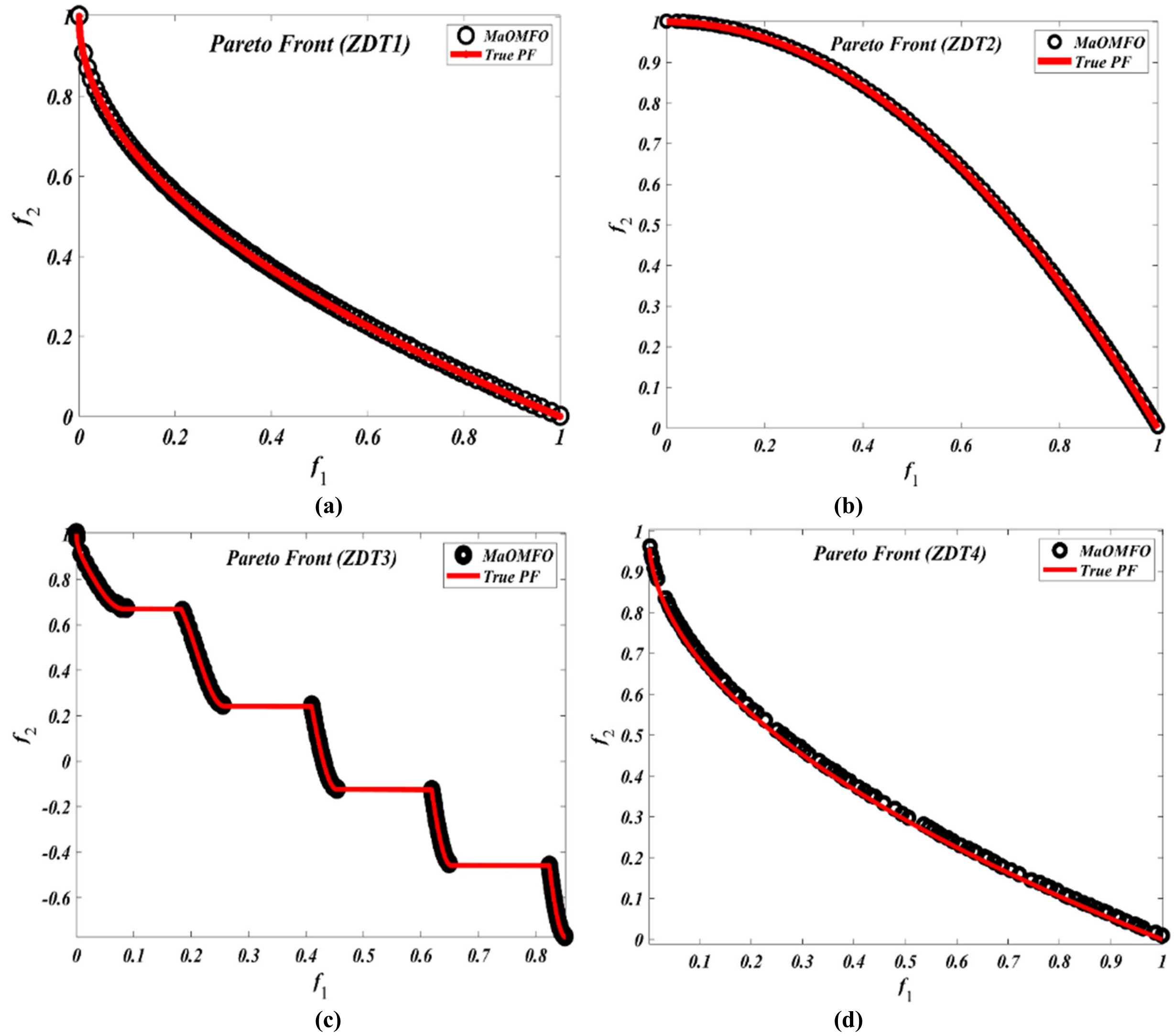
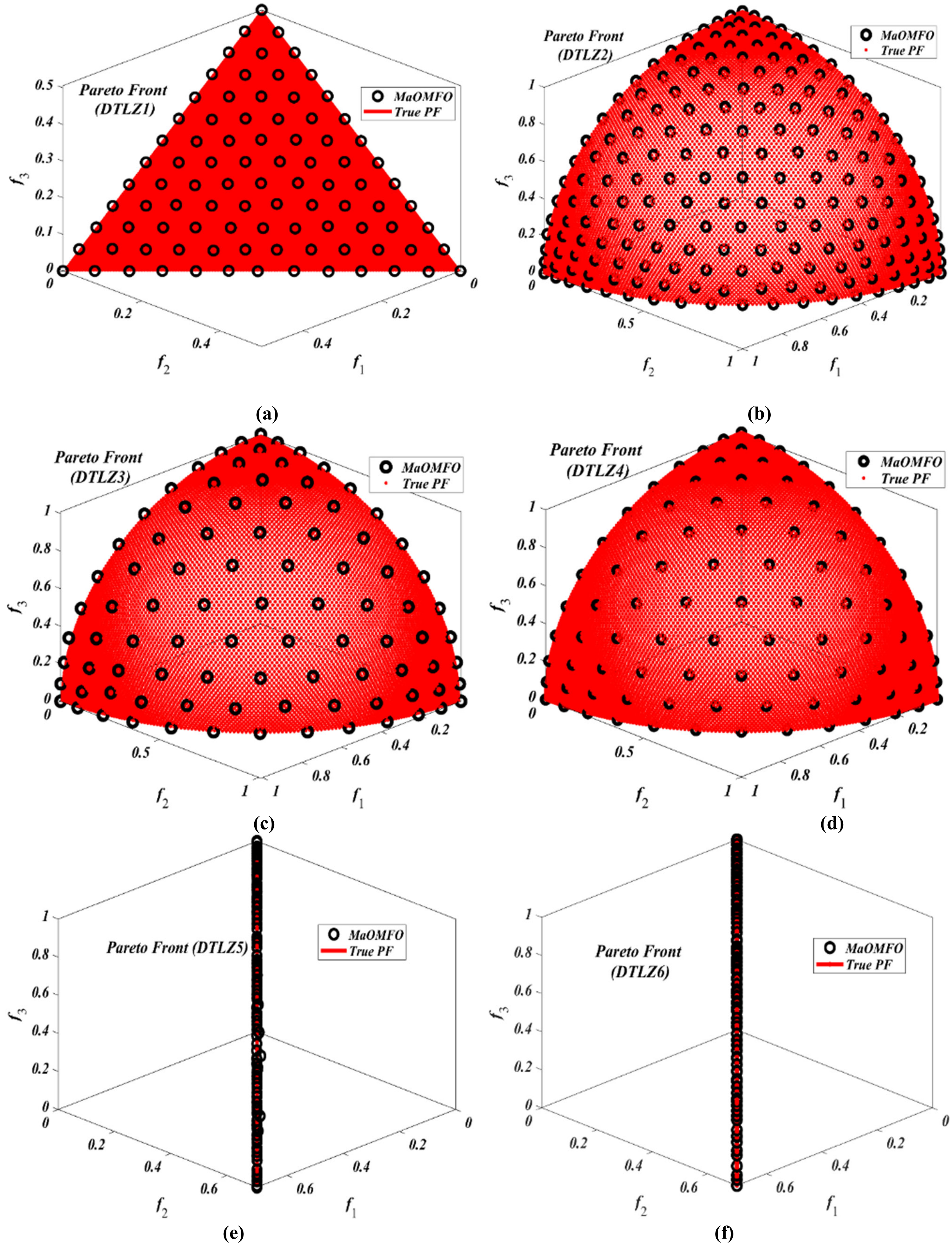


Fig. 3. Pareto fronts obtained by MaOMFO algorithm for ZDT problems with 2 objectives; (a) ZDT1, (b) ZDT2, (c) ZDT3, (d) ZDT4


Pareto Front (DTLZ1)
MaOMFO
True PF
f3
f2
f1
(a)
Pareto Front (DTLZ2)
MaOMFO
True PF
(b)
Pareto Front (DTLZ3)
MaOMFO
True PF
(c)
Pareto Front (DTLZ4)
MaOMFO
True PF
(d)
Pareto Front (DTLZ5)
MaOMFO
True PF
(e)
Pareto Front (DTLZ6)
MaOMFO
True PF
(f)

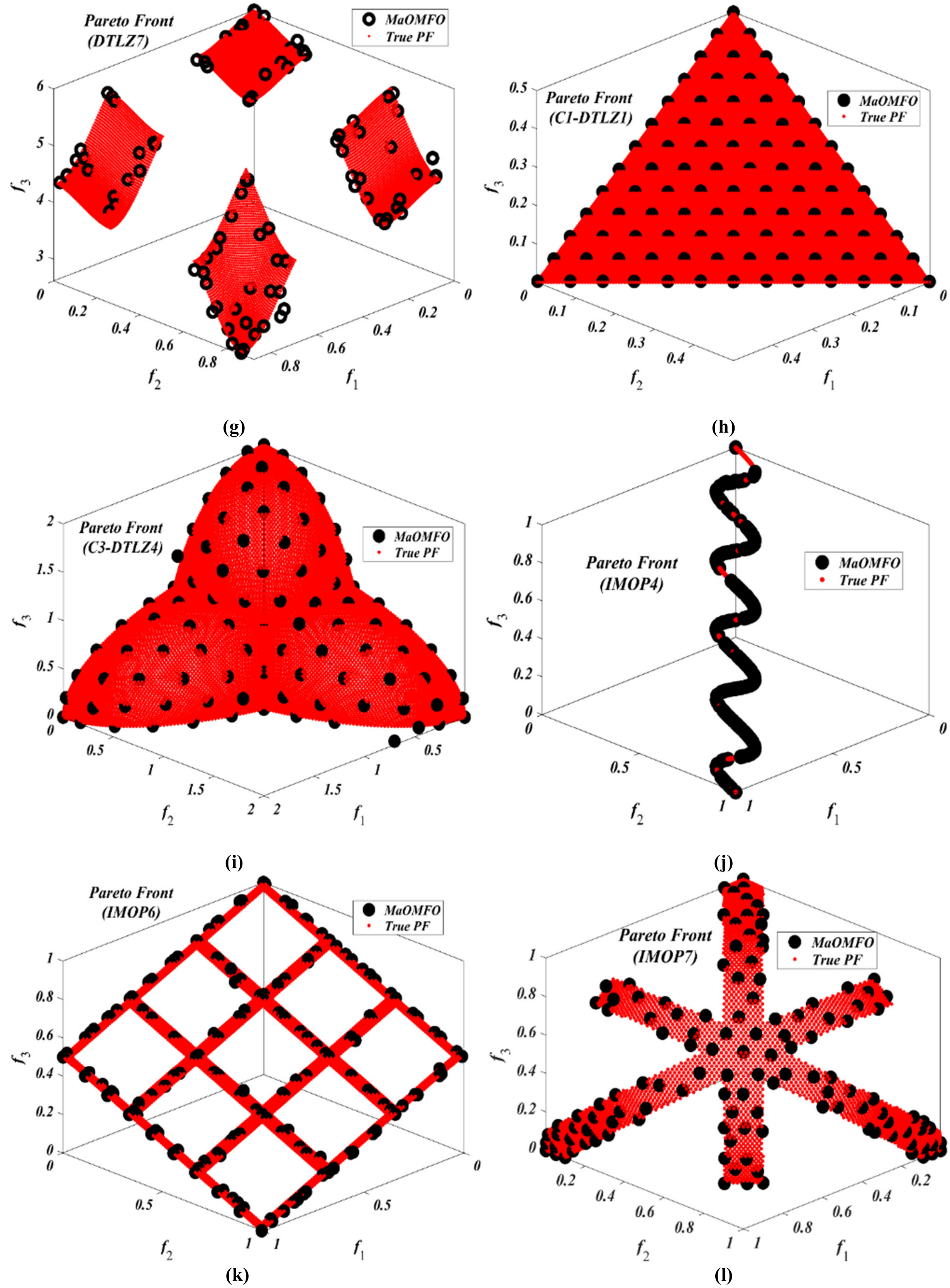
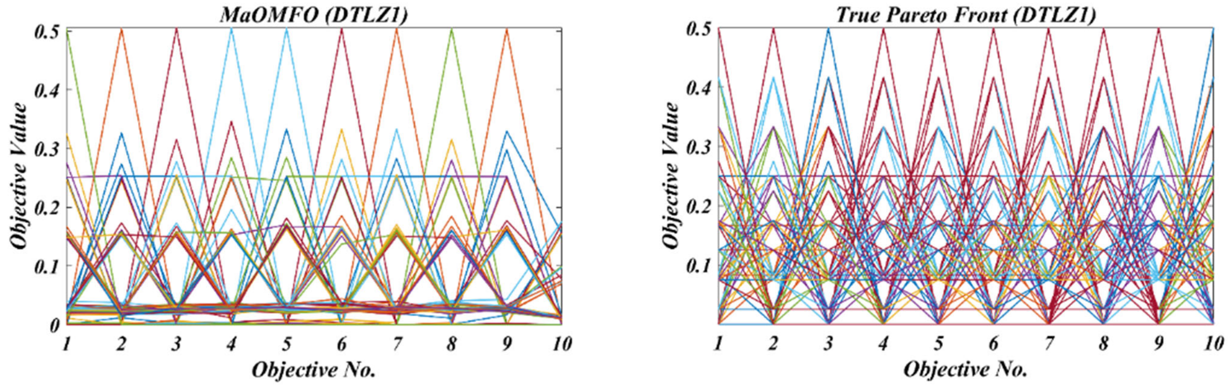
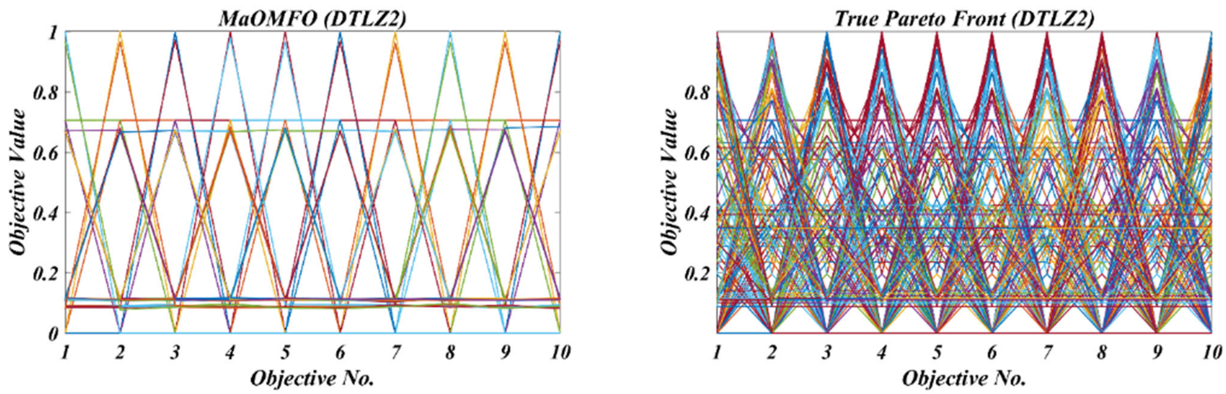


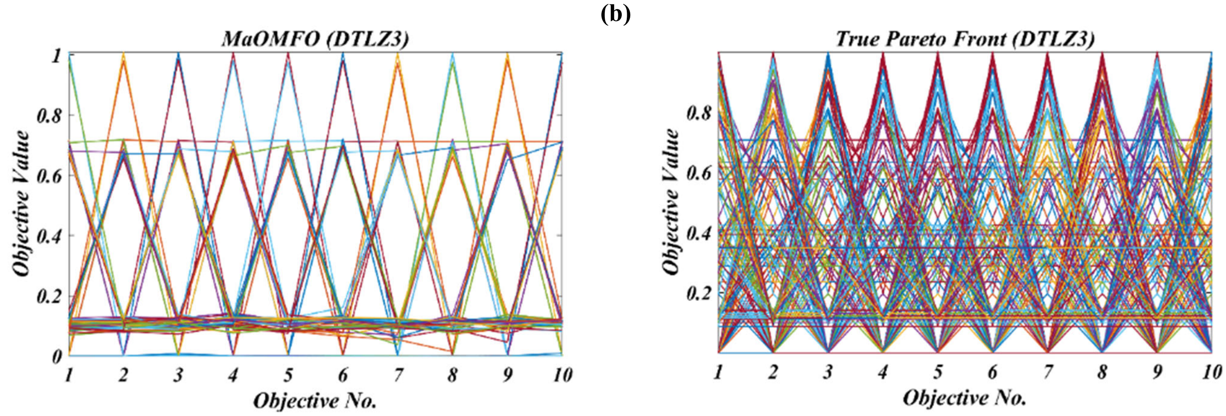
Fig. 4. Pareto optimal front obtained by the MaOMFO algorithm on DTLZ and IMOP with 3-objectives; (a) DTLZ1, (b) DTLZ2, (c) DTLZ3, (d) DTLZ4, (e) DTLZ5, (f) DTLZ6, (g) DTLZ7, (h) C1-DTLZ1, (i) C3-DTLZ4, (j) IMOP4, (k) IMOP6 and (l) IMOP7


MaOMFO (DTLZ1)
True Pareto Front (DTLZ1)
Objective Value
Objective No.

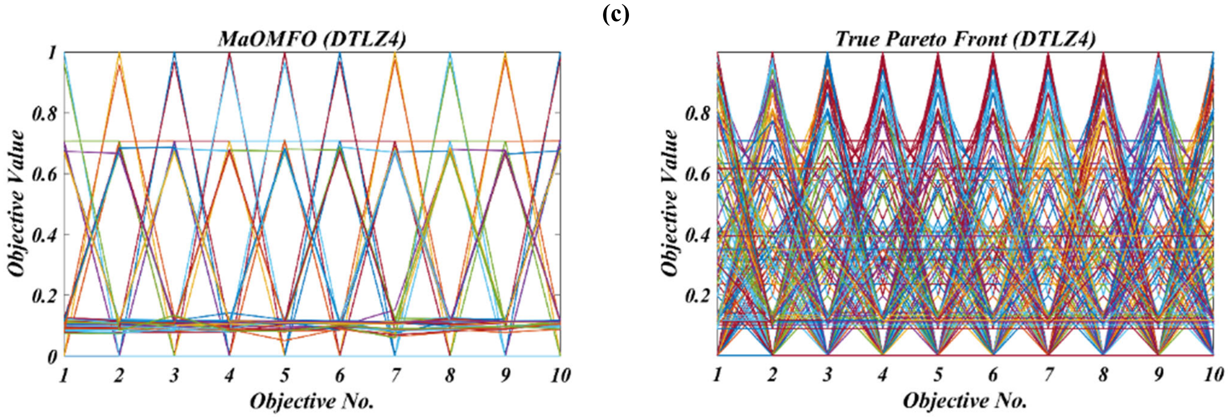
(a)


MaOMFO (DTLZ2)
True Pareto Front (DTLZ2)
Objective Value
Objective No.

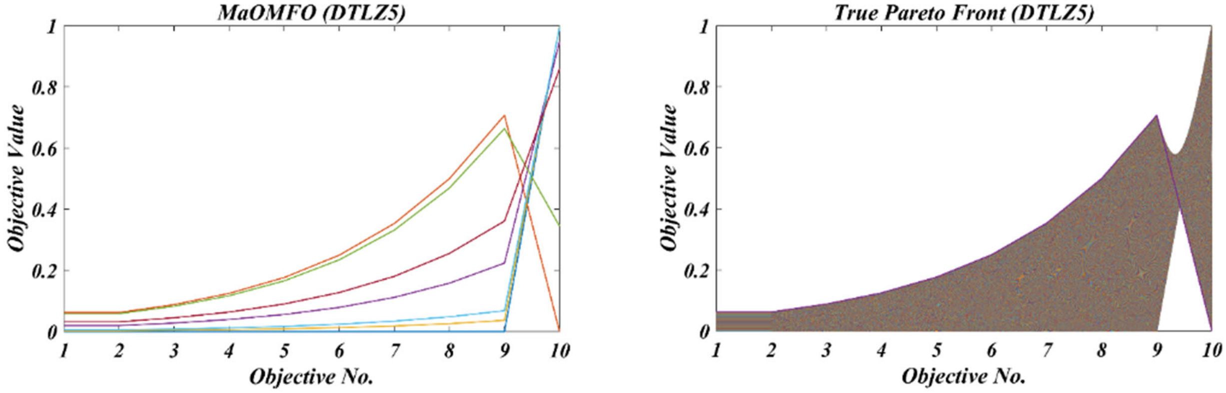
(b)


MaOMFO (DTLZ3)
True Pareto Front (DTLZ3)
Objective Value
Objective No.

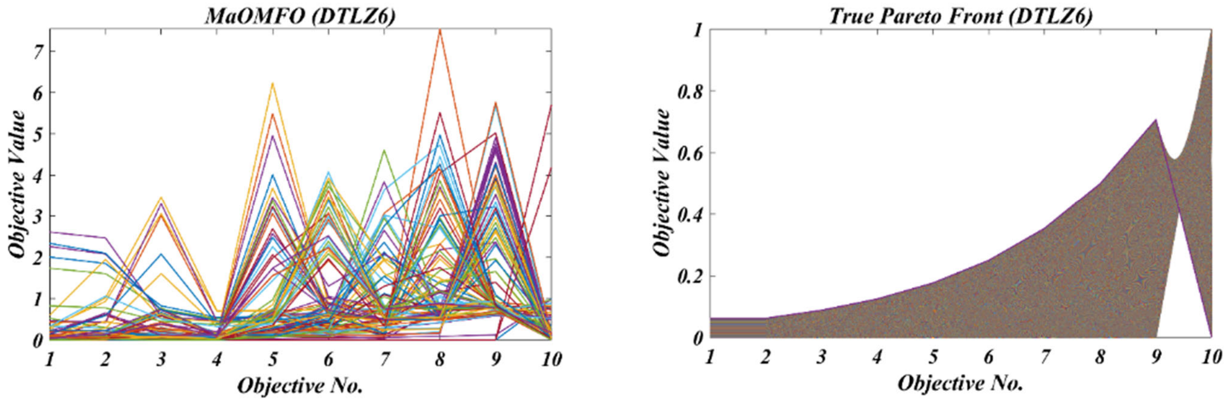
(c)


MaOMFO (DTLZ4)
True Pareto Front (DTLZ4)
Objective Value
Objective No.

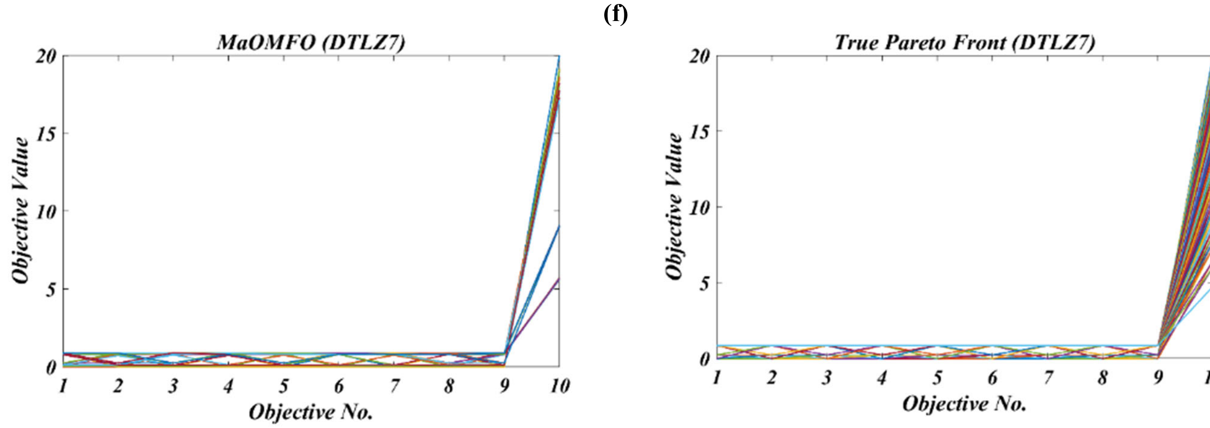
(d)



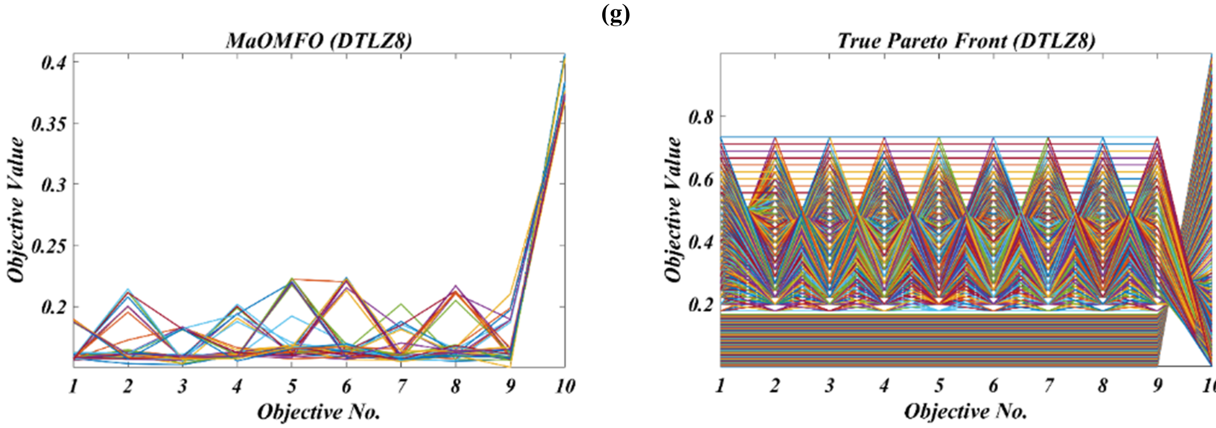
(e)



(f)



(g)



(h)

Fig. 5. True Pareto front and Pareto front obtained by MaOMFO algorithm on DTLZ problems with 10 objectives; (a) DTLZ1, (b) DTLZ2, (c) DTLZ3, (d) DTLZ4, (e) DTLZ5, (f) DTLZ6, (g) DTLZ7, (h) DTLZ8

The solution diversity of the proposed MaOMFO algorithm is high, demonstrating that the MaOMFO can deal with high-complexity problems and discover Pareto optimal solutions in various feasible search spaces. In order to visualize the performance metrics, such as GD, IGD, HV, and coverage, the plots are provided for each test function. Fig. 6 shows the performance metrics of all ZDT1-ZDT4 problems with two objectives, Fig. 7 shows the performance metrics of DTLZ1-DTLZ3 and DTLZ5-DTLZ6 problems with 3 objectives, and Fig. 8 shows the performance metrics of DTLZ7, C_DTLZ1, IMOP4, IMOP6, and IMOP7 problems with 3 objectives. Figs. 6-8 are useful to understand how the proposed algorithm converges to the global optimal values and how well the Pareto fronts are distributed. From Figs. 6-8, it is observed that the proposed algorithm is comparable to recently proposed many-objective algorithms.

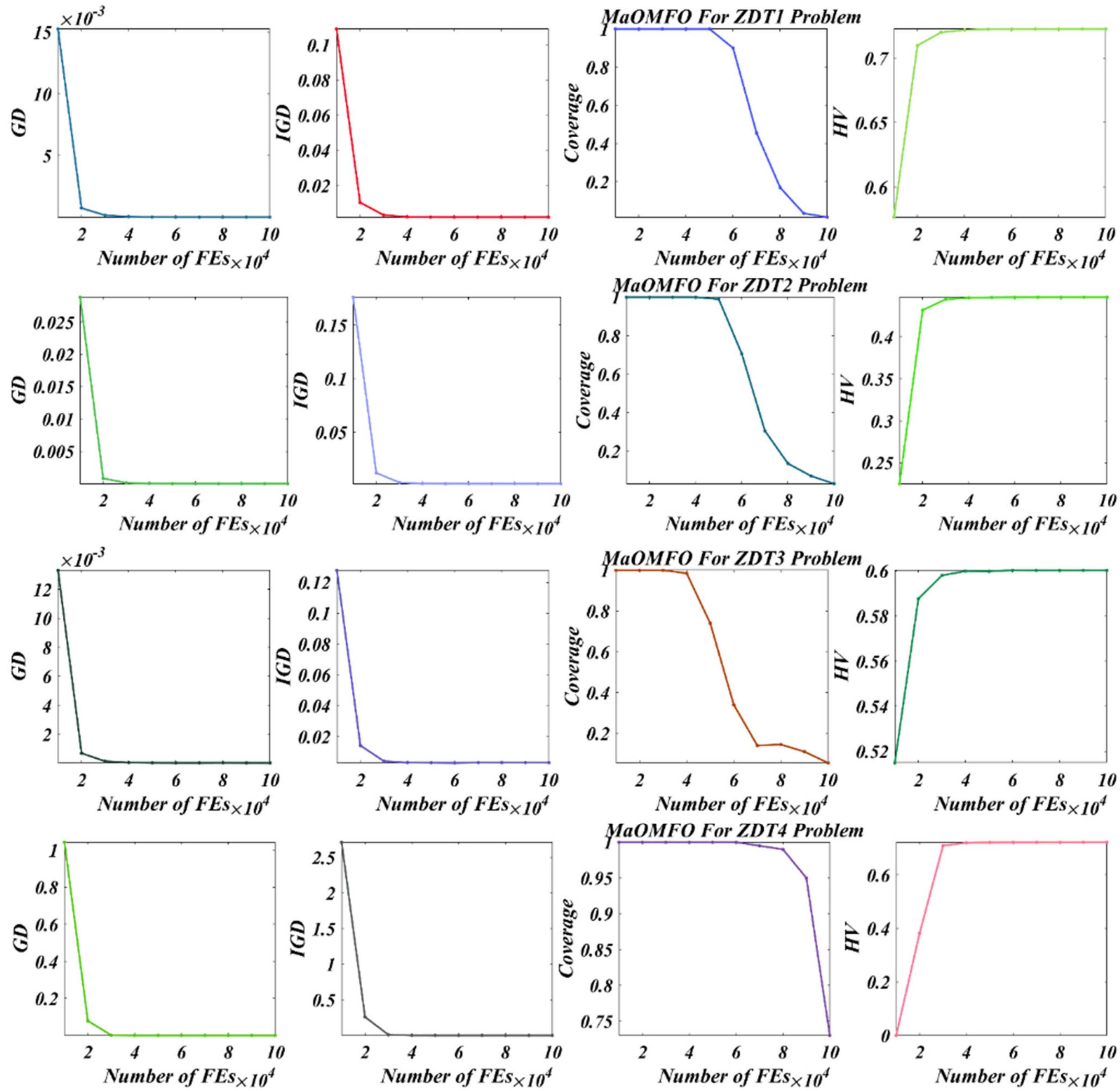
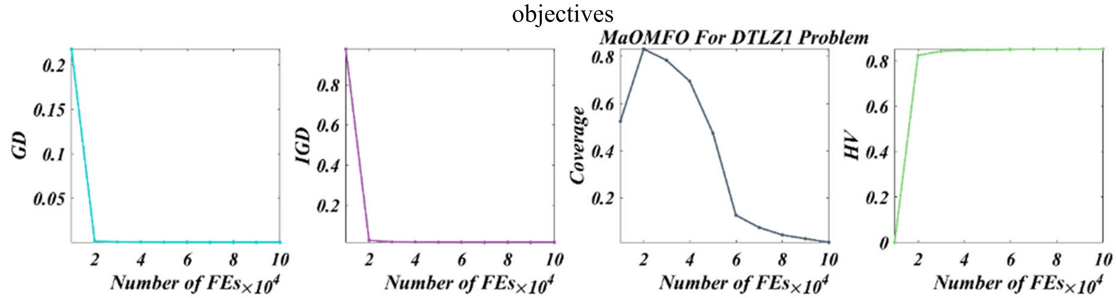


Fig. 6. GD, IGD, Coverage, and HV curved obtained by MaoMFO on ZDT1, ZDT2, ZDT3, and ZDT4 problems with 2-objectives



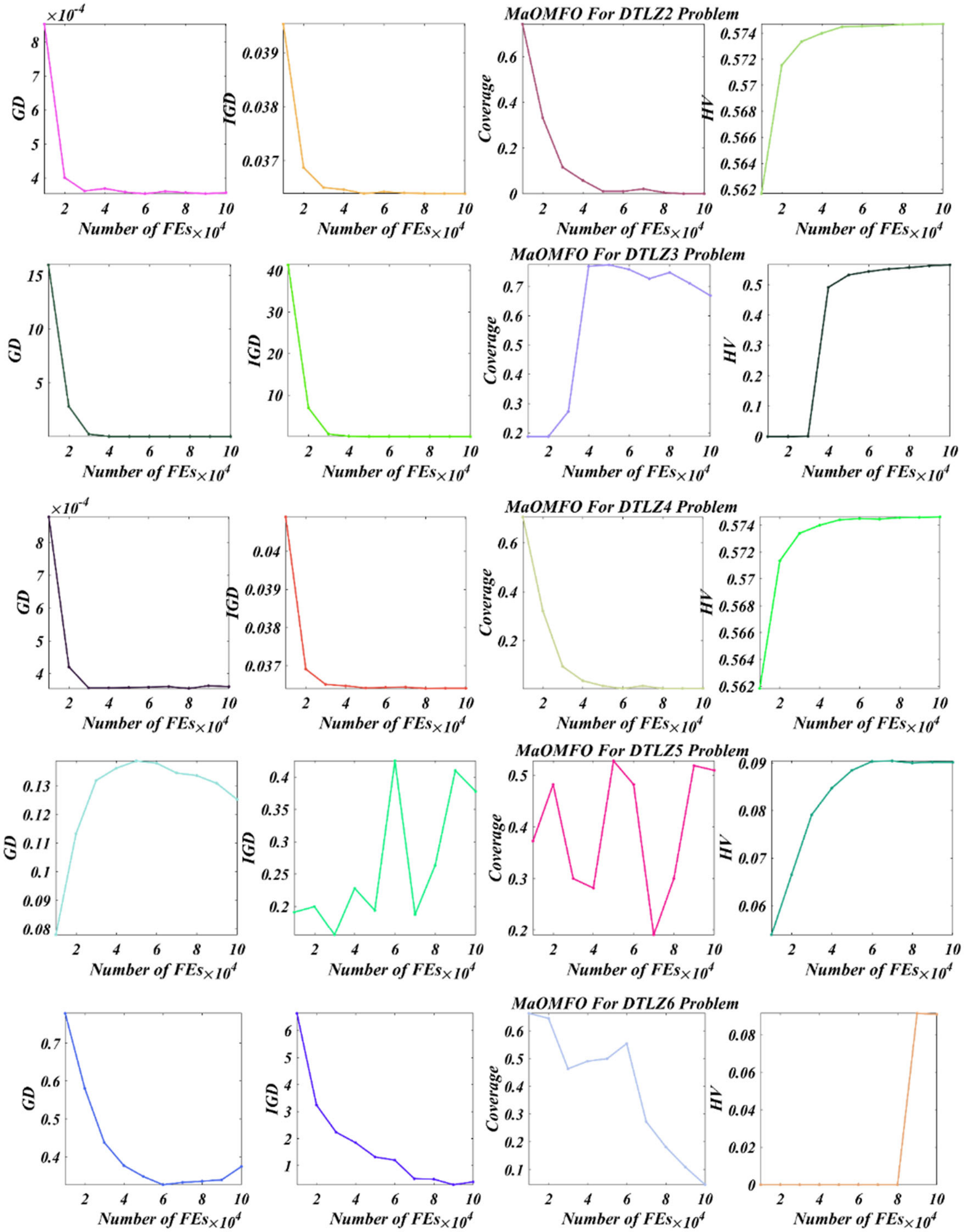
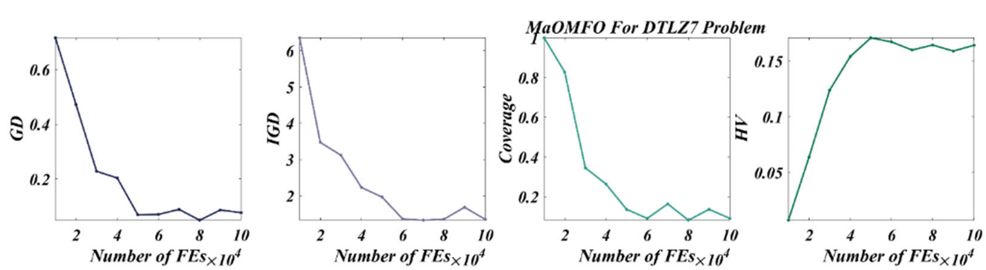


Fig. 7. GD, IGD, Coverage, and HV curved obtained by MaoMFO on DTLZ1, DTLZ2, DTLZ3, DTLZ5, and DTLZ6 problems with 3-objectives



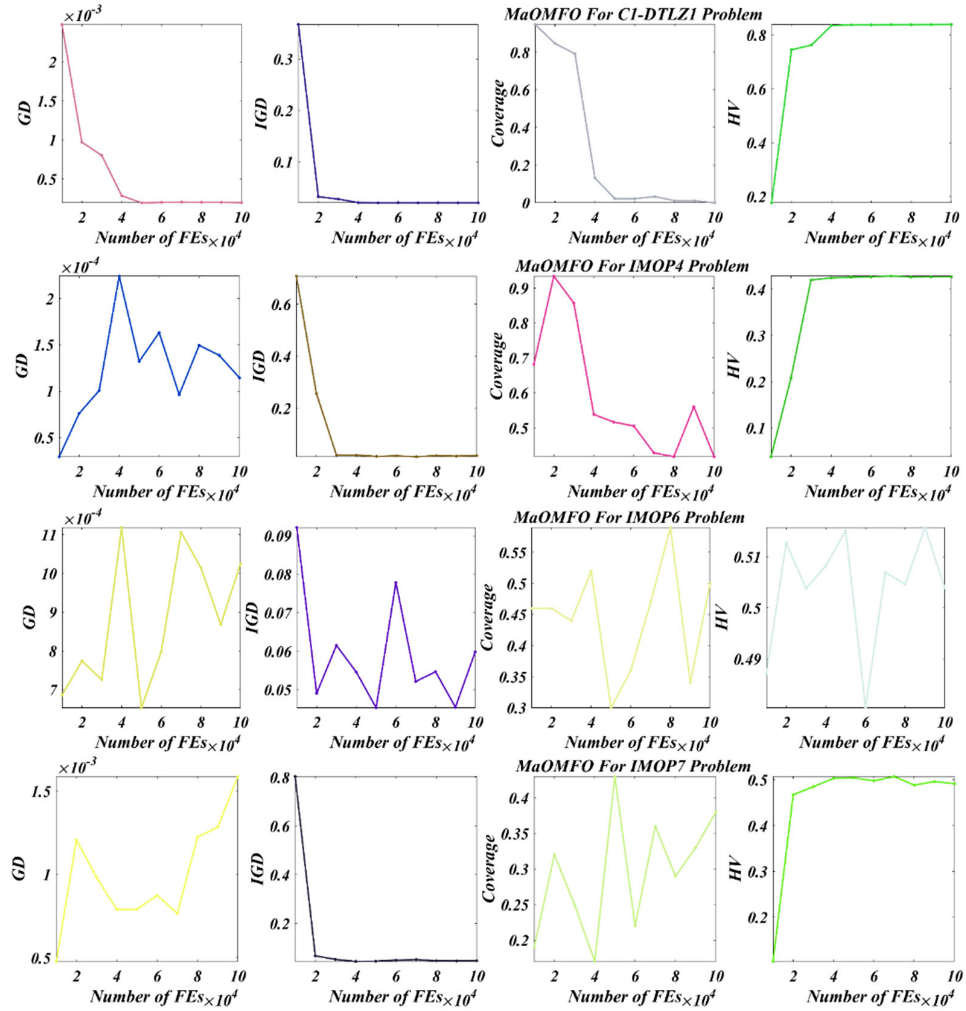


Fig. 6. GD, IGD, Coverage, and HV curved obtained by MaoMFO on DTLZ7, C1-DTLZ2, IMOP4, IMOP6, and IMOP7 problems with 3-objectives

Table 1
GD values (mean and STD) obtained by all selected algorithms for DTLZ, IMOP, and ZDT problems

Problem	M	D	NSGA-III	NMPSO	MOEA-DD	MaOMFO
	3	7	5.1799e-4 (9.48e-5)	2.3619e-4 (3.29e-5)	**2.0862e-4 (1.05e-5)**	2.3566e-4 (3.25e-5)
DTLZ1	10	14	8.4165e-1 (1.44e+0)	6.5568e-3 (1.18e-3)	4.6288e-3 (4.81e-4)	**4.5784e-3 (1.54e-3)**
	3	12	**4.9888e-4 (9.82e-6)**	5.0499e-4 (3.50e-6)	5.0751e-4 (3.78e-6)	5.2043e-4 (1.15e-5)
DTLZ2	10	19	1.4368e-2 (2.18e-3)	1.0669e-2 (1.91e-3)	4.4595e-3 (4.00e-3)	**2.8527e-3 (1.59e-3)**
	3	12	1.0908e-1 (1.05e-1)	1.7203e-2 (6.67e-3)	2.5714e-1 (1.06e-1)	**2.8352e-3 (7.13e-4)**
DTLZ3	10	19	9.3765e+0 (5.07e+0)	4.5562e-1 (7.58e-1)	1.4482e+0 (2.17e+0)	**7.4369e-3 (1.27e-3)**
	3	12	5.5305e-4 (8.62e-5)	5.7975e-4 (8.53e-5)	5.0524e-4 (4.60e-6)	**4.9843e-4 (1.18e-5)**
DTLZ4	10	19	1.2458e-2 (1.95e-3)	9.3741e-3 (1.76e-3)	8.1329e-3 (2.54e-3)	**7.0424e-3 (8.75e-4)**
	3	12	2.8367e-4 (7.56e-5)	3.7595e-2 (1.06e-2)	3.9435e-4 (2.22e-4)	**1.2002e-5 (4.66e-6)**
DTLZ5	10	19	1.1131e-1 (1.20e-2)	2.7157e-1 (2.53e-1)	7.9522e-2 (1.30e-2)	**2.8415e-6 (5.50e-7)**
	3	12	4.5728e-6 (2.59e-7)	1.6572e-1 (2.40e-1)	4.1921e-2 (1.62e-2)	**4.2295e-6 (4.83e-7)**
DTLZ6	10	19	5.8955e-1 (1.64e-1)	6.1494e-1 (3.88e-1)	2.9899e-1 (4.11e-2)	**1.9207e-6 (9.63e-7)**
	3	22	2.3212e-3 (4.75e-4)	5.9920e-3 (1.99e-3)	2.8669e-3 (4.15e-4)	**7.1097e-4 (2.81e-5)**
DTLZ7	10	29	2.5192e-1 (1.41e-1)	6.3230e-2 (3.94e-3)	9.9162e-2 (3.22e-2)	**2.3351e-2 (1.73e-3)**
DTLZ8	10	100	2.2110e-2 (3.63e-3)	5.5044e-2 (7.91e-3)	2.7208e-2 (2.95e-3)	**1.0075e-2 (1.24e-3)**
C1_DTLZ1	3	7	2.6455e-4 (6.81e-5)	2.9537e-4 (9.49e-5)	3.1086e-4 (1.77e-4)	**2.1066e-4 (1.51e-5)**
C3_DTLZ4	3	12	2.8131e-3 (2.40e-4)	5.0552e-3 (8.59e-4)	8.1621e-3 (1.97e-3)	**2.7581e-3 (5.53e-4)**
IMOP6	3	10	4.8070e-4 (5.60e-5)	1.1729e-2 (3.38e-3)	**4.0083e-4 (2.58e-5)**	4.4730e-4 (8.63e-6)
IMOP7	3	10	5.4020e-4 (4.56e-4)	5.4888e-4 (3.40e-4)	4.1395e-4 (2.77e-4)	**2.5008e-4 (8.71e-6)**
IMOP4	3	10	1.1066e-4 (9.27e-6)	1.0098e-4 (9.23e-5)	1.5760e-4 (2.18e-4)	**8.7593e-6 (4.16e-6)**
ZDT1	2	30	**3.2891e-5 (5.97e-6)**	1.9651e-3 (1.07e-4)	1.7958e-4 (4.75e-5)	8.0461e-5 (5.25e-5)
ZDT2	2	30	4.3302e-5 (2.24e-5)	4.3269e-3 (7.97e-4)	1.8689e-4 (6.29e-5)	**1.6900e-5 (6.88e-6)**
ZDT3	2	30	4.7446e-5 (1.49e-5)	3.2522e-3 (2.16e-4)	1.4374e-4 (4.74e-5)	**3.3465e-5 (4.12e-6)**
ZDT4	2	10	3.0612e-4 (1.11e-4)	1.5256e-3 (2.19e-4)	6.0579e-4 (2.67e-4)	**1.7584e-4 (5.64e-5)**

Table 2
IGD values (mean and STD) obtained by all selected algorithms for DTLZ, IMOP, and ZDT problems

Problem	M	D	NSGA-III	NMPSO	MOEA-DD	MaOMFO
	3	7	2.1541e-2 (3.08e-4)	2.0717e-2 (9.84e-5)	2.0904e-2 (1.27e-4)	**2.0639e-2 (1.96e-5)**
DTLZ1	10	14	3.2521e-1 (2.31e-1)	1.4756e-1 (3.84e-3)	2.9710e-1 (5.79e-3)	**1.3577e-1 (2.32e-3)**
	3	12	5.4495e-2 (1.56e-5)	**5.4467e-2 (1.41e-6)**	5.4472e-2 (4.41e-6)	5.4786e-2 (1.63e-4)
DTLZ2	10	19	6.1446e-1 (2.98e-2)	5.0585e-1 (5.64e-3)	6.9186e-1 (1.03e-1)	**5.0064e-1 (2.21e-3)**
	3	12	8.6040e-1 (7.13e-1)	1.2445e-1 (9.36e-3)	1.1372e+0 (9.19e-1)	**1.0869e-1 (6.91e-2)**
DTLZ3	10	19	1.4676e+1 (7.63e+0)	1.1178e+0 (1.01e-2)	1.8642e+0 (1.16e+0)	**8.2545e-1 (4.78e-1)**
	3	12	2.1679e-1 (2.81e-1)	2.1687e-1 (2.81e-1)	**5.4485e-2 (2.31e-5)**	5.5157e-2 (9.31e-5)
DTLZ4	10	19	7.1959e-1 (1.84e-2)	7.3516e-1 (3.77e-2)	6.7731e-1 (1.56e-1)	**5.4656e-1 (2.78e-2)**
	3	12	1.2255e-2 (9.25e-4)	7.9813e-2 (1.66e-2)	3.1109e-2 (7.16e-4)	**1.0457e-2 (1.80e-3)**
DTLZ5	10	19	3.1246e-1 (4.61e-2)	4.3756e-1 (2.55e-1)	**1.5095e-1 (4.92e-3)**	6.9012e-1 (4.80e-2)
	3	12	1.9476e-2 (3.13e-3)	7.4403e-2 (7.00e-3)	3.3565e-2 (4.86e-4)	**1.5130e-2 (1.26e-3)**
DTLZ6	10	19	3.1290e+0 (1.68e+0)	1.9297e-1 (4.13e-2)	**1.5199e-1 (3.29e-3)**	7.1318e-1 (3.68e-2)
	3	22	**7.8830e-2 (2.56e-3)**	1.0799e-1 (2.86e-3)	4.3421e-1 (3.34e-1)	1.8419e-1 (1.51e-1)
DTLZ7	10	29	2.2081e+0 (8.12e-2)	5.4382e+0 (4.66e-2)	2.3426e+0 (5.48e-1)	**1.7784e+0 (2.71e-1)**
DTLZ8	10	100	8.0649e-1 (7.20e-2)	8.7529e-1 (1.37e-2)	7.7571e-1 (1.96e-2)	**4.7718e-1 (2.17e-3)**
C1_DTLZ1	3	7	**2.0427e-2 (4.40e-4)**	2.1869e-2 (2.46e-3)	2.1325e-2 (1.87e-3)	2.2973e-2 (1.01e-4)
C3_DTLZ4	3	12	1.0640e-1 (1.49e-3)	9.7195e-2 (1.42e-3)	1.2438e-1 (6.63e-3)	**9.5746e-2 (5.45e-4)**
IMOP6	3	10	4.2409e-2 (1.14e-3)	5.2965e-1 (1.43e-3)	2.0350e-1 (2.65e-1)	**4.1808e-2 (6.98e-4)**
IMOP7	3	10	6.0676e-1 (4.87e-1)	9.0151e-1 (5.55e-2)	5.7271e-1 (4.58e-1)	**6.6339e-2 (3.60e-3)**
IMOP4	3	10	1.8064e-2 (1.74e-3)	5.0268e-2 (4.50e-3)	7.7413e-1 (2.93e-3)	**1.6932e-2 (4.11e-4)**
ZDT1	2	30	5.3942e-2 (3.28e-2)	1.7560e-2 (4.52e-4)	4.3487e-3 (1.07e-4)	**3.9205e-3 (1.09e-5)**
ZDT2	2	30	8.2237e-2 (1.34e-1)	2.9909e-2 (5.11e-3)	4.2264e-3 (9.92e-5)	**3.8712e-3 (5.93e-5)**
ZDT3	2	30	8.4508e-3 (6.68e-4)	3.2231e-2 (2.61e-3)	7.0174e-2 (1.26e-4)	**6.1495e-3 (1.36e-4)**
ZDT4	2	10	2.1453e-1 (1.46e-1)	5.8317e-2 (7.75e-2)	7.4004e-3 (1.96e-3)	**5.6354e-3 (5.28e-4)**

Table 3
HV values (mean and STD) obtained by all selected algorithms for DTLZ, IMOP, and ZDT problems

Problem	M	D	NSGA-III	NMPSO	MOEA-DD	MaOMFO
	3	7	8.3487e-1 (1.47e-3)	8.4011e-1 (9.21e-4)	8.3950e-1 (8.90e-4)	**8.4071e-1 (1.89e-4)**
DTLZ1	10	14	6.6061e-1 (5.62e-1)	7.1404e-1 (3.24e-3)	9.8948e-1 (2.90e-4)	**9.9342e-1 (3.23e-3)**
	3	12	5.5940e-1 (4.66e-5)	5.5905e-1 (1.54e-4)	5.5940e-1 (9.71e-5)	**5.5946e-1 (2.36e-5)**
DTLZ2	10	19	8.4631e-1 (9.18e-3)	8.1187e-1 (8.42e-2)	9.3643e-1 (3.71e-3)	**9.3910e-1 (1.95e-4)**
	3	12	1.6918e-1 (2.93e-1)	4.1859e-1 (1.99e-2)	2.6033e-2 (3.49e-2)	**4.8557e-1 (5.75e-2)**
DTLZ3	10	19	0.0000e+0 (0.00e+0)	**5.9363e-1 (5.14e-1)**	1.6997e-1 (2.94e-1)	2.0803e-1 (3.87e-2)
	3	12	4.8643e-1 (1.27e-1)	4.8639e-1 (1.26e-1)	5.5900e-1 (1.72e-4)	**5.5946e-1 (3.56e-5)**
DTLZ4	10	19	7.2499e-1 (1.60e-2)	7.8501e-1 (3.96e-2)	8.1439e-1 (1.35e-1)	**9.2228e-1 (1.48e-2)**
	3	12	**1.9425e-1 (5.17e-4)**	1.5036e-1 (9.37e-3)	1.8318e-1 (1.71e-4)	1.8984e-1 (2.17e-5)
DTLZ5	10	19	5.8992e-2 (5.11e-2)	9.1314e-2 (4.39e-4)	9.1961e-2 (1.33e-3)	**9.4507e-2 (1.14e-3)**
	3	12	1.9104e-1 (1.62e-3)	1.5434e-1 (1.03e-2)	1.8204e-1 (2.14e-4)	**1.9419e-1 (1.27e-3)**
DTLZ6	10	19	0.0000e+0 (0.00e+0)	9.1003e-2 (1.32e-4)	**9.3840e-2 (2.58e-4)**	9.1457e-2 (8.86e-4)
	3	22	**2.6794e-1 (1.26e-3)**	2.6182e-1 (3.43e-3)	2.2346e-1 (2.83e-2)	2.6434e-1 (2.02e-2)
DTLZ7	10	29	1.2784e-1 (9.14e-3)	1.3362e-1 (2.36e-3)	4.0679e-5 (1.58e-5)	**1.3744e-1 (6.95e-4)**
DTLZ8	10	100	1.0095e-1 (1.26e-2)	9.8957e-2 (2.81e-3)	**1.1666e-1 (6.51e-3)**	2.0964e-2 (1.01e-2)
C1_DTLZ1	3	7	8.2662e-1 (7.02e-3)	7.9787e-1 (1.10e-3)	8.2045e-1 (2.42e-2)	**8.3700e-1 (5.31e-3)**
C3_DTLZ4	3	12	**7.9102e-1 (2.78e-4)**	7.8956e-1 (1.08e-3)	7.7060e-1 (5.63e-3)	7.8093e-1 (1.36e-3)
IMOP6	3	10	5.1454e-1 (1.54e-3)	8.6864e-2 (1.87e-3)	3.8154e-1 (2.27e-1)	**5.1779e-1 (3.26e-4)**
IMOP7	3	10	2.3339e-1 (2.39e-1)	9.3145e-2 (3.80e-3)	2.3747e-1 (2.33e-1)	**4.9879e-1 (3.04e-3)**
IMOP4	3	10	1.3965e-1 (1.34e-1)	4.0991e-1 (5.92e-4)	2.3165e-2 (5.83e-4)	**4.2552e-1 (1.06e-3)**
ZDT1	2	30	6.8609e-1 (1.99e-2)	6.9968e-1 (4.48e-4)	7.1897e-1 (4.45e-5)	**7.1994e-1 (7.55e-5)**
ZDT2	2	30	3.6888e-1 (1.21e-1)	4.0733e-1 (6.02e-3)	4.4389e-1 (3.04e-4)	**4.4454e-1 (1.68e-4)**
ZDT3	2	30	5.8202e-1 (4.31e-5)	5.5409e-1 (1.31e-3)	5.0751e-1 (2.66e-4)	**5.9852e-1 (4.04e-3)**
ZDT4	2	10	**7.1611e-1 (9.64e-4)**	6.7622e-1 (5.06e-2)	7.1313e-1 (3.54e-3)	5.9010e-1 (8.60e-2)

Table 1, Table 2, Table 3, and Table 4 shows that the MaOMFO algorithm outperforms NSGA-III, NMPSO, and MOEA/DD in most of the test cases with different objectives. The solution diversity is uniform for both MaOMFO and NSGA-III, showing high convergence and coverage. There is a hole in the Pareto front acquired by the NSGA-III, which adversely impacts the inclusion of this optimization. C_DTLZ has an inward moulded Pareto front and is continually trying collection-based strategies. These outcomes show that MaOMFO can productively fit with the true front with a high convergence and coverage. As can be seen, the results vary significantly in this case due to the complexity of such problems. However, the superiority of the proposed MaOMFO is still evident.

Table 4
Coverage values (mean and STD) obtained by all selected algorithms for DTLZ, IMOP, and ZDT problems

Problem	M	D	NSGA-III	NMPSO	MOEA-DD	MaOMFO
	3	7	6.2637e-1 (4.40e-2)	1.2088e-1 (1.35e-1)	**4.7619e-2 (3.36e-2)**	9.5279e-2 (1.10e-1)
DTLZ1	10	14	**5.1282e-3 (8.88e-3)**	5.1282e-3 (8.88e-3)	3.0769e-2 (2.66e-2)	1.8018e-2 (3.12e-2)
	3	12	1.0989e-2 (1.10e-2)	1.0989e-2 (1.10e-2)	7.3260e-3 (1.27e-2)	**7.3260e-3 (6.34e-3)**
DTLZ2	10	19	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**0.0000e+0 (0.00e+0)**
	3	12	6.7033e-1 (4.79e-2)	7.2233e-1 (6.40e-2)	7.5233e-1 (8.60e-2)	**5.4172e-1 (9.02e-2)**
DTLZ3	10	19	3.5897e-2 (1.78e-2)	3.7506e-2 (8.18e-3)	2.0693e-1 (1.21e-1)	**0.0000e+0 (0.00e+0)**
	3	12	**3.6630e-3 (6.34e-3)**	1.0989e-2 (1.90e-2)	3.6630e-3 (6.34e-3)	1.8315e-2 (1.68e-2)
DTLZ4	10	19	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	0.0000e+0 (0.00e+0)	**0.0000e+0 (0.00e+0)**
	3	12	8.9377e-1 (3.86e-2)	1.0000e+0 (0.00e+0)	6.3063e-1 (1.39e-1)	**1.7901e-1 (6.68e-2)**
DTLZ5	10	19	3.0769e-1 (5.55e-2)	4.3590e-1 (2.00e-1)	**1.6611e-1 (2.63e-2)**	5.1313e-1 (3.02e-1)
	3	12	5.4945e-2 (0.00e+0)	6.2217e-1 (3.30e-1)	1.1740e-1 (3.83e-2)	**2.3919e-2 (7.24e-4)**
DTLZ6	10	19	4.8205e-1 (6.94e-2)	3.2882e-1 (2.23e-1)	2.6906e-1 (1.95e-2)	**2.0526e-1 (2.70e-1)**
	3	22	7.4725e-1 (3.96e-2)	8.0406e-1 (8.27e-2)	3.2085e-1 (9.41e-2)	**4.0765e-2 (1.56e-3)**
DTLZ7	10	29	2.6667e-1 (7.27e-2)	6.5359e-2 (8.90e-2)	**0.0000e+0 (0.00e+0)**	2.1420e-2 (9.39e-3)
DTLZ8	10	100	7.1795e-1 (2.09e-1)	3.2085e-1 (9.41e-2)	4.1806e-1 (2.02e-1)	**1.6768e-1 (1.46e-1)**
C1_DTLZ1	3	7	2.3443e-1 (2.20e-1)	2.0369e-1 (1.89e-1)	2.7839e-1 (4.06e-1)	**3.3211e-2 (3.33e-2)**
C3_DTLZ4	3	12	**2.4542e-1 (2.54e-2)**	3.8030e-1 (5.24e-2)	6.2942e-1 (7.85e-2)	5.2015e-1 (2.54e-2)
IMOP6	3	10	1.1355e-1 (4.96e-2)	5.2240e-1 (2.27e-2)	5.9426e-2 (6.78e-2)	**3.6686e-2 (1.20e-2)**
IMOP7	3	10	1.0623e-1 (1.84e-1)	6.1033e-2 (1.06e-1)	1.1905e-2 (2.06e-2)	**0.0000e+0 (0.00e+0)**
IMOP4	3	10	6.0073e-1 (8.32e-2)	1.8725e-1 (4.37e-2)	**0.0000e+0 (0.00e+0)**	4.2231e-1 (3.04e-1)
ZDT1	2	30	9.9333e-1 (5.77e-3)	1.0000e+0 (0.00e+0)	1.0000e+0 (0.00e+0)	**4.3000e-1 (2.65e-2)**
ZDT2	2	30	1.0000e+0 (0.00e+0)	1.0000e+0 (0.00e+0)	1.0000e+0 (0.00e+0)	**6.6440e-1 (1.78e-1)**
ZDT3	2	30	6.4333e-1 (7.37e-2)	1.0000e+0 (0.00e+0)	9.9167e-1 (1.44e-2)	**2.1768e-1 (3.97e-2)**
ZDT4	2	10	1.0000e+0 (0.00e+0)	1.0000e+0 (0.00e+0)	1.0000e+0 (0.00e+0)	**1.0000e+0 (0.00e+0)**

4.1. Discussions

The ZDT, DTLZ & IMOP test suite is considered as one of the most crucial testing benchmarks sets in writing and incorporates exceptionally unimodal, multimodal, rotated, hybrid, and composite test features. The optimal Pareto front of such features is of various shapes and coherence. This subsection discusses the MaOMFO findings of these test suites and contrasts the outcomes with NSGA-III, NMPSO, and MOEA/DD. The outcomes are given in Table 1, Table 2, Table 3 & Table 4. Note that to prove the MaOMFO findings further, it is contrasted with NSGA-III, NMPSO, and MOEA/DD on this test suite with 2, 3 & 10 objectives. After careful observations of the results recorded in Tables 1-4, it is apparent that MaOMFO shows the best outcomes on most of the benchmarks and gives outstanding results when handling difficult objectives of the ZDT, DTLZ & IMOP test suite. These outcomes display the proposed MaOMFO performance when measuring the test sizes with challenges. High convergence and coverage are the principal points of interest in the proposed MaOMFO performance in correlation with NSGA-III, NMPSO, and MOEA/DD. Better convergence is expected than determining the primary solution, in which outstanding amongst other non-dominated solutions consistently update the position. Another favourable feature is the high coverage of the MaOMFO, which is a direct result of both the history support system and the choice of solution update.

The MaOMFO is based on the NSGA-III and adds the history data of individuals from earlier iterations to the production of offspring. The individuals chosen in this method may be excellent or undesirable since they are selected arbitrarily or in a predetermined manner rather than the best individuals in the population. This would somewhat slow down the algorithm's convergence rate and prevent the algorithm from reaching a local optimum. Regardless, MaOMFO can also be used to solve problems with three or more objective functions. Due to the Pareto dominated-based solution in MaOMFO turns out to be less successful for the higher objective problems. This is because of the way that in problems with many objectives, an enormous number of solutions is non-dominated, so the archive becomes full. In this way, the MaOMFO performance is reasonable for handling problems with more than four objectives. The outcomes demonstrated that MaOMFO could be extremely successful for multi/many-objective problems. A better combination of MaOMFO is expected than the moth's position update around the best non-dominated solutions with reference point mechanism. The high coverage of MaOMFO is a direct result of the archive maintenance and solution update. When the archive is full, non-dominated solutions are discarded by MaOMFO, which improves the solution diversity along the complete front. The proposed MaOMFO has the features such as high search accuracy, neighbourhood solution distribution, exploitation, and quick convergence. The MaOMFO can keep away from nearby fronts and coverage towards the best Pareto front shown in Figure 3, Figure 4, and Figure 5. The obtained results and above-all discussions demonstrate that the MaOMFO algorithm is reliable and simple too to handle the multi/many-objective optimization problems with different search spaces.

5. Conclusion

This paper proposes a simple and new many-objective algorithm called Many-Objective Moth Flame Optimizer (MaOMFO) based on the original version of the MFO, reference point strategy, and non-dominated sorting mechanism. This study presents a unique MaOMFO for processing MaOPs to enhance the overall performance in terms of convergence and diversity. An MFO algorithm with outstanding convergence ability is used as the MaOMFO's optimization procedure.

Both strong convergence speed and better diversity are characteristics of the MaOMFO algorithm. Experimental studies contrasting the MaOMFO with the other three most reputed algorithms while running them on the DTLZ, IMOP, and ZDT test cases demonstrate the MaOMFO-efficiency. The findings of the experiments reveal that the suggested MaOMFO algorithm succeeds well in most of the DTLZ, IMOP, and ZDT test cases that we investigated, and the generated solution set shows good convergence and diversity. The statistical metrics, such as GD, IGD, HV, and MS, demonstrate that the proposed MaOMFO algorithm performs better than other state-of-the-art many-objective optimizers. However, the result also demonstrates that none of the algorithms can surpass any other algorithms in any instance. This highlights the significance of making an informed decision regarding which algorithms to use while attempting to solve the MaOPs.

In addition, to further validate MaOMFO's efficacy, it is suggested to extend it to be able to tackle constrained many-objective problems by different constraint handling mechanisms. This will allow MaOMFO to handle several objectives of real-world many-objective engineering design problems simultaneously.

Acknowledgements

The authors would like to acknowledge the blind reviewers for their constructive comments to improve the paper.

Conflicts of Interest

We declare that the authors have no competing interests as defined by the publisher or other interests that might be perceived to influence the results and/or discussion reported in this paper.

References

Abbasi, M., Abbasi, E., & Mohammadi-Ivatloo, B. (2021). Single and multi-objective optimal power flow using a new differential-based harmony search algorithm. *Journal of Ambient Intelligence and Humanized Computing, 12*(1), 851–871.

Behmanesh, R., Rahimi, I., Amir, ·, & Gandomi, H. (2021). Evolutionary Many-Objective Algorithms for Combinatorial Optimization Problems: A Comparative Study. *Archives of Computational Methods in Engineering, 28,* 673–688.

Bi, X., & Wang, C. (2018). A niche-elimination operation based NSGA-III algorithm for many-objective optimization. *Applied Intelligence, 48*(1), 118–141.

Branke, J., Kaußler, T., & Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software, 32*(6), 499–507.

Brest, J., Maučec, M. S., & Bošković, B. (2017). Single objective real-parameter optimization: Algorithm jSO. *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings,* 1311–1318.

Castro, O. R., Santana, R., Lozano, J. A., & Pozo, A. (2017). Combining CMA-ES and MOEA/DD for many-objective optimization. *2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings*, 1451–1458.

Coello Coello, C. A., González Brambila, S., Figueroa Gamboa, J., Castillo Tapia, M. G., & Hernández Gómez, R. (2019). Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems, 6*(2), 221–236.

Deb, K., & Jain, H. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation, 18*(4), 577–601.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). Scalable multi-objective optimization test problems. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), 1*, 825–830.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. *In: Evolutionary Multiobjective Optimization*, 105–145.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26*(1), 29–41.

Duan, H., Zhao, W., Wang, G., & Feng, X. (2012). Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO. *Mathematical Problems in Engineering, 2012,* 712752.

Elsayed, S., & Sarker, R. (2016). Differential evolution framework for big data optimization. *Memetic Computing, 8*(1), 17–33.

Erickson, M., Mayer, A., & Horn, J. (2001). The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 1993, 681–695.

Fan, Z., Wang, T., Cheng, Z., Li, G., & Gu, F. (2017). An Improved Multiobjective Particle Swarm Optimization Algorithm Using Minimum Distance of Point to Line. *Shock and Vibration, 2017*, 25-45.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization. *Proceedings of the 5th International Conference on Genetic Algorithms*, 416–423.

Guo, X., & Wang, X. (2019). A Novel Objective Grouping Evolutionary Algorithm for Many-Objective Optimization Problems. *International Journal of Pattern Recognition and Artificial Intelligence, 34*(6), 2059018.

He, Z., & Yen, G. G. (2014). Many-Objective Evolutionary Algorithms and Hybrid Performance Metrics. *In Frontiers of Intelligent Control and Information Processing,* 335–364.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 82–87 vol.1.

Ibrahim, A., Rahnamayan, S., Martin, M. V., & Deb, K. (2016). EliteNSGA-III: An improved evolutionary many-objective optimization algorithm. *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, 973–982.

Jain, H., & Deb, K. (2014). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation, 18*(4), 602–622.

Jangir, P., Manoharan, P., Pandya, S & Sowmya, R. (2023). MaOTLBO: Many-objective teaching-learning-based optimizer for control and monitoring the optimal power flow of modern power systems. *International Journal of Industrial Engineering Computations, 14*(2), 293-308.

Johari, N. F., Zain, A. M., Noorfa, M. H., & Udin, A. (2013). Firefly Algorithm for Optimization Problem. *Applied Mechanics and Materials, 421,* 512–517.

Ke, L., Zhang, Q., & Battiti, R. (2013). MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony. *IEEE Transactions on Cybernetics, 43*(6), 1845–1859.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science, 220(4598),* 671–680.

Kumar, B. V., Oliva, D., & Suganthan, P. N. (Eds.). (2022). Differential Evolution: From Theory to Practice. 1009.

Kumar, S., Jangir, P., Tejani, G. G., & Premkumar, M. (2022). A Decomposition based Multi-Objective Heat Transfer Search algorithm for structure optimization. *Knowledge-Based Systems, 253,* 109591.

Kumar, S., Jangir, P., Tejani, G. G., Premkumar, M., & Alhelou, H. H. (2021). MOPGO: A New Physics-Based Multi-Objective Plasma Generation Optimizer for Solving Structural Optimization Problems. *IEEE Access, 9,* 84982–85016.

Li, B., Li, J., Tang, K., & Yao, X. (2015). Many-Objective Evolutionary Algorithms. *ACM Computing Surveys (CSUR), 48(1),* 2792984.

Li, H., & Zhang, Q. (2009). Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation, 13*(2), 284–302.

Li, M., Yang, S., Liu, X., & Shen, R. (2013). A comparative study on evolutionary algorithms for many-objective optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7811 LNCS*, 261–275.

Lin, Q., Liu, S., Zhu, Q., Tang, C., Song, R., Chen, J., Coello, C. A. C., Wong, K. C., & Zhang, J. (2018). Particle Swarm Optimization with a Balanceable Fitness Estimation for Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation, 22*(1), 32–46.

Liu, Y., Gong, D., Sun, X., & Zhang, Y. (2017). Many-objective evolutionary optimization based on reference points. *Applied Soft Computing Journal, 50,* 344–355.

Luo, W., Shi, L., Lin, X., & Coello Coello, C. A. (2019). The g-dominance Relation for Preference-Based Evolutionary Multi-Objective Optimization. *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings,* 2418–2425.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89,* 228–249.

Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications, 27*(4), 1053–1073.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software, 69,* 46–61.

Mousa, A. A., El-Shorbagy, M. A., & Abd-El-Wahed, W. F. (2012). Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. *Swarm and Evolutionary Computation, 3,* 1–14.

Premkumar, M., Jangir, P., Kumar, B. S., Alqudah, M. A., & Nisar, K. S. (2022). Multi-objective grey Wolf optimization algorithm for solving real-world bldc motor design problem. *Computers, Materials and Continua, 70*(2), 2435–2452.

Premkumar, M., Jangir, P., Santhosh Kumar, B., Sowmya, R., Haes Alhelou, H., Abualigah, L., Riza Yildiz, A., Mirjalili, S., & Premkumar, M. (2021). A New Arithmetic Optimization Algorithm for Solving Real-World Multiobjective CEC-2021 Constrained Optimization Problems: Diversity Analysis and Validations. *IEEE Access, 9,* 84263–84295.

Premkumar, M., Jangir, P., & Sowmya, R. (2021). MOGBO: A new Multiobjective Gradient-Based Optimizer for real-world structural optimization problems. *Knowledge-Based Systems, 218,* 106856.

Premkumar, M., Jangir, P., Sowmya, R., & Elavarasan, R. M. (2021). Many-Objective Gradient-Based Optimizer to Solve Optimal Power Flow Problems: Analysis and Validations. *Engineering Applications of Artificial Intelligence, 106,* 104479.

Premkumar, M., Kumar, C., Sowmya, R., & Pradeep, J. (2021). A novel salp swarm assisted hybrid maximum power point tracking algorithm for the solar photovoltaic power generation systems. *Automatika, 62*(1), 14-25.

Premkumar, M., Sowmya, R., Jangir, P., Haes Alhelou, H., Heidari, A. A., & Huling Chen. (2021). MOSMA: Multi-Objective Slime Mould Algorithm Based on Elitist Non-Dominated Sorting. *IEEE Access, 9,* 3229–3248.

Premkumar, M., Sowmya, R., Jangir, P., & Siva Kumar, J. S. V. (2020). A New and Reliable Objective Functions for Extracting the Unknown Parameters of Solar Photovoltaic Cell Using Political Optimizer Algorithm. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*, 1-8.

Premkumar, M., & Sumithira, R. (2018). Humpback whale assisted hybrid maximum power point tracking algorithm for partially shaded solar photovoltaic systems. *Journal of Power Electronics, 18*(6), 1805–1818.

Rao, R. V. (2020). Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *International Journal of Industrial Engineering Computations, 11*(1), 107–130.

Schaffer, J. D. (1984). Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms (Artificial Intelligence, Optimization, Adaptation, Pattern Recognition).

Sierra, M., & Coello, C. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ∈-Dominance. *Lecture Notes in Computer Science*, 3410.

Stanovov, V., Akhmedova, S., & Semenkin, E. (2020). Differential Evolution with Linear Bias Reduction in Parameter Adaptation. *Algorithms, 13*(11)*,* 283.

Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A matlab platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine, 12*(4), 73–87.

Vachhani, V. L., Dabhi, V. K., & Prajapati, H. B. (2016). Improving NSGA-II for solving multi objective function optimization problems. *2016 International Conference on Computer Communication and Informatics, ICCCI 2016*, 1-8.

Venkata Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations, 7*(1), 19–34.

Vesikar, Y., Deb, K., & Blank, J. (2019). Reference Point Based NSGA-III for Preferred Solutions. *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018,* 1587–1594.

Wang, S., Zhou, Y., & Zhang, J. (2018). An Improved NSGA-III Approach to Many-Objective Optimal Power Flow Problems. *2018 Chinese Automation Congress (CAC),* 2664–2669.

Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions On Evolutionary Computation, 1*(1), 67–82.

Yang, X.-S. (2021). Genetic Algorithms. *Nature-Inspired Optimization Algorithms*, 91–100.

Zhang, J., Wang, S., Tang, Q., Zhou, Y., & Zeng, T. (2019). An improved NSGA-III integrating adaptive elimination strategy to solution of many-objective optimal power flow problems. *Energy, 172,* 945–957.

Zhang, Q., & Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation, 11*(6), 712–731.

Zheng, W., Tan, Y., Meng, L., & Zhang, H. (2018). An improved MOEA/D design for many-objective optimization problems. *Applied Intelligence, 48*(10), 3839–3861.

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhangd, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation, 1*(1), 32–49.

Zhou, A., Zhang, Q., & Zhang, G. (2012). A multiobjective evolutionary algorithm based on decomposition and probability model. *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, 1-8.

Zhou, J., Wang, C., Li, Y., Wang, P., Li, C., Lu, P., & Mo, L. (2017). A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security. *Applied Mathematical Modelling, 45,* 684–704.