Information Sciences 563 (2021) 70–90

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Many-Objective Evolutionary Algorithm with Adaptive Reference Vector

Maoqing Zhang [a], Lei Wang [a,*], Wuzhao Li [b], Bo Hu [a], Dongyang Li [a], Qidi Wu [a]

[a] School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
[b] School of Intelligent Manufacturing, Jiaxing Vocational Technology College, 314036 Zhejiang, China

ARTICLE INFO

*Article history:*
Received 31 July 2020
Received in revised form 1 January 2021
Accepted 6 January 2021
Available online 26 January 2021

*Keywords:*
Many-objective optimization problems
Convergence
Spread
Adaptive reference vector strategy
Hierarchical clustering strategy

ABSTRACT

Convergence is always a major concern for many-objective optimization problems. Over the past few decades, various methods have been designed for measuring the convergence. However, according to our mathematical and empirical analyses, most of these methods are more focused on the convergence, and may neglect the exploration of boundary solutions, resulting in the incomplete Pareto fronts and the poor extent of spread achieved among the obtained non-dominated solutions. Regarding this issue, this paper proposes a Many-Objective Evolutionary Algorithm with Adaptive Reference Vector (MaOEA-ARV). In MaOEA-ARV, an adaptive reference vector strategy is designed to dynamically adjust the reference vectors according to the current distribution of candidate solutions for ensuring the spread and convergence simultaneously. Additionally, a hierarchical clustering strategy is employed to adaptively partition candidate solutions into multiple clusters for the diversity of candidate solutions. Experimental results on DTLZ, BT, ZDT and WFG test suites with up to 12 objectives demonstrate the effectiveness of MaOEA-ARV.


## 1. Introduction

In various engineering applications, there exist many optimization problems [15,7,6], which generally have multiple conflicting objectives. For example, a soft development team would like to reduce the software development costs as much as possible, and improve the software development quality at the same time [3,20]. Such problems are termed as Multi-objective Optimization Problems(MOPs) [33,44], which can be mathematically formulated as follows:

$$\begin{aligned} &Min\boldsymbol{F}(\boldsymbol{X}) = (f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \ldots, f_M(\boldsymbol{X})) \\ &s.t.\boldsymbol{X} \in \Omega \end{aligned} \tag{1}$$

where $\boldsymbol{X}$ is the decision vector, and $f_M(\boldsymbol{X})$ is the $M$-th objective function. In principle, for MOPs, there does not exist one best solution which can minimize all the objectives simultaneously, but a set of trade-off solutions. $\boldsymbol{X}$ strictly Pareto dominates $\boldsymbol{Y}$, expressed as $\boldsymbol{X} \prec \boldsymbol{Y}$, *iff* $\forall i \in \{1,2,3,\ldots,M\} : f_i(\boldsymbol{X}) \leqslant f_i(\boldsymbol{Y})$and $\exists j \in \{1,2,3,\ldots,M\} : f_j(\boldsymbol{X}) < f_j(\boldsymbol{Y})$. $\boldsymbol{X}^*$ is Pareto optimal if no solution dominates it. A set of Pareto optimal solutions constitutes of the Pareto Set (PS), whose objectives in the objective space correspond to the Pareto Front(PF).

* Corresponding author.
*E-mail address:* wanglei@tongji.edu.cn (L. Wang).

https://doi.org/10.1016/j.ins.2021.01.015
0020-0255/

Over the past few decades, researchers have proposed various methods [18,1,19] for tackling MOPs, such as mathematical programming methods [2], local search algorithms [41] and genetic evolution based techniques [11,40]. Genetic evolution based techniques are inspired from the idea of evolution in nature, which are able to obtain a set of trade-off solutions in a single run [17]. Due to this characteristic, genetic evolution based techniques are widely applied to MOPs. Typical evolutionary methods include NSGA-II [9] and MOEA/D [45], both of which have exhibited outstanding performance on MOPs. However, the increasing number of objectives of MOPs inevitably results in the inefficiency of Pareto dominance in distinguishing excess trade-off solutions. MOPs having more than three objectives are generally termed as Many-objective Optimization Problems (MaOPs). For MOPs with three objectives, the probability of one solution dominating the other one is $1/2^{M-1} = 0.25$, where $M$ is the number of objectives. However, for MaOPs, the probability exponentially decreases with the increasing $M$, which is known as Dominance Resistance(DR).

To remedy the phenomenon of DR, various efforts have been made to enhance the selection pressure, which can be grouped into the following categories. The first category is to modify the Pareto dominance relation, such as the strengthened Pareto dominance [38], adaptive controlling dominance [36], generalized Pareto optimality [47], $L$-dominance [50], and fuzzy dominance [14]. These modified dominance metrics generally enhance the selection pressure by relaxing the dominance areas. The second category is to combine Pareto dominance with additional selection criteria. To be specific, a large number of non-dominated solutions is further compared with each other using a secondary criterion. The third category is to utilize an indictor to measure each solution. The representative method is HypE [4], which ranks solutions with the Hypervolume(HV) indicator. Inspired by HypE, KnEA [46] employs a knee point strategy to determine solutions contributing more to HV values. Similar to KnEA, VaEA [43] defines a convergence indicator by simply adding up normalized objectives to distinguish the non-dominated solutions. The combination of Pareto dominance and decomposition strategy also falls into the third category, such as MOEA/DD [28] and BCE-MOEA/D [30].

Recently, there exists a research interest in quantitatively defining the convergence degree of one solution. Both the strengthened Pareto dominance [38] and convergence information [43] measure the convergence degree of one solution by the sum of each objective. In other words, all the objectives are treated equally. In principle, without the prior knowledge of user preference, it is reasonable to conduct such a treatment. However, according to our observations and analyses in later sections, the equal treatment of each objective may cause an unbalanced selection of candidate solutions. To put it in another way, more attention is paid to solutions with better convergence values, and boundary solutions with poor performance in terms of the convergence are likely to be neglected, resulting in the poor spread and incomplete Pareto fronts. Thus, it is reasonable to remedy such a phenomenon to realize a relatively fair selection. Based on the considerations above, this paper proposes a Many-Objective Evolutionary Algorithm with Adaptive Reference Vector (MaOEA-ARV), where the reference vector can be dynamically adjusted according to the current population. Therefore, the contributions of this paper are detailed as follows:

1) An adaptive reference vector strategy is proposed. According to the analyses of representative methods for measuring the convergence degree of each solution, the observation is obtained that the equal treatment of all the objectives may result in the preference for solutions with better convergence values, directly affecting the spread of current candidate solutions. To avoid this phenomenon, the adaptive reference vector strategy is designed by defining a new form of reference vectors, which is capable to adaptively adjust the reference vectors according to the current population.
2) MaOEA-ARV is proposed by incorporating the adaptive reference vector strategy into the evolutionary algorithm. The adaptive reference vector strategy is used to ensure the convergence and spread, and a hierarchical clustering strategy is included into MaOEA-ARV to improve the solution diversity. In the hierarchical clustering strategy, the number of clusters are dynamically adjusted.
3) MaOEA-ARV is comprehensively tested with some state-of-art methods tailored for MOPs and MaOPs on DTLZ [10], BT [26], ZDT [48] and WFG [16] test suites with up to 12 objectives using Spread [42], IGD [5] and HV [49] indicators. Experimental results demonstrate that MaOEA-ARV outperforms its competitors, such as the latest NSGA-II/SDR [38], as well as other representative methods.

The structure of this paper is detailed as follows. In Section 2, related work and analyses are presented. Section 3 introduces the core idea of the adaptive reference vector strategy. Following that, the framework of MaOEA-ARV is presented in detail in Section 4. Experiments and analyses are given in Section 5. Conclusions are drawn in Section 6.

## 2. Related work and analyses

Due to the inefficiency of Pareto dominance in tackling MaOPs, many researchers attempt to define some convergence metrics to measure the convergence of candidate solutions to the true Pareto fronts. More recently, a strengthened dominance relation [38] is proposed to measure the convergence of any one solution by the sum of all the objectives. Similarly, BiGE [29] converts MOPs into a bi-goal (objective) optimization problem regarding the proximity and diversity, where the proximities of solutions are computed by simply adding up all the objectives. Inspired by BiGE, VaEA [43] proposes to measure solutions with the sum of normalized objectives. Further, Liu et al. [32] analyze various convergence indicators, such as the sum of all the objectives, the Chebyshev distance to the ideal point, and the Euclidean distance to the ideal point, and then present their pros and cons. Typically, these categories above can be summarized as follows:

1) The first one is the sum of all the objectives [38,29]:

$$Con(\boldsymbol{X}) = \sum_{m=1}^{M} f_m(\boldsymbol{X}) \tag{2}$$

where $Con(\boldsymbol{X})$ indicates the convergence of solution $\boldsymbol{X}$ , $f_m(\boldsymbol{X})$ means the $m$-th objective and $M$ represents the number of objectives.

2) The next one is the Chebyshev distance to the ideal point, which can be defined as follows [32,45]:

$$Con(\boldsymbol{X}) = \max_{1 \leqslant m \leqslant M} |f_m(\boldsymbol{X}) - z_m^*| \tag{3}$$

where $\boldsymbol{z}^* = (z_1^*, z_2^*, \ldots, z_M^*)^T$ indicates the ideal point, and $z_m^* = \min_{\boldsymbol{X} \in Q} f_m(\boldsymbol{X})$ .

3) Similar to the Chebychev distance, the third one is the Euclidean distance to the ideal point [32,45]:

$$Con(\boldsymbol{X}) = \sqrt{\sum_{m=1}^{M} (f_m(\boldsymbol{X}) - z_m^*)^2} \tag{4}$$

4) The last one is the Euclidean distance to the Nadir point as follows [32,45]:

$$Con(\boldsymbol{X}) = 1 \Big/ \sqrt{\sum_{m=1}^{M} (f_m(\boldsymbol{X}) - z_m^{nad})^2} \tag{5}$$

where $\boldsymbol{z}^{nad} = (z_1^{nad}, z_2^{nad}, \ldots, z_M^{nad})^T$ and $z_m^{nad} = \max_{\boldsymbol{X} \in Q} f_m(\boldsymbol{X})$.

The four categories above have been widely researched in many papers [32,27]. Various pros and cons of them have been summarized. For example, Eq. (2) is considered to be more suitable for problems with concave Pareto optimal fronts. Eq. (3) is believed to be effective in tackling problems with concave fronts, as well as discrete problems. Eq. (4) is outstanding on concave problems, especially for problems whose Pareto optimal fronts are part of hypersphere. Different from Eq. (4), Eq. (5) is able to evolve the population toward the entire Pareto optimal fronts.

From Eqs. (2)–(5), it can be seen that, although various convergence metrics have been defined to indicate the proximity of any one solution, they have one thing in common that all the objectives are treated equally. That is to say, no matter how the true Pareto optimal fronts vary, the importance of each objective is the same. Mathematically, Eqs. (2)–(5) can be represented with the following more general expression:

$$Con(\boldsymbol{X}) = f(\boldsymbol{F}(\boldsymbol{X})|\boldsymbol{R}, \boldsymbol{z}) \tag{6}$$

where $f(\cdot)$ means various functions as Eqs. (2)–(5) indicate, $\boldsymbol{F}(\boldsymbol{X})$ is the objective vector of solution $\boldsymbol{X}, \boldsymbol{R} = (1, 1, \ldots, 1)$ is the reference vector of $M$ dimensions, where '1' means all the objectives are equally important. $\boldsymbol{z}$ represents $\boldsymbol{z}^*$ in Eq. (3) or $\boldsymbol{z}^{nad}$ in Eq. (5). To be specific, take Eq. (2) as an example:

$$Con(\boldsymbol{X}) = f(\boldsymbol{F}(\boldsymbol{X})|\boldsymbol{R}, \boldsymbol{z}) = \boldsymbol{F}(\boldsymbol{X}) \cdot \boldsymbol{R} = \sum_{m=1}^{M} (f_m(\boldsymbol{X}) \times 1) = \sum_{m=1}^{M} f_m(\boldsymbol{X}) \tag{7}$$

where $\boldsymbol{F}(\boldsymbol{X}) \cdot \boldsymbol{R}$ indicates the dot product of $\boldsymbol{F}(\boldsymbol{X})$ and $\boldsymbol{R} = (1, 1, \ldots, 1)$. Further, Eq. (7) can be re-expressed with the following formulation:

$$Con(\boldsymbol{X}) = f(\boldsymbol{F}(\boldsymbol{X})|\boldsymbol{R}, \boldsymbol{z}) = ||\boldsymbol{F}(\boldsymbol{X})||_2 \times ||\boldsymbol{R}||_2 \times \cos\alpha \tag{8}$$

where $||\boldsymbol{F}(\boldsymbol{X})||_2$ indicates the Euclidean norm of $\boldsymbol{F}(\boldsymbol{X})$ , and $\alpha$ is the acute angle between $\boldsymbol{F}(\boldsymbol{X})$ and $\boldsymbol{R}$ . $||\boldsymbol{R}||_2 = \sqrt{M}$if $\boldsymbol{R} = (1, 1, \ldots, 1)$, and it is actually a constant for given MaOPs. From Eq. (8), it can be seen that $Con(\boldsymbol{X})$ is determined by $||\boldsymbol{F}(\boldsymbol{X})||_2 \times \cos\alpha$, which actually is the projection of $\boldsymbol{F}(\boldsymbol{X})$ on reference vector $\boldsymbol{R}$.

Intuitively, it is reasonable to treat all the objectives equally because it can have an unbiased selection of candidate solutions in the objective space. However, the fact may not follow the intuition perfectly. Actually, such a treatment may result in unbalanced selections and incomplete Pareto fronts. Take Fig. 1 as an example to illustrate our analyses.

As can be seen, Fig. 1 shows a two-dimensional objective space, where point $\boldsymbol{A} \sim \boldsymbol{F}$ are candidate solutions. Evidently, according to the definition of Pareto dominance, these candidate solutions are non-dominated with each other. The solid lines with two arrows mean the convergence values, and they are $Con(\boldsymbol{A}) = 11$, $Con(\boldsymbol{B}) = 9$, $Con(\boldsymbol{C}) = 7$, $Con(\boldsymbol{D}) = 8$, $Con(\boldsymbol{E}) = 10$, $Con(\boldsymbol{F}) = 11$, all of which are calculated using Eq. (8) with $\boldsymbol{R} = (1, 1)$. As a result, the sorted convergence sequence is $Con(\boldsymbol{C}) < Con(\boldsymbol{D}) < Con(\boldsymbol{B}) < Con(\boldsymbol{E}) < Con(\boldsymbol{A}) = Con(\boldsymbol{F})$. Such a sequence indicates that solution $\boldsymbol{C}$ and $\boldsymbol{D}$ are preferred to solution $\boldsymbol{B}$ and $\boldsymbol{E}$. As a result of such a priority, the convergence of the current solution set can be ensured as much as possible, which has been discussed and verified in many papers [38,43]. However, one fact is neglected that the boundary solution $\boldsymbol{A}$ and $\boldsymbol{F}$ have a small chance to be selected as promising solutions. A small probability of choosing boundary solutions may affect the spread of solutions in the future evolutionary process, and prevent the exploration of boundaries of MaOPs, which is verified by the visual comparisons on multiple test suites in subsections 5.3 and 5.4. Although choosing boundary solutions may transitorily deteriorate the convergence of the current solution set $\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D},\boldsymbol{E},\boldsymbol{F}\}$, it is necessary and significant to perform such an operation to obtain the complete Pareto fronts. In addition, more diverse solutions can be therefore obtained in future evolutionary process, and the overall performance can also be improved consequently.

