Applied Intelligence (2023) 53:7423–7438
https://doi.org/10.1007/s10489-022-03545-w

A reference vector adaptive strategy for balancing diversity and convergence in many-objective evolutionary algorithms

Lin Zhang[1] · Liping Wang[1] · Xiaotian Pan[1] · Qicang Qiu[2]

Accepted: 21 March 2022 / Published online: 20 July 2022

Abstract

Decomposition-based multiobjective evolutionary algorithms (MOEA/D) have achieved great success in the field of evolutionary multiobjective optimization, and their outstanding performance in solving for the Pareto-optimal solution set has attracted attention. This type of algorithm uses reference vectors to decompose the multiobjective problem into multiple single-objective problems and searches them collaboratively, hence the choice of reference vectors is particularly important. However, predefined reference vectors may not be suitable for dealing with many-objective optimization problems with complex Pareto fronts (PFs), which can affect the performance of MOEA/D. To solve this problem, we introduce a reference vector initialization strategy, namely, scaling of the reference vectors (SRV), and also propose a new reference vector adaptation strategy, that is, transformation of the solution positions (TSP) based on the ideal point solution, to deal with irregular PFs. The TSP strategy can adaptively redistribute the reference vectors through periodic adjustment to endow that the solution set with better convergence and a better distribution. Both strategies are introduced into a representative MOEA/D, called θ-DEA-TSP, which is compared with five state-of-the-art algorithms to verify the effectiveness of the proposed TSP strategy.

Keywords Many-objective optimization problems · Decomposition · Complex Pareto fronts · Reference vector adaptation

1 Introduction

Multiobjective evolutionary algorithms (MOEAs) have received extensive attention in recent decades. Due to their superior performance in both research and practical problem optimization, they have been widely used in the fields of engineering [1–4] and scientific research [5–8]. In these fields, an increasing number of multiobjective optimization problems (MOPs) are emerging in which it is necessary to optimize three or more conflicting objectives simultaneously. Therefore, studying such many-objective optimization problems (MaOPs) has important practical significance.

Without loss of generality, the minimum MOP can be defined as:

$$\begin{aligned} &minimize\ F(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T \\ &Subject\ to\ x \in \Omega \end{aligned} \tag{1}$$

where x=$(x_1, x_2, \cdots, x_m)^T$ is the decision vector, $f_i(x)$ represents the i-th objective function in the objective space, m denotes the number of objective functions, and Ω is the feasible region in the decision space. An MOP can also be called an MaOP if m>3 [9–11]. So-called PFs are generated by these MOEAs to help decision-makers choose a solution that strikes a desirable balance among the various conflicting objectives.

In recent years, the emergence of many excellent algo rithms, such as the adaptive-reference-point-based nondominated sorting genetic algorithm (ANSGA-III) [12], MOEA/D [13], and the reference-vector-guided evolutionary algorithm (RVEA) [14], has enabled many MaOPs to be solved. In paticular MOEA/D, which was first proposed by Zhang,

✉ Liping Wang
690589058@qq.com

✉ Qicang Qiu
qiuqc@zhejianglab.com

Lin Zhang
916289813@qq.com

Xiaotian Pan
332827678@qq.com

[1] College of Computer Science and Technology, College of software, Zhejiang University of Technology, Hangzhou 310023, China

[2] Zhejiang Lab, Hangzhou 310023, China

has been a trending research topic in recent decades. Without prior knowledge, this algorithm generates a series of uniformly distributed reference vectors in the first quadrant. The algorithm then uses this set of predefined reference vectors to decompose the problem into several subproblems. Reference vectors in different directions are used to search different areas of the objective space. As described by Liu [15], the predefined reference vectors are constructed from a set of reference points on the hyperplane $\sum_{i=1}^{m} f_i = 1$ in the normalized m-dimensional objective space, called the normal boundary intersection (NBI) [16]. The basic assumption of MOEA/D is that a uniform distribution of the reference vectors can ensure the diversity of the solution set, as shown in Fig. 1(a).

However, such predefined reference vectors do not allow satisfactory handling of MaOPs with irregular PFs. Because the generated reference vectors are not evenly distributed along the true PF, the performance of MOEA/D can be greatly affected. Under the assumptions that the PF is symmetric and the m objectives have been normalized to $0<f_i<1$, the curvature of the PF can be expressed as $f_1^p + f_2^p + \cdots + f_n^p = 1$, where $p>0$. When $p>1$, the PF is concave, and most of the solutions obtained are sparse in the central area and dense at the corners. When $p<1$, the PF is convex, and most of the solutions obtained are dense in the center area and sparse at the corners. For the concave and convex cases, the solutions specified by ten uniformly distributed reference vectors on the PF of a simple two-objective problem are shown in Fig. 1(b) and (c) respectively.

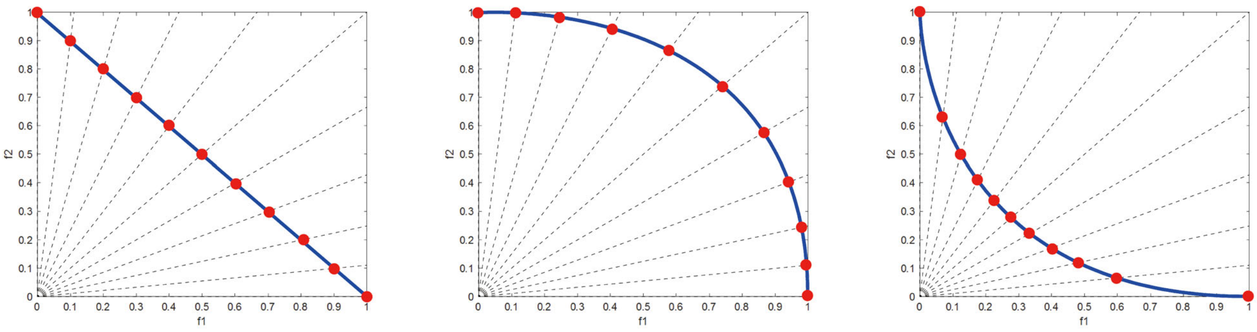
According to Pan [17], irregular PFs also include discontinuous and degenerate Pareto surfaces. A discontinuous PF is defined as a PF whose geometry is incomplete and the PF of a degenerate MOP has fewer than m-1 dimensionals.

Therefore, to solve such problems, many scholars have set out to study how to ensure that the reference vectors in MOEA/D are evenly distributed on the real PF.

The improved MOEA/D (MOEA/D-TPN) proposed by Jiang [18] adopts a two-phase strategy. In the first phase, the unevenness of the PF of the MOP is judged, and then the reference vectors are adjusted in accordance with the results of the first stage and a corresponding decision on how to allocate computational resources, however, this algorithm introduces too many parameters, making it heavily dependent on the values of these parameters.

Dong [19] proposed an adaptive reference vector adjustment strategy based on chain segmentation, in which a chain topology relationship is established for the subproblems, the step length between adjacent individuals is calculated to divide the chain equally, and then, the positions of the reference vectors are determined through a distance strategy and the calculation of the ratios between two adjacent individuals. Finally, vector addition is used to obtain a more uniformly distributed reference vector set. However, the structure assumed for the topological relationship is suitable only for two-dimensional and three-dimensional MOPs.

Qi [20] proposed an improved MOEA/D with adpative weight adjustment (MOEA/D-AWA), in which the reference points are dynamically adjusted during the optimization process. The entire optimization process is also divided into two stages to balance convergence and diversity. Specifically, a sparsity level calculation formula is introduced to calculate the sparsity of each subproblem; then, overcrowded subproblems are periodically deleted in accordance with a preset value, and new reference vectors are generated in relatively sparse positions. However, the external archive mechanism and the complicated sparsity level calculation seriously affect the efficiency of the algorithm.



(a) Solutions specified by ten uniformly distributed reference vectors on a regular PF

(b) Solutions specified by ten uniformly distributed reference vectors on a concave PF

(c) Solutions specified by ten uniformly distributed reference vectors on a convex PF

Fig. 1 Ten uniformly distributed reference vectors on a regular PF, a concave PF, and a convex PF. (a) Ten uniformly distributed reference vectors on a regular PF. (b) Ten uniformly distributed reference vectors on a concave PF. (c) Ten uniformly distributed reference vectors on a convex PF.

On the other hand, some scholars have combined machine learning methods to generate new reference vectors. Gu [21] proposed a reference point design method based on a self-organizing map (SOM). This method uses the weights of the neurons as the reference vectors and uses the objective vector of the nearest population to periodically train an SOM network of N neurons. Wu [22] proposed a learning decomposition method to adaptively determine the reference vectors. This method is divided into a learning module and an optimization module. Specifically, in the learning module, the current nondominated solutions are used as training data, and Gaussian process (GP) regression is used to train an analysis model. In the optimization module, feature vectors are extracted to adjust the reference vectors. However, the learning process is time consuming and can easily lead to overfitting.

For MaOPs with convex, concave, degenerate, or discontinuous PFs, many algorithms based on reference vectors may not be able to solve them well or may require considerable computational resources to solve them [23]. To solve such problems more effectively, we propose our TSP strategy, which can be combined with other decomposition-based MOEAs.

The main contributions of this paper can be summarized as follows.

1. The proposed TSP strategy can be easily integrated into other algorithms to improve the convergence of the population.
2. θ-DEA-TSP is proposed, which incorporates the TSP strategy and considers the diversity and convergence of the population throughout the whole execution of the algorithm, instead of focusing on the diversity of the population at the end of the algorithm, to avoid the algorithm falling into a local optimum in the early stage. At the same time, the algorithm has lower computational complexity.
3. This research is expected to provide better solutions for MaOPs facing similar problems.

