Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **8**, Number **4**, November **2023**

Tolerance-based Elite Mutation Firefly Algorithm

Juan Chen
School of Information Engineering
Nanchang Institute of Technology
Nanchang Key Laboratory of IoT Perception and Collaborative Computing for Smart City
1989058628@qq.com

Ren-Xian Zeng
School of Information Engineering
Nanchang Institute of Technology
zengrx@163.com

Shen-Yu Qiu
College of Science
Nanchang Institute of Technology
931761781@qq.com

Ping Kang
School of Information Engineering
Nanchang Institute of Technology
1721621494@qq.com

Jia Zhao*
School of Information Engineering
Nanchang Institute of Technology
Nanchang Key Laboratory of IoT Perception and Collaborative Computing for Smart City
zhaojia925@163.com

*Corresponding author: Jia Zhao

Received July 17, 2023, revised September 19, 2023, accepted October 18, 2023.

Abstract. *For the firefly algorithm in the solution of complex functions are prone to low accuracy and "premature" phenomenon. In this paper, we propose a tolerance-based elite mutation firefly algorithm (MOFA-TEM). The algorithm employs non-dominated sorting and congestion distance to select the elite leader , which is fused with the individual historical optimal solution to form a more comprehensive integrated information of the population, and jointly guide the movement of the fireflies, which not only enhances the interactivity of the information between fireflies in the population, but also improves the algorithm's ability of global exploration. Secondly, a tolerance-based firefly state judgment mechanism is designed, which will exponentially increase the probability of updating the elite leader as the number of individual firefly stagnation increases, which both fully exploits the potential of 's search for excellence and prevents ineffective searching of the population. Finally, if the firefly triggers the tolerance mechanism, the elite mutation strategy will be used to update the and select a new elite learning object to avoid the algorithm from stagnating. Experiments on 19 test functions and validation using Friedman's test show that MOFA-TEM gives better results compared to other optimization algorithms.*
Keywords: Firefly algorithm; elite learning; tolerance mechanism; elite mutation

---

1. **Introduction.** Multi-objective optimization problems (MOPs) [1] is a category of problems with multiple conflicting optimization objective functions, without the existence of all the objectives optimal at the same time, so the MOPs is usually obtained as a set of compromise solutions, the Pareto solution set (PS) [2]. When tackling intricate MOPs, traditional optimization approaches suffer from the disadvantages of low solution accuracy, slow convergence, long time-consumption, and high cost. With the improvement of the requirements for the solution of MOPs and the limitations of traditional optimization algorithms in solving complex problems, many scholars have proposed swarm intelligence algorithm (SIA) [3], which has been rapidly developed in the past two decades and has become one of the most active algorithmic research fields at present. Swarm intelligence is the property of individuals with simple intelligence to exhibit group intelligent behavior through mutual collaboration and organization. As a bio-heuristic algorithm, SIA expands the search scope and enhances the search efficiency through competition, learning, and interaction among groups to find better individuals. Because of its global search performance, it is suitable for solving complex MOPs and has become a mainstream method for MOPs. It has been successfully applied in the fields of transportation network prediction [4], remote sensing imagery [5], route planning [6], and Internet of Things [7]. Nowadays, there are many novel SIA, such as the tumbleweed optimization algorithm [8], the krill herd algorithm [9], the ant lion optimizer algorithm [10], the crow search algorithm [11], the Phasmatodea Population Evolution algorithm [12], and the bamboo forest growth optimization algorithm [13].

Yang proposed the multi-objective firefly algorithm (MOFA) [14]. The idea of the algorithm comes from the biological habit of glowing fireflies in nature to fly towards fireflies brighter than themselves. Fireflies are discretely distributed in a certain activity range, each firefly emits light of varying intensity due to different amounts of fluorescein in its body, and over time, the fireflies in the range gradually gather near the brightest fireflies to form a number of aggregation centers of similar brightness, similar to the process of searching for superiority. Although MOFA is simple, efficient and easy to implement, and has found extensive applications across various domains such as feature selection and neural network training [15, 16, 17, 18], it still suffers from the defects of falling into local optimum, slow convergence speed, and poor convergence accuracy. For this reason, scholars have made many optimizations and improvements to MOFA, which are studied in the following three areas: (1) Optimization and improvement of the iterative strategy of the algorithm, including parameter adjustment and iterative solution optimization. Zhao et al. [19] enhances population initialization and introduces the Maximin learning strategy, which accelerates convergence and expands the exploration area of the population. He et al. [20] improves the convergence speed of the algorithm by introducing inertia weights based on the attention mechanism in the learning formula. (2) Optimizing and improving the learning structure of the algorithm to utilize more population information. Zhao et al. [21] creates a central particle derived from the historically optimal position, subject it to one-dimensional deep learning iterations, and let the acquired knowledge guide the population's evolution. Lai et al. [22] not only divides the populations, but also divides the evolutionary stages and use different learning strategies at different stages of learning. Zhao et al. [23] divides fireflies into self-learning particles and ordinary particles; self-learning particles use 3 learning strategies to generate candidate solutions and select the optimal solution, while ordinary particles learn from 2 particles that outperform them. (3) Integration of MOFA with other algorithms to improve the MOFA by utilizing the advantages of other algorithms. Aydilek [24] proposes a hybrid algorithm combining FA and PSO, combining the strengths of both algorithms, PSO is employed for global search while FA is utilized for local search.

In summary, although the above improved algorithm enhances the exploration capability to a certain extent, there is still room for improvement in effectively balancing convergence and diversity. We propose a tolerance-based elite mutation firefly algorithm (MOFA-TEM). The algorithm during the search for the best solution, each firefly exists individual behavior and group behavior, fireflies through the learning of the population's flight experience and draw on their own flight experience to discover the best solution, and save the optimal information searched for in the individual optimum, not only to maximize the utilization of the population's high-quality information, to improve the quality of the elite leader, and to ensure that the individual and the population of the co-evolutionary. In order to monitor the movement status of fireflies, a tolerance-based judgment mechanism is introduced. As the tolerance of individual fireflies, that is, the frequency of stagnation occurrences rises, the probability of updating the elite leader of the population will increase exponentially, which will help the population to update the learning object at the right time and ensure that the population maximizes the utilization of the learning potential of the elite leader without falling into stagnation. Finally, the elite mutation strategy is used to update the elite leader , generating a new learning object to prevent the algorithm from being stuck in local optima.

2. **MOFA.**

The MOFA draws inspiration from the natural behavior of fireflies, with two key elements: brightness and attractiveness. The attractiveness is defined as follow:

$$\beta_{ij} = \beta_0 \cdot e^{-\gamma \cdot r_{ij}} \tag{1}$$

where β_0 is the most attractive, usually $\beta_0 = 1$; γ is the light absorption coefficient, usually $\gamma \in [0.01, 100]$; r_{ij} is the Euclidean distance between any two fireflies.

Assuming that firefly i moves towards firefly j, the movement formula of firefly i is as follows:

$$X_i(t+1) = X_i(t) + \beta_{ij}(r_{ij})(X_j(t) - X_i(t)) + \alpha \cdot \varepsilon_i \tag{2}$$

where t represents the current iteration number; α is the step factor, usually take the random number between $[0, 1]$; ε_i represents a random number vector.

3. **MOFA-TEM.**

3.1. **Improvement of Learning Strategy.**

In standard MOFA, as shown in Formula (2), the firefly is only affected by the firefly that dominates it and the random term, so the selection of a better quality learning object for the firefly plays a crucial role in accelerating the convergence of the population and improving the accuracy of the solution. MOFA-TEM updates the position of fireflies through the elite leader *gbest* and the individual history of optimal *pbest* . *gbest* records the optimal position of the population throughout the search phase, and each firefly saves the optimal information of the search in it, so its structure is superior in multiple dimensions. *pbest* stores the best position of each firefly throughout the search history, incorporating the firefly's own experience of the search. MOFA-TEM searches for potentially better solutions in its neighborhood centered on the firefly's own current position, and add *gbest* and *pbest* to guide the search towards achieving optimal results, learning from the population and its own experience, preventing the firefly from conducting ineffective searches and wasting the number of evaluations in other directions. The position update formula follows:

$$X_i(t+1) = wX_i(t) + a_1\beta_{gi}(gbest - X_i(t)) + a_2\beta_{pi}(pbest_i - X_i(t)) + alpha(t) \tag{3}$$

where ω is the weighting factor; $gbest$ is elite leader; $pbest_i$ is the historical optimum of firefly i ; β_{ig} is the attraction between fireflies i and $gbest$; β_{pi} is the attraction between fireflies i and $pbest$; $a_1, a_2 \in (0,1)$ and $a_1 + a_2 = 1$; $alpha(t) = 0.9 \cdot alpha(t-1)$, $alpha(1) = 0.2$, is a random perturbation factor that reduces as the iterations progress.

3.2. **Tolerance-based firefly state judgment mechanism.** In MOFA-TEM, the optimization search direction of fireflies is determined by $gbest$ and $pbest$, and fireflies in the population complete the position iteration by learning from the superior individuals, and if $gbest$ is a locally optimal position in the solution space, other fireflies in the population may stagnate together with $gbest$ after several searches. To avoid stagnation of the population, this paper introduces a tolerance-based firefly state judgment mechanism, by judging the current evolutionary state of the firefly and updating the firefly's learning object $gbest$ at the appropriate time, to avoid the firefly falling into the "precocious maturity" due to the wrong search direction.