## 3. Proposed adaptive reference vector strategy

To remedy the issue above, this paper attempts to propose an adaptive reference vector strategy. In other words, the reference vector $\boldsymbol{R}$ in Eq. (8) is adjusted adaptively, and the boundary solutions have higher probabilities of being selected. Mathematically, for one solution set $\{\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_K\}$, the reference vector $\boldsymbol{R}$ is defined as follows:

$$\boldsymbol{R} = \frac{\boldsymbol{F}(\boldsymbol{X}_i) - \boldsymbol{Z}}{||\boldsymbol{F}(\boldsymbol{X}_i) - \boldsymbol{Z}||_2} \tag{9}$$

where $\boldsymbol{Z}$ is defined with Eq. (10), and $\boldsymbol{F}(\boldsymbol{X}_i)$ is the objective vector of solution $\boldsymbol{X}_i$ , which is mathematically expressed with Eq. (11).

$$\boldsymbol{Z} = \frac{\boldsymbol{z}^{nad} + \boldsymbol{z}^{*}}{2} \tag{10}$$

where $\boldsymbol{z}^{nad} = (z_1^{nad}, z_2^{nad}, \ldots, z_M^{nad})^T$ and $\boldsymbol{z}^{*} = (z_1^{*}, z_2^{*}, \ldots, z_M^{*})^T$ .

$$\boldsymbol{X}_i = \arg\min_{k=1}^{K}\left(\sum_{q=1,q\neq k}^{K}\sum_{m=1}^{M}((f_m(\boldsymbol{X}_k) - f_m(\boldsymbol{X}_q)) \times C_m)\right) \tag{11}$$

$$s.t.\ C_m = \begin{cases} 1, & f_m(\boldsymbol{X}_k) > f_m(\boldsymbol{X}_q) \\ 0, & f_m(\boldsymbol{X}_k) \leqslant f_m(\boldsymbol{X}_q) \end{cases}$$

where $M$ indicates the number of objectives. $K$ is the amount of solutions.

Note that $\boldsymbol{X}_i$ in Eq. (11) is actually defined as the knee point of solution set $\{\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_K\}$. Moreover, the knee point determined by Eq. (11) is different from that in KnEA. Specifically, the knee point in KnEA [46] is the solution with the maximum distance from itself to the line determined by the candidate boundary solutions. However, there exist two issues with the knee point in KnEA. Firstly, the candidate boundary points during the evolutionary process are always changing, and may be far away from the true boundary points. According to the descriptions of KnEA in the original paper [46], the knee point strategy prefers solutions contributing more to HV values, rather than the candidate boundary points. Hence, it is evident that the candidate boundary solutions may be abandoned during the evolutionary process, resulting in the constant changes of boundary solutions. Secondly, even if the candidate boundary solutions are able to get close to the true boundary solutions as much as possible, the determination of the knee points may also be affected by the candidate boundary solutions. The reason is that, as the true Pareto fronts of MaOPs are various, the true Pareto fronts may have the same boundary solutions, but have different inner Pareto fronts. That is, the knee points are measured with the contribution to HV values, without considering the extent of the spread achieved by the current candidate solutions. Therefore, it can be said that the performance of KnEA is largely limited to the features of the true Pareto fronts and may perform poorly on problems with multi-feature Pareto fronts. The phenomenon will be observed and analyzed in subsections 5.3 and 5.4. Evidently different from KnEA, Eq. (11) determines the knee point by considering position relations of all solutions in the current solution set, and is able to effectively avoid the phenomenon caused by the candidate boundary points.

Indeed, the adaptive reference vector strategy attempts to optimize the convergence due to the involvement of the knee points determined by Eq. (11). However, it also tries to take into consideration the extent of the spread achieved by candidate solutions. Fig. 2 illustrates the core idea of the adaptive reference vector strategy, which has the same candidate solutions as Fig. 1 shows. According to Eq. (11), the knee point of solution set $\{\mathbf{A},B,C,D,E,F\}$ is solution $C(3,4)$. $Z=(5.5,5.5)$ according to Eq. (10). Therefore, the revised reference vector for the current solution set is $R'=(5\sqrt{34}/34,3\sqrt{34}/34)$. The convergence values of these solutions are $Con'(A)=35\sqrt{34}/34$, $Con'(B)=31\sqrt{34}/34$, $Con'(C)=\ 27\sqrt{34}/34$, $Con'(D)=\sqrt{34}$, $Con'(E)=23\sqrt{34}/17$, and $Con'(F)=53\sqrt{34}/34$, respectively. Then, the sorted convergence sequence is $Con'(C)<Con'(B)<Con'(D)<Con'(A)<Con'(E)<Con'(F)$. Compared with the original convergence sequence $Con(C)<Con(D)<Con(B)<Con(E)<Con(A)=Con(F)$ analyzed in Fig. 1, it can be seen that the priority of boundary solution $A$ becomes higher, indicating that boundary solution $A$ has a higher probability of being selected and the spread of the current solution set can be strengthened to some extent. In addition, the key solution $C$, $D$, and $B$, which are of importance in

evolving the current population toward the true Pareto fronts, still have higher priorities than solution $A$, $E$ and $F$, implying that the convergence is also ensured. Considering the two advantages above, it can be said the proposed adaptive reference vector strategy can take into account the convergence and spread simultaneously.

To test the analyses and the adaptive reference vector strategy above, subSection 5.2 is designed by comparing MaOEA-ARV with the representative methods with no adaptive reference vector strategy in terms of the spread. Note that, this paper is intended to design the effective adaptive reference vector strategy, not to verify the differences of Eqs. (2)–(5). Similar comparisons have been conducted in many papers [32,27]. Therefore, in later sections, Eq. (8) with the adaptive reference vector strategy is used as the main method for calculating the convergence indicator.

It should be noted that, Eqs. (2)–(5) have been applied to many decomposition-based methods by incorporating randomly generated or predefined weight vectors, such as MOEA/D and its variants [28,30]. Various effective methods have been also proposed to adaptively adjust the weight or reference vectors. For example, MOEA/D-AWA [35] utilizes an elite population to regulate the distribution of the weight vectors of subproblems periodically. W-MOEA/D [12] employs the linear interpolation of non-dominated solutions to approximate the Pareto fronts, and then uniformly generates some points on the Pareto fronts for the adaptive weight design. Similarly, based on MOEA/D, Li et al. [25] propose to employ simulated annealing for diversifying non-dominated solutions. Inspired by $\varepsilon$-dominance, $pa\lambda$-MOEA/D [24] designs a Pareto-adaptive weight vector strategy, which is the mixture of uniform design and hypervolume metric. As can be seen, the discussed methods above for adaptively adjusting the reference vectors are essentially different from the adaptive reference vector strategy in this paper. The differences lie in two aspects. The first one is that the methods above are based on MOEA/D, while the proposed adaptive reference vector strategy is based on the traditional Pareto dominance. The second one is that all the discussed methods above are designed to remedy the inefficiency of uniformly generated weight vectors for approximating the complete true Pareto fronts, and are focused on the adaptive adjustment of a set of weight or reference vectors to approximate the true Pareto fronts. Different from them, the proposed reference vector strategy is designed to extend the spread and ensure the convergence simultaneously.

## 4. MaOEA-ARV

This section is to introduce the details of MaOEA-ARV. The basic framework of MaOEA-ARV is firstly presented for the clear understanding of all the major components of MaOEA-ARV. Then, each component is explained sequentially. In the end, the computational complexity of MaOEA-ARV is analyzed to exhibit its computational efficiency.

### 4.1. General framework of MaOEA-ARV

The main framework of MaOEA-ARV is presented in Algorithm 1. As can be seen, the major components of MaOEA-ARV include the adaptive reference vector strategy, mating selection, non-dominated sorting and environmental selection. Firstly, a population of $N$ individuals are initialized randomly. Then, the non-dominated sorting is conducted to divide the current population into multiple non-dominated levels. After that, the adaptive reference vector strategy is performed to calculate the convergence of each individual. The mating selection is used to generate the mating pool of $N$ individuals. $N$ individuals are generated using *Variation*, which are then combined with parent population into $P'$. The environmental selection is to select $N$ individuals from $P'$ to form the next generation. These operations above are repeated until the termination criterion is reached.

**Algorithm 1**. General Framework of MaOEA-ARV

**Input:** $N$(population size), $M$(number of objectives)
**Output:** $P$(population)
$P \leftarrow Initialize(N)$
$F \leftarrow Non\!-\!dominated\_sorting(P, N, M)$ % Various non-dominated levels are generated; $F_i$ is the first Pareto front such that $|F_1 \cup F_2 ... \cup F_i| \geq N$
**while** *termination criterion not reached* **do**
  $Con \leftarrow Adaptive\_reference\_vector(P, N, M)$
  $R \leftarrow Mating\_selection(P, F, Con, N)$
  $P' \leftarrow P \cup Variation(R, N)$
  $F \leftarrow Non\!-\!dominated\_sorting(P', 2N, M)$
  $P \leftarrow Environmental\_selection(F, N, M)$
**end**

In the following subsections, this paper will detail the adaptive reference vector strategy, mating selection and environmental selection, which are essentially the main features of MaOEA-ARV.

### 4.2. Adaptive reference vector strategy

The adaptive reference vector strategy is to compute the adaptive reference vectors as Eq. (9) indicates and then calculate the convergence of each individual using Eq. (8), where $R$ is replaced with the resultant adaptive reference vectors. The details of the adaptive reference vector strategy is shown in Algorithm 2. As can be seen from Algorithm 2, $z^{nad}$ and $z^{*}$ are firstly obtained in preparation for the calculation of Eq. (9), referring to line 2–6. In the sequential, the knee-point solution is determined according to Eq. (11), referring to line 7–29. After that, the adaptive reference vector for the current population is determined, referring to line 30. Finally, the convergence of each individual is obtained on the basis of the adaptive reference vector.

**Algorithm 2**. Adaptive reference vector($P, N, M$)

```
   Input: P (population), N (population size), M (number of objectives)
   Output: Con
1  Z ← zeros(M, 1)
2  for m ← 1 to M do
3      z*_m ← min_{X∈P} f_m(X)
4      z^nad_m ← max_{X∈P} f_m(X)
5      Z_m ← (z*_m + z^nad_m)/2
6  end
7  Temp ← 0, T ← 0 %Calculate the knee point as Eq.(11) indicates
8  for i ← 1 to N do
9      T_1 ← 0
10     for j ← 1 to N do
11         T_2 ← 0
12         for m ← 1 to M do
13             if f_m(X_i) > f_m(X_j) then
14                 T_3 ← (f_m(X_i) − f_m(X_j))
15             else
16                 T_3 ← 0
17             end
18             T_2 ← T_3 + T_2
19         end
20         T_1 ← T_1 + T_2
21     end
22     if i = 1 then
23         Temp ← T_1, T ← i
24     else
25         if Temp > T_1 then
26             Temp ← T_1, T ← i
27         end
28     end
29 end
30 R ← (F(X_T) − Z) / ||F(X_T) − Z||_2 % Calculate the reference vector as Eq.(9) indicates
31 for k ← 1 to N do
32     Con(X_k) ← F(X_k) · R % Calculate the convergence value of solution X_k
33 end
```

### 4.3. Mating selection

The mating selection in MaOEA-ARV is a binary tournament selection strategy, which includes the dominance comparison and convergence comparison. As shown in Algorithm 3, two individuals are randomly selected from the current population, referring to line 3. If the previous one dominates the later one, then the previous one is preferred, referring to line 4 and 5. However, if they are non-dominated with each other, the convergence can be used to further distinguish them. The better one can be determined according to their convergence values, referring to line 9–12. Otherwise, randomly select one of them, referring to line 14–17. Repeat the procedure above until $N$ individuals are determined.

**Algorithm 3**. Mating Selection $(P, F, Con, N)$

```
Input: P (population), F (multiple Pareto levels), Con, N
Output: R (mating pool)
R ← ∅
while |R| ≤ N do
    randomly choose X_a and X_b from P
    if X_a ≺ X_b then
        R ← R ∪ {X_a}
    else if X_b ≺ X_a then
        R ← R ∪ {X_b}
    else
        if Con(X_a) < Con(X_b) then
            R ← R ∪ {X_a}
        else if Con(X_b) < Con(X_a) then
            R ← R ∪ {X_b}
        else
            if rand() < 0.5 then
                R ← R ∪ {X_a}
            else
                R ← R ∪ {X_b}
            end
        end
    end
end
```

### 4.4. Environmental selection

Environmental selection is to choose $N$ individuals from the combined population to form the next generation. The details of environmental selection are presented in Algorithm 4. As can be seen, the number of individuals to be selected from the last Pareto front is firstly determined, referring to line 2. As discussed in many papers [31], both convergence and diversity are great of importance. The diversity may affect the convergence to some extent. Therefore, balancing the convergence and diversity is a key issue. Thus, to ensure the diversity of the next generation, the hierarchical clustering strategy [34] is used to divide the $i$-th Pareto front into $k$ solution sets, where the Euclidean distance is used to measure the similarity of individuals. After that, for each set, calculate the convergence of each individual, and choose the individual with the minimum conver-

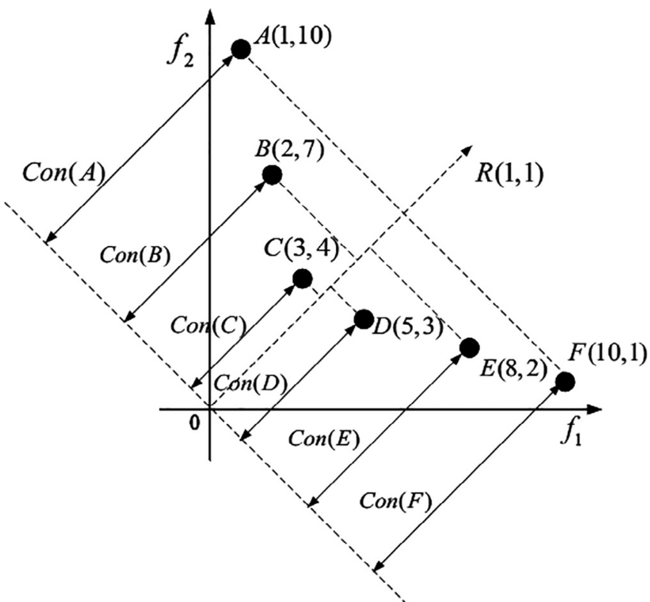



**Fig. 1.** Illustration of our analyses.

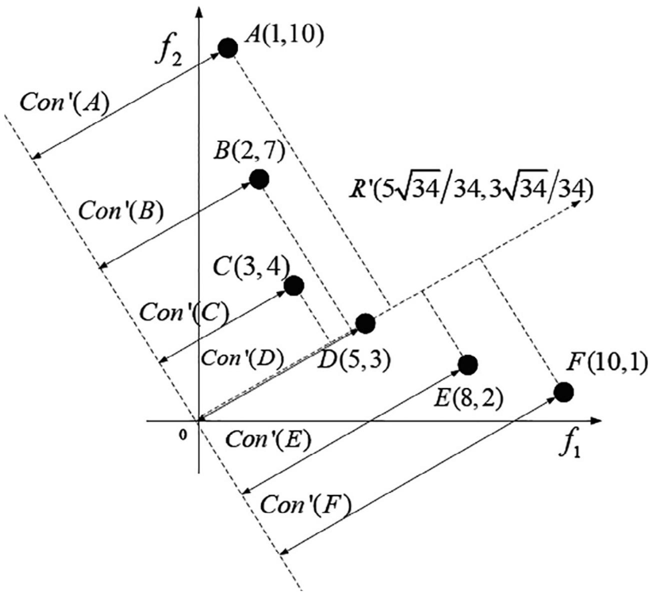



**Fig. 2.** Illustration of the adaptive reference vector strategy.

gence value as the member of next population, referring to line 5–7. Note that the parameter $k$ is adjusted adaptively, which is determined by the number of individuals to be selected from the $i$-th Pareto front.