The rest of this paper is organized as follows. In Section 2, we introduce the background of our work, including the SRV strategy and the previously proposed strategy for the transformation of the solution locations (TSL), as well as the motivation for our work. In Section 3, we describe in detail the framework of the proposed TSP strategy and θ-DEA-TSP. In Section 4, our experimental results illustrate the effectiveness of the proposed strategy and the comprehensive performance of θ-DEA-TSP on MaOPs with complex PFs. Finally, in Section 5, this paper is summarized, and future research topics are proposed.

2 Related work

To solve MaOPs with complex PFs, Liang [24] proposed the SRV and TSL strategies for modifying the reference vectors.

Specifically, the SRV strategy scales the reference vectors generated via the NBI method on the basis of a predesignated center vector, to ensure that the reference vectors can be evenly distributed on PFs that are concave or convex in nature.

In the NBI method, the number of reference vectors N is counted as follows:

$$N = \binom{H+m-1}{m-1} \tag{2}$$

where m is the number of objectives, and H represents the the number of divisions on each axis. The N reference points are evenly distributed on the entire normalized hyperplane with m objectives.

Therefore, the reference points are mostly generated near the boundary, and only a few points are generated on the inner part of the hyperplane, where $H \geq m$. In the SRV stragety, divides the boundary and inner layers are divided using values of as H_1 and H_2, respectively, to generate two-layer reference points. The N reference vectors are then generated as follows:

$$N = \binom{H_1+m-1}{m-1} + \binom{H_2+m-1}{m-1} \tag{3}$$

However, when the intent is to solve an MaOP with a concave or convex PF, the reference vectors thus generated still will not be evenly distributed on the PF, especially in the case of MaOPs. Therefore, a scaling function (SF) is applied to the original reference vectors, as expressed below:

$$W_i = \frac{W_i \circ r_i + V_c \circ (1 - r_i)}{||W_i \circ r_i + V_c \circ (1 - r_i)||} \tag{4}$$

$$r_i = p_i^k \tag{5}$$

$$p_i = \frac{max|W_i - V_c|}{max_{i=1}^{N}\{max\{|W_i - V_c|\}\}} \tag{6}$$

Here, $i = 1, 2, \cdots, N$, W_i represents the i-th reference vector. V_c represents the designated center vector. The $\circ$ operator denotes the Hadamard product and $||\cdot||$ denotes the calculation of the norm. r_i is the scale of the i-th reference vector, and $p_i \in$[0,1] represents the similarity between the i-th reference vector and the center vector.

To solve MaOPs with degenerate and discontinuous PFs, Liang [24] also proposed the TSL strategy. Specifically, this strategy consists of comparing the Euclidean distances between pair of solutions, filtering out less diverse solutions based on their the Euclidean distances, removeing the overcrowded solutions and transforming the remaining highly diverse solutions into a new set of reference vectors to deal with the degenerate or discontinuous PF. This strategy solves the problem that the reference vectors cannot be evenly distributed on the PF in the case of complex MaOPs. However, we have found that the TSL strategy considers only the diversity of the solutions when deleting overcrowded solutions. It ignores the convergence of the solutions, leading to population degradation. In the proposed algorithm θ-DEA*, the TSL strategy is applied at the end of the algorithm to increase the diversity of the solution set. However, this can cause the algorithm to fall into a local optimum early.

In this article, to solve the problem that the TSL strategy results in degeneration of the population in the face of complex MaOPs, we propose the TSP strategy as an alternative. In the TSP strategy, when a new set of reference vectors is being constructed two solutions to be deleted are compared with the current ideal point by calculating the Euclidean distance, and the better solution is retained to prevent population degradation. In addition, the TSP strategy is applied to adjust the reference vectors throughout the whole execution of θ-DEA-TSP, thereby guaranteeing good diversity of the population while improving convergence.

3 Methodologies

In this section, we describe the TSP strategy. Specifically, to better deal with MaOPs with degraded or discontinuous PFs, the positions of solutions are converted into new reference vectors to prevent the degradation of the population while maintaining its diversity. This strategy can be integrated along with the SRV strategy into popular MOEAs based on reference vectors to enhance the evolutionary search process.

3.1 Transformation of solution positions based on the ideal point solution

Oriented toward MaOPs with degenerate or discontinuous PFs, we propose the TSP strategy to fit an irregular PF by transforming the position of a solution with good distribution and convergence characteristics to obtain a new reference vector. Specifically, in the TSP strategy, the distances between pairwise solutions are initially calculated. If such a distance is less than a preset threshold, then the corresponding solution with poorer diversity is selected to calculate the Euclidean distance between this solution and the ideal point. Solutions with larger Euclidean distances, that is, poorer convergence are deleted, and the remaining solutions are converted into reference vectors.

The TSP process is shown in Algorithm 1. First, the algorithm initializes a population archive PA as the entire population, in which each individual is normalized. Second, the Euclidean distances $Dist$ of the pairwise solutions in PA are calculated. The Euclidean distance is expressed as follows:

$$\begin{aligned} D(X, Y) &= \sqrt{\sum_{m}^{i=1}(x_i - y_i)^2} \\ Subject\ to\ X&, Y \in \omega \end{aligned} \tag{7}$$

where X=$(x_1, x_2, \cdots, x_m)^T$ and Y=$(y_1, y_2, \cdots, y_m)^T$ are the decision vectors, and ω is the feasible region in the decision space. Lines 3-10 of the algorithm filter out the two solutions corresponding to the minimum Euclidean distance based on $Dist$, calculate the Euclidean distances between these two solutions and the current ideal point, and delete the solution corresponding to the larger distance. This process is repeated until $Dist_{min}$ is greater than the threshold ϵ. The aim of Lines 11-15 is to find the solution among the remaining solutions that is nearest to be better solution of each solution pair processed as described above and generate a new solution S at the midpoint of the two. The new solution S is stored in PA, $Dist$ is updated, and the process is repeated until the number of solutions in PA is N. In Lines 17-19, the solutions in PA are converted into new reference vectors to guide the generation of new offspring.

To illustrate the process more clearly, we present an example in Fig. 2. In Fig. 2(a), the algorithm finds four points corresponding to two Euclidean distances smaller than our preset threshold ϵ, namely, points B and C and points D and E. Second, the Euclidean distances between B, C, D and E, and the current ideal point Z^* are calculated, as shown in Fig. 2(b). Next, the algorithm deletes the individuals farther from the ideal point between B and C and between D and E, that is, B and D, as shown in Fig. 2(c). Finally, the individuals closest to B and D other than C and E are found, namely, A and F. New individuals B' and D' are generated at the midpoints between A and C and between E and F, as shown in Fig. 2(d).

3.2 The framework of θ-DEA-TSP

The SRV and TSP strategies are two independent strategies that can be easily integrated into MOEAs. The algorithm framework is shown in Algorithm 2. First, the population Pop is initialized and reference vectors W are generated using the NBI method. In Lines 2-3, the SRV strategy mentioned in Section 2 is used to scale the reference vectors, and then, the objective normalization flag is set. Lines 4-14 are the main part of the algorithm. The purpose of line 5 is to generate N offspring. Then, based on the Boolean value of the objective normalization flag, it is judged whether it is necessary to deal with the problem of different ranges for each objective. Subsequently, the SRV strategy is used to adjust the generated reference vectors. We introduce a frequency parameter f_r to control the frequency of application of the TSP strategy so that it will be applied throughout the entire execution of the algorithm instead of taking effect only at the end of the algorithm. Finally, N individuals are obtained through environmental selection.

Algorithm 1 Transformation of Solution Positions based on the ideal point solution

```
Input: $\epsilon$: Euclidean distance threshold; $z*$: the ideal point; $PA$: population archive;
Output: $new\_W$: new reference vectors
Initiate population archive $PA$ by copying the population
/*Calculate the distance value Dist between every two solutions in $PA$*/
for all $x_i$ in $PA$ do
    for all $x_j$ in $PA$ do
        $W_{i,j}$=D($x_i$, $x_j$) //using (7)
        if i==j then
            $W_{i,j}$=inf
        end if
    end for
end for
while $Dist_{min}$<$\epsilon$ do
    /*Find the two solutions $A$ and $B$ corresponding to $Dist_{min}$*/
    ($A$, $B$)=minW($\omega$)
    if D(A,z*)>D(B,z*) then// compare the distance between $A$ and $z*$ with the distance between $B$ and $z*$
        remove $A$
    else
        remove $B$
    end if
end while
while PA<N do
    /*Find the two solutions $A'$ and $B'$ corresponding to $Dist_{max}$*/
    Create a new solution $S$ at the midpoint of $A'$ and $B'$
    $PA$=$PA \cup S$
    Update $Dist$
end while
for all $x_i$ in $PA$ do
    new_$W_i$=$\frac{f(x_i)}{||f(x_i)||}$
end for
return $new\_W$
```

3.3 Computational complexity

Since it is still necessary to calculate the Euclidean distances between pairs of solutions, the computational complexity of the TSP strategy is the same as that of the original TSL strategy, which is O(N^2). However, the overall algorithm frameworks differ between the TSL and TSP strategies. In the overall framework of θ-DEA*, the TSL strategy takes effect every constant generation at the end of the algorithm, and the algorithm complexity is O($N^2 * \frac{T_{max} - 0.75 * T_{max}}{C}$), where T_{max} is the maximum number of function evaluations (FEs), and C is the preset constant of the algorithm. In contrast, this article introduces the frequency parameter f_r, and the complexity of θ-DEA-TSP is O($N_2 * \frac{T_{max}}{f_r * T_{max}}$). In the case of many FEs, the TSL strategy will take effect frequently. Therefore, in this case, the complexity of θ-DEA-TSP will be better than that of θ-DEA*.