The tolerance-based firefly state judgment mechanism defines tolerance as the number of times a firefly has been in stagnation, denoted as S_i . If the tolerance S_i increases by 1, it means that firefly individuals learn from the current $gbest$, and the fitness value does not get better, that is, the current $gbest$ is not conducive to population evolution. Obviously, the larger the S_i value, the greater the number of individual firefly stagnations and the greater the probability that the population's evolutionary capacity is in stagnation. Thus MOFA-TEM determines the evolutionary status of each individual firefly by using the individual's own evolutionary experience to update the elite leader $gbest$ according to the tolerance probability. For firefly i , the probability of updating $gbest$ is expressed as :

$$P_i = \frac{\exp(S_i - 1)}{\exp(5) - 1} \tag{4}$$

This algorithm determines the referentiality of $gbest$ according to the method of increasing P_i exponentially with increasing tolerance S_i . When $P_i > rand()$, the elite leader of the current firefly is considered to be less referential and highly susceptible to stagnation, therefore stops learning from the current learning object $gbest$. Conversely, if $P_i < rand()$, consider the current firefly's elite leader to still be referential and continue to learn from the current learning object $gbest$.

The probability P_i will be recalculated after each position update, and the probability P_i of the MOFA-TEM adjustment updating $gbest$ increases exponentially as the tolerance S_i increases. When S_i is smaller, meaning that the number of stagnation is smaller, the value of P_i is also smaller, indicating that the population learns with a higher probability still toward the current learning object $gbest$, fully exploiting the potential of the current search direction. When S_i is large, meaning that the number of stagnations is high, the value of P_i increases exponentially, indicating that the current direction of search is almost uninformative and the population should stop learning from the current learning object $gbest$.

3.3. **Elite Mutation.** The elite leader $gbest$ as the elite solution that guides the evolution of the population and is the object of study for all fireflies in the population. When a firefly triggers the tolerance-based firefly state judgment mechanism, it indicates that the old learning object cannot lead the population to evolve. Therefore, a mutation operation is executed on the elite leader $gbest$, and if the mutated new individual $gbest^*$ has a superior fitness value compared to the original $gbest$ fitness value, then $gbest^*$ replaces $gbest$ and participates in a new round of evolutionary process. The new elite leader $gbest^*$ will act as a new learning object during subsequent evolution, guiding the population movement and thus helping stagnant individuals to escape from the local optimum. The formula follows:

$$gbest_j^* = unifrnd(gl, gu) \tag{5}$$

$$\begin{cases} gl = gbest_j - rand() \cdot (VarMax - VarMin) \\ gu = gbest_j + rand() \cdot (VarMax - VarMin) \end{cases} \tag{6}$$

where j denotes the j th dimension of $gbest$; $unifrnd(gl, gu)$ denotes a random number between $[gl, gu]$; $rand()$ denotes a random number between $[0, 1]$; $VarMax$, $VarMin$ denote the upper and lower limits of the firefly.

3.4. **Algorithm description.** Algorithm 1 gives the pseudocode for MOFA-TEM. Firstly, the firefly population was randomly initialized, the fitness value of each firefly was calculated, and $gbest$ was selected according to the non-dominated sorting and crowding distance. In the iteration, the fireflies update their positions according to the improved learning Formula (3) and record the historical optimal solution for each firefly. According to Formula (4), the probability of updating $gbest$ for each firefly is calculated based on the tolerance, if $P_i > rand()$, a new learning object $gbest^*$ is generated according to Formula (5), and if $gbest^*$'s fitness value is better than $gbest$, $gbest$ is updated. Otherwise, learning continues according to Formula (3) until the condition is satisfied. The basic flow of MOFA-TEM is shown in Figure 1.

Algorithm 1 Algorithm1: Pseudo-code of MOFA-TEM

```
Input: Population size N_pop , maximum iterations MaxIt , external archive Rep
Output: PS
Initialize firefly population
Calculate the fitness value of fireflies
while (t ≤ MaxIt) do
   Selecting gbest based on non-dominated sorting and congestion distance
   for i = 1 : N_pop do
      Move firefly i according to Formula (3)
      Calculate the tolerance S_i of firefly i
      P_i is calculated according to according to Formula (4)
      if (P_i > rand()) then
         Generate a new gbest* according to Formula (5)
      end if
   end for
   Update the PS in Rep
   if (Rep is full) then
      Remove overflow solutions according to the congestion distance mechanism
   end if
end while
return PS
```

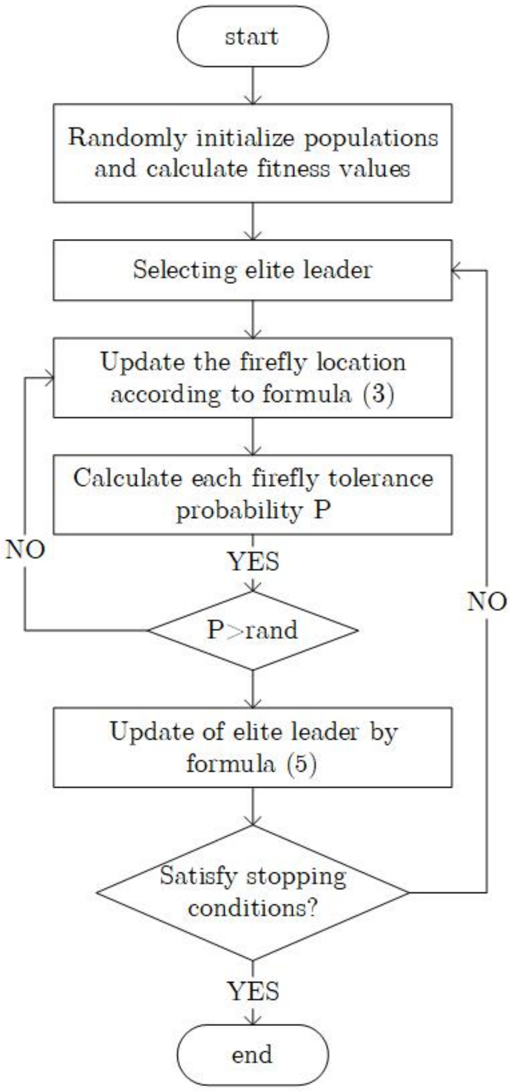


FIGURE 1. Flowchart of MOFA-TEM

4. **Simulation experiment and result analysis.** In order to validate the performance of MOFA-TEM in dealing with MOPs, section 4.2 compares MOFA-TEM with 5 classical multi-objective evolutionary algorithms (MOEAs) [25], namely PESA-II [26], MOPSO [27], MOEA/D [28], MOFA [14] and NSGA-III [29]. Section 4.3 compares MOFA-TEM with seven recent MOEAs, namely MONSFA [30], MOEA/D-ACD [31], NSGA-II-SDR [32], CFMOFA [33], Top [34], DGEA [35] and HVMA-M [36].

4.1. **Test Functions.** In order to test the ability of MOFA-TEM to solve different problems, 19 test functions from DTLZ, Viennet, Deb and ZDT are selected for comparative analysis in this paper, as Table 1. The ZDT and Deb test sets are two-objective test problems, and the Viennet and DTLZ test sets are three-objective test functions. These test functions have different properties and complex Pareto front features, such as concavity, multimodality, and irregular Pareto front shapes, which can effectively demonstrate the algorithm's reliability and efficiency.

Table 1. Test function set

Problem	Objective	Constraints
ZDT1	2	$n = 30, 0 \le x_i \le n$
ZDT2	2	$n = 30, 0 \le x_i \le n$
ZDT3	2	$n = 30, 0 \le x_i \le n$
ZDT4	2	$n = 10, 0 \le x_1 \le 1$, -5 $\le x_i \le 5, i = 2, \cdots, n$
ZDT6	2	$n = 10, 0 \le x_i \le 1, i = 1, \cdots, n$
Deb1	2	$0 \le x_1, x_2 \le 1$
Deb2	2	$0 \le x_1, x_2 \le 1$
Deb3	2	$0 \le x_1, x_2 \le 1$
Viennet1	3	$-2 \le x, y \le 2$
Viennet2	3	$-4 \le x, y \le 4$
Viennet3	3	$-3 \le x, y \le 3$
Viennet4	3	$-3 \le x, y \le 3$
DTLZ1	3	$n = 7, 0 \le x_i \le 1, i = 1, \cdots, n$
DTLZ2	3	$n = 12, 0 \le x_i \le 1, i = 1, \cdots, n$
DTLZ3	3	$n = 12, 0 \le x_i \le 1, i = 1, \cdots, n$
DTLZ4	3	$n = 12, 0 \le x_i \le 1, i = 1, \cdots, n$
DTLZ5	3	$n = 12, 0 \le x_i \le 1, i = 1, \cdots, n$
DTLZ6	3	$n = 12, 0 \le x_i \le 1, i = 1, \cdots, n$
DTLZ7	3	$n = 12, 0 \le x_i \le 1, i = 1, \cdots, n$

4.2. **Comparison with classical MOEAs.** MOFA-TEM was compared with 5 classical MOEAs, and the algorithm parameters are specified in Table 2. The inverted generation distance (IGD) [37] is an evaluation metric used to assess algorithm convergence and diversity. A smaller IGD indicates better comprehensive performance of the algorithm. The results of the experiment were statistically calculated by Friedman's test to see if there was a significant difference between the methods. N_{pop} is 50, Rep is 100, $MaxIt$ is 300, the algorithm is executed in 30 independent runs, and the mean (Mean) and standard deviation (Std) are recorded, shown in Table 3. Where Total denotes the number of optimal times, Ranking denotes the ranking mean obtained by Friedman's test, Final rank denotes the ranking of the algorithm, and the shaded data in the table denote the optimal value.