**Algorithm 4**. Environmental selection $(F, N, M)$

```
  Input: F (various non-dominated levels), N M
  Output: P
1 P ← ∪_{j=1}^{i-1} F_j
2 K ← N − |P|
3 S ← Divide F_i into K sets using the hierarchical clustering strategy
4 for all S_k ∈ S do
5 |  Con ← Adaptive_reference_vector(S_k, |S_k|, M)
6 |  X_k ← arg min_{X∈S_k} (Con)
7 |  P ← P ∪ {X_k}
8 end
```

*4.5. Analysis of computational complexity*

As presented above, MaOEA-ARV has the following operations: 1) Non-dominated sorting; 2) Adaptive reference vector strategy; 3) Mating selection; 4) Genetic variation and 5) Environmental selection. For a population of $N$ individuals, and an optimization problem with $M$ objectives and $D$ decision variables, the computational complexity of non-dominated sorting is $O(MN^2)$ . The adaptive reference vector strategy includes Eqs. (9)–(11). The computational complexity of Eq. (10) is $O(MN)$ , and Eq. (11) requires a computational complexity of $O(MN^2)$ in the worst case. Eq. (9) needs a computational complexity of $O(M)$ . Therefore, the computational complexity of the adaptive reference vector strategy is $O(MN^2)$. The mating selection needs a runtime of $O(N)$ to form a mating pool of size $N$. For genetic variations, the simulated binary crossover and polynomial mutation are performed, which need $O(DN)$ to generate $N$ individuals. The environmental selection is to determine $N$ individuals from the combined population. In the worst case, $N$ individuals should be selected from the last Pareto front of size $2N$. Therefore, the upper boundary of the computational complexity of environmental selection is $O(MN^2)$ . In summary, considering all the components of MaOEA-ARV, the total computational complexity is $O(MN^2)$ within one generation. Compared with some representative methods, such as NSGA-III and KnEA, both of which need $O(MN^2)$ , MaOEA-ARV is competitive.

## 5. Experiments and analyses

The experimental section is divided into five subsections. Subsection 5.1 is to introduce the experimental settings, including the comparison algorithms, the test suites and the performance indicators. After that, subsection 5.2 is designed to verify our motivation for the adaptive reference vector strategy. The performance of MaOEA-ARV is tested on MOPs in subsection 5.3. Then, the effectiveness of MaOEA-ARV is verified on MaOPs in subsection 5.4. Finally, parameter $k$ in the hierarchical clustering strategy is investigated in subsection 5.5.

### 5.1. Experimental settings

#### 5.1.1. Comparison algorithms

To evaluate MaOEA-ARV, this paper uses NSGA-II/SDR [38], KnEA [46], NSGA-III [9] and MOEA/D [45] as comparison algorithms. NSGA-II/SDR, as discussed in Section 2, mainly relies on the sum of objectives as the convergence indicator, of which the drawback is the lack of the consideration of the unbalanced selections, while MaOEA-ARV is designed to utilize the adaptive reference vectors to remedy the drawback above. Therefore, it is reasonable and necessary to compare NSGA-II/SDR with MaOEA-ARV to show their differences and the effectiveness of the adaptive reference vector strategy. KnEA is famous for its

**Table 1**
Average Spread values of optimizers on DTLZ test suite

| Problem | M | MaOEA-ARV | NSGA-III | MOEA/D | NSGA-II/SDR | KnEA |
|---|---|---|---|---|---|---|
| C1_DTLZ1 | 8 | **1.6154e−1 (1.28e−2)** | 3.7733e−1 (2.55e−1) | 3.1692e−1 (8.04e−2) | 5.8167e−1 (1.15e−1) | 3.9294e−1 (1.05e−2) |
| | 10 | **1.6909e−1 (1.18e−2)** | 3.8843e−1 (3.03e−1) | 4.6966e−1 (3.62e−2) | 6.5025e−1 (1.83e−1) | 3.3939e−1 (5.07e−2) |
| | 12 | **2.0314e−1 (3.48e−2)** | 5.9576e−1 (5.52e−2) | 1.0327e+0 (5.80e−2) | 9.3818e−1 (2.85e−1) | 2.7944e−1 (3.68e−2) |
| C1_DTLZ2 | 8 | **1.6988e−1 (2.50e−2)** | 4.0975e−1 (5.66e−2) | 4.9715e−1 (4.92e−1) | 5.8378e−1 (1.32e−1) | 4.5289e−1 (1.17e−1) |
| | 10 | **1.3864e−1 (7.91e−3)** | 3.3341e−1 (2.10e−2) | 2.8070e−1 (7.73e−2) | 7.0134e−1 (1.70e−1) | 2.1692e−1 (4.59e−2) |
| | 12 | **1.2085e−1 (2.25e−3)** | 4.7359e−1 (1.60e−2) | 6.2691e−1 (3.60e−1) | 7.3399e−1 (5.37e−2) | 2.1968e−1 (7.99e−2) |
| DTLZ1 | 8 | 1.7152e−1 (8.10e−3) | 4.3401e−1 (2.52e−1) | **1.3993e−1 (4.09e−2)** | 4.2960e−1 (2.12e−2) | 9.7949e−1 (4.38e−1) |
| | 10 | **1.5842e−1 (1.92e−2)** | 5.0766e−1 (5.08e−1) | 2.4908e−1 (5.64e−2) | 6.7444e−1 (1.32e−1) | 3.1504e−1 (8.30e−2) |
| | 12 | **1.3048e−1 (9.30e−3)** | 1.2394e+0 (5.67e−1) | 8.3337e−1 (2.22e−2) | 5.3299e−1 (7.24e−2) | 2.7918e−1 (1.64e−1) |
| DTLZ2 | 8 | **1.3371e−1 (9.26e−3)** | 1.3788e−1 (6.50e−3) | 1.4102e−1 (1.72e−3) | 5.9138e−1 (2.09e−1) | 1.5134e−1 (4.66e−2) |
| | 10 | **1.5486e−1 (2.29e−2)** | 5.2971e−1 (2.13e−1) | 3.8244e−1 (1.35e−1) | 1.6981e+0 (7.13e−1) | 2.2746e−1 (1.10e−1) |
| | 12 | 1.2794e−1 (4.25e−3) | 6.3715e−1 (4.80e−1) | 1.9805e−1 (1.76e−2) | 4.2718e−1 (6.12e−2) | **1.2005e−1 (1.16e−1)** |
| DTLZ3 | 8 | **1.9370e−1 (1.59e−2)** | 8.5021e−1 (5.30e−1) | 4.9736e−1 (5.42e−1) | 5.8995e−1 (1.28e−1) | 4.9350e−1 (1.87e−1) |
| | 10 | **1.6883e−1 (2.90e−2)** | 1.2871e+0 (2.76e−1) | 6.2979e−1 (3.84e−1) | 8.9741e−1 (1.78e−1) | 2.2670e−1 (1.00e−1) |
| | 12 | **1.0971e−1 (1.67e−3)** | 1.3506e+0 (2.53e−2) | 8.3976e−1 (3.08e−1) | 6.6442e−1 (1.46e−1) | 2.0074e−1 (1.38e−1) |
| DTLZ4 | 8 | 1.3896e−1 (1.51e−3) | 2.5861e−1 (2.13e−1) | 1.2777e+0 (1.58e−1) | 8.6018e−1 (2.42e−1) | **8.9694e−2 (6.32e−2)** |
| | 10 | 1.4841e−1 (2.59e−2) | 8.9753e−1 (2.53e−1) | 1.3803e+0 (6.47e−2) | 9.9997e−1 (3.18e−5) | **5.2849e−2 (1.24e−2)** |
| | 12 | 1.2197e−1 (3.43e−3) | 3.8350e−1 (4.81e−1) | 1.1847e+0 (2.32e−2) | 8.2119e−1 (3.10e−1) | **8.1261e−2 (1.82e−2)** |
| DTLZ5 | 8 | **1.7484e−1 (2.54e−2)** | 5.8071e−1 (4.93e−2) | 1.6682e+0 (1.73e−2) | 4.4522e−1 (2.57e−2) | 3.3125e−1 (4.68e−2) |
| | 10 | **1.4485e−1 (1.20e−2)** | 7.3913e−1 (2.21e−1) | 1.8224e+0 (4.89e−2) | 4.9573e−1 (1.43e−1) | 3.3262e−1 (3.05e−2) |
| | 12 | **1.3167e−1 (1.08e−2)** | 5.7003e−1 (5.17e−2) | 1.4944e+0 (4.55e−2) | 4.6924e−1 (1.72e−2) | 2.7324e−1 (4.82e−2) |
| DTLZ6 | 8 | **1.4882e−1 (7.42e−3)** | 4.9699e−1 (3.12e−2) | 1.7697e+0 (2.51e−2) | 6.9266e−1 (1.04e−1) | 3.3065e−1 (3.10e−2) |
| | 10 | **2.0474e−1 (6.78e−2)** | 7.1827e−1 (1.11e−1) | 1.8555e+0 (7.59e−2) | 9.1192e−1 (1.50e−1) | 3.0940e−1 (1.26e−1) |
| | 12 | **1.1843e−1 (8.18e−3)** | 5.9346e−1 (5.06e−2) | 1.5259e+0 (6.94e−2) | 5.0983e−1 (1.51e−1) | 1.6097e−1 (5.67e−2) |
| DTLZ7 | 8 | **1.2802e−1 (4.90e−3)** | 5.3707e−1 (1.38e−2) | 9.6270e−1 (1.03e−2) | 4.4747e−1 (3.37e−2) | 3.4429e−1 (9.06e−2) |
| | 10 | 1.5994e−1 (4.79e−2) | 5.0072e−1 (9.98e−2) | 1.0095e+0 (3.16e−2) | 4.8830e−1 (9.86e−2) | **1.3772e−1 (9.54e−2)** |
| | 12 | **1.0951e−1 (3.50e−3)** | 5.5533e−1 (9.49e−2) | 1.0033e+0 (2.01e−2) | 5.4497e−1 (1.62e−1) | 2.5192e−1 (1.63e−2) |
| DTLZ8 | 8 | **2.0751e−1 (6.71e−3)** | 6.1392e−1 (2.80e−2) | 5.3442e−1 (4.52e−2) | 3.5437e−1 (5.64e−3) | 3.7329e−1 (7.09e−3) |
| | 10 | **3.4617e−1 (1.30e−1)** | 7.6691e−1 (2.31e−1) | 6.5543e−1 (3.12e−1) | 3.8135e−1 (1.51e−1) | 3.7003e−1 (1.45e−1) |
| | 12 | **2.1645e−1 (8.63e−3)** | 6.4070e−1 (3.82e−2) | 6.9413e−1 (5.35e−2) | 2.9443e−1 (8.54e−3) | 2.6739e−1 (9.69e−3) |
| DTLZ9 | 8 | **2.0192e−1 (8.74e−3)** | 4.9362e−1 (2.13e−2) | 3.5463e−1 (2.43e−2) | 2.1428e−1 (9.55e−3) | 2.7494e−1 (3.35e−2) |
| | 10 | **3.6097e−1 (1.18e−1)** | 7.1448e−1 (2.37e−1) | 5.1351e−1 (3.84e−1) | 4.9249e−1 (2.72e−1) | 3.7491e−1 (1.56e−1) |
| | 12 | **2.1784e−1 (1.40e−2)** | 5.3297e−1 (1.22e−2) | 4.6484e−1 (3.48e−2) | 2.4749e−1 (5.24e−3) | 3.0543e−1 (5.09e−3) |
| IDTLZ1 | 8 | 9.4567e−1 (8.66e−1) | 6.3540e−1 (2.11e−2) | 1.2792e+0 (7.33e−2) | **2.0552e−1 (7.09e−2)** | 5.1967e−1 (1.29e−1) |
| | 10 | 9.7753e−1 (8.01e−1) | 5.3141e−1 (1.97e−2) | 1.3773e+0 (7.19e−2) | **2.1167e−1 (4.14e−2)** | 3.9793e−1 (5.82e−2) |
| | 12 | 9.7446e−1 (7.99e−1) | 6.5901e−1 (4.16e−2) | 1.3628e+0 (1.17e−1) | **2.4828e−1 (9.96e−2)** | 4.2235e−1 (7.98e−2) |
| IDTLZ2 | 8 | **1.2744e−1 (1.77e−2)** | 5.7172e−1 (3.52e−2) | 5.3586e−1 (1.59e−2) | 2.9969e−1 (8.84e−2) | 2.1631e−1 (2.28e−2) |
| | 10 | **1.2236e−1 (2.33e−2)** | 4.9969e−1 (6.19e−2) | 6.3614e−1 (1.14e−2) | 2.4271e−1 (1.46e−2) | 2.0510e−1 (6.43e−2) |
| | 12 | **1.4641e−1 (8.14e−3)** | 8.2851e−1 (7.44e−2) | 8.9948e−1 (3.28e−2) | 2.3641e−1 (4.86e−2) | 1.9689e−1 (1.80e−3) |
| SDTLZ1 | 8 | **2.7537e−1 (2.75e−2)** | 9.9253e−1 (3.30e−1) | 9.5521e−1 (6.39e−3) | 8.1053e−1 (5.03e−2) | 1.1391e+0 (1.50e−1) |
| | 10 | **2.5398e−1 (6.98e−2)** | 1.0855e+0 (3.10e−1) | 1.0070e+0 (1.32e−3) | 9.3429e−1 (2.31e−2) | 8.1712e−1 (1.49e−1) |
| | 12 | **2.1990e−1 (3.82e−2)** | 1.5554e+0 (1.58e−1) | 1.0009e+0 (4.42e−5) | 9.9232e−1 (2.00e−3) | 7.6925e−1 (7.97e−2) |
| SDTLZ2 | 8 | **1.6623e−1 (1.25e−2)** | 8.0583e−1 (1.59e−2) | 1.0199e+0 (2.02e−2) | 8.1727e−1 (2.17e−2) | 6.3583e−1 (7.39e−2) |
| | 10 | **1.5360e−1 (7.42e−3)** | 9.5116e−1 (6.96e−2) | 1.0118e+0 (1.94e−2) | 9.3651e−1 (1.09e−2) | 8.6509e−1 (4.16e−2) |
| | 12 | **1.6065e−1 (1.75e−2)** | 9.3037e−1 (6.20e−2) | 1.0011e+0 (7.92e−4) | 9.9342e−1 (3.98e−3) | 9.5099e−1 (1.85e−1) |
| Ranking | | **1.33** | 3.87 | 4.18 | 3.38 | 2.24 |

knee point strategy and has attracted a lot of attentions due to its outstanding performance on MaOPs. The adaptive reference vector strategy in MaOEA-ARV also relies on the knee points as Eq. (11) indicates. By comparing KnEA and MaOEA-ARV, the differences between them can be shown to some extent. Parameter $T$ in KnEA is set according to the original paper [46]. NSGA-III and MOEA/D are classical methods in the evolutionary computation field, and have been applied to many practical applications. NSGA-III is featured with its reference point strategy. However, the reference points generated by the reference point strategy are likely to fail to properly guide the current population because of the lack of considerations of problem features. On the contrary, MaOEA-ARV is designed to adaptively adjust the reference vectors and is able to take into account the problem features as many as possible. MOEA/D is popular due to its decomposition strategy. In this paper, NSGA-III and MOEA/D are used as the base line to highlight the effectiveness of MaOEA-ARV. For NSGA-III and MOEA/D, the two-layered reference vector strategy [9] is adopted to generate the evenly distributed vectors. $(p_1, p_2)$, which is used to control the numbers of reference vectors along the outer and inner layers, is set to (99,0) for bi-objective test instances, (3,2) for 8- and 10-objective test instances, and (2,2) for 12-objective test instances. The crossover probability is set to 1.0 and the mutation probability is set to $1/D$, where $D$ is the length of decision variables. For MOEA/D, the range of neighborhood is set to $N/10$ for all test instances, and weighted sum approach is used as the aggregation function, where $N$ is the population size. All the optimizers are implemented with Matlab language and performed on PlatEMO [37], which is an open source platform.

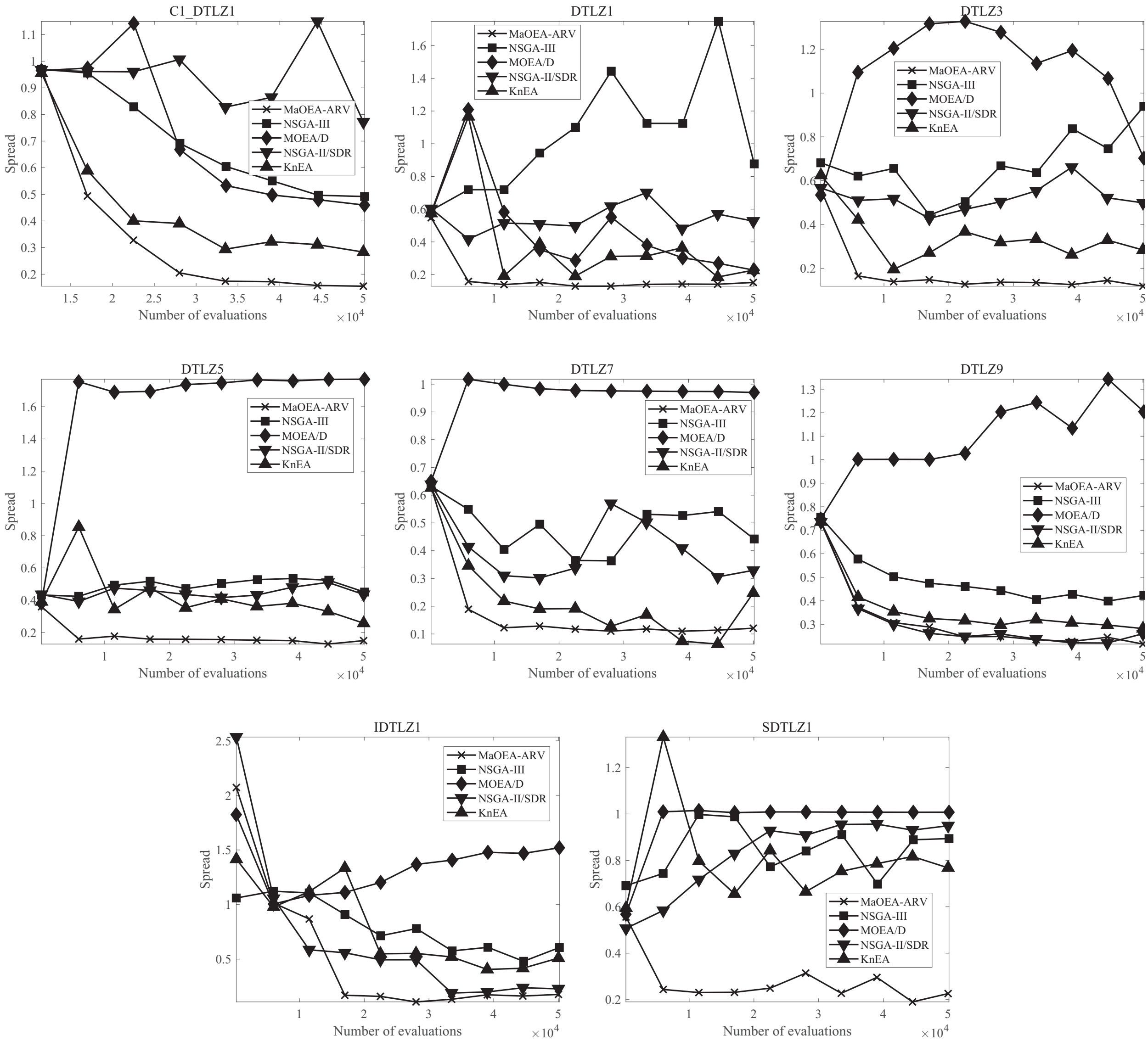



**Fig. 3.** Dynamic observations of Spread values during evolutionary process on DTLZ test suite.

### 5.1.2. Test suites

To test MaOEA-ARV from various perspectives, this paper employs ZDT [48], BT [26], DTLZ [10] and WFG [16] test suites. ZDT is a bi-objective test suite, including six test instances, ZDT1∼ZDT6, of which the features cover convexity, discrete Pareto fronts, deception and multimodality. The length of decision vector ($D$) is set to 30 for ZDT1∼ZDT3, 10 for ZDT4 and ZDT6, and 80 for ZDT5. BT test suite includes nine test instances, BT1∼BT8 with two objective and BT9 with three objectives. For BT1∼BT9, $D$ is set to 30. BT test suite is designed to test the ability of MOEAs to tackle various bias features. As paper [16] pointed out, bias features indicate the slopes of objective function and may change dramatically in the vicinities of certain solutions. Therefore, BT test suite is very suitable to verify the adaptive reference vector strategy. DTLZ test suite has nine test instances, DTLZ1∼DTLZ9, all of which are scalable to any number of objectives. In addition, the scaled DTLZ1∼DTLZ2 [9], convex DTLZ1∼DTLZ2 [23] and inverted DTLZ1∼DTLZ2 [23] are also included into the experiments. Because of these characteristics, DTLZ test suite is widely used as benchmarking problems. In DTLZ test suite, $D$ is set to $M + k - 1$, where $M$ indicates the number of objectives, and $k$ is set to 5 for DTLZ1, SDTLZ1 and IDTLZ1, 20 for DTLZ7, and 10 for the others. WFG toolkit consists of nine scalable test problems, WFG1∼WFG9, all of which have complex features, such as scalable number of parameters, dissimilar parameter domains, dissimilar tradeoff ranges, bias and so on. WFG toolkit is able to comprehensively examine the performance of any optimizers. Regarding these nine problems above, $D$ is set to $K + L$, where $K$ is set to $M - 1$ and $L$ is set to 10. The maximum generation is set to 800 for BT test suite and 200 for others.

### 5.1.3. Population sizes

For ZDT and BT test suites, the population size is set to 100. The population size is set to 230 for 8-objective test instances, 240 for 10-objective test instances and 260 for 12-objective test instances.

**Table 2**
Comparison of average IGD values of all optimizers on BT and ZDT test suites.

| Problem | M | MaOEA-ARV | NSGA-III | MOEA/D | NSGA-II/SDR | KnEA |
|---|---|---|---|---|---|---|
| BT1 | 2 | **1.5213e−1 (1.29e−1)** | 4.7777e−1 (2.77e−1) | 1.3438e+0 (2.22e−1) | 2.2494e−1 (1.07e−1) | 2.3160e−1 (6.65e−2) |
| BT2 | 2 | **1.2296e−1 (4.62e−3)** | 1.4022e−1 (2.83e−2) | 1.9443e−1 (2.07e−2) | 2.4886e−1 (1.35e−2) | 3.3954e−1 (5.92e−2) |
| BT3 | 2 | 8.4889e−2 (2.78e−2) | 3.1390e−2 (1.53e−2) | 3.0806e−1 (1.07e−1) | **1.5314e−2 (1.34e−3)** | 3.5505e−2 (1.23e−2) |
| BT4 | 2 | 3.4412e−2 (1.48e−3) | 2.6040e−2 (1.86e−3) | 1.9046e−1 (6.94e−2) | **2.4927e−2 (3.82e−3)** | 3.1484e−2 (3.28e−4) |
| BT5 | 2 | **1.1187e−1 (6.28e−2)** | 3.2850e−1 (6.66e−2) | 1.6562e+0 (4.04e−2) | 4.5197e−1 (3.91e−3) | 2.3730e−1 (3.58e−2) |
| BT6 | 2 | **2.5009e−1 (2.49e−2)** | 2.9365e−1 (1.35e−2) | 3.3859e−1 (2.41e−2) | 3.2275e−1 (5.07e−3) | 2.8339e−1 (2.33e−3) |
| BT7 | 2 | **1.4248e−1 (1.32e−2)** | 2.2531e−1 (1.12e−1) | 3.7990e−1 (5.65e−2) | 4.0823e−1 (2.26e−1) | 2.6950e−1 (7.08e−2) |
| BT8 | 2 | 4.6033e−1 (1.62e−1) | 4.2121e−1 (9.34e−2) | 1.1708e+0 (3.33e−1) | 4.2949e−1 (9.84e−2) | **3.4295e−1 (5.33e−2)** |
| BT9 | 3 | **9.7980e−2 (7.57e−3)** | 2.4779e−1 (1.05e−1) | 7.9236e−1 (1.09e−1) | 1.5159e−1 (7.63e−2) | 1.6489e−1 (6.08e−2) |
| ZDT1 | 2 | 1.3741e−2 (2.52e−3) | **4.4404e−3 (1.32e−4)** | 1.9591e−2 (7.58e−3) | 6.8173e−3 (3.10e−4) | 3.3441e−2 (2.16e−2) |
| ZDT2 | 2 | **4.7031e−3 (1.57e−4)** | 4.8479e−3 (2.64e−4) | 1.3225e−1 (4.09e−2) | 5.7596e−3 (6.85e−4) | 8.2486e−2 (7.58e−3) |
| ZDT3 | 2 | **5.8846e−3 (9.44e−5)** | 6.6172e−3 (1.58e−4) | 5.1766e−2 (2.90e−2) | 1.1508e−2 (1.83e−3) | 9.2906e−3 (1.84e−3) |
| ZDT4 | 2 | **1.6568e−2 (1.12e−2)** | 2.8712e−2 (1.14e−2) | 9.2977e−2 (6.42e−2) | 4.4525e−2 (3.44e−2) | 2.6506e−1 (1.28e−1) |
| ZDT5 | 2 | 1.7031e+0 (5.16e−1) | **6.1795e−1 (6.77e−2)** | 7.7340e+0 (3.29e−1) | 7.8194e+0 (2.80e−2) | 5.3203e+0 (1.47e+0) |
| ZDT6 | 2 | **4.3552e−3 (1.02e−3)** | 9.9362e−3 (3.13e−3) | 1.5752e−2 (2.58e−3) | 6.8108e−3 (2.97e−3) | 8.9336e−3 (1.92e−3) |
| Ranking | | **2.13** | 2.4 | 4.6 | 2.67 | 3.2 |

**Table 3**
Comparison of average HV values of all optimizers on BT and ZDT test suites.

| Problem | M | MaOEA-ARV | NSGA-III | MOEA/D | NSGA-II/SDR | KnEA |
|---|---|---|---|---|---|---|
| BT1 | 2 | **5.2593e−1 (1.57e−1)** | 2.3722e−1 (2.51e−1) | 4.2342e−1 (1.23e−1) | 4.4731e−1 (1.38e−1) | 4.5699e−1 (1.16e−1) |
| BT2 | 2 | 4.1978e−1 (1.10e−2) | 5.4220e−1 (3.35e−2) | 4.7261e−1 (2.15e−2) | **5.6040e−1 (7.16e−3)** | 3.6216e−1 (2.91e−2) |
| BT3 | 2 | 5.8092e−1 (4.80e−2) | 6.6388e−1 (3.38e−2) | 3.3665e−1 (7.60e−2) | **6.9603e−1 (2.57e−3)** | 6.5659e−1 (2.13e−2) |
| BT4 | 2 | **6.9248e−1 (3.37e−3)** | 6.8200e−1 (5.05e−3) | 4.3858e−1 (9.39e−2) | 6.8986e−1 (4.40e−3) | 6.7996e−1 (3.79e−3) |
| BT5 | 2 | **5.1211e−1 (7.86e−2)** | 2.8313e−1 (6.05e−2) | 3.9567e−1 (5.34e−2) | 1.7957e−1 (1.31e−2) | 3.9791e−1 (5.14e−2) |
| BT6 | 2 | **3.9405e−1 (9.84e−3)** | 3.6020e−1 (1.25e−2) | 3.6314e−1 (5.27e−2) | 3.8615e−1 (1.83e−2) | 3.7811e−1 (7.70e−3) |
| BT7 | 2 | **5.3173e−1 (1.26e−2)** | 4.9093e−1 (5.66e−2) | 2.9762e−1 (6.34e−2) | 3.4630e−1 (2.28e−1) | 4.5314e−1 (5.60e−2) |
| BT8 | 2 | 1.9372e−1 (1.47e−1) | 2.4257e−1 (1.46e−1) | 1.6373e−1 (1.57e−1) | 2.2956e−1 (6.23e−2) | **3.2348e−1 (9.32e−2)** |
| BT9 | 3 | **4.8225e−1 (1.26e−2)** | 2.3657e−1 (1.22e−1) | 3.5257e−1 (1.87e−1) | 3.6650e−1 (1.55e−1) | 3.3784e−1 (1.38e−1) |
| ZDT1 | 2 | **7.1832e−1 (3.51e−4)** | 7.0886e−1 (7.46e−4) | 7.0042e−1 (6.67e−3) | 7.1500e−1 (6.48e−4) | 6.9843e−1 (1.35e−2) |
| ZDT2 | 2 | **4.4275e−1 (4.19e−4)** | 4.4199e−1 (5.34e−4) | 3.0308e−1 (3.01e−2) | 4.4180e−1 (1.07e−3) | 3.6763e−1 (8.16e−3) |
| ZDT3 | 2 | **6.3333e−1 (4.11e−2)** | 5.8087e−1 (3.13e−4) | 5.8181e−1 (1.05e−3) | 5.7976e−1 (5.43e−4) | 5.8127e−1 (4.23e−4) |
| ZDT4 | 2 | 6.8316e−1 (2.86e−2) | 6.9171e−1 (9.94e−3) | 6.0521e−1 (8.10e−2) | **7.0376e−1 (9.03e−3)** | 5.6035e−1 (7.74e−2) |
| ZDT5 | 2 | **8.1451e−1 (1.26e−2)** | 8.0856e−1 (1.33e−2) | 7.3272e−1 (5.77e−3) | 7.7184e−1 (1.13e−2) | 6.5971e−1 (3.70e−2) |
| ZDT6 | 2 | **3.8549e−1 (1.54e−3)** | 3.7758e−1 (4.30e−3) | 3.6820e−1 (4.02e−3) | 3.8248e−1 (5.33e−3) | 3.8249e−1 (1.79e−3) |
| Ranking | | **4.20** | 3.07 | 1.67 | 3.47 | 2.60 |

#### 5.1.4. Performance indicators

To reflect the overall performance of MaOEA-ARV, this paper employs Spread [42], IGD [5] and HV [49] as performance indicators. IGD and HV are two indicators widely used for evaluating the convergence and diversity. For the calculation of IGD, 10000 sampling points on the Pareto front of each test instance are generated using Das and Denniss approach [8,39] with two layers. For the calculation of HV, the reference point is set to $(1, 1, \ldots, 1)$, and the objective values are nor-

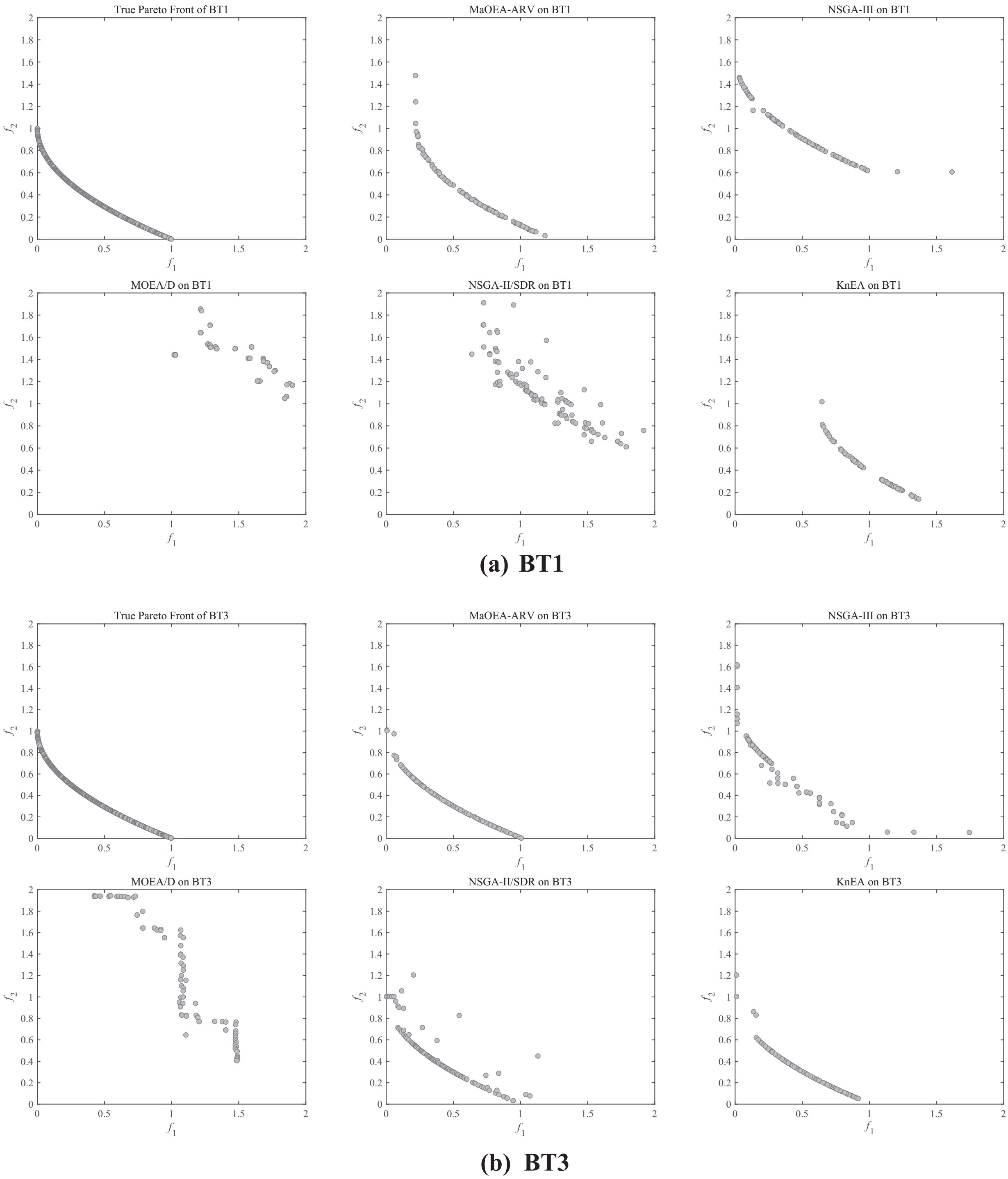
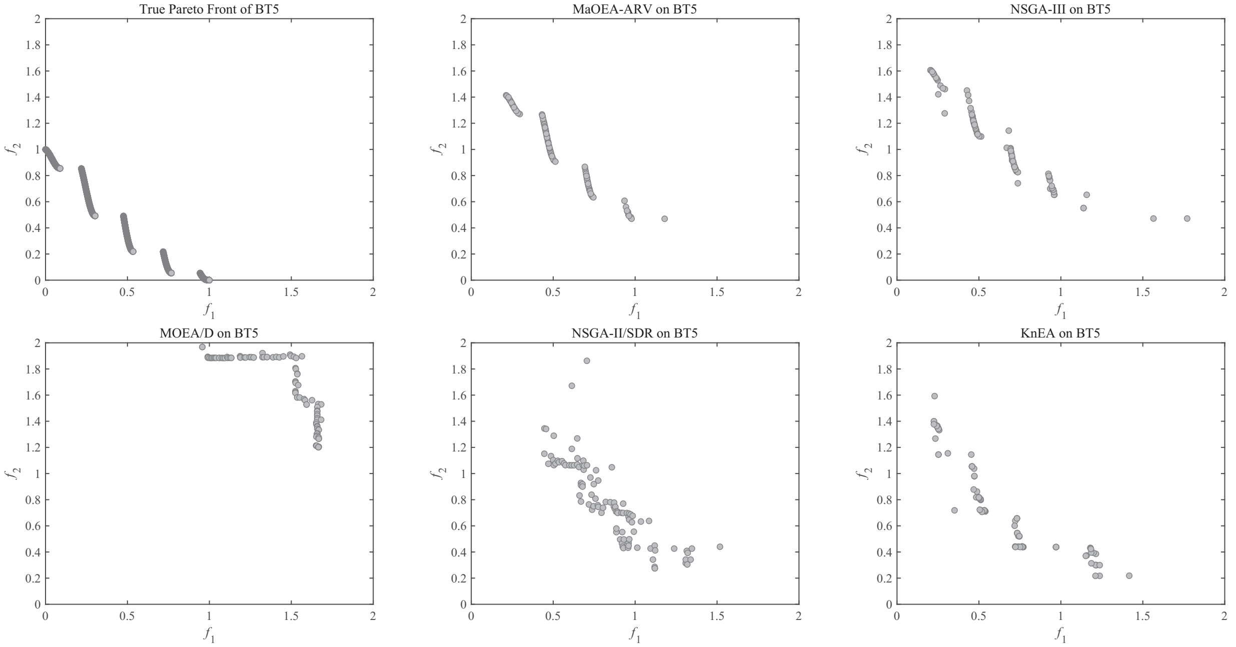



**Fig. 4.** Comparison of Pareto fronts on BT test suite.




**(c) BT5**

**Fig. 4** (*continued*)

malized by $1.1 \times \mathbf{z}^{nad}$, where $\mathbf{z}^{nad}$ is the nadir point. For test instances with more than 5 objectives, the Monte Carlo method is used for calculating HV. Spread is used to measure the extent of spread achieved among the obtained non-dominated solutions. All optimizers run 30 times for comparison purpose. The best values are highlighted in bold. Moreover, the Friedman test is used to statistically compare these optimizers. Corresponding resultant rankings are listed in each table. It is worth noting that the sampling points of WFG3 for IGD calculation in PlatEMO are regenerated according to its formulations because paper [22] has pointed out that the true Pareto front of WFG3 is not a degenerate Pareto front. In addition, it should be specially mentioned that both IGD-based and HV-based performance comparison results depend on the specification of parameter settings to some extent, such as the sampling points for IGD calculation and the reference points for HV estimation. Different comparison results may be obtained from different specifications of parameter settings. This paper just follows the general settings of IGD and HV indicators, which are also accepted in many mainstream papers.

### 5.2. Verification of our motivation

As discussed in Section 2, the proposed adaptive reference vector strategy is able to strengthen the spread. To verify our motivation, Table 1 tabulates the averages and standard deviations of spread values of all optimizers on DTLZ test suite. From Table 1, we can roughly see that MaOEA-ARV achieves the best results on most of test instances compared with MOEA/D, NSGA-II/SDR, NSGA-III and KnEA. However, MOEA/D performs the best on 8-objective DTLZ1. NSGA-II/SDR is the best algorithm on IDTLZ1 with 8, 10 and 12 objectives. KnEA also shows its superiority over other optimizers on DTLZ2 with 12 objectives, DTLZ4 with 8, 10 and 12 objectives, and DTLZ7 with 10 objectives. In terms of the number of the best values, it is evident that MaOEA-ARV outperforms its competitors. From the rankings in the last row, it can be seen that MaOEA-ARV obtains the smallest ranking value, which statistically shows the advantage of MaOEA-ARV over other optimizers. The reason for the phenomenon above can be explained as follows. KnEA, as analyzed in Section 2, suffers from the uncertainty of candidate boundary solutions and the preference to solutions with more contributions to HV. NSGA-II/SDR performs poorly due to the fixed reference vectors. For MOEA/D, the evenly generated weight vectors can not ensure the extent of the spread in the objective space due to the complexity of the true Pareto fronts. Similarly, although the reference point strategy in NSGA-III is able to generate a set of evenly distributed reference points, it obviously neglects the considerations of various features of the true Pareto fronts, resulting in the poor extent of the spread.

Fig. 3 shows the dynamic changes of spread values during the evolutionary processes of all the optimizers on 10-objective C1_DTLZ1, DTLZ1, DTLZ3, DTLZ5, DTLZ7, DTLZ9, IDTLZ1 and SDTLZ1. On C1_DTLZ1, DTLZ1, DTLZ3, DTLZ5, SDTLZ1, it is evident that MaOEA-ARV has the best convergence speed than NSGA-II/SDR, KnEA, NSGA-III and MOEA/D. On DTLZ7, MaOEA-ARV is slightly inferior to KnEA, but is still better than NSGA-III, NSGA-II/SDR and MOEA/D in terms of the convergence speed. On DTLZ9, MaOEA-ARV and NSGA-II/SDR exhibit the similar convergence process. As analyzed in Section 2, the adaptive reference vector strategy in MaOEA-ARV is able to take into account the boundary solutions and improve the spread considerably. Due to the simple sum of objectives, NSGA-II/SDR fails to maintain the spread during the evolutionary process on most test problems.

In summary, from the comparison results in Table 1 and the dynamic convergence profiles in Fig. 3, it can be concluded that the proposed MaOEA-ARV is significantly effective in improving the extent of the spread during the evolutionary process, empirically illustrating the effectiveness of our motivation.

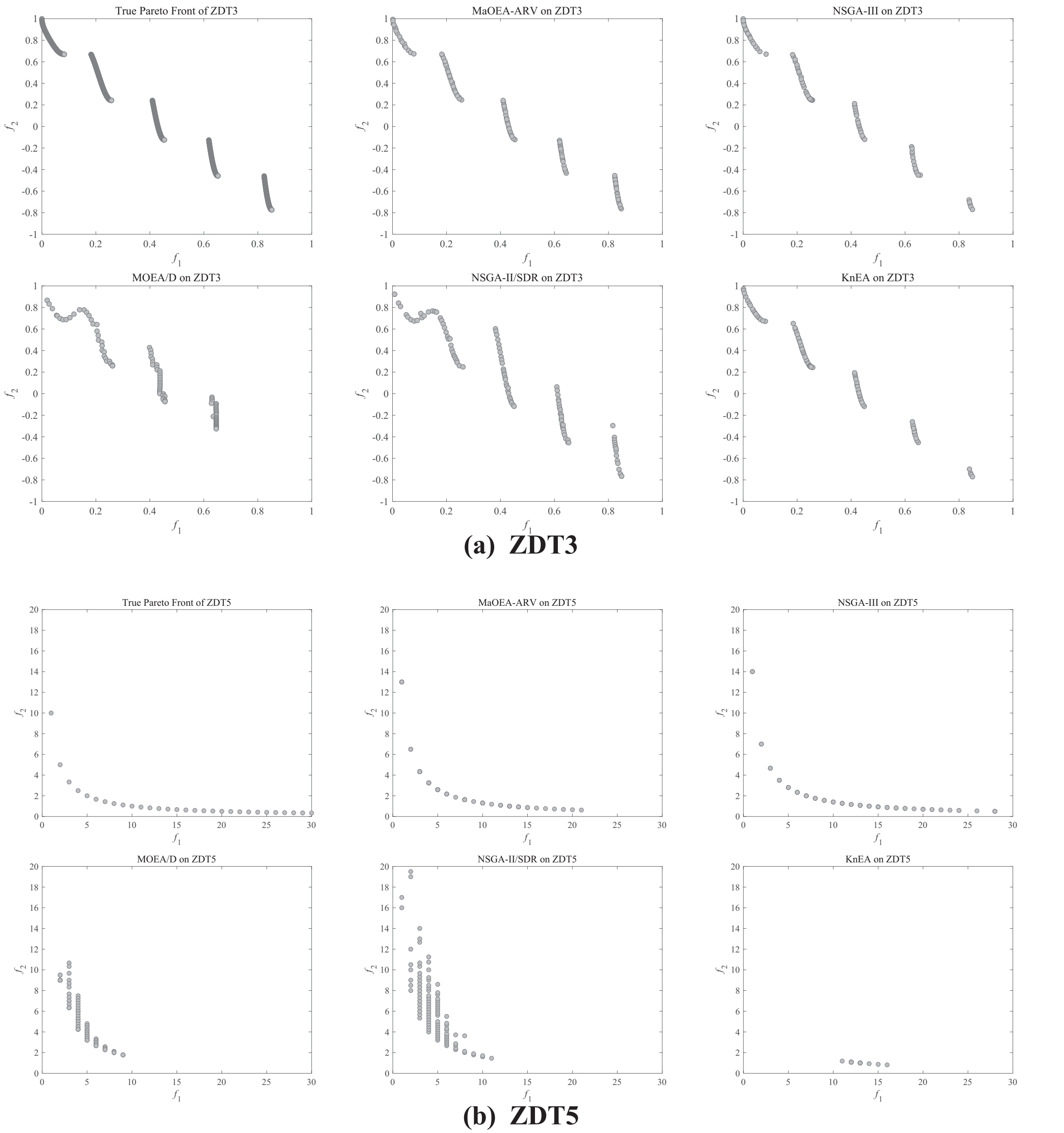



**Fig. 5.** Comparison of Pareto fronts on ZDT test suite.

### 5.3. Experiments on MOPs

Table 2 lists the averages and standard deviations of IGD values of all optimizers on BT and ZDT test suites, where the best values are in bold. From Table 2, it is evident that MaOEA-ARV wins in 10 out of 15 instances. However, NSGA-III obtains the best results on ZDT1 and ZDT5, and is superior to MaOEA-ARV. In addition, NSGA-II/SDR achieves the best on BT3 and BT4. KnEA performs the best on BT8 compared with MaOEA-ARV, NSGA-III, MOEA/D and KnEA. Table 3 exhibits the experimental HV values of all optimizers. It can be seen that, from Table 3, MaOEA-ARV performs the best on 11 test instances, and loses to NSGA-III on ZDT1, NSGA-II/SDR on BT3 and BT4, and KnEA on BT8, respectively. Besides, from Tables 2 and 3, it can be further observed that MaOEA-ARV is always inferior to NSGA-II/SDR on BT3, which is an instance with the position-related bias according to its definition [26]. This observation illustrates that MaOEA-ARV is less effective in dealing with such problems. However, in terms of the overall performance of MaOEA-ARV on BT and ZDT test suites, the advantage of MaOEA-ARV is outstanding. Additionally, from the Friedman test results listed in the last rows of Tables 2 and 3, it can be observed that MaOEA-ARV is statistically the best optimizer in terms of the convergence and diversity.

Figs. 4 and 5 plot the Pareto fronts obtained by all optimizers on BT1, BT3, BT5, ZDT3 and ZDT5. For easy comparisons, this paper unifies the ranges of all the axes. On BT1, it can be seen that both NSGA-II/SDR and MOEA/D perform very poor, and do not obtain the complete Pareto fronts. Different from NSGA-II/SDR and MOEA/D, NSGS-III and KnEA exhibit the similar shapes as the true Pareto front of BT1 shows. However, both of them are inferior to MaOEA-ARV because MaOEA-ARV exhibits a smooth and compete Pareto front as the true Pareto front shows. Similarly, NSGA-II/SDR and MOEA/D are consistently the worst optimizers compared with the true Pareto fronts of BT3. On BT3, NSGA-III and KnEA are slightly better than NSGA-II/SDR and MOEA/D, but are still worse than MaOEA-ARV. The same phenomenon can also be observed from the comparisons on BT5. Both MOEA/D and NSGA-II/SDR fail to obtain the similar Pareto fronts to the true Pareto fronts of BT5. The Pareto front obtained by MaOEA-ARV is significantly better than those obtained by NSGA-III and KnEA in terms of the spread. On ZDT3, it is observed that both NSGA-II/SDR and MOEA/D fail to distinguish the disconnected Pareto fronts. MaOEA-ARV obtains the same Pareto fronts as the true Pareto front of ZDT3, and NSGA-III obtains an incomplete Pareto front compared with the true Pareto front of ZDT3. Moreover, on ZDT3, the bottom-right area of the Pareto front obtained by KnEA is fragmentary. On ZDT5, whose true Pareto front is evidently different from test instances above, MaOEA-ARV is still superior to other optimizers in terms of the convergence, diversity and spread. KnEA only obtains partial Pareto fronts. Both NSGA-II/SDR and MOEA/D evidently fail to achieve the approximation of the true Pareto front of ZDT5. From the comparisons of the obtained Pareto fronts of various instances, it can be concluded, in terms of the spread, the adaptive reference vector strategy

**Table 4**
Comparison of average IGD values of all optimizers on WFG test suite.

| Problem | M | MaOEA-ARV | NSGA-III | MOEA/D | NSGA-II/SDR | KnEA |
|---|---|---|---|---|---|---|
| WFG1 | 8 | 2.2219e+0 (3.42e−1) | 1.4879e+0 (7.68e−2) | 1.6412e+0 (1.01e−1) | 1.5241e+0 (3.86e−2) | **1.2543e+0 (5.53e−2)** |
| | 10 | 2.9479e+0 (2.77e−2) | 2.0387e+0 (3.97e−2) | 2.3256e+0 (1.48e−1) | 1.6934e+0 (1.28e−1) | **1.4238e+0 (9.39e−2)** |
| | 12 | 3.3035e+0 (2.96e−2) | 1.9482e+0 (2.30e−1) | 1.9519e+0 (3.83e−2) | 1.5719e+0 (1.08e−2) | **1.5511e+0 (1.24e−1)** |
| WFG2 | 8 | **1.0072e+0 (3.26e−2)** | 1.2925e+0 (4.07e−1) | 1.8531e+0 (1.96e−2) | 1.3246e+0 (1.03e−1) | 1.1504e+0 (9.91e−2) |
| | 10 | 1.3914e+0 (2.63e−2) | 1.5227e+0 (1.22e−1) | 2.0728e+0 (4.13e−2) | 1.7359e+0 (5.63e−2) | **1.1679e+0 (6.08e−3)** |
| | 12 | 1.4161e+0 (2.79e−2) | **1.2843e+0 (1.28e−1)** | 1.7947e+0 (2.92e−2) | 1.6862e+0 (3.07e−3) | 1.3155e+0 (7.80e−2) |
| WFG3 | 8 | 2.8953e+0 (3.63e−1) | **7.8941e−1 (1.28e−1)** | 4.1442e+0 (1.31e−1) | 1.8522e+0 (4.82e−1) | 1.1243e+0 (1.11e−1) |
| | 10 | 2.8356e+0 (3.01e−1) | 1.5656e+0 (1.31e+0) | 7.1353e+0 (8.13e−1) | 1.7252e+0 (5.67e−1) | **1.5316e+0 (1.59e−1)** |
| | 12 | 3.1522e+0 (2.74e−1) | 2.8724e+0 (5.58e−1) | 6.2835e+0 (2.48e−1) | 2.4172e+0 (1.81e−1) | **1.6321e+0 (1.75e−1)** |
| WFG4 | 8 | **2.8648e+0 (1.63e−2)** | 2.9622e+0 (3.07e−3) | 7.1056e+0 (8.00e−3) | 2.9057e+0 (2.59e−2) | 2.9954e+0 (1.13e−2) |
| | 10 | **4.3127e+0 (6.76e−2)** | 4.7486e+0 (1.15e−2) | 9.5062e+0 (5.07e−2) | 4.4727e+0 (1.82e−2) | 4.4441e+0 (7.05e−2) |
| | 12 | **5.8513e+0 (2.07e−2)** | 6.6877e+0 (8.08e−2) | 1.2838e+1 (1.95e−1) | 6.1382e+0 (1.12e−1) | 5.9034e+0 (3.91e−3) |
| WFG5 | 8 | **2.7994e+0 (4.67e−2)** | 2.9390e+0 (1.10e−2) | 6.5755e+0 (1.58e−1) | 3.0004e+0 (5.25e−2) | 3.0029e+0 (2.30e−2) |
| | 10 | **4.1954e+0 (4.16e−2)** | 4.6377e+0 (2.85e−2) | 9.1288e+0 (1.13e−1) | 4.5329e+0 (1.25e−1) | 4.4250e+0 (3.73e−2) |
| | 12 | **5.6331e+0 (3.60e−2)** | 6.6480e+0 (2.41e−3) | 1.2324e+1 (1.15e−1) | 6.2479e+0 (5.90e−2) | 5.8832e+0 (6.84e−2) |
| WFG6 | 8 | **2.8631e+0 (4.19e−2)** | 2.9694e+0 (2.23e−2) | 7.1543e+0 (1.44e−1) | 3.1099e+0 (8.08e−2) | 3.0811e+0 (4.80e−3) |
| | 10 | **4.3307e+0 (4.51e−2)** | 4.7647e+0 (2.79e−2) | 9.5255e+0 (1.75e−1) | 4.6941e+0 (4.91e−2) | 4.6266e+0 (1.78e−2) |
| | 12 | **5.8147e+0 (3.26e−2)** | 6.8317e+0 (2.43e−2) | 1.2598e+1 (1.99e−1) | 6.3725e+0 (9.85e−2) | 6.1301e+0 (4.23e−2) |
| WFG7 | 8 | **2.8931e+0 (1.20e−2)** | 2.9625e+0 (1.33e−2) | 7.0131e+0 (1.72e−1) | 3.0200e+0 (6.21e−2) | 2.9796e+0 (3.22e−2) |
| | 10 | **4.2679e+0 (4.23e−2)** | 4.6820e+0 (3.29e−2) | 9.4773e+0 (1.55e−2) | 4.5346e+0 (3.94e−2) | 4.3691e+0 (4.21e−2) |
| | 12 | **5.8212e+0 (7.86e−2)** | 7.1882e+0 (4.01e−1) | 1.3132e+1 (3.34e−2) | 6.0353e+0 (3.41e−2) | 5.9009e+0 (1.95e−2) |
| WFG8 | 8 | **3.0605e+0 (1.78e−2)** | 3.3394e+0 (2.69e−1) | 6.4868e+0 (8.06e−2) | 3.0934e+0 (9.88e−2) | 3.1493e+0 (2.45e−2) |
| | 10 | **4.4247e+0 (3.69e−2)** | 5.0426e+0 (5.64e−1) | 8.7669e+0 (1.19e−1) | 4.6248e+0 (2.56e−2) | 4.5777e+0 (9.46e−2) |
| | 12 | **5.8847e+0 (8.37e−2)** | 6.4458e+0 (2.55e−2) | 1.1627e+1 (1.74e−1) | 6.2970e+0 (1.48e−2) | 5.9794e+0 (1.12e−1) |
| WFG9 | 8 | **2.8675e+0 (3.58e−2)** | 2.9577e+0 (3.20e−2) | 6.7502e+0 (9.77e−2) | 2.8908e+0 (2.73e−2) | 2.9266e+0 (3.87e−2) |
| | 10 | **4.2211e+0 (2.77e−2)** | 4.4186e+0 (2.21e−2) | 9.0714e+0 (3.26e−1) | 4.4011e+0 (4.21e−3) | 4.2544e+0 (4.43e−2) |
| | 12 | **5.4979e+0 (8.70e−2)** | 6.5063e+0 (1.01e−1) | 1.2048e+1 (1.00e+0) | 5.9749e+0 (1.54e−1) | 5.6553e+0 (8.26e−2) |
| Ranking | | **1.89** | 3.2 | 4.87 | 2.98 | 2.06 |

in MaOEA-ARV is evidently superior to the sum of objectives in NSGA-II/SDR on MOPs. Regarding the convergence and diversity, it is evident that MaOEA-ARV is more effective than NSGA-II/SDR, NSGA-III, MOEA/D and KnEA in tackling MOPs, especially for MOPs with bias features.

### 5.4. Experiments on MaOPs

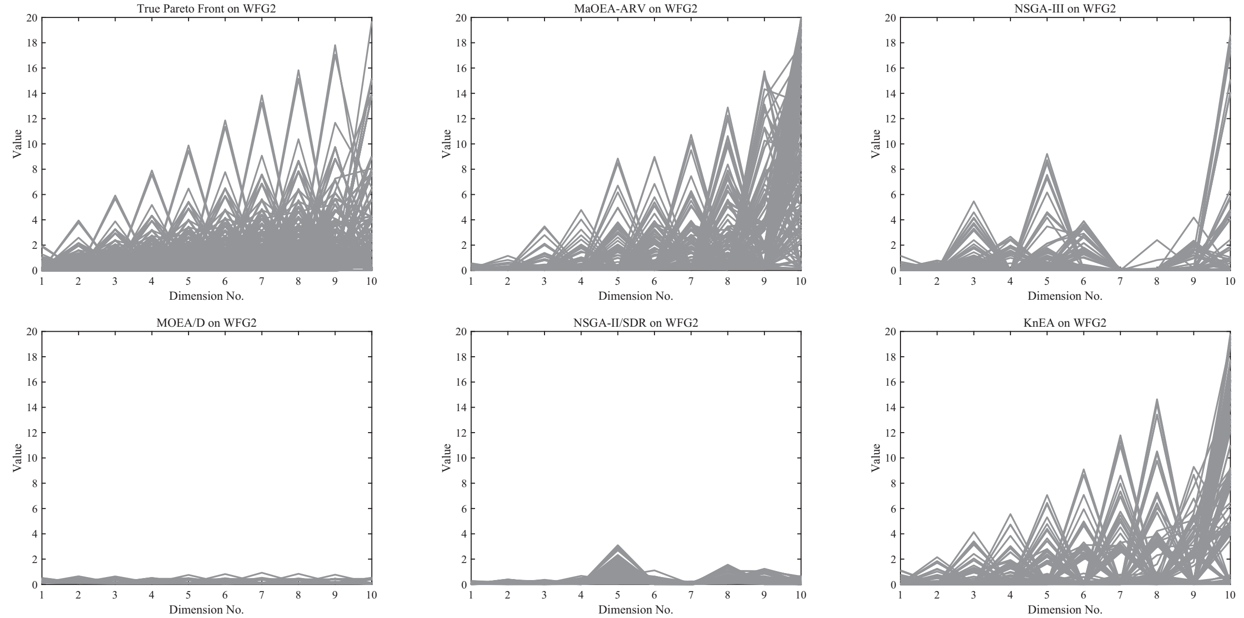
This section is to verify MaOEA-ARV on many-objective WFG test suite. The average IGD values are tabulated in Table 4. As can be seen from Table 4, MaOEA-ARV performs the best on WFG4∼WFG9 with 8, 10 and 12 objectives. On WFG1 with 8, 10 and 12 objectives, WFG2 with 10 objectives, WFG3 with 10 objectives, KnEA is the best optimizer compared with others. On WFG2 with 12 objectives and WFG3 with 8 objectives, NSGA-III shows its superiority over MaOEA-ARV, MOEA/D and NSGA-II/SDR. Table 5 further lists the average spread values of all optimizers. The fact can be observed that MaOEA-ARV is the best algorithm compared with others on all instances except for WFG2 with 12 objectives, WFG7 with 8 objectives and WFG9 with 10 objectives. Besides, the Friedman test results in the last rows of Tables 4 and 5, further confirm the analyses above from the statistical perspective.

Fig. 6 visually presents the Pareto fronts obtained by all the optimizers on 10-objective WFG2, WFG4, WFG6 and WFG8. On WFG2, MaOEA-ARV obtains the similar Pareto front as the true Pareto front exhibits. The Pareto fronts obtained by NSGA-II/SDR and MOEA/D are still far from satisfactory. Both NSGA-III and KnEA trend to converge to the true Pareto fronts, but are still less effective compared with MaOEA-ARV. On WFG4, MaOEA-ARV is very outstanding in terms of the convergence, diversity and spread according to the obtained Pareto front. In addition, the evenness of the Pareto front obtained by NSGA-III is worse than that of MaOEA-ARV. The same observation also goes to KnEA and NSGA-II/SDR. As for MOEA/D, it is still less satisfying as it exhibits on other instances. On WFG6, MaOEA-ARV consistently performs the best compared with other optimizers. On WFG8, both MaOEA-ARV and NSGA-III exhibit the similar performance, but are better than NSGA-II/SDR and KnEA. MOEA/D is the worst optimizer compared with others.

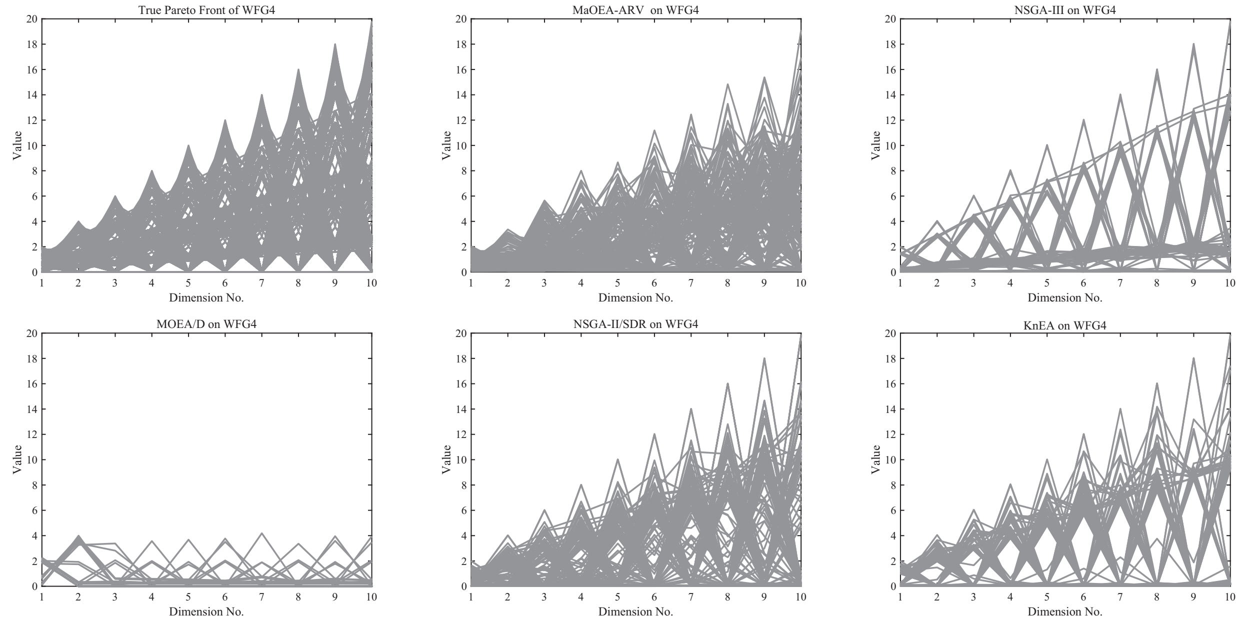
In summary, from the experimental results and the visual exhibitions of the obtained Pareto fronts, it can be concluded that MaOEA-ARV performs the best in dealing with MOPs and MaOPs. On BT test suite with bias features, MaOEA-ARV shows its superiority over other optimizers. On WFG test suite, MaOEA-ARV is also able to outperform some state-of-art optimizers. As explained in Section 2, the core idea of MaOEA-ARV is just the proposed adaptive reference vector strategy, and the adaptive reference vector strategy is demonstrated to be very effective in ensuring the spread and convergence. To sum up, the overall performance of MaOEA-ARV is also outstanding as the analyses above show.

**Table 5**
Comparison of average Spread values of all optimizers on WFG test suite.

| Problem | M | MaOEA-ARV | NSGA-III | MOEA/D | NSGA-II/SDR | KnEA |
|---|---|---|---|---|---|---|
| WFG1 | 8 | **3.1721e−1 (9.31e−3)** | 6.5245e−1 (1.72e−2) | 8.7568e−1 (5.71e−2) | 6.9897e−1 (3.67e−2) | 6.3399e−1 (5.20e−2) |
| | 10 | **2.9871e−1 (1.58e−2)** | 7.3453e−1 (6.77e−2) | 9.5170e−1 (8.01e−2) | 7.8511e−1 (4.54e−2) | 6.4925e−1 (1.42e−2) |
| | 12 | **3.1907e−1 (5.94e−3)** | 8.0933e−1 (2.80e−2) | 9.9188e−1 (1.66e−3) | 8.1510e−1 (7.12e−2) | 6.5805e−1 (7.15e−3) |
| WFG2 | 8 | **1.9550e−1 (1.32e−2)** | 7.6058e−1 (6.89e−2) | 7.8679e−1 (1.51e−2) | 7.0433e−1 (2.71e−2) | 5.6529e−1 (4.56e−2) |
| | 10 | **2.1444e−1 (9.30e−3)** | 7.7510e−1 (2.11e−2) | 8.5759e−1 (2.72e−2) | 7.5907e−1 (1.46e−2) | 5.4206e−1 (4.82e−3) |
| | 12 | 5.4872e−1 (2.10e−2) | 9.2428e−1 (7.54e−2) | 1.0136e+0 (9.25e−3) | 8.1467e−1 (4.07e−3) | **2.0427e−1 (4.85e−3)** |
| WFG3 | 8 | **1.5903e−1 (6.60e−3)** | 6.6954e−1 (1.78e−2) | 5.5509e−1 (7.21e−3) | 4.1269e−1 (4.12e−2) | 2.8623e−1 (2.75e−2) |
| | 10 | **1.6535e−1 (1.12e−2)** | 6.5345e−1 (4.74e−2) | 7.4771e−1 (2.34e−2) | 4.5129e−1 (7.67e−2) | 2.9596e−1 (3.39e−2) |
| | 12 | **1.4553e−1 (1.21e−2)** | 6.7642e−1 (5.31e−2) | 1.2195e+0 (2.83e−2) | 4.7466e−1 (7.37e−2) | 3.5730e−1 (1.75e−2) |
| WFG4 | 8 | **1.1845e−1 (2.89e−3)** | 2.7856e−1 (6.30e−3) | 1.0266e+0 (2.41e−1) | 3.7280e−1 (4.58e−2) | 3.4249e−1 (1.69e−2) |
| | 10 | **1.1642e−1 (1.00e−2)** | 3.5474e−1 (1.76e−3) | 8.5461e−1 (1.79e−1) | 4.2219e−1 (8.59e−3) | 3.3317e−1 (6.06e−3) |
| | 12 | **1.0819e−1 (4.47e−3)** | 1.6793e−1 (2.00e−2) | 1.1041e+0 (4.39e−2) | 5.0548e−1 (1.18e−1) | 3.7187e−1 (2.68e−2) |
| WFG5 | 8 | **1.0753e−1 (8.64e−3)** | 2.7287e−1 (3.73e−3) | 1.1729e+0 (1.55e−2) | 4.6885e−1 (5.50e−2) | 3.8603e−1 (4.92e−2) |
| | 10 | **1.1283e−1 (3.25e−3)** | 3.6966e−1 (7.27e−3) | 1.2349e+0 (4.63e−2) | 4.9510e−1 (1.06e−1) | 3.4297e−1 (3.89e−2) |
| | 12 | **1.2062e−1 (5.54e−3)** | 1.4362e−1 (1.73e−2) | 1.1219e+0 (1.15e−1) | 6.9503e−1 (8.76e−2) | 4.0796e−1 (2.87e−2) |
| WFG6 | 8 | **1.1229e−1 (7.77e−3)** | 2.6953e−1 (5.06e−3) | 7.6694e−1 (1.03e−1) | 5.4939e−1 (5.85e−2) | 4.7366e−1 (2.46e−2) |
| | 10 | **1.2155e−1 (3.71e−3)** | 3.5778e−1 (4.82e−3) | 9.7866e−1 (1.42e−1) | 4.6461e−1 (1.30e−1) | 3.9945e−1 (4.64e−2) |
| | 12 | **1.2314e−1 (3.42e−3)** | 1.8083e−1 (1.21e−3) | 1.0751e+0 (1.10e−1) | 5.0794e−1 (1.83e−1) | 4.8452e−1 (2.07e−2) |
| WFG7 | 8 | 2.5975e−1 (4.74e−3) | **1.1035e−1 (1.45e−2)** | 1.1426e+0 (3.52e−1) | 5.3072e−1 (9.84e−2) | 3.8679e−1 (1.30e−2) |
| | 10 | **1.1749e−1 (8.02e−3)** | 3.5518e−1 (1.15e−2) | 9.3235e−1 (3.44e−1) | 3.9353e−1 (6.45e−2) | 3.1140e−1 (1.69e−2) |
| | 12 | **1.2433e−1 (3.74e−4)** | 7.5352e−1 (1.04e−1) | 1.1102e+0 (1.65e−2) | 3.8051e−1 (1.64e−2) | 3.2320e−1 (1.08e−2) |
| WFG8 | 8 | **1.1356e−1 (5.30e−3)** | 5.0057e−1 (7.08e−2) | 1.1547e+0 (1.20e−1) | 3.8088e−1 (7.39e−3) | 5.5212e−1 (4.63e−2) |
| | 10 | **1.1997e−1 (2.85e−3)** | 5.8581e−1 (1.36e−1) | 1.1246e+0 (2.16e−1) | 3.9184e−1 (4.17e−2) | 4.8629e−1 (3.60e−2) |
| | 12 | **1.2320e−1 (5.33e−3)** | 6.5830e−1 (1.33e−1) | 1.1451e+0 (4.50e−2) | 6.1899e−1 (2.08e−1) | 4.9973e−1 (9.37e−3) |
| WFG9 | 8 | **1.2440e−1 (7.53e−3)** | 3.0612e−1 (3.80e−2) | 1.2532e+0 (3.63e−2) | 3.8742e−1 (3.39e−2) | 2.7880e−1 (4.10e−2) |
| | 10 | 3.5723e−1 (9.05e−3) | **1.1375e−1 (3.71e−3)** | 1.2917e+0 (2.31e−2) | 4.3380e−1 (2.26e−2) | 2.8583e−1 (7.91e−3) |
| | 12 | **1.1799e−1 (1.51e−3)** | 3.8196e−1 (1.83e−1) | 1.1718e+0 (7.04e−2) | 5.5543e−1 (6.62e−2) | 3.2527e−1 (2.22e−2) |
| Ranking | | **1.15** | 2.96 | 4.96 | 3.56 | 2.37 |




**(a) WFG2**

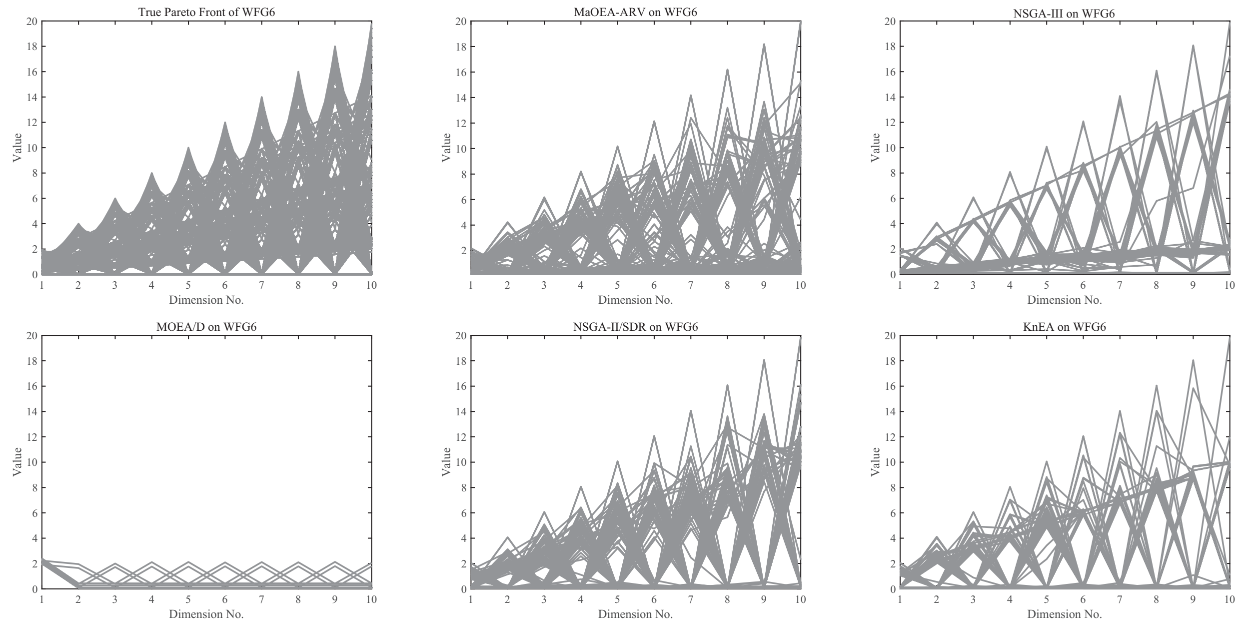



**(b) WFG4**

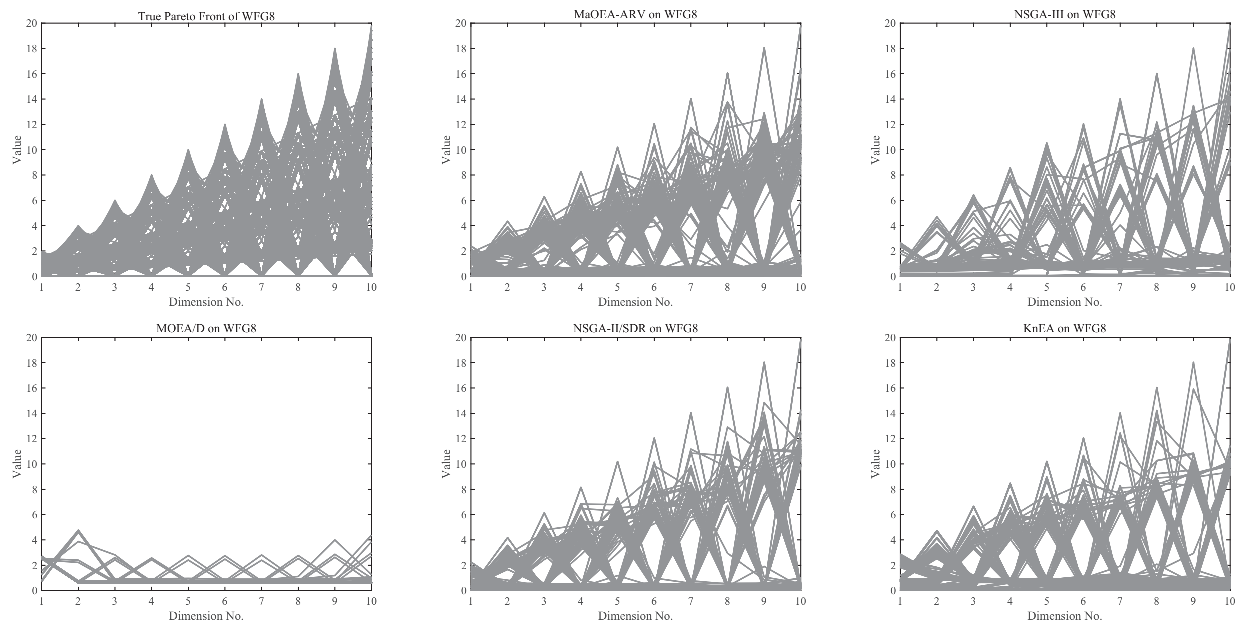
**Fig. 6.** Comparison of Pareto fronts on WFG test suite.

### 5.5. Effectiveness of parameter k

This section is to exhibit the effectiveness of the adaptive parameter $k$ in the hierarchical clustering strategy. The employed instances are bi-objective BT9 and 10-objective WFG6. Other parameters, such as the population size, the number of decision variables, and the maximum generation, remain the same as explained above. In addition, the number of non-dominated solutions in the last Pareto front generated by the non-dominated sorting strategy is also included into the comparisons as the base line.



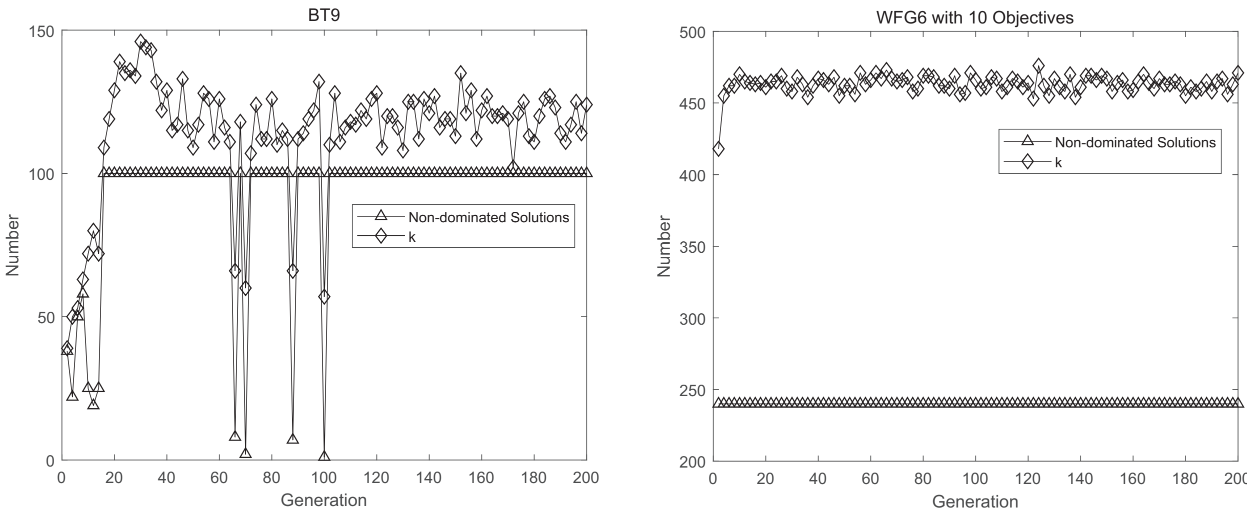

**(c) WFG6**




**(d) WFG8**

**Fig. 6** (*continued*)

As can be seen from Fig. 7(a), when the number of non-dominated solutions in the last Pareto front is less than the population size, namely 100, the parameter $k$ is adjusted adaptively to select certain number of solutions. Further, the parameter $k$ is fixed to the population size when the number of non-dominated solutions is more than the predefined population size. This mechanism is simple yet effective in determining the parameter $k$. From Fig. 7(b), it can be observed that parameter $k$ is consistently fixed to the population size, which indicates that there exists a large amount of non-dominated solutions in the last Pareto front. Actually, the last Pareto front is the only one Pareto front because most of the solutions during the evolutionary process are non-dominated with each other. It can be concluded from the analyses that parameter $k$ can be adaptively adjusted to properly control the population size under various conditions.




**(a) BT9** **(b) WFG6**

**Fig. 7.** Dynamic illustration of parameter $k$ on BT9 and WFG6 test instances.

## 6. Conclusion and future work

Various methods for measuring the convergence have been proposed over the past three decades. However, according to the analyses of these representative methods, it is found that these methods may result in unbalanced selections of candidate solutions, thus affecting the extent of the spread among the achieved non-dominated solutions. To remedy this issue, this paper designs an adaptive reference vector strategy, which is able to adaptively adjust the reference vectors according to the current solutions. After that, the hierarchical clustering strategy is used to partition current population into multiple clusters for the even distribution of the next generation. Based on the adaptive reference vector strategy, MaOEA-ARV is proposed accordingly. Extensive experiments are conducted on DTLZ, BT, ZDT, WFG with 2 to 12 objectives. The experimental results are analyzed using IGD, HV and Spread performance indicators. The performance of MaOEA-ARV is superior to some state-of-art algorithms according to the experimental analyses.

However, as discussed above, the true Pareto fronts of practical MaOPs are various. Although MaOEA-ARV performs very well on most test instances, it should be recognized that the adaptive reference vector strategy is ineffective on some special problems, such as WFG1 and WFG3. In addition, as discussed in recent papers [13,21], both DTLZ and WFG test suites can only cover limited problem characteristics, and the proposed adaptive reference vector strategy should be further assessed and verified on more test suites and practical engineering problems for demonstrating its usefulness and effectiveness. Therefore, the further work should be focused on the following two aspects. The first one is on the investigation and research of realistic problems with more diverse Pareto fronts. The second aspect lies in the further improvement and assessment of the adaptive reference vector strategy.

## CRediT authorship contribution statement

**Maoqing Zhang:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Lei Wang:** Supervision, Funding acquisition, Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Writing - review & editing. **Wuzhao Li:** Supervision, Funding acquisition, Conceptualization, Data curation, Formal analysis, Methodology, Project administration, Resources, Writing - review & editing. **Bo Hu:** Data curation, Writing - review & editing. **Dongyang Li:** Data curation, Writing - review & editing. **Qidi Wu:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Writing - review & editing.

## Declaration of Competing Interest

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

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 71771176 and 61503287), Natural Science Foundation of Shanghai (No. 19ZR1479000), Science and Technology Winter Olympic Project (No. 2018YFF0300505), National Science Foundation of Zhejiang Province (No. LY18F030010), Open Research Project of the State Key Laboratory of Industrial Control Technology (No. ICT20003), and Jiaxing public welfare research project (No. 2020AY10028).

Authors would express our thanks to the editors and reviewers for their insightful comments in improving the quality of this paper.

## References

[1] L. Abualigah, M. Abd Elaziz, A.G. Hussien, B. Alsalibi, S.J. Jalali, A.H. Gandomi, Lightning search algorithm: a comprehensive survey, Applied Intelligence (2020) 1–24. .
[2] A. Almunif, L. Fan, Optimal pmu placement for modeling power grid observability with mathematical programming methods, International Transactions on Electrical Energy Systems 30 (2) (2020) e12182.
[3] A.S. Assiri, A.G. Hussien, M. Amin, Ant lion optimization: variants, hybrids, and applications, IEEE Access 8 (2020) 77746–77764.
[4] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (1) (2011) 45–76.
[5] C.A.C. Coello, N.C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genetic Programming and Evolvable Machines 6 (2) (2005) 163–190.
[6] Z. Cui, X. Xu, F. Xue, X. Cai, Y. Cao, W. Zhang, J. Chen, Personalized recommendation system based on collaborative filtering for iot scenarios, IEEE Transactions on Services Computing 13 (4) (2020) 685–695.
[7] Z. Cui, F. Xue, S. Zhang, X. Cai, Y. Cao, W. Zhang, J. Chen, A hybrid blockchain-based identity authentication scheme for multi-wsn, IEEE Transactions on Services Computing 13 (2) (2020) 241–251.
[8] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM Journal on Optimization 8 (3) (1998) 631–657.
[9] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints, IEEE Transactions on Evolutionary Computation 18 (4) (2013) 577–601.
[10] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: Evolutionary Multiobjective Optimization, Springer, 2005, pp. 105–145.
[11] S. Dwivedi, M. Vardhan, S. Tripathi, A.K. Shukla, Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection, Evolutionary Intelligence 13 (1) (2020) 103–117.
[12] F. Gu, H. Liu, A novel weight design in multi-objective evolutionary algorithm, in: 2010 International Conference on Computational Intelligence and Security, IEEE, 2010, pp. 137–141..
[13] A. Habib, H.K. Singh, T. Chugh, T. Ray, K. Miettinen, A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization, IEEE Transactions on Evolutionary Computation 23 (6) (2019) 1000–1014.
[14] Z. He, G.G. Yen, J. Zhang, Fuzzy-based pareto optimality for many-objective evolutionary algorithms, IEEE Transactions on Evolutionary Computation 18 (2) (2013) 269–285.
[15] Z. Hou, C. He, R. Cheng, Reformulating preferences into constraints for evolutionary multi-and many-objective optimization, Information Sciences 541 (2020) 1–15.
[16] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506.
[17] A.G. Hussien, M. Amin, M. Abd El Aziz, A comprehensive review of moth-flame optimisation: variants, hybrids, and applications, Journal of Experimental & Theoretical Artificial Intelligence (2020) 1–21.
[18] A.G. Hussien, M. Amin, M. Wang, G. Liang, A. Alsanad, A. Gumaei, H. Chen, Crow search algorithm: theory, recent advances, and applications, IEEE Access 8 (2020) 173548–173565.
[19] A.G. Hussien, A.E. Hassanien, E.H. Houssein, Swarming behaviour of salps algorithm for predicting chemical compound activities, in: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), IEEE, 2017, pp. 315–320. .
[20] A.G. Hussien, D. Oliva, E.H. Houssein, A.A. Juan, X. Yu, Binary whale optimization algorithm for dimensionality reduction, Mathematics 8 (10) (2020) 1821.
[21] H. Ishibuchi, L. He, K. Shang, Regular pareto front shape is not realistic. In 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 2034–2041..
[22] H. Ishibuchi, H. Masuda, Y. Nojima, Pareto fronts of many-objective degenerate test problems, IEEE Transactions on Evolutionary Computation 20 (5) (2015) 807–813.
[23] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach, IEEE Transactions on Evolutionary Computation 18 (4) (2013) 602–622.
[24] S. Jiang, Z. Cai, J. Zhang, Y. Ong, Multiobjective optimization by decomposition with pareto-adaptive weight vectors, in: 2011 Seventh International Conference on Natural Computation, vol. 3, IEEE, 2011, pp. 1260–1264..
[25] H. Li, D. Landa, An adaptive evolutionary multi-objective approach based on simulated annealing, Evolutionary Computation 19 (4) (2011) 561–595.
[26] H. Li, Q. Zhang, J. Deng, Biased multiobjective optimization and decomposition algorithm, IEEE Transactions on Cybernetics 47 (1) (2016) 52–66.
[27] J. Li, X. Tao, B. Jia, Y. Han, C. Liu, P. Duan, Z. Zheng, H. Sang, Efficient multi-objective algorithm for the lot-streaming hybrid flowshop with variable sub-lots, Swarm and Evolutionary Computation 52 (2020) 100600.
[28] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Transactions on Evolutionary Computation 19 (5) (2014) 694–716.
[29] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artificial Intelligence 228 (2015) 45–65.
[30] M. Li, S. Yang, X. Liu, Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2015) 645–665.
[31] W. Li, X. Meng, Y. Huang, Z. Fu, Multipopulation cooperative particle swarm optimization with a mixed mutation strategy, Information Sciences 529 (2020) 179–196.
[32] Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, IEEE Transactions on Cybernetics 47 (9) (2017) 2689–2702.
[33] Y. Liu, N. Zhu, M. Li, Solving many-objective optimization problems by a pareto-based evolutionary algorithm with preprocessing and a penalty mechanism, IEEE Transactions on Cybernetics (2020) 1–10.
[34] E.L. Piedrahita, Hierarchical clustering for anomalous traffic conditions detection in power substations, Cienciae Ingeniería Neogranadina 30 (1) (2020).
[35] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evolutionary Computation 22 (2) (2014) 231–264.
[36] H. Sato, H. Aguirre, K. Tanaka, Self-controlling dominance area of solutions in evolutionary many-objective optimization, in: Asia-Pacific Conference on Simulated Evolution and Learning, Springer, 2010, pp. 455–465..
[37] Y. Tian, R. Cheng, X. Zhang, Y. Jin, Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum], IEEE Computational Intelligence Magazine 12 (4) (2017) 73–87.

[38] Y. Tian, R. Cheng, X. Zhang, Y. Su, Y. Jin, A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization, IEEE Transactions on Evolutionary Computation 23 (2) (2018) 331–345.
[39] Y. Tian, X. Xiang, X. Zhang, R. Cheng, Y. Jin, Sampling reference points on the pareto fronts of benchmark multi-objective optimization problems, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–6..
[40] H. Wang, W. Wang, L. Cui, H. Sun, J. Zhao, Y. Wang, Y. Xue, A hybrid multi-objective firefly algorithm for big data optimization, Applied Soft Computing 69 (2018) 806–815.
[41] Y. Wang, S. Cai, J. Chen, M. Yin, Sccwalk: An efficient local search algorithm and its improvements for maximum weight clique problem, Artificial Intelligence 280 (2020) 103230.
[42] Y. Wang, L. Wu, X. Yuan, Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure, Soft Computing 14 (3) (2010) 193.
[43] Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, IEEE Transactions on Evolutionary Computation 21 (1) (2016) 131–152.
[44] M. Zhang, L. Wang, W. Guo, W. Li, D. Li, B. Hu, Q. Wu, Many-objective evolutionary algorithm based on relative non-dominance matrix, Information Sciences 547 (2021) 963–983.
[45] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
[46] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (6) (2014) 761–776.
[47] C. Zhu, L. Xu, E. Goodman, Generalization of pareto-optimality for many-objective evolutionary optimization, IEEE Transactions on Evolutionary Computation 20 (2) (2015) 299–315.
[48] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2) (2000) 173–195.
[49] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.
[50] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38 (5) (2008) 1402–1412.