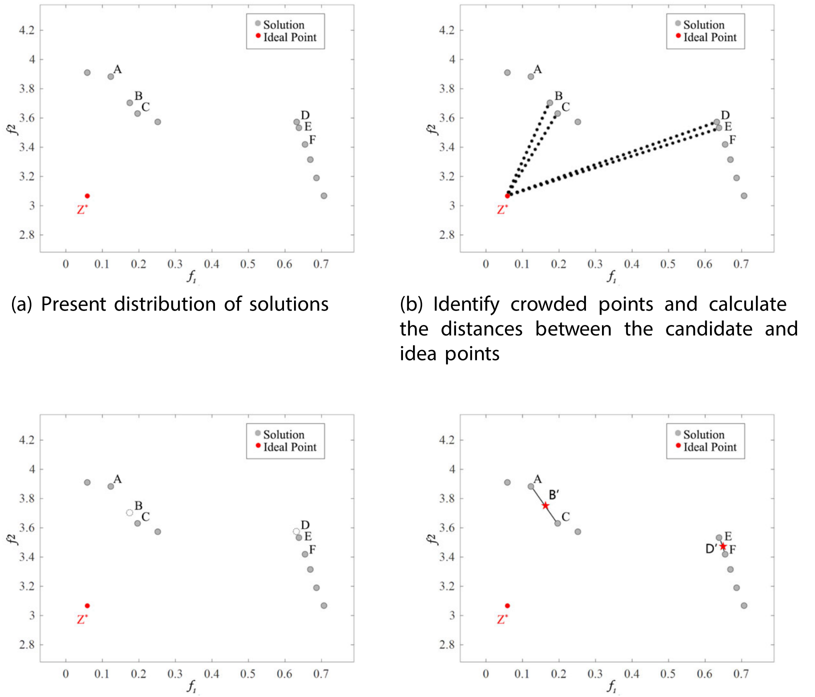
Algorithm 2 Framework of θ-DEA-TSP

```
Input: $t$: generation index; $tmax$: maximal number of generations; $f_r$: frequency of employment of the TSP strategy; $Pop$: population
Output: $final\_Pop$: final population
Create a population $Pop$ and initialize reference vectors $W$ using (2)
$new\_W$=SRV(W) in (3-6)
set $Objective\_normalization\_flag$=True
while the termination criteria are not met do
    $Q$=Offspring_creation($Pop$)
    if $Objective\_normalization\_flag$=True then
        Normalize $Q$
    end if
    if ($\frac{t}{t\_max}$ mod $f_r$)=0 then
        TSP($new\_W$) Algorithm 1
        Set $Objective\_normalization\_flag$=false
    end if
    $final\_Pop$=Environment_Selection($Pop$)
    $t$=$t$+1
end while
Return $final\_Pop$
```

4 Experiment study

In this section, we combine the proposed strategy with θ-DEA to obtain the resulting θ-DEA-TSP algorithm for comparison with several typical algorithms: θ-DEA* [24], ANSGA-III [25], RVEA [14], and the guided nondominated sorting genetic algorithm (gNSGA-II) [26]. θ-DEA* is the algorithm proposed by Liang, which integrates the SRV and TSL strategies into θ-DEA. θ-DEA* is an algorithm based on decomposition. It incorporates the TSL strategy and selects a set of excellent solutions to guide the reference vectors for self-adaptation, enabling some

Fig. 2 An example illustrating the whole process of screening, deleting, and generating solutions in accordance with the TSP strategy



(a) Present distribution of solutions

(b) Identify crowded points and calculate the distances between the candidate and idea points

(c) Delete the solutions with poor diversity and convergence

(d) Create new solutions

improvements in solving MaOPs. ANSGA-III is adaptive in the dynamic deletion and inclusion of new reference points. It is an algorithm proposed by Deb based on the non-dominated sorting genetic algorithm version II (NSGA-II) framework for solving MaOPs, providing denser PFs under the same computational workload and demonstrating excellent performance in dealing with constrained and unconstrained MaOPs. RVEA is also an optimization algorithm based on decomposition. It applies an adaptive strategy for dynamic adjustment of the reference vector distribution in accordance with the objective function, and reference vector regeneration strategies are used to deal with irregular PFs. Many papers have proven its effectiveness in solving MaOPs. gNSGA-II is an evolutionary algorithm based on dominance that introduces the concept of g-dominance. Through comparisons with this algorithm, the performance of the g-dominance method for solving MaOPs has been proven. Since DTLZ1 is an MOP with a regular PF, we test the above mentioned 5 algorithms on DTLZ2-7 [27]. Each test problem is run independently 20 times, and the average and standard deviation are taken. The above tests are all carried out on the PlatEMO [28] platform. To ensure fairness, the parameters of the algorithms mentioned above are all set to their default values in PlatEMO.

In the following sections, we first give the parameter settings for the simulation experiments. Second, we briefly introduce the evaluation metrics used in the comparative study. Then, we analyze the influence of the value of the parameter f_r on the performance of our algorithm. Finally, we compare the experimental results obtained by independently running each algorithm 20 times on each test problem. In these comparisons, the symbol "-" indicates that θ-DEA-TSP is significantly better than the other algorithm considered for comparison, the symbol "+" indicates that θ-DEA-TSP is significantly inferior to the other compared algorithm, and the symbol "$\approx$" indicates that the result obtained is not significantly different from that of the other algorithm.

4.1 Parameter setting

The parameters of all algorithms considered for comparison are set as follows: In θ-DEA-TSP, θ-DEA*, ANSGA-III, RVEA, and gNSGA-II, the simulated binary crossover (SBX) operator and polynomial mutation are used to solve the test problems, and the detailed settings are shown in Table 1. To ensure fairness, the default parameters values

Table 1 Parameter settings for crossover and mutation

Parameters	Values
crossover probability (P_c)	1.0
mutation Probability (P_m)	1/N
cross distribution index (η_c)	20
variation distribution index (η_m)	20

are used for binary crossover and polynomial mutation. The relevant parameter settings of the test function are summarized in Table 2, where M is the number of objectives; D is the number of decision variables, and k is a parameter used to artificially change the length of a decision variable.

4.2 Performance metrics

In the experiments, we measure the performance of various algorithms on the test problems. We use the following three metrics to evaluate the convergence and comprehensive performance of the solution set.

(1) Generation Distance (GD) [34]

The GD metric is used to calculate the distance between the obtained solution set and the true Pareto solution set, which is defined as follows:

$$GD = \frac{\sum_{i=1}^{n}(d_i^p)^{\frac{1}{p}}}{n} \tag{8}$$

where n denotes the number of obtained solutions, d_i is the Euclidean distance between the obtained solution i and its nearest true solution, and p denotes the number of objectives. The smaller the GD value is, the stronger the algorithm performance.

(2) Inverse Generation Distance (IGD) [35]

The IGD indicator is calculated as the average value of the distance from each reference point to the nearest solution, and can be used to simultaneously evaluate the convergence and distribution of the solution set calculated by an algorithm. The indicator is defined as follows:

$$IGD(P, P^*) = \frac{\sum_{x^* \in P^*} d(x^*, P)}{|P^*|} \tag{9}$$

where P represents the solution set obtained by the algorithm, P^* denotes the solution set of the ideal PF, and $d(x^*, P)$ denotes the minimum Euclidean distance from x^* to P. The smaller the IGD value is, the stronger the algorithm performance.

(3) Hypervolume Distance (HV) [36]

HV is another way to measure the comprehensive performance. It calculates the volume of the objective space enclosed by the reference points and the i-th non-dominated solution. The larger the HV value, the better the comprehensive performance. The metric can be defined as follows:

$$HV(S) = \text{volume}(\bigcup_{i=1}^{|S|} a^i) \tag{10}$$

where a^i is the hypercube formed by the solution x and the origin of the coordinate axes as diagonal lines. The space covered by a^i is $volume(a^i)$.

Notably, the widely used evaluation metric HV is not selected for the DTLZ problems because HV and IGD show certain contradictions on concave PFs [37]. For MaOPs whose PFs are nonconvex, HV has a certain error [36]. The PF of DTLZ2-6 is concave, as shown in Table 2. GD is an indicator of convergence and, can still measure the convergence of an algorithm even if the diversity of the solution set is not good. To prove the effectiveness of the convergence enhancement strategy, we choose the GD and IGD metrics to evaluate the results on the DTLZ problems.