Table 2. Algorithm parameter setting

Algorithm	Parameter setting	Reference
PESA-II	$w = 0.4, c_1, c_2 = \text{rand}[0, 1]$	Corne 2001
MOPSO	$pCrossover = 0.5$ $nCrossover = 2 * \text{round}, pCrossover * N_{\text{pop}}/2$	Colleo 2004
MOEA/D	$\gamma = 0.5$	Zhang 2007
MOFA	$pCrossover = 0.5, \beta = 1, \gamma = 2$ $nCrossover = 2 * \text{round}[pCrossover * N_{\text{pop}}/2]$	Yang 2013
NSGA-III	$\alpha = 0.2, \beta_0 = 1, \gamma = 1$	Deb 2014
MOFA-TEM	$alpha(1) = 0.2, alpha(t) = 0.9 \cdot alpha(t-1), \beta_0 = 1, \gamma = 1$	

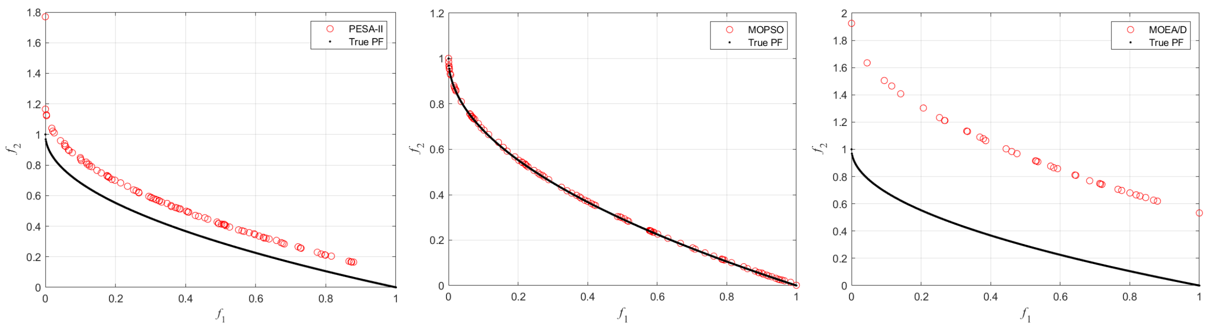
As shown in Table 3, for the ZDT, Deb, and Viennet test sets, MOFA-TEM obtains the IGD optimum, and on the ZDT test set not only the IGD optimums were obtained, but they were all 1-2 orders of magnitude higher than the suboptimal values. For the DTLZ test set, MOFA-TEM obtains the IGD optimum on the DTLZ5-DTLZ7 test functions. On the DTLZ1 and DTLZ4 test problems, MOFA-TEM did not obtain optimal values, but its mean IGD values: 1.76E+02 and 7.39E-01, are of the same order of magnitude (10^2,10^{-1}) as the optimal mean IGD values: 1.47E+02 and 1.05E-01, indicating that there is not much difference between the values. Among them, PESA-II and NSGA-III each obtained 1 IGD optimum, MOFA obtained 2 IGD optimums, and neither MOPSO nor MOEA/D obtained one IGD optimum. MOFA-TEM obtained the smallest Ranking value of 1.84, indicating a more significant overall performance of MOFA-TEM on these test functions. In summary, from the IGD and Friedman test results, MOFA-TEM has better overall performance compared to the comparison algorithms, and better convergence and diversity of solutions can be obtained in solving the MOPs.

Table 3. Comparison of MOFA-TEM and 5 classical MOEAs on IGD

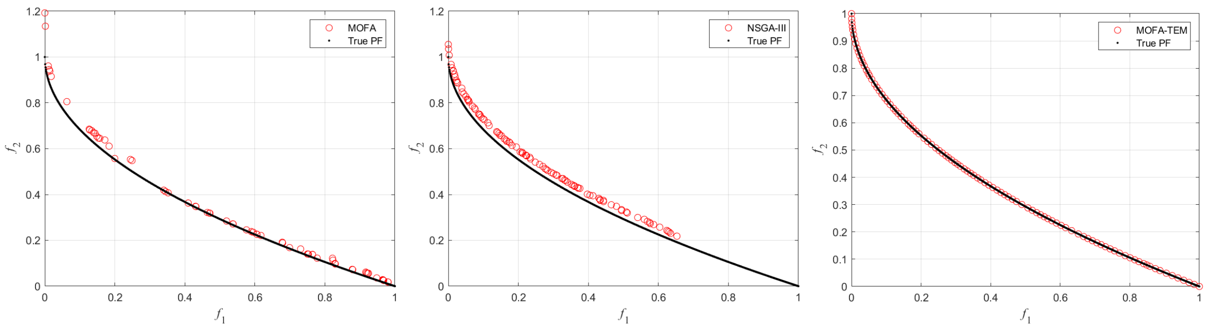
Instances	PESA-II	MOPSO	MOEA/D
	Mean±Std.	Mean±Std.	Mean±Std.
ZDT1	1.33E-01±1.93E-02	1.09E-02±2.32E-03	3.14E-01±1.24E-01
ZDT2	1.03E-01±2.34E-02	7.28E-01±8.78E-01	1.45E+00±5.16E-01
ZDT3	1.49E-01±3.07E-02	5.00E-02±7.28E-02	5.17E-01±2.55E-01
ZDT4	1.48E+00±7.31E+00	5.48E+00±6.40E+00	4.95E+00±2.03E+00
ZDT6	1.33E-02±5.85E-03	1.02E-01±1.44E-01	2.85E+00±1.15E+00
Deb1	9.55E-03±1.47E-03	8.37E-03±9.03E-04	1.56E-02±3.28E-03
Deb2	9.60E-03±1.30E-03	8.64E-03±4.57E-04	8.06E-02±4.47E-02
Deb3	8.12E-03±1.33E-03	7.42E-03±5.76E-04	3.19E-02±1.76E-02
Viennet1	1.58E-01±7.37E-03	1.55E-01±6.84E-03	4.69E-01±4.74E-02
Viennet2	1.93E-02±3.85E-03	2.15E-02±2.46E-03	1.38E-01±6.22E-02
Viennet3	9.53E-02±1.65E-02	7.03E-02±1.01E-02	2.41E+00±1.77E-01
Viennet4	1.96E-01±2.38E-02	2.21E-01±2.17E-02	8.47E-01±1.64E-01
DTLZ1	1.71E+02±4.12E+00	1.71E+02±2.77E+00	1.57E+02±5.43E-01
DTLZ2	8.80E-02±2.76E-03	9.52E-02±3.56E-03	1.08E-01±1.22E-02
DTLZ3	1.68E+02±2.31E+01	1.99E+02±5.24E+00	1.32E+02±3.69E+02
DTLZ4	**1.05E-01±1.01E-02**	1.06E-01±9.37E-03	5.58E-01±7.00E-04
DTLZ5	7.14E-01±8.36E-02	7.01E-01±1.26E-01	1.28E+00±2.95E-01
DTLZ6	3.88E+00±5.23E-01	1.89E+00±1.00E+00	3.17E+00±5.17E-01
DTLZ7	3.05E-01±6.65E-02	8.31E-02±5.68E-03	2.19E+00±1.03E+00
Total	1	0	0
Ranking	3.29	3.03	5.21
Final	3	2	6

Instances	MOFA	NSGA-III	MOFA-TEM
	Mean±Std.	Mean±Std.	Mean±Std.
ZDT1	2.62E-02±3.69E-03	4.70E-02±7.06E-03	**3.87E-03±1.55E-05**
ZDT2	4.37E-02±2.14E-02	1.35E-01±3.40E-01	**3.94E-03±3.88E-05**
ZDT3	3.35E-02±1.04E-02	7.65E-02±3.77E-02	**4.44E-03±3.02E-05**
ZDT4	1.73E-01±1.52E-01	1.98E+00±8.02E-01	**3.85E-03±2.95E-05**
ZDT6	2.63E-01±5.01E-02	1.47E-01±1.93E-01	**4.30E-03±1.20E-05**
Deb1	1.82E-02±5.15E-03	1.81E-02±9.46E-03	**3.85E-03±2.75E-06**
Deb2	1.46E-02±1.16E-03	3.79E-02±1.98E-02	**3.85E-03±2.75E-06**
Deb3	1.64E-02±3.46E-03	1.66E-02±8.87E-03	**3.97E-03±1.89E-04**
Viennet1	1.67E-01±4.94E-03	3.09E-01±5.43E-02	**1.30E-01±2.42E-03**
Viennet2	3.42E-02±3.39E-03	2.94E-02±6.03E-03	**1.21E-02±1.36E-04**
Viennet3	9.77E-01±6.33E-01	2.36E+00±4.15E-01	**6.44E-02±1.02E-02**
Viennet4	2.57E-01±2.66E-02	2.69E-01±1.90E-02	**1.71E-01±6.41E-03**
DTLZ1	**1.47E+02±1.66E+00**	1.56E+02±4.37E+00	1.76E+02±2.59E+00
DTLZ2	**6.87E-02±1.41E-03**	1.27E-01±2.49E-02	3.11E-01±2.98E-02
DTLZ3	9.40E+01±4.19E+01	**1.12E+01±4.27E+00**	1.23E+02±9.50E+00
DTLZ4	7.81E-01±1.31E-01	3.62E-01±3.34E-01	7.39E-01±2.00E-02
DTLZ5	5.95E-01±1.09E-01	8.35E-01±4.17E-01	**5.04E-01±4.87E-02**
DTLZ6	7.68E+00±4.91E-01	5.22E+00±3.34E+00	**6.69E-01±1.22E-01**
DTLZ7	1.48E-01±3.18E-02	2.44E-01±1.36E-01	**6.05E-02±6.69E-04**
Total	2	1	15
Ranking	3.47	4.16	1.84
Final	4	5	1

To more visually compare the convergence and distribution of MOFA-TEM with PESA-II, MOPSO, MOEA/D, MOFA, and NSGA-III, Pareto front fitting plots were plotted, as in Figure 2-Figure 14. where the red circles denote the solution sets obtained by the corresponding algorithms and the black regions denote the Pareto front. The closer the red circles are to the black region, the better the algorithm converges, and the more evenly the red circles are distributed over the black region, the better the algorithm is distributed. From the figure, MOFA-TEM has better optimization performance compared to all other algorithms in obtaining PS with higher accuracy and uniform distribution on different test functions.

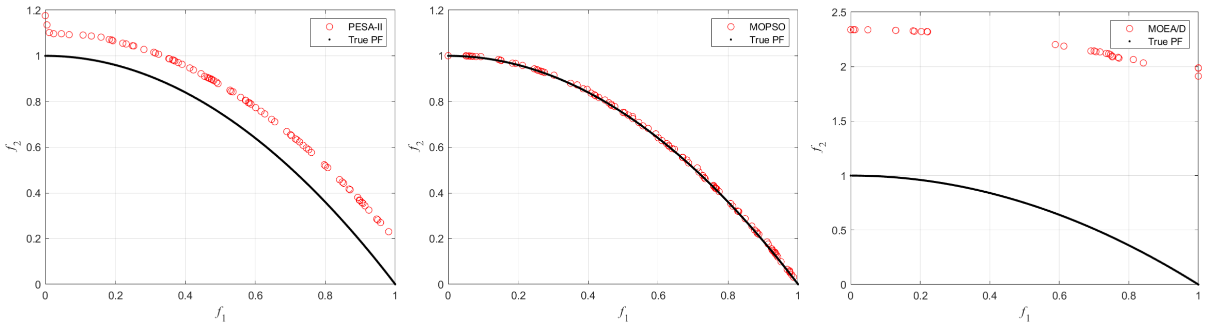


(a) PESA-II (b) MOPSO (c) MOEA-D

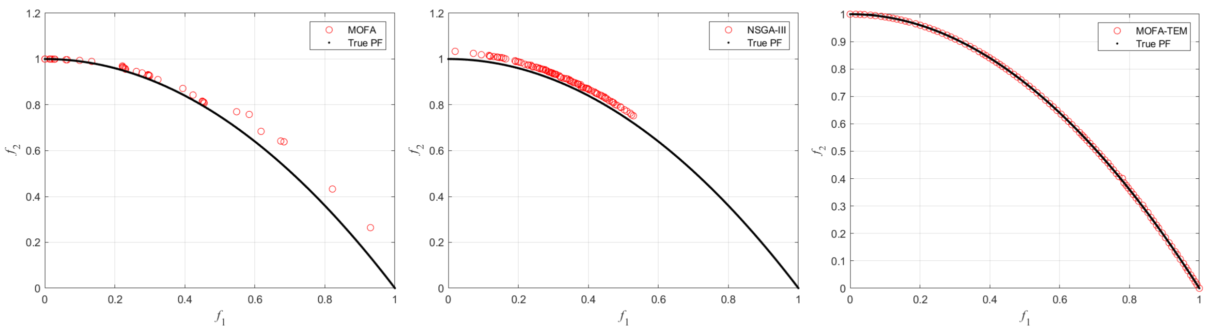


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 2. Frontier fitting plot for ZDT1

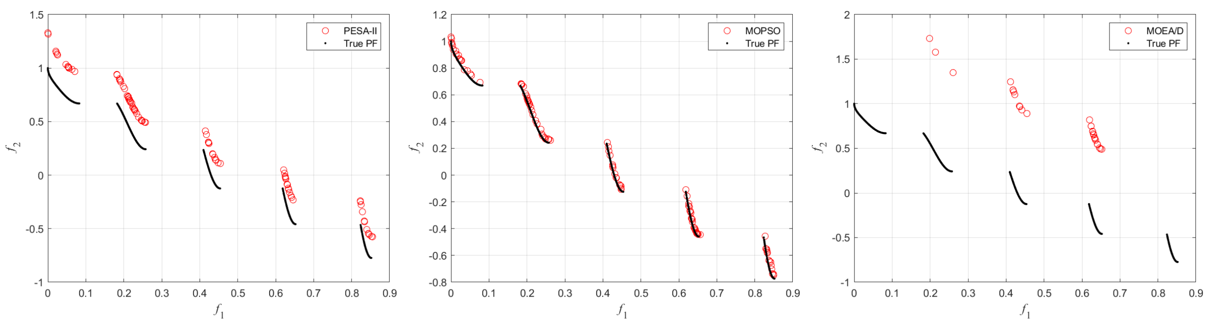


(a) PESA-II (b) MOPSO (c) MOEA-D

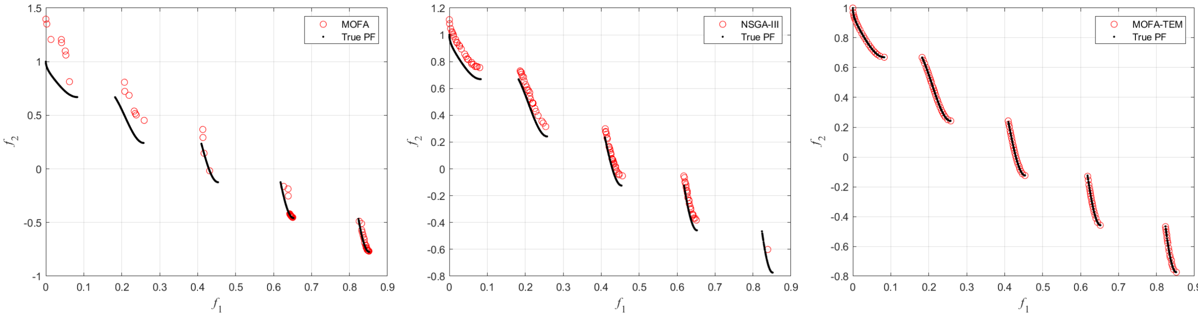


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 3. Frontier fitting plot for ZDT2

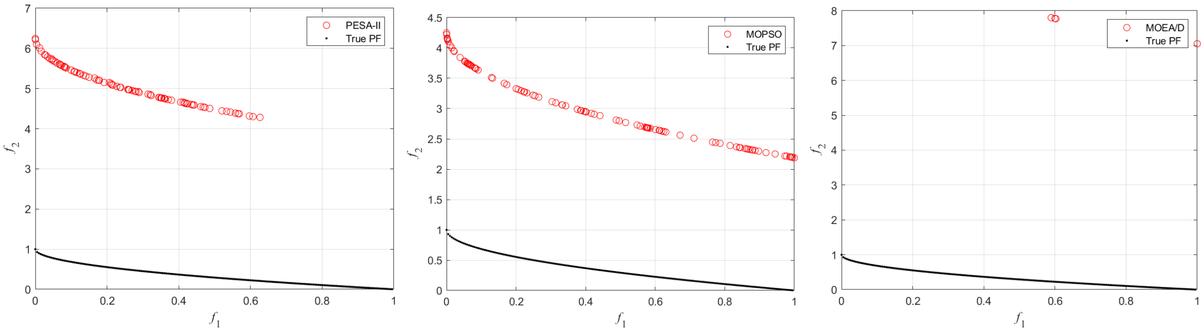


(a) PESA-II (b) MOPSO (c) MOEA-D

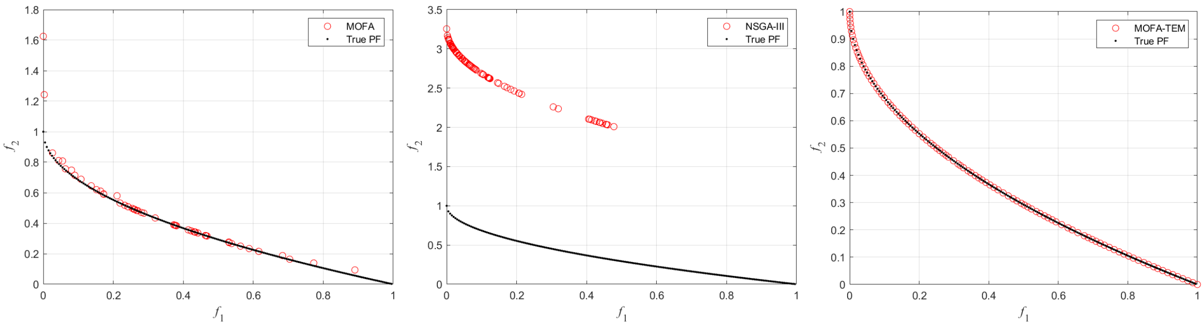


(d) MOFA (e) NSGA-III (f) MOFA-TEM

Figure 4. Frontier fitting plot for ZDT3

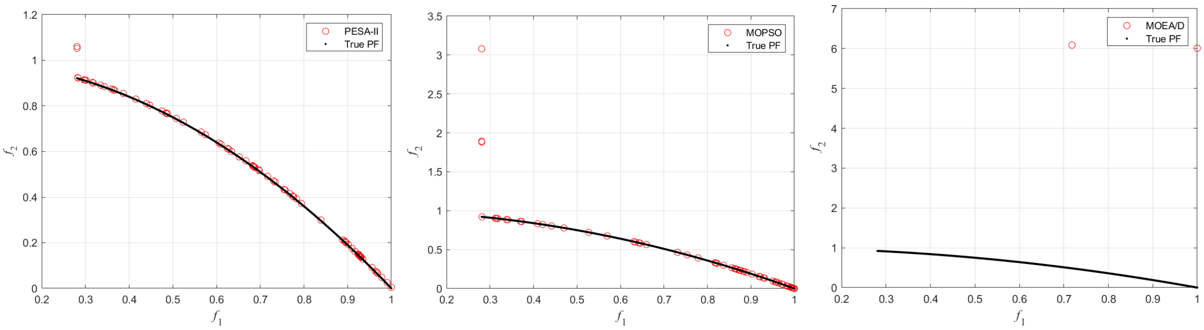


(a) PESA-II (b) MOPSO (c) MOEA-D

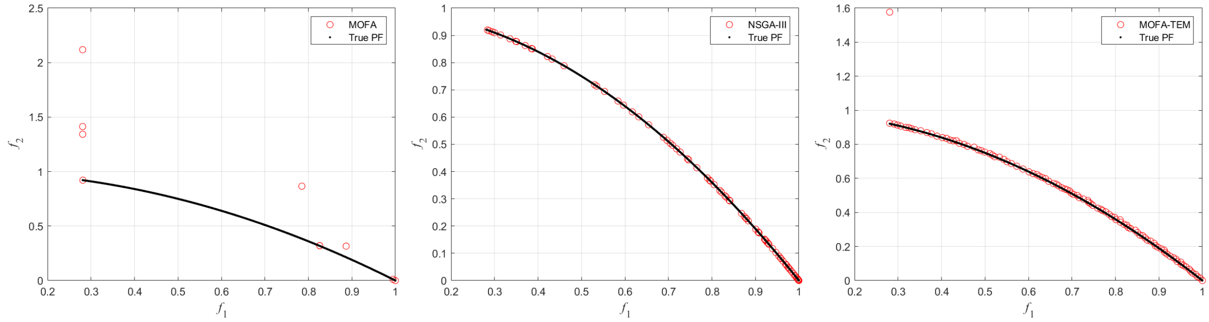


(d) MOFA (e) NSGA-III (f) MOFA-TEM

Figure 5. Frontier fitting plot for ZDT4

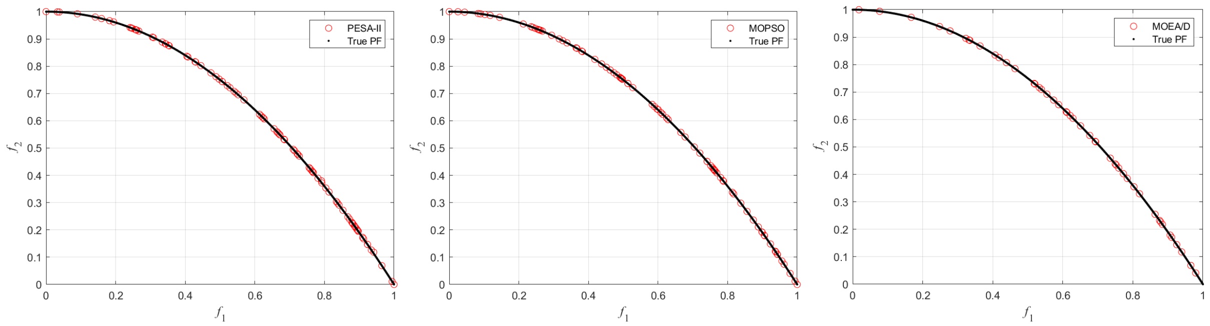


(a) PESA-II (b) MOPSO (c) MOEA-D

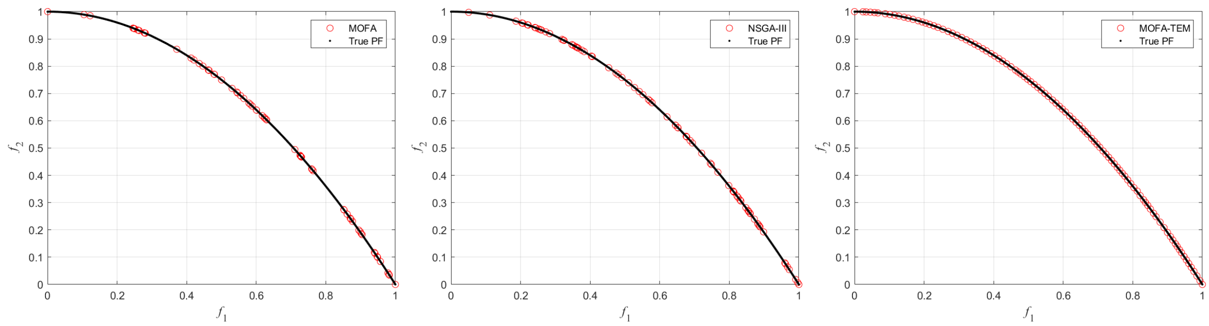


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 6. Frontier fitting plot for ZDT6

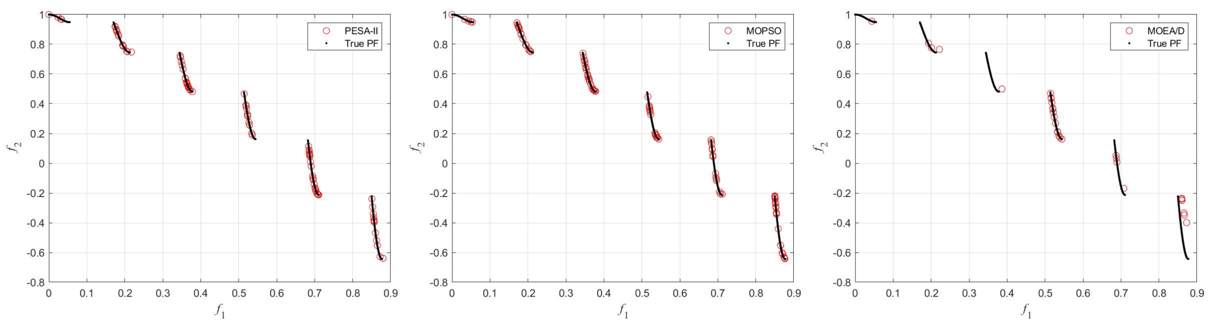


(a) PESA-II (b) MOPSO (c) MOEA-D

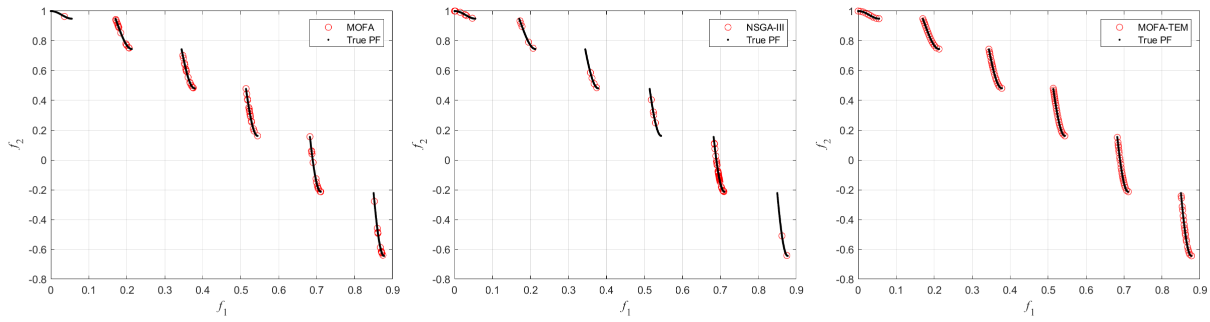


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 7. Frontier fitting plot for Deb1

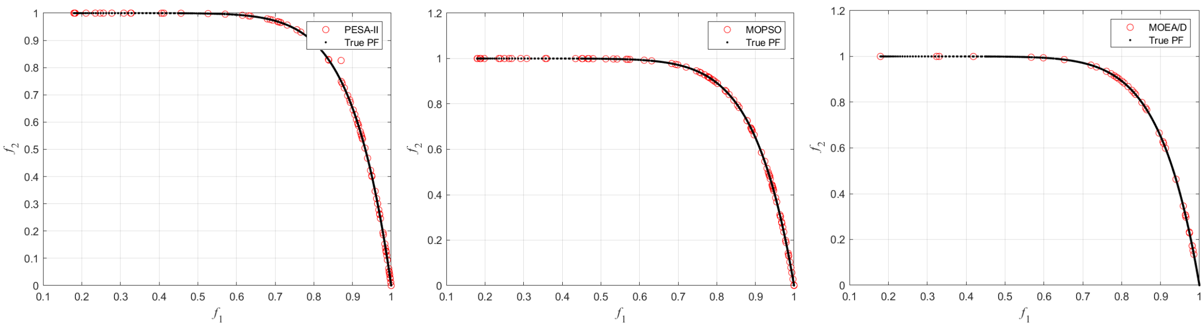


(a) PESA-II (b) MOPSO (c) MOEA-D

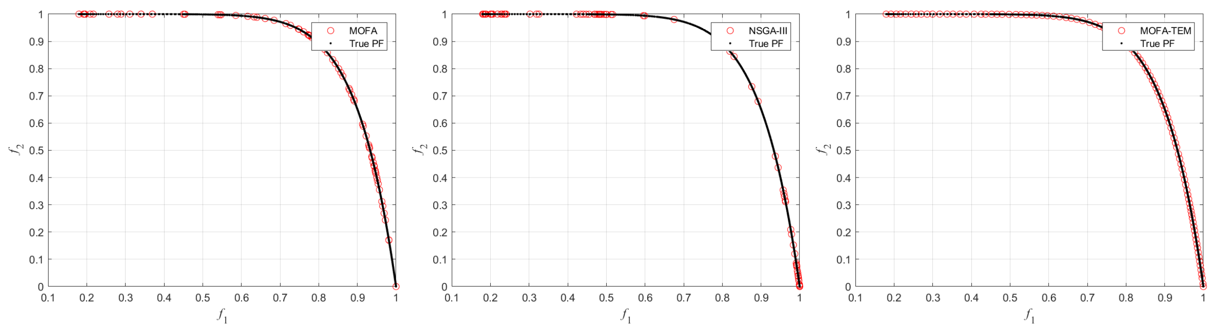


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 8. Frontier fitting plot for Deb2

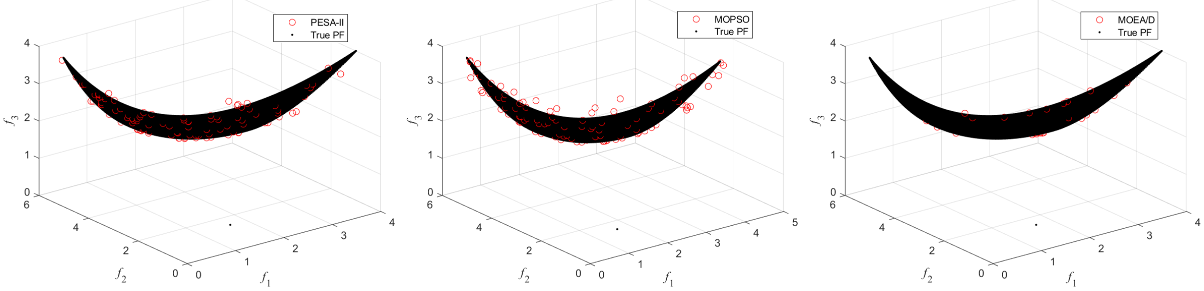


(a) PESA-II (b) MOPSO (c) MOEA-D

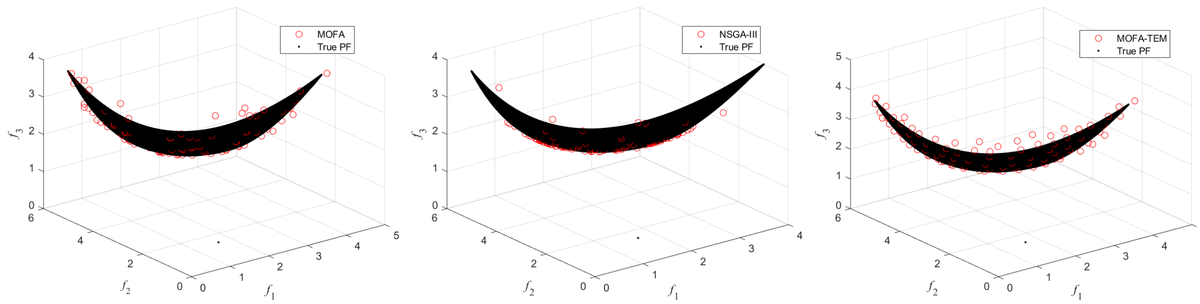


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 9. Frontier fitting plot for Deb3

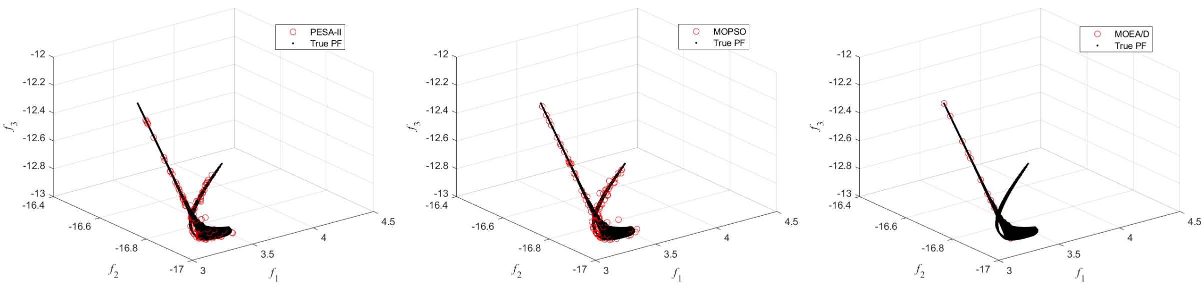


(a) PESA-II (b) MOPSO (c) MOEA-D

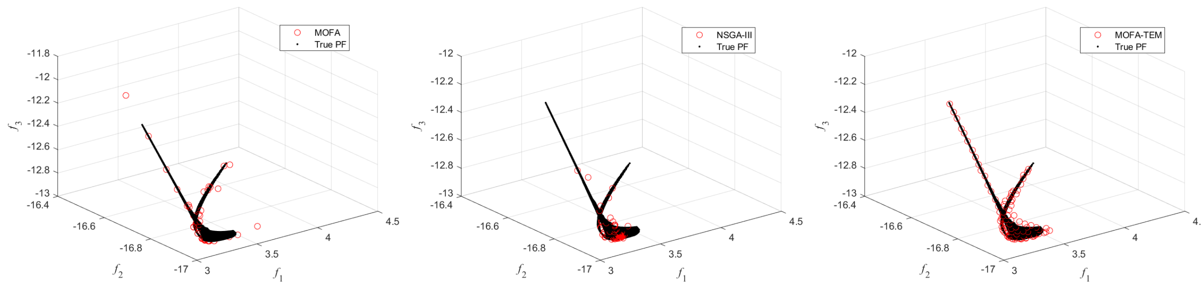


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 10. Frontier fitting plot for Viennet1

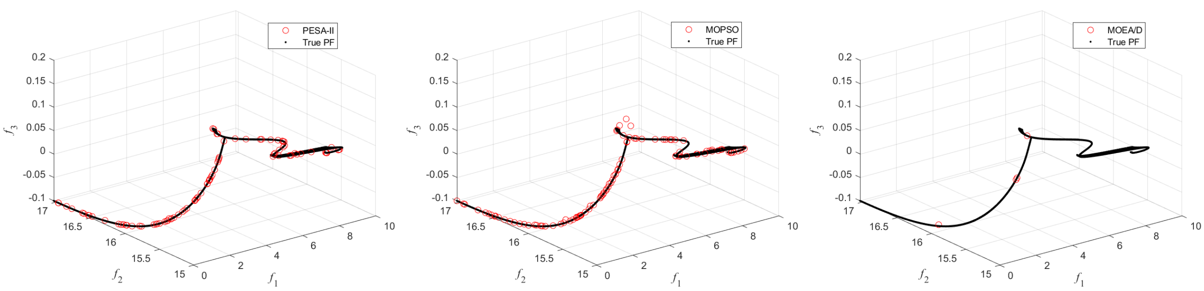


(a) PESA-II (b) MOPSO (c) MOEA-D

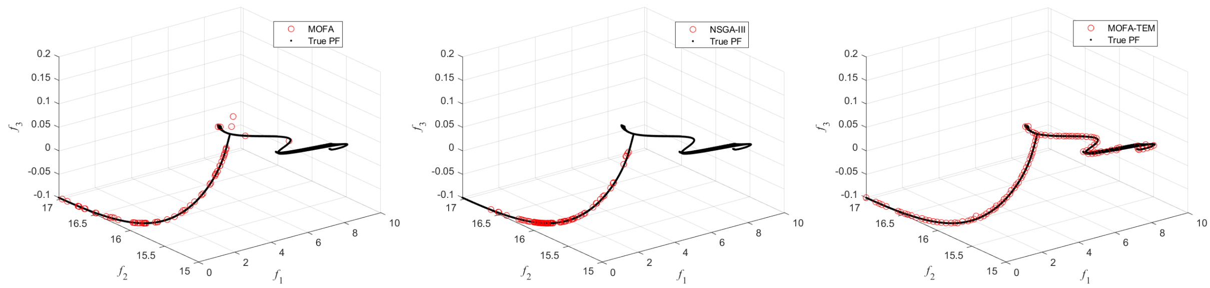


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 11. Frontier fitting plot for Viennet2

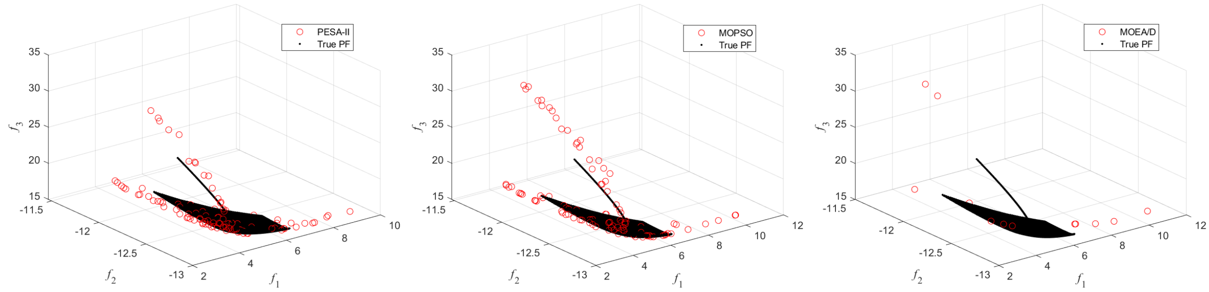


(a) PESA-II (b) MOPSO (c) MOEA-D

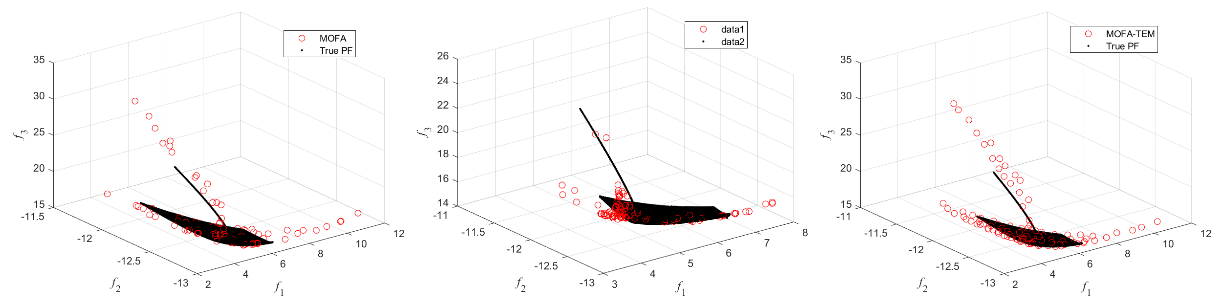


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 12. Frontier fitting plot for Viennet3

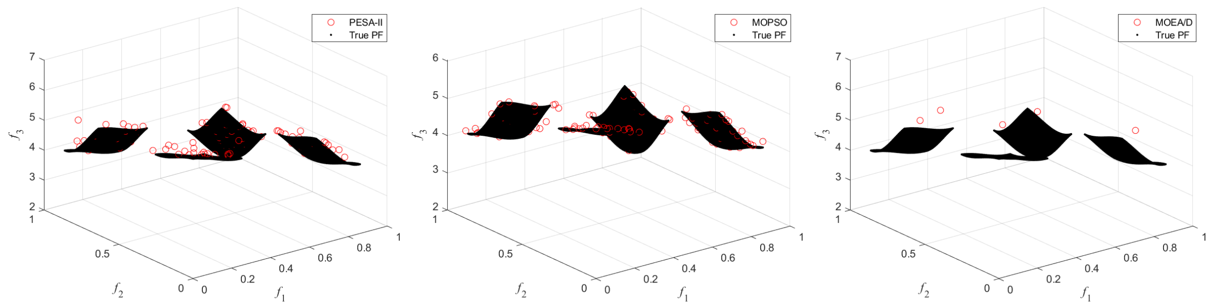


(a) PESA-II (b) MOPSO (c) MOEA-D

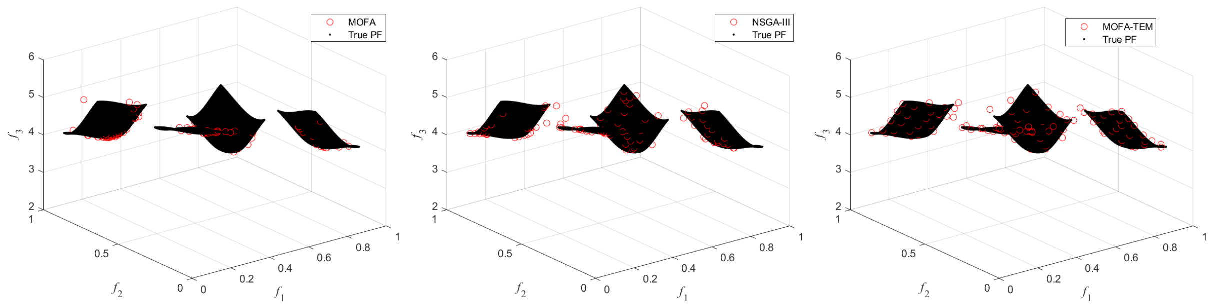


(d) MOFA (e) NSGA-III (f) MOFA-TEM

FIGURE 13. Frontier fitting plot for Viennet4



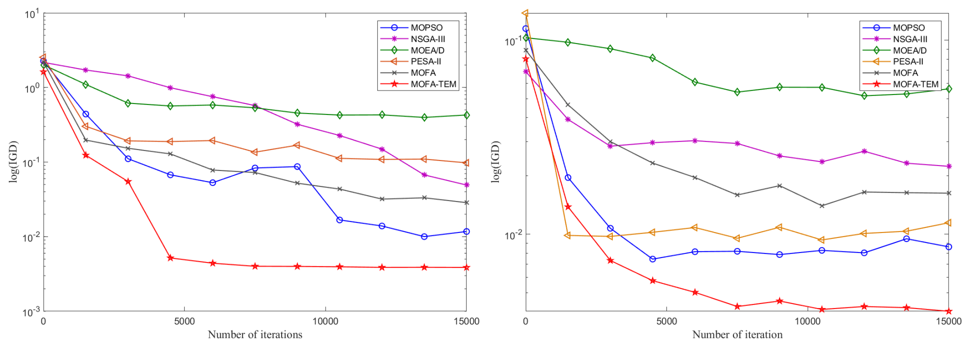
(a) PESA-II (b) MOPSO (c) MOEA-D



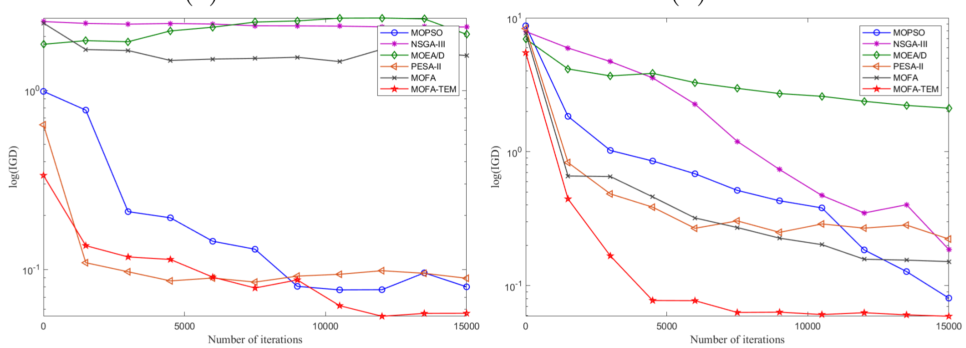
(d) MOFA (e) NSGA-III (f) MOFA-TEM

Figure 14. Frontier fitting plot for DTLZ7

To visualize the search performance of MOFA-TEM, the convergence curves of MOFA-TEM with MOPSO, NSGA-III, MOEA/D, PESA-II and MOFA are plotted as shown in Figure 15. Where the horizontal coordinate represents the number of evaluations of the algorithm and the vertical coordinate represents the convergence accuracy of the algorithm, and the vertical coordinate is logarithmic for ease of observation. From the figure, due to the different characteristics of different functions, the speed and accuracy of the algorithms are different, but MOFA-TEM has higher accuracy and faster optimization search speed. Because MOFA-TEM adopts a learning strategy with more comprehensive population information, it prevents fireflies from conducting ineffective searches in other directions and wasting evaluation times, which makes MOFA-TEM converge faster in the early iterations. When other algorithms stagnate, MOFA-TEM triggers a tolerance-based firefly state judgment mechanism to keep the population in the right search direction and continuously search for the optimal solution. This demonstrates the better ability of MOFA-TEM to consistently seek optimization and jump out of local extreme.



(a) ZDT1 (b) Deb2



(e) Viennet3 (f) DTLZ7

Figure 15. Convergence curves of the 6 MOEAs

4.3. **Comparison with recent MOEAs.** MOFA-TEM was compared with 7 recent MOEAs to verify the validity and competitiveness of the algorithm. Among them, the parameter settings of HVMA-M, CFMOFA, MOEA/D-ACD and MONSFA are taken from the related literature, and the remaining algorithm parameters are consistent with the PlatEMO platform, shown in Table 4. For the 2-objective function, N_{pop} is 100, *Rep* is 100, $MaxIt$ is 300; For the 3-objective function, N_{pop} is 200, *Rep* is 200, $MaxIt$ is 600. The algorithm was run independently for 30 times and the experimental results are shown in Table 5.

Table 4. Algorithm parameter setting

Algorithm	Parameter setting
NSGA-II-SDR [32]	Adoption of the Parameter Settings within the PlatEMO Platform
Top [34]	
DGEA [35]	
MONSFA [30]	$alpha = 0.25, \beta_0 = 1, \gamma = 1$
MOEA/D-ACD [31]	$CR = 1.0, F = 0.5, p_m = 1/n,$ $\eta_m = 20, T = 20, \delta = 0.9, n_r = 2$
CFMOFA [33]	$\alpha = 0.2, \beta_0 = 1, \gamma = 1$
HVMA-M [36]	$\alpha = 0.2, \beta_0 = 1, \gamma = 1$
MOFA-TEM	$alpha(1) = 0.2, alpha(t) = 0.9$ $alpha(t-1), \beta_0 = 1, \gamma = 1$

Table 5. Experimental results of MOFA-TEM and 7 recent MOEAs on IGD

Instances	MONSFA	MOEA/D-ACD	NSGA-II-SDR	NSGA-II-SDR
	Mean±Std.	Mean±Std.	Mean±Std.	Mean±Std.
ZDT1	2.62E-02±3.69E-03	4.70E-02±7.06E-03	3.87E-03±1.55E-05	2.60E-02±4.18E-03
ZDT2	4.37E-02±2.14E-02	1.35E-01±3.40E-01	3.94E-03±3.88E-05	9.13E-03±1.14E-03
ZDT3	3.35E-02±1.04E-02	7.65E-02±3.77E-02	4.44E-03±3.02E-05	1.36E-02±4.25E-03
ZDT4	1.73E-01±1.52E-01	1.98E+00±8.02E-01	3.85E-03±2.95E-05	8.90E-03±1.03E-03
ZDT6	2.63E-01±5.01E-02	1.47E-01±1.93E-01	4.30E-03±1.20E-05	1.88E-02±1.31E-02
DTLZ2	**6.87E-02±1.41E-03**	1.27E-01±2.49E-02	3.11E-01±2.98E-02	5.41E-02±9.46E-04
DTLZ3	9.40E+01±4.19E+01	1.12E+01±4.27E+00	**1.23E+02±9.50E+00**	3.74E+01±1.27E+00
DTLZ4	7.81E-01±1.31E-01	**3.62E-01±3.34E-01**	7.39E-01±2.00E-02	1.43E-01±4.55E-02
DTLZ5	5.95E-01±1.09E-01	**8.35E-01±4.17E-01**	5.04E-01±4.87E-02	5.99E-01±2.65E-02
DTLZ6	7.68E+00±4.91E-01	5.22E+00±3.34E+00	6.69E-01±1.22E-01	8.29E-01±2.16E-01
DTLZ7	1.48E-01±3.18E-02	2.44E-01±1.36E-01	6.05E-02±6.69E-04	5.90E-02±5.28E-03
Total	1	2	1	0
Ranking	4.50	4.55	3.91	4.45
Final rank	4	5	2	3

Instances	Top	DGEA	HVMA-M	MOFA-TEM
	Mean±Std.	Mean±Std.	Mean±Std.	Mean±Std.
ZDT1	5.41E-02±2.01E-02	1.17E-01±1.02E-01	1.18E-02±4.19E-03	**3.88E-03±1.23E-05**
ZDT2	1.32E-01±4.94E-02	8.91E-03±5.55E-03	1.25E-02±5.30E-03	**3.83E-03±5.40E-06**
ZDT3	1.55E-01±5.81E-02	3.00E-01±1.38E-01	1.15E-02±2.15E-03	**4.49E-03±8.58E-05**
ZDT4	1.01E+01±5.01E+00	9.44E-01±6.67E-01	9.40E-03±1.47E-03	**3.89E-03±6.68E-05**
ZDT6	3.82E-03±1.92E-04	**3.10E-03±9.74E-07**	1.34E-01±9.31E-02	3.96E-03±1.50E-04
DTLZ2	5.83E-02±1.77E-03	3.93E-02±5.66E-04	1.97E-01±6.78E-02	1.27E-01±1.83E-03
DTLZ3	2.53E+00±5.06E+00	6.95E+01±5.66E+01	2.34E+01±1.44E+01	7.82E+01±4.61E+00
DTLZ4	5.99E-02±2.24E-03	4.15E-02±4.73E-03	2.73E-01±7.86E-02	5.47E-01±9.78E-02
DTLZ5	4.06E-03±2.37E-04	4.19E-02±1.96E-03	7.39E-01±6.45E-02	4.16E-01±3.47E-02
DTLZ6	**2.62E-03±1.07E-04**	2.51E-02±4.60E-03	3.33E+00±2.72E+00	5.06E-01±2.01E-03
DTLZ7	1.09E-01±8.44E-02	1.31E-01±1.26E-01	2.17E-01±1.11E-02	**3.99E-02±3.82E-05**
Total	1	1	0	5
Ranking	4.50	4.73	5.55	3.82
Final rank	4	6	7	1

In Table 5, MOFA-TEM obtained better results on ZDT1-ZDT4 and DTLZ7 respectively, with a total of 5 optimums. Among them, MOEA/D-ACD obtained 2 optimums on DTLZ4 and DTLZ5, MONSFA obtained 1 optimum on DTLZ2, NSGA-II-SDR obtained 1 optimum on DTLZ3, Top obtained 1 optimum on DTLZ6, and DGEA obtained 1 optimum on ZDT6, whereas CFMOFA and HVMA-M were both obtained. On ZDT6, the IGD mean value of MOFA-TEM: 3.96E-03 has the same order of magnitude (10^{-3}) as the optimal IGD mean value: 3.10E-03, indicating that the difference between the two values is small. Secondly, MOFA-TEM obtained the smallest Ranking value of 3.82, indicating a more significant global performance of MOFA-TEM. From IGD and Friedman test results, MOFA-TEM obtained optimal results in Total, Ranking and Final rank, which verified its validity and competitiveness in the recent MOEAs.

5. **Conclusion.** Individuals in a classical MOFA are updated mainly by learning the global optimum individual, which causes the population to stagnate in evolution when the global optimum individual falls into a local optimum, and the algorithm will converge prematurely. Especially when the MOPs become complex, the phenomenon of premature convergence of MOFA will be more and more frequent. When facing optimization problems with complex search space such as nonconvex and discontinuous, the classical MOFA cannot solve such problems effectively, so this paper proposes a tolerance-based elite mutation firefly algorithm (MOFA-TEM). MOFA-TEM changes the single learning object approach of learning only from the optimal individual, and reduces the ineffective search of the population in unfavorable directions by learning from $gbest$, which contains the population's flight experience, and $pbest$, which has its own flight experience. To prevent premature population stagnation, a tolerance-based judgment mechanism for firefly status is introduced. As the number of firefly stagnation increases, the tolerance probability increases exponentially, and when it exceeds $rand()$, it indicates that the current learning object is worthless as a reference, and it will stop learning from the current elite leader $gbest$, and utilize the elite mutation strategy to update $gbest$ and produce a new elite leader $gbest^*$. By comparing MOFA-TEM with five classical as well as seven recent MOEAs by on IGD and Friedman test, and plotting Pareto frontier fitting graphs and convergence curves, it is verified that MOFA-TEM obtains better results when dealing with different problems, which fully demonstrates the effectiveness and superiority of

the algorithm. MOFA-TEM can be used to solve complex optimization problems in the future.

Acknowledgment. This work was supported by the Jiangxi Province Department of Education Science and Technology Project with No. GJJ201915; The National Natural Science Foundation of China with No. 52069014.

REFERENCES

[1] Y.-C. Hua, Q.-Q. Liu, K.-R. Hao, and Y.-C. Jin, "A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 303-322, 2021.

[2] L.-P. Wang, X.-T. Pan, X. Shen, P.-P. Zhao, and Q.-C. Qiu, "Balancing convergence and diversity in resource allocation strategy for decomposition-based multi-objective evolutionary algorithm," *Applied Soft Computing*, vol. 100, pp. 106968, 2021.

[3] W.-L. Wang, *Artificial intelligence and its applications*, Higher Education Press, Beijing, 2020.

[4] F.-Q. Zhang, T.-Y. Wu, Y.-O. Wang, R. Xiong, G.-Y. Ding, P. Mei, and L.-Y. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[5] A.-L.-H.-P. Shaik, M.-K. Manoharan, A.-K. Pani, R.-R. Avala, and C.-M. Chen, "Gaussian Mutation–Spider Monkey Optimization (GM-SMO) Model for Remote Sensing Scene Classification," *Remote Sensing*, vol. 14, no. 24, pp. 6279, 2022.

[6] C.-M. Chen, S. Lv, J. Ning, and J.-M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, no. 1, pp. 124, 2023.

[7] L. Kang, R.-S. Chen, N.-X. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, "Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things," *IEEE Access*, vol. 7, pp. 59504-59513, 2019.

[8] T.-Y Wu, A.-K. Shao, and J.-S. Pan, "CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm," *Mathematics*, vol. 11, no. 10, pp. 2339, 2023.

[9] A. H. Gandomi, and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831-4845, 2012.

[10] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.

[11] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," *Computers and Structures*, vol. 169, pp. 1-12, 2016.

[12] T.-Y. Wu, H.-N. Li, and S.-C. Chu, "PPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps," *Mathematics*, vol. 11, no. 9, pp. 1977, 2023.

[13] S.-C. Chu, Q. Feng, J. Zhao, and J.-S. Pan, "BFGO: Bamboo Forest Growth Optimization Algorithm," *Journal of Internet Technology*, vol. 24, no. 1, pp. 1-10, 2023.

[14] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, no. 2, pp. 175-184, 2013.

[15] X.-J. Cai, S.-J. Geng, J.-B. Zhang, D. Wu, Z.-H. Cui, W.-S. Zhang, and J.-J. Chen, "A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7650-7658, 2021.

[16] J. Zhao, J.-J Tang, A. Shi, T.-H. Fan, and L.-Z. Xu, "Improved density peaks clustering based on firefly algorithm," *International Journal of Bio-Inspired Computation*, vol. 15, no. 1, pp. 24-42, 2020.

[17] J. Zhao, D.-D. Chen, R.-B. Xiao, J. Chen, J.-S. Pan, Z.-H. Cui, and H. Wang, "Multi-objective Firefly Algorithm with Adaptive Region Division," *Applied Soft Computing*, vol. 147, pp. 110796, 2023.

[18] J. Zhao, G. Wang, J.-S. Pan, T.-H. Fan, and I. Lee, "Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets," *Pattern Recognition*, vol. 139, pp. 109406, 2023.

[19] J. Zhao, D.-D. Chen, R.-B. Xiao, Z.-H. Cui, H. Wang, and I. Lee, "Multi-strategy ensemble firefly algorithm with equilibrium of convergence and diversity," *Applied Soft Computing*, vol. 123, pp. 108938, 2022.

[20] Z. He, P. Kang, Q.-P. Li, X.-M. Liu, S.-W. Li, and J. Zhao, "Firefly Algorithm with combination multi-strategies," *Journal of Nanchang Institute of Technology*, vol. 42, no. 01, pp. 80-87, 2023.

[21] J. Zhao, Z.-F. Xie, L. Lu, H. Wang, H. Sun, and X. Yu, "Firefly Algorithm with Deep Learning," *Acta Electonica Sinica*, vol. 46, no. 11, pp. 2633, 2018.

[22] Z.-Z. Lai, R.-X. Wu, Q. Li, Y. Zhen, S.-X. Zhang, H.-Q. Ou, C. Huo, and J. Zhao, "Multi-objective firefly algorithm for group leaching," *Journal of Nanchang Institute of Technology*, vol. 42, no. 03, pp. 73-81, 2023.

[23] J. Zhao, W.-P. Chen, R.-B. Xiao, and H. Wang, "Firefly algorithm based on self-learning for multi-peak optimization problem," *Control and Decision*, vol. 37, no. 8, pp. 1971-1980, 2022.

[24] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Applied Soft Computing*, vol. 66, pp. 232-249, 2018.

[25] X.-F. Hong, M.-F. Jiang, and J.-L. Yu, "Fine-grained ensemble of evolutionary operators for objective space partition based multi-objective optimization," *IEEE Access*, vol. 9, pp. 400-411, 2020.

[26] B. Gadhvi, V. Savsani, and V. Patel, "Multi-objective optimization of vehicle passive suspension system using NSGA-II, SPEA2 and PESA-II," *Procedia Technology*, vol. 100, no. 23, pp. 361-368, 2016.

[27] C. A. C Coello, G.T. Pulido, and M.S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256-279, 2004.

[28] Y.-T. Qi, X.-L. Ma, F. Liu, L.-C. Jiao, J.-Y. Sun, and J.-S. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231-264, 2014.

[29] K. Deb, and H. Jain, "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*, vol. 18, no. 4, pp. 577-601, 2014.

[30] C.W. Tsai, Y. T. Huang, M.C. Chiang, "A non-dominated sorting firefly algorithm for multi-objective optimization," in *2014 14th International Conference on Intelligent Systems Design and Applications*. IEEE, 2015, pp. 62-67.

[31] L.-P. Wang, Q.-F. Zhang, A.-M. Zhou, M.-G. Gong, and L.-C. Jiao, "Constrained subproblems in a decomposition-based multi-objective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 475-480, 2016.

[32] Y. Tian, R. Cheng, X.-Y. Zhang, Y.-S. Su, and Y.-C. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 331-345, 2018.

[33] L. Lv, J. Zhao, J.-Y. Wang, and T.-H. Fan, "Multi-objective firefly algorithm based on compensation factor and elite learning," *Future Generation Computer Systems*, vol. 91, pp. 37-47, 2019.

[34] Z.-Z. Liu, and Y. Wang, "Handling constrained multi-objective optimization problems with constraints in both the decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 870-884, 2019.

[35] C. He, R. Cheng, and D. Yazdani, "Adaptive offspring generation for evolutionary large-scale multi-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 786-798, 2020.

[36] J. Zhao, D.-D. Chen, R.-B. Xiao, T.-H. Fan, "A heterogeneous variation firefly algorithm with maximin strategy," *CAAI Transactions on Intelligent Systems*, vol. 17, no. 1, pp. 116–130, 2022.

[37] S. Zapotecas, and C. A. C. Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. 2011, pp. 69-76.