4.3 Sensitive analysis of parameter f_r

f_r is an important parameter in the reference vector adaptation strategy. Its value will affect the frequency of reference vector adaptation, and thus, in turn, affect the convergence and distribution of the solution set found by the algorithm. To study the impact of this parameter on the algorithm performance, with f_r=0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1, we tested the algorithm on DTLZ2-7 with 6, 8, 10, and 15 objectives, and all parameters except f_r remaining unchanged. Fig. 3 shows the comprehensive performance on the test problems when f_r takes each value. With taking different values of f_r, the IGD ranking of θ-DEA-TSP fluctuates greatly on the DTLZ2-7 test problems. However, it can be seen from Fig. 3 that when f_r=0.1 and M=8, θ-DEA-TSP has the best IGD ranking on the DTLZ2-7 test

Table 2 Parameter settings for the test problems

Problem	Feature	M	D	k	Function evaluations(FEs)
DTLZ2	Concave [29]	6, 8 ,10 ,15	M+k-1 [30]	10	15000*M
DTLZ3	Concave, Multi-modal [29]				
DTLZ4	Concave, Biased [29]				
DTLZ5	Concave, Degenerate [31]				
DTLZ6	Concave, Degenerate [32]				
DTLZ7	Mixed, Disconnected, Multi-modal, Scaled [29]			20	
IDTLZ1	Non-concave [12]			5	
IDTLZ2	Non-concave [12]			10	
IDTLZ5IM	Non-concave [33]				

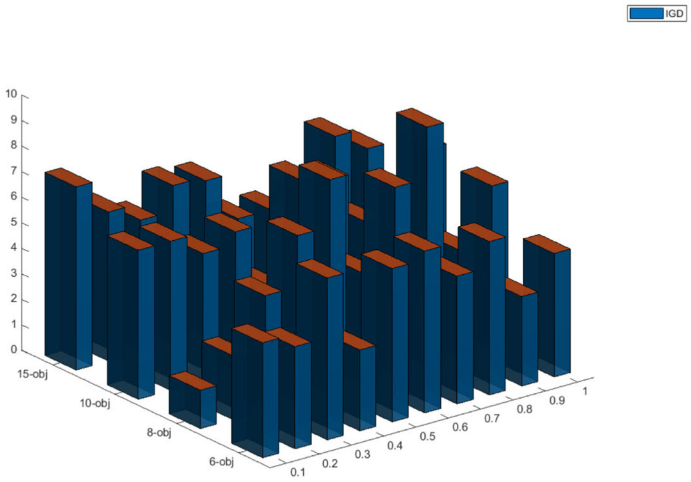


Fig. 3 The IGD performance of θ-DEA-TSP on the 6-, 8-, 10-, and 15-objective versions of DTLZ2-7 when f_r is 0.1-1

problems. Therefore, for the comparative experiment, we choose f_r=0.1 as the value of the frequency parameter of θ-DEA-TSP.

4.4 Comparison experience

4.4.1 Test problems

In this section, θ-DEA-TSP is compared with θ-DEA*, ANSGA-III, RVEA, and gNSGA-II on DTLZ2-7 [38], IDTLZ1-2 [12] and DTLZ5IM [33]. Here, the numbers of objectives (m values) of DTLZ2-7 are 6, 8, 10, and 15. Tables 3 and 4 show the performance of the algorithms in terms of the GD metric and the IGD metric, respectively, on DTLZ2-7.

Experiments were run 20 times independently on the DTLZ2-7 test problems to compare the convergence performance of θ-DEA-TSP, θ-DEA*, ANSGA-III, RVEA, and gNSGA-II as represented by the GD metric. The mean and variance of the GD values are listed in Table 3. In Table 3, the average GD value of θ-DEA-TSP is better than those of the other algorithms on 50% of the 24 problems tested. In particular, θ-DEA-TSP shows obviously superior performance on DTLZ4-7. Especially for the concave and multipeak DTLZ6 problem [39], the results for the 6-, 8-, 10-, and 15-objective versions of this problem are all better than those of the other four algorithms. On 8-objecitve DTLZ6, the GD value of θ-DEA-TSP is 64% lower than that of the second-ranking algorithm θ-DEA*. On the 10-objective DTLZ6 problem, the GD value of θ-DEA-TSP is 54% lower than that of the second-ranking algorithm θ-DEA*.

The experimental results in terms of the IGD values on DTLZ2-7 are shown in Table 4. We can see that when solving DTLZ4-7, θ-DEA-TSP exhibits excellent performance. Even when the number of objectives is large, the IGD value of this algorithm is still better than those of the other algorithms. It is not difficult to see that θ-DEA-TSP is highly very competitive on such problems. In addition, it can be seen from the experimental results that RVEA shows excellent performance on DTLZ2 and DTLZ3 and on other test problems with regular PFs. However, on DTLZ5-7, which have discontinuous or degraded PFs, the performance of our algorithm is significantly better than the other four, thus proving the effectiveness and pertinence of the TSP strategy proposed in this article. To more intuitively illustrate the performance of the five algorithms on DTLZ2-7 for different numbers of objectives, we plot their ranking histograms in Fig. 4. The x-axis represents the number of objectives, the y-axis represents the algorithm, and the z-axis represents the average ranking. The lower the column height in this figure, the higher the ranking of the corresponding algorithm, and the better its performance. We can see that the performance of the θ-DEA-TSP and RVEA algorithms is relatively stable, while the ranking of θ-DEA* fluctuates greatly for problem with 8 objectives and 15 objectives. The performance of the ANSGA-III worsens as the number of objectives increases. Compared with the other four algorithms, gNSGA-II performs poorly on these problems. The optimization results of the 5 compared algorithms for 15-objective DTLZ5 and 6-objective DTLZ7 are shown in Figs. 5 and 6.

θ-DEA* selects excellent solution sets only in the middle and late stages of the algorithm and generates reference vectors to guide the algorithm. A strategy for screening the solution sets that places too much emphasis on diversity may result in the degradation of the solution set. RVEA optimizes the reference vectors globally, which is not beneficial

Table 3 The average GD values of the θ-DEA-TSP algorithm and other algorithms on the DTLZ2-7 test problems

Problem	M	θ-DEA-TSP Mean(std)	θ-DEA* Mean(std)	RVEA Mean(std)	ANSGA-III Mean(std)	gNSGA-II Mean(std)
DTLZ2	6	**7.6804e-3 (1.87e-4)**	8.3461e-3 - (8.91e-5)	9.0739e-3 - (6.15e-6)	8.2397e-3 - (3.92e-4)	2.0645e-1 - (3.37e-2)
DTLZ2	8	**8.0090e-3 (9.00e-4)**	9.0228e-3 - (6.79e-4)	2.3838e-2 - (1.19e-5)	1.1489e-2 - (1.48e-3)	2.4273e-1 - (1.50e-3)
DTLZ2	10	7.8146e-3 (1.18e-3)	7.4077e-3 - (9.21e-4)	**9.4849e-4 + (1.74e-3)**	1.0894e-2 - (6.68e-3)	2.1609e-1 - (1.01e-2)
DTLZ2	15	3.2867e-2 (6.41e-3)	2.0690e-2 + (5.17e-5)	**1.5044e-2 + (1.80e-2)**	4.0926e-2 - (6.33e-3)	2.2068e-1 - (5.25e-3)
DTLZ3	6	4.6792e-1 (1.09e+0)	**8.1135e-3 + (2.12e-4)**	9.0896e-3 + (9.18e-5)	1.3030e+0 - (2.19e+0)	1.5449e+2 - (8.22e+0)
DTLZ3	8	9.5252e-1 (1.52e+0)	1.6177e-1 + (5.94e-1)	**2.3772e-2 - (1.24e-4)**	7.5663e+0 - (3.16e+0)	1.7710e+2 - (4.75e+0)
DTLZ3	10	6.3082e-1 (1.25e+0)	9.1695e-2 + (3.75e-1)	**1.7666e-3 + (1.68e-3)**	9.0507e+0 - (5.43e+0)	1.7365e+2 - (3.58e+0)
DTLZ3	15	1.9341e+0 (3.26e+0)	1.2888e+0 - (2.99e+0)	**2.5969e-2 + (2.84e-2)**	1.8333e+1 - (8.62e+0)	1.7958e+2 - (2.78e+0)
DTLZ4	6	**7.3612e-3 (2.19e-4)**	8.3169e-3 - (6.69e-5)	8.8159e-3 - (3.35e-4)	7.8012e-3 - (1.23e-3)	1.4819e-2 - (1.79e-3)
DTLZ4	8	**6.5104e-3 (1.28e-3)**	7.8258e-3 - (7.84e-4)	2.3145e-2 - (8.82e-4)	1.2695e-2 - (4.52e-3)	2.8101e-2 - (4.07e-4)
DTLZ4	10	8.5447e-3 (1.39e-3)	7.9675e-3 - (9.96e-4)	**5.9096e-3 + (4.33e-3)**	9.7986e-3 - (5.38e-3)	3.9587e-2 - (1.33e-2)
DTLZ4	15	2.2079e-2 (1.38e-3)	**2.0660e-2 + (4.88e-5)**	3.4487e-2 - (1.05e-2)	2.6342e-2 - (2.90e-3)	4.9793e-2 - (1.70e-2)
DTLZ5	6	**8.0904e-2 (3.31e-2)**	1.2930e-1 - (4.05e-2)	3.0446e-1 - (1.05e-1)	1.0090e-1 - (1.67e-2)	2.2109e-1 - (3.13e-2)
DTLZ5	8	9.9914e-2 (2.51e-2)	1.8842e-1 - (2.38e-2)	**6.5053e-2 + (1.62e-1)**	1.4077e-1 - (1.70e-2)	2.6245e-1 - (3.39e-3)
DTLZ5	10	**1.0208e-1 (1.71e-2)**	1.9744e-1 - (2.46e-2)	2.7461e-1 - (2.95e-1)	1.5860e-1 - (3.74e-2)	2.5777e-1 - (5.06e-3)
DTLZ5	15	1.3419e-1 (4.79e-2)	1.0788e-1 + (5.28e-2)	**5.7012e-4 + (1.23e-3)**	1.6419e-1 - (5.51e-2)	2.7900e-1 - (3.49e-3)
DTLZ6	6	**2.1564e-1 (3.41e-2)**	3.4821e-1 - (6.61e-2)	4.9344e-1 - (5.43e-2)	3.4803e-1 - (4.74e-2)	9.5961e-1 - (1.49e-2)
DTLZ6	8	**2.7804e-1 (6.07e-2)**	4.5814e-1 - (6.46e-2)	8.9401e-1 - (1.15e-1)	5.8214e-1 - (1.01e-1)	9.9223e-1 - (9.76e-3)
DTLZ6	10	**2.8410e-1 (4.34e-2)**	4.3772e-1 - (5.94e-2)	4.8703e-1 - (8.98e-2)	7.5169e-1 - (1.69e-1)	9.9092e-1 - (6.24e-3)
DTLZ6	15	**4.4762e-1 (7.97e-2)**	4.8621e-1 - (1.03e-1)	9.7167e-1 - (1.05e+0)	6.5784e-1 - (5.20e-1)	1.0059e+0 - (7.94e-3)
DTLZ7	6	1.0939e-2 (3.30e-3)	**9.6202e-3 + (1.28e-3)**	3.3385e-2 - (6.21e-3)	2.5260e-2 - (5.51e-3)	3.9332e-1 - (3.82e-1)
DTLZ7	8	**1.9700e-2 (1.63e-3)**	2.5023e-2 - (3.31e-3)	2.9006e-2 - (3.71e-3)	5.3229e-2 - (1.29e-2)	7.9152e-1 - (1.98e-1)
DTLZ7	10	**3.0158e-2 (3.32e-3)**	3.5813e-2 - (3.39e-3)	7.5955e-2 - (4.48e-3)	5.3000e-2 - (1.12e-2)	3.1994e+0 - (3.10e-1)
DTLZ7	15	1.5797e-1 (3.02e-2)	1.4210e-1 + (9.64e-3)	1.7819e-1 - (4.35e-2)	**1.2366e-1 + (2.08e-2)**	5.7681e+0 - (6.91e-1)
+/-/=			8/16/0	8/16/0	1/23/0	0/24/0

For each instance, the best average value among the algorithms is indicated bold

for subregion-specific solution sets. Although ANSGA-III modifies the traditional concept of dominance applied in NSGA-II, it also introduces extreme point calculation and target normalization strategies to enhance the convergence of the algorithm. However, the experimental results are still not very good. When θ-DEA-TSP solves a problem whose PF has degenerate or discontinuous characteristics, it will dynamically screen the excellent solution set throughout the whole process, delete locally inferior solutions, and generate reference vectors from the results to better guide the evolution of the algorithm. Hence, the θ-DEA-TSP algorithm performs better on DTLZ2-7 in terms of both the GD and IGD evaluation metrics.

In Fig. 5, on the one hand, we can see that the convergence of gNSGA-II, ANSGA-III, and θ-DEA* is poor, which may be because they lack convergence improvement methods or place too much emphasis on the diversity of the solution set. On the other hand, RVEA has poor diversity on this problem, which may be because, in high-dimensional situations, its current reference vectors tend to be excessively concentrated in a certain location, and its adaptive reference vector strategy dose not improve this situation. In comparison, θ-DEA-TSP does not fully converge, but because it considers both convergence and diversity, it performs significantly better than the algorithms. In Fig. 6, with an increase in the number of dimensions, the selection pressure of the dominance-based algorithms decreases sharply, preventing them from converging to the optimal PF, and gNSGA-II does not fully converge. In contrast, the convergence and diversity performance of θ-DEA-TSP, θ-DEA*, RVEA and ANSGA-III on 6-objective DTLZ7 are all good, and θ-DEA-TSP is slightly better than the other two.

To further analyze and compare the experimental results, we present boxplots of the experimental results for some test problems, namely, 8-objective DTLZ5, 10-objective

Table 4 The average IGD values of the θ-DEA-TSP algorithm and other algorithms on the DTLZ2-7 test problems

Problem	M	θ-DEA-TSP Mean(std)	θ-DEA* Mean(std)	RVEA Mean(std)	ANSGA-III Mean(std)	gNSGA-II Mean(std)
DTLZ2	6	3.1761e-1 (1.80e-3)	3.1940e-1 = (1.47e-3)	**2.7903e-1 + (2.27e-5)**	3.2338e-1 = (5.46e-2)	1.6406e+0 - (3.38e-1)
DTLZ2	8	4.2307e-1 (2.51e-3)	4.2324e-1 = (2.04e-3)	**3.8701e-1 + (5.85e-5)**	6.0857e-1 - (5.82e-2)	2.3477e+0 - (1.21e-1)
DTLZ2	10	5.3686e-1 (4.57e-3)	5.3706e-1 = (2.67e-3)	**5.0066e-1 + (1.38e-3)**	6.4260e-1 - (9.55e-2)	1.6684e+0 - (4.83e-1)
DTLZ2	15	6.6251e-1 (3.39e-2)	**6.4946e-1 = (2.34e-4)**	7.3878e-1 - (1.21e-1)	9.3486e-1 - (2.47e-2)	1.4094e+0 - (1.23e-1)
DTLZ3	6	3.8954e-1 (3.78e-2)	3.2050e-1 + (3.36e-3)	**2.8022e-1 + (8.79e-4)**	3.7299e-1 = (9.03e-2)	8.3693e+2 - (1.96e+2)
DTLZ3	8	5.1822e-1 (5.39e-2)	4.3925e-1 + (1.95e-2)	**3.8842e-1 + (1.28e-3)**	1.5925e+0 - (1.14e+0)	1.0060e+3 - (2.14e+2)
DTLZ3	10	6.1182e-1 (4.75e-2)	5.7047e-1 + (6.46e-2)	**5.0290e-1 + (3.79e-3)**	1.7893e+0 - (1.34e+0)	7.4378e+2 - (1.81e+2)
DTLZ3	15	9.8920e-1 (6.50e-2)	1.0046e+0 = (7.81e-2)	**7.7462e-1 + (1.38e-1)**	1.1239e+0 - (5.40e-2)	7.2881e+2 - (1.10e+2)
DTLZ4	6	**3.1837e-1 (9.11e-4)**	3.1909e-1 = (9.63e-4)	3.3310e-1 - (7.55e-2)	4.0090e-1 - (1.31e-1)	5.2182e-1 - (9.75e-2)
DTLZ4	8	**4.3170e-1 (2.21e-2)**	4.3564e-1 = (2.59e-2)	4.3358e-1 = (5.69e-2)	5.8987e-1 - (9.60e-2)	6.1214e-1 - (3.67e-3)
DTLZ4	10	**5.4030e-1 (1.01e-2)**	5.4102e-1 = (1.80e-2)	5.4490e-1 = (5.71e-2)	6.5484e-1 - (9.85e-2)	6.4147e-1 - (4.04e-2)
DTLZ4	15	6.5272e-1 (4.37e-3)	**6.4952e-1 + (1.22e-4)**	7.9036e-1 - (3.81e-2)	9.3220e-1 - (3.68e-2)	7.6786e-1 - (3.91e-2)
DTLZ5	6	**1.1595e-1 (3.72e-2)**	1.6814e-1 - (7.34e-2)	3.9731e-1 - (6.32e-2)	1.9949e-1 - (6.95e-2)	1.3561e+0 - (3.73e-1)
DTLZ5	8	**2.1051e-1 (6.16e-2)**	2.9780e-1 - (7.72e-2)	5.8335e-1 - (2.03e-1)	4.9863e-1 - (1.78e-1)	2.0176e+0 - (2.37e-1)
DTLZ5	10	**2.2635e-1 (5.06e-2)**	2.9674e-1 - (7.93e-2)	4.4081e-1 - (1.74e-1)	4.6443e-1 - (2.17e-1)	1.8984e+0 - (2.06e-1)
DTLZ5	15	**2.2745e-1 (8.24e-2)**	2.3108e-1 = (8.01e-2)	6.5999e-1 - (1.54e-1)	2.6038e-1 = (1.31e-1)	2.4692e+0 - (1.16e-1)
DTLZ6	6	**1.9600e-1 (6.69e-2)**	3.4520e-1 - (7.92e-2)	2.7970e-1 - (6.12e-2)	4.9770e-1 - (2.37e-1)	8.3732e+0 - (5.36e-1)
DTLZ6	8	**2.6892e-1 (8.04e-2)**	3.9209e-1 - (7.16e-2)	2.7590e-1 = (8.88e-2)	2.7197e+0 - (1.01e+0)	8.5943e+0 - (4.63e-1)
DTLZ6	10	3.1422e-1 (1.14e-1)	4.0310e-1 - (8.38e-2)	**2.0154e-1 + (5.22e-2)**	4.4331e+0 - (1.67e+0)	8.6551e+0 - (3.37e-1)
DTLZ6	15	**3.9969e-1 (1.08e-1)**	4.1291e-1 = (1.16e-1)	4.4277e-1 - (2.18e-1)	1.9507e+0 - (1.72e+0)	8.8434e+0 - (5.11e-1)
DTLZ7	6	**5.1649e-1 (1.23e-1)**	5.3737e-1 - (8.66e-2)	9.8615e-1 - (1.81e-1)	5.3433e-1 - (6.27e-2)	1.3045e+0 - (9.17e-1)
DTLZ7	8	**9.7826e-1 (2.28e-1)**	1.3852e+0 - (3.09e-1)	1.7811e+0 - (2.27e-1)	1.2882e+0 - (2.36e-1)	1.3591e+0 - (1.48e-1)
DTLZ7	10	**1.5927e+0 (4.10e-1)**	2.0604e+0 - (3.51e-1)	3.4305e+0 - (4.59e-1)	1.8892e+0 - (3.64e-1)	1.3013e+1 - (3.04e+0)
DTLZ7	15	8.9100e+0 (1.58e+0)	1.0059e+1 - (7.00e-1)	**6.3893e+0 + (1.44e+0)**	1.0516e+1 - (6.44e-1)	2.7328e+1 - (1.13e+1)
+/-/=			4/10/10	9/12/3	0/21/3	0/24/0

For each instance, the best average value among the algorithms is indicated in bold

DTLZ5, 8-objective DTLZ6, and 10-objective DTLZ7 in Fig. 7. The average IGD values of our algorithm on 8-objective DTLZ5 and 10-objective DTLZ5 are significantly better than those of the other four algorithms. It is not difficult to see in Fig. 6(a) and (b) that the stability of θ-DEA-TSP is also significantly better than that of the other four algorithms. It can be seen from the boxplot for 8-objectives DTLZ6 that the IGD values of θ-DEA-TSP, θ-DEA*, and RVEA are relatively close and the fluctuations of the IGD values of these three algorithms are relatively small. The boxplot for 10-objective DTLZ7 shows that, except for the outliers, the IGD values of θ-DEA-TSP are relatively close to those of ANSGA-III. Among the compared algorithms, the fluctuation in the performance of gNSGA-II on this test problem is particularly obvious. In addition, RVEA yields too many outliers, although the fluctuation in the IGD values is similar to that of the other three algorithms.

The experimental results for the IGD values on DTLZ5IM and IDTLZ1-2 for all dimensionalities are shown in Table 5. For these test problems in all numbers of dimensions, 58.33% of the experimental results of θ-DEA-TSP are better than those of the other algorithms considered for comparison. Especially for DTLZ5IM and 6-objective IDTLZ2, θ-DEA-TSP shows excellent performance. A feature of IDTLZ1-2 is that many reference points created on the standardized hyperplane do not have a Pareto optimal advantage. θ-DEA-TSP, θ-DEA* and ANSGA-III all benefit from their adaptive strategies for adaptively screening reference points and reallocating them to obtain Pareto-optimal solutions. Therefore, they perform well on these two test problems. However, gNSGA-II and RVEA lack strategies for local adaptation, leading to unsatisfactory results.

The experimental results for the HV values on DTLZ5IM and IDTLZ1-2 for each dimensionality are shown in Table 6.

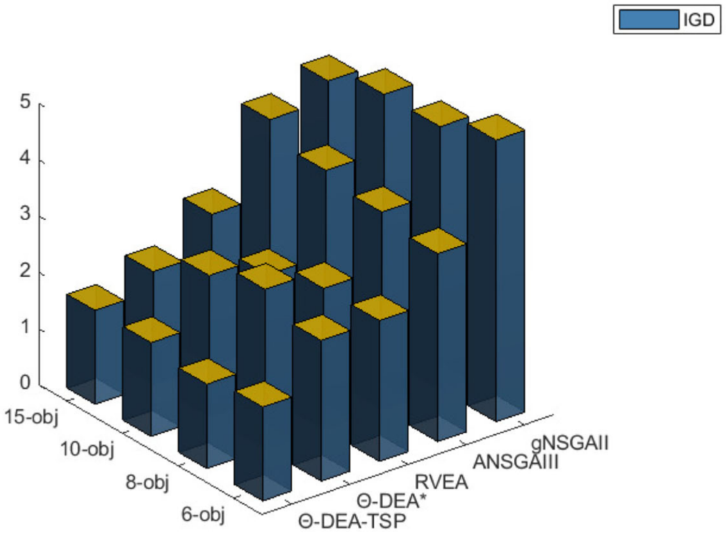
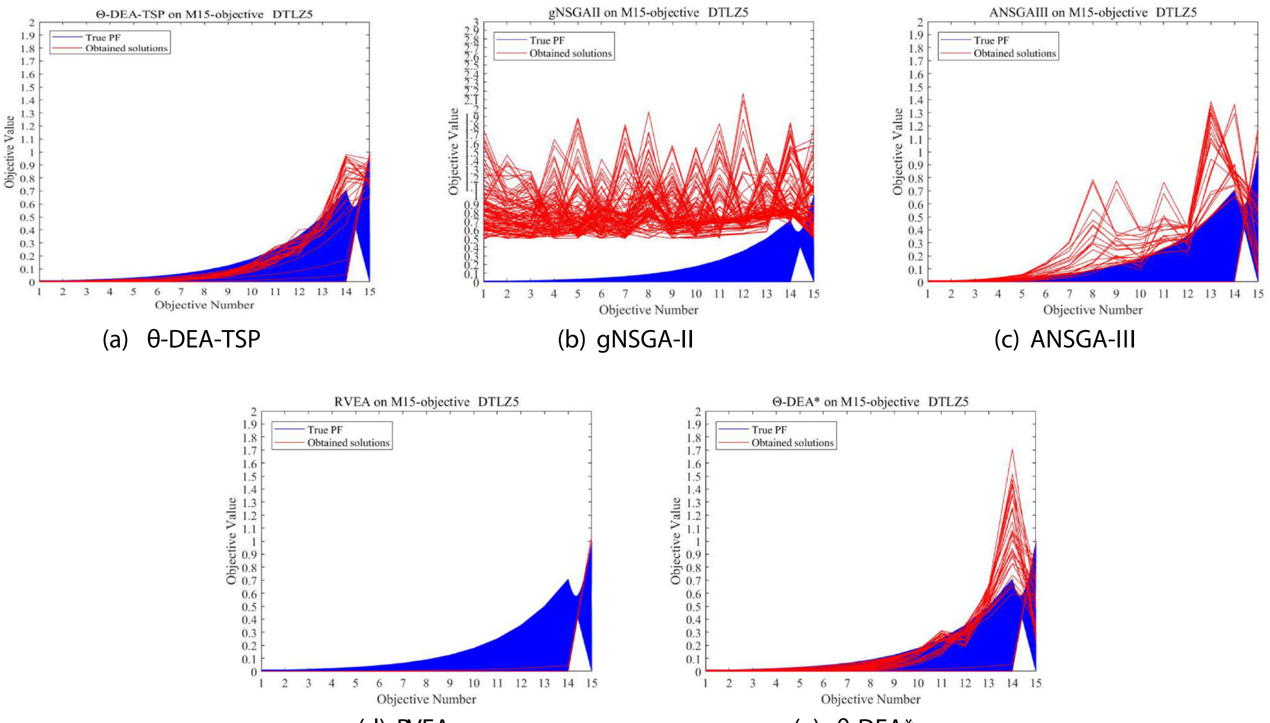


Fig. 4 Bar chart of the average rankings of the five algorithms considered for comparison in terms of the IGD metric on the 6-, 8-, 10-, and 15-objective versions of the DTLZ2-7 test problems

The experimental results of θ-DEA-TSP are better than those of the other compared algorithms on half of these test instances. The PF of DTLZ5IM is nonconvex. By comparing the experimental HV and IGD results, it can be seen that the HV and IGD values may contradict each other in the testing of the PF. Nearly half of the HV values for gNSGA-II are 0, because all obtained solutions do not form a supervolume with the lowest point.



(a) θ-DEA-TSP

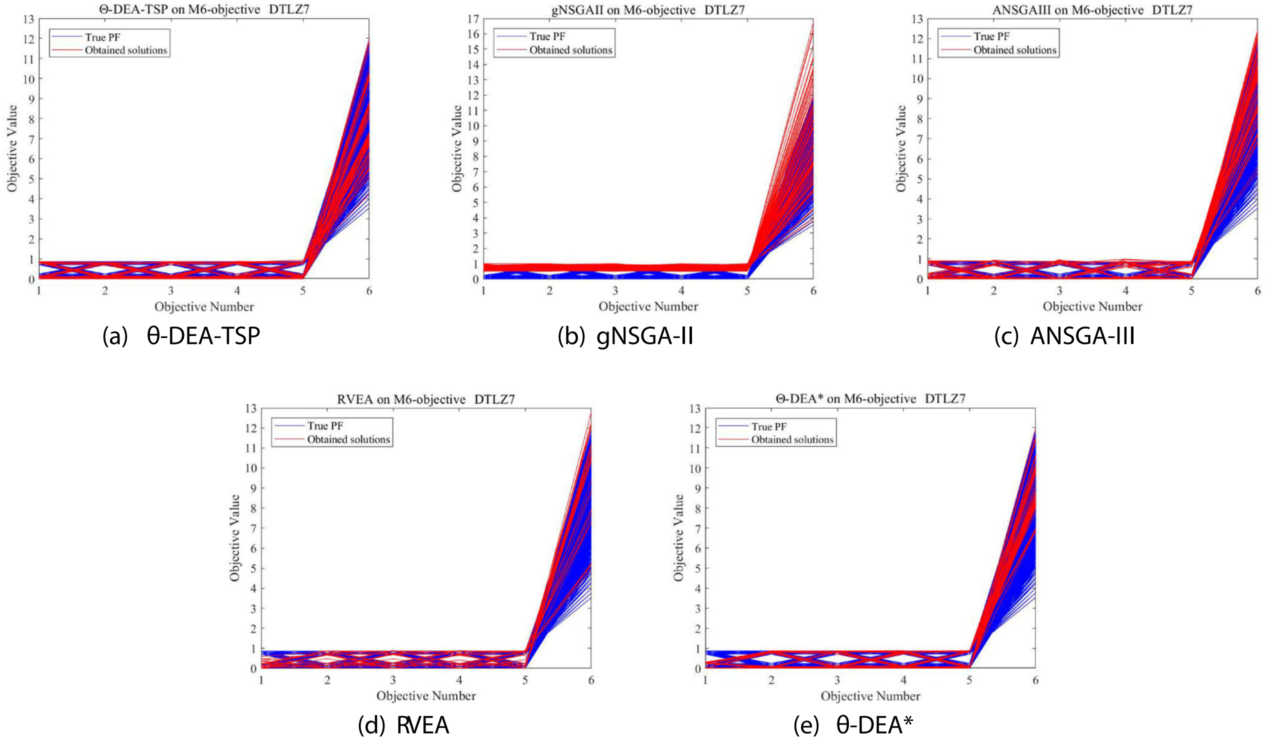
(b) gNSGA-II

(c) ANSGA-III

(d) RVEA

(e) θ-DEA*

Fig. 5 The solutions obtained by the five compared algorithms on 15-objective DTLZ5



(a) θ-DEA-TSP (b) gNSGA-II (c) ANSGA-III

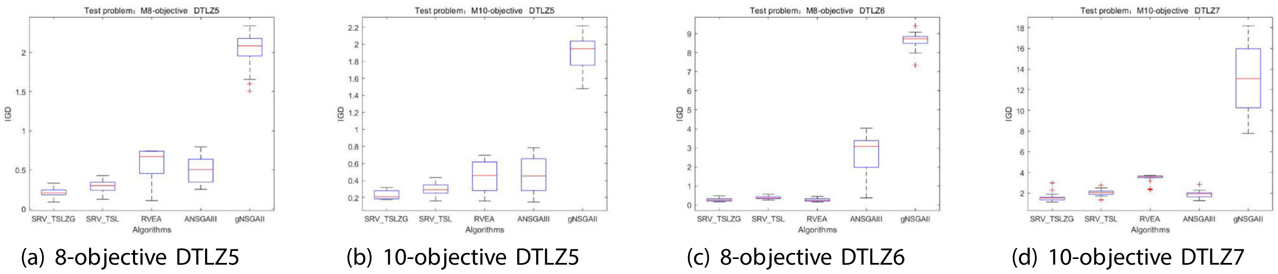
(d) RVEA (e) θ-DEA*

Fig. 6 The solutions obtained by the five compared algorithms on 6-objective DTLZ7

4.4.2 Real world problem

To verify the performance of the proposed algorithm on real problems, we also conduct a comparative analysis of this algorithm on the multipoint distance minimization problem (M-DMP) [40]. The problem of minimizing the distance between a point and a set of predetermined fixed target points has many advanced applications in navigation and layout optimization [30]. In this section, the D value of the problem is set to 2, and the upper and lower bounds are 100 and -100 [40], respectively. The other experimental parameters are the same as above. For real problems, we usually choose HV to measure the quality of the solution set.

The experimental results for versions of M-MDP with different numbers of dimensions are shown in Table 7. It can be seen that the proposed algorithm performs very well on this real problem, and the experimental results of gNSGA-II are also surprisingly good, perhaps because the changes to the traditional concept of dominance introduced by the g-dominance concept applied in this algorithm allow it to perform only slightly worse than θ-DEA-TSP on this problem. In contrast, because θ-DEA* and ANSGA-III sacrifice their convergence to preserve the diversity of the solution set, their solution sets do not form a hypervolume with the reference point of the HV metric for 15-objective M-DMP, resulting in a result of 0.



(a) 8-objective DTLZ5 (b) 10-objective DTLZ5 (c) 8-objective DTLZ6 (d) 10-objective DTLZ7

Fig. 7 Boxplots of the IGD values of the five compared algorithms on the 8-objective DTLZ5, 10-objective DTLZ5, 8-objective DTLZ6, and 10-objective DTLZ7 test problems

Table 5 The average IGD values of the θ-DEA-TSP algorithm and other algorithms on other many-objective test problems

Problem	M	θ-DEA-TSP Mean(std)	θ-DEA* Mean(std)	RVEA Mean(std)	ANSGA-III Mean(std)	gNSGA-II Mean(std)
DTLZ5IM	6	**3.9123e-2 (4.80e-2)**	1.1640e-1 - (7.86e-2)	1.4799e-1 - (1.95e-1)	4.0074e-2 = (3.00e-2)	1.9450e+0 - (1.15e+0)
DTLZ5IM	8	**9.2284e-2 (1.01e-1)**	1.0393e-1 - (6.40e-2)	3.6694e-1 - (2.92e-1)	1.1781e-1 - (9.19e-2)	6.2592e+1 - (3.69e+1)
DTLZ5IM	10	**1.6796e-1 (1.51e-1)**	1.8885e-1 - (1.22e-1)	2.2170e-1 = (1.95e-1)	3.9733e-1 - (1.57e-1)	9.1263e+1 - (3.13e+1)
DTLZ5IM	15	**3.1388e-1 (4.14e-2)**	3.2713e-1 = (1.54e-1)	6.0368e-1 - (2.25e-1)	4.9579e-1 - (2.16e-1)	1.4307e+2 - (4.61e+1)
IDTLZ1	6	1.4453e-1 (3.05e-2)	1.2669e-1 + (6.15e-3)	2.3069e-1 - (3.48e-2)	**1.1759e-1 + (3.32e-3)**	5.1524e-1 - (4.63e-1)
IDTLZ1	8	1.6993e-1 (1.68e-2)	1.6479e-1 = (4.51e-3)	2.7067e-1 - (2.43e-2)	**1.5655e-1 = (1.51e-3)**	7.5478e-1 - (5.47e-1)
IDTLZ1	10	**1.6084e-1 (4.72e-3)**	1.8035e-1 - (7.78e-3)	3.1158e-1 - (3.63e-2)	1.6826e-1 - (7.27e-3)	7.3982e-1 - (5.12e-1)
IDTLZ1	15	**1.8314e-1 (4.48e-3)**	1.8668e-1 = (9.20e-3)	3.4481e-1 - (5.03e-2)	1.9139e-1 - (8.08e-3)	7.6855e-1 - (5.61e-1)
IDTLZ2	6	**3.7507e-1 (1.17e-2)**	4.6668e-1 - (1.52e-2)	4.7025e-1 - (1.01e-2)	4.0559e-1 - (1.21e-2)	7.2202e-1 - (8.31e-2)
IDTLZ2	8	6.6488e-1 (1.63e-2)	6.3489e-1 + (1.44e-2)	7.7892e-1 - (3.42e-2)	**5.8464e-1 + (1.31e-2)**	7.6956e-1 - (4.99e-2)
IDTLZ2	10	7.1381e-1 (5.89e-3)	7.4165e-1 - (1.48e-2)	7.4738e-1 - (2.01e-2)	**7.0176e-1 + (1.23e-2)**	7.9541e-1 - (4.75e-2)
IDTLZ2	15	8.5849e-1 (9.53e-3)	8.5484e-1 = (1.17e-2)	9.4955e-1 - (3.34e-2)	8.5090e-1 = (6.53e-3)	**8.2826e-2 + (3.41e-2)**
+/-/=			2/6/4	0/11/1	3/6/3	1/11/0

For each instance, the best average value among the algorithms is indicated in bold

Table 6 The average HV values of the θ-DEA-TSP algorithm and other algorithms on other many-objective test problems

Problem	M	θ-DEA-TSP Mean(std)	θ-DEA* Mean(std)	RVEA Mean(std)	ANSGA-III Mean(std)	gNSGA-II Mean(std)
DTLZ5IM	6	1.4278e-3 (4.69e-4)	**1.5674e-3 + (4.33e-4)**	1.5375e-3 + (1.44e-3)	1.4196e-3 = (4.89e-4)	0.0000+0 - (0.00e+0)
DTLZ5IM	8	1.8510e-5 (8.26e-6)	2.0041e-5 + (6.47e-6)	**2.8678e-5 + (1.60e-5)**	1.6030e-5 - (2.55e-6)	0.0000e+0 - (0.00e+0)
DTLZ5IM	10	6.2954e-8 (2.37e-8)	**6.4827e-8 + (1.92e-8)**	3.2441e-8 - (5.00e-8)	5.9923e-8 - (1.05e-8)	0.0000e+0 - (0.00e+0)
DTLZ5IM	15	6.6912e-17 (4.33e-17)	7.8683e-17 + (5.29e-17)	**1.6674e-16 + (1.86e-16)**	8.9989e-17 + (3.99e-17)	0.0000e+0 - (0.00e+0)
IDTLZ1	6	2.5564e-5 (2.39e-5)	2.1445e-5 - (8.34e-6)	4.8517e-6 - (6.28e-6)	**2.9778e-5 + (8.87e-6)**	1.9836e-5 - (6.58e-5)
IDTLZ1	8	8.9610e-8 (5.74e-8)	1.3295e-7 + (4.28e-8)	9.4814e-9 - (7.13e-9)	**1.5796e-7 + (7.09e-8)**	0.0000e+0 - (0.00e+0)
IDTLZ1	10	**7.8832e-10 (8.95e-10)**	3.9173e-10 - (1.38e-10)	1.6812e-11 + (2.70e-11)	5.5478e-10 - (2.82e-10)	0.0000e+0 - (0.00e+0)
IDTLZ1	15	**3.1001e-16 (4.63e-16)**	1.0787e-16 - (1.59e-16)	2.3351e-18 - (3.31e-18)	2.0367e-16 - (5.45e-16)	0.0000e+0 - (0.00e+0)
IDTLZ2	6	**5.6369e-2 (1.57e-2)**	4.2315e-2 - (1.01e-2)	1.8627e-2 - (8.79e-3)	2.5565e-2 - (1.15e-2)	2.3267e-4 - (5.44e-4)
IDTLZ2	8	**5.6218e-3 (1.19e-3)**	5.5051e-3 - (1.18e-3)	7.3479e-4 - (7.15e-4)	3.9949e-3 - (2.11e-3)	4.6283e-6 - (1.22e-5)
IDTLZ2	10	**4.9093e-4 (9.72e-5)**	4.2846e-4 - (6.13e-5)	2.6033e-4 - (1.42e-4)	4.5314e-4 - (1.02e-4)	5.5942e-7 - (9.43e-7)
IDTLZ2	15	**4.7652e-7 (1.17e-7)**	4.5133e-7 - (1.30e-7)	1.5472e-8 - (1.88e-8)	4.5781e-7 - (1.21e-7)	8.4903e-11 - (2.52e-10)
+/-/=			5/7/0	4/8/0	3/8/1	0/12/0

For each instance, the best average value among the algorithms is indicated in bold

Table 7 The average HV values of the θ-DEA-TSP algorithm and other algorithms on M-DMP

Problem	M	θ-DEA-TSP Mean(std)	θ-DEA* Mean(std)	RVEA Mean(std)	ANSGA-III Mean(std)	gNSGA-II Mean(std)
M-DMP	6	**8.6271e+0 (2.49e+0)**	7.7359e+0 - (1.76e+0)	5.3380e+0 - (3.28e+0)	7.1251e+0 - (2.36e+0)	8.0056e+0 - (8.89e-1)
M-DMP	8	**1.4214e+1 (6.14e+0)**	1.4136e+1 = (5.83e+0)	6.1359e+0 - (5.18e+0)	1.0914e+1 - (4.60e+0)	1.3143e+1 - (3.49e+0)
M-DMP	10	**2.6582e+1 (2.13e+1)**	1.6852e+1 - (9.97e+0)	6.9134e+0 - (9.28e+0)	1.4776e+1 - (1.16e+1)	2.2878e+1 - (6.12e+0)
M-DMP	15	3.2693e+1 (9.78e+1)	0.0000e+0 - (0.00e+0)	1.0481e+1 - (1.46e+1)	0.0000e+0 - (0.00e+0)	**5.9407e+1 + (3.81e+1)**
+/-/=			0/3/1	0/4/0	0/4/0	1/3/0

For each instance, the best average value among the algorithms is indicated in bold

5 Conclusion

We propose the TSP strategy to balance the convergence and diversity of the population during evolution and correspondingly propose the θ-DEA-TSP algorithm. The proposed algorithm has a lower computational cost on MaOPs, and can better solve problems with complex PFs and balance the convergence and diversity of the solution set throughout the whole execution of the algorithm. For classic test problems, recent test problems and a real world problem with different numbers of objectives, the GD, IGD and HV values are better than those of other algorithms on at least half the tested problem instances, although some of the algorithms considered for comparison already perform extremely well on MaOPs, thereby confirming the effectiveness of the proposed MOEA. However, as described by Aggarwal [41], in the case of high dimensions, the Euclidean distance is not a good metric. Therefore, we must admit that in MaOPs, the proposed TSP strategy will still have shortcomings. In future work, we need to find another method of measuring the diversity and convergence between individuals to replace the calculation of the Euclidean distance in the TSP strategy.

Acknowledgements This work was supported in part by Natural Science Foundation of Zhejiang Province (LQ20F020014), in part by the National Natural Science Foundation of China (61472366, 61379077), in part by the Natural Science Foundation of Zhejiang Province (LY17F020022), in part by Key Projects of Science and Technology Development Plan of Zhejiang Province (2018C01080).

Declarations

Human participants This study does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Alkebsi K, Du W (2021) Surrogate-assisted multi-objective particle swarm optimization for the operation of co2 capture using vpsa. Energy 224(3):120078
2. Tawhid MA, Savsani V (October 2018) A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems. Appl Intell 48(10):3762?3781. https://doi.org/10.1007/s10489-018-1170-x
3. Mirjalili S, Jangir P, Saremi S (January 2017) Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. Appl Intell 46(1):79?-95. https://doi.org/10.1007/s10489-016-0825-8
4. Khan I, Maiti MK, Basuli K (2020) Multi-objective traveling salesman problem: an abc approach. Appl Intell, 2
5. Li H, Deb K, Zhang Q, Suganthan PN, Chen L (2019) Comparison between moea/d and nsga-iii on a set of many and multi-objective benchmark problems with challenging difficulties. Swarm and Evolutionary Computation
6. Bugingo E, Zhang D, Chen Z, Zheng W (2020) Towards decomposition based multi-objective workflow scheduling for big data processing in clouds. Clust Comput, pp 1–25
7. Panagant N, Pholdee N, Bureerat S, Yildiz AR, Mirjalili S (2021) A comparative study of recent multi-objective metaheuristics for solving constrained truss optimisation problems. Archives of Computational Methods in Engineering, pp 1–17
8. Lu C, Gao L, Pan Q, Li X, Zheng J (2019) A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution. Appl Soft Comput 75:728–749
9. Liang Z, Hu K, Ma X, Zhu Z (2019) A many-objective evolutionary algorithm based on a two-round selection strategy. IEEE Transactions on Cybernetics, PP(99)
10. Zhou J, Yao X, Gao L, Hu C (2021) An indicator and adaptive region division based evolutionary algorithm for many-objective optimization. Appl Soft Comput 99:106872
11. Luo J, Huang X, Yang Y, Li X, Feng J (2019) A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. Inf Sci, 514
12. Fellow, IEEE, Jain H, Deb K (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. IEEE Trans Evol Comput 18(4):602–622
13. Zhang Q, Li H (2007) Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731. https://doi.org/10.1109/TEVC.2007.892759
14. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput, 20(5)
15. Liu H-L, Gu F, Cheung Y (2010) T-moea/d: Moea/d with objective transform in multi-objective problems. In: 2010 international conference of information science and management engineering, vol 2, IEEE, pp 282–285
16. Das BI, Dennis JE (1998) Normal boundary intersection: A new method for generating pareto optimal points in multicriteria optimization problems
17. Pan L, He C, Tian Y, Su Y, Zhang X (2017) A region division based diversity maintaining approach for many-objective optimization. Integrated Computer-Aided Engineering 24(3):279–296
18. Jiang S, Yang S (2015) An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts. IEEE transactions on cybernetics 46(2):421–437
19. Dong Z, Wang X, Tang L (2020) Moea/d with a self-adaptive weight vector adjustment strategy based on chain segmentation. Inf Sci 521:209–230
20. Qi Y, Ma X, Liu F, Jiao L, Sun J, Wu J (2014) Moea/d with adaptive weight adjustment. Evolutionary computation 22(2):231–264
21. Gu F, Cheung Y-M (2017) Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. IEEE Trans Evol Comput 22(2):211–225
22. Wu M, Li K, Kwong S, Zhang Q, Zhang J (2018) Learning to decompose: A paradigm for decomposition-based multiobjective optimization. IEEE Trans Evol Comput 23(3):376–390

23. Zhang Q, Zhu W, Liao B, Chen X, Cai L (2018) A modified pbi approach for multi-objective optimization with complex pareto fronts. Swarm and Evolutionary Computation 40:216–237
24. Liang Z, Hou W, Huang X, Zhu Z (2019) Two new reference vector adaptation strategies for many-objective evolutionary algorithms. Inf Sci 483:332–349
25. Jain H, Deb K (2013) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. IEEE Transactions on evolutionary computation 18(4):602–622
26. Molina J, Santana LV, Hernández-Díaz AG, Coello CAC, Caballero R (2009) g-dominance: Reference point based dominance for multiobjective metaheuristics. Eur J Oper Res 197(2):685–692
27. Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), vol 1, IEEE, pp 825–830
28. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87
29. Yuan Y, Xu H, Wang B, Yao X (2015) A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 20(1):16–37
30. Tian Y, Cheng R, Zhang X, Su Y, Jin Y (2018) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. Evolutionary Computation, IEEE Transactions on
31. Guo X, Wang X, Wei Z (2015) Moea/d with adaptive weight vector design. In: 2015 11th international conference on computational intelligence and security (CIS), IEEE, pp 291–294
32. Tian Y, Cheng R, Zhang X, Cheng F, Jin Y (2017) An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. IEEE Trans Evol Comput 22(4):609–622
33. Deb K, Saxena DK (2005) On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-ob jective optimization problems
34. Van Veldhuizen DA, Lamont GB (2000) On measuring multiobjective evolutionary algorithm performance. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512), vol 1, IEEE, pp 204–211
35. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms?a comparative case study. In: International conference on parallel problem solving from nature, Springer, pp 292–301
36. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271
37. Jiang S, Ong Y, Zhang J, Feng L (2014) Consistencies and contradictions of performance metrics in multiobjective optimization. IEEE Transactions on Cybernetics 44(12):2391–2404
38. Deb K, Thiele L, Laumanns M, Zitzler E (2006) Scalable test problems for evolutionary multi-objective optimization
39. Wang L, Pan X, Shen X, Zhao P, Qiu Q (2020) Balancing convergence and diversity in resource allocation strategy for decomposition-based multi-objective evolutionary algorithm. Appl Soft Comput 100:106968
40. Xu J, Deb K, Gaur A (2015) Identifying the pareto-optimal solutions for multi-point distance minimization problem in manhattan space. Comput. Optim. Innov.(COIN) Lab., East Lansing, MI, USA, COIN Tech. Rep. 2015018
41. Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: International conference on database theory, Springer, pp 420–434

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Lin Zhang received the B.S. degree from Jiaxing University Nanhu college, Zhejiang, China, in 2019. And now pursuing the M.S degree from Zhejiang University of Technology, Hangzhou, China. His main research direction is computational intelligence, decision optimization.

Liping Wang received the B.S. degree from East China Normal University, Shanghai, China, in 1986, the M.S. degree from Zhejiang University, Zhejiang, China, in 1997, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2005. She is currently a full professor with College of Computer Science and Technology, Zhejiang University of Technology, China. And she is also the Director of Institute of Information Intelligent and Decision Optimization in Zhejiang University of Technology. She is CCF member and No.98913M. Her main research interests include computational intelligence with applications to optimization, learning, data mining.

Xiaotian Pan received the B.S. degree from Nankai University Binhai college, Tianjin, China, in 2015, the M.S. degree from Zhengzhou University of Aeronautics, Zhengzhou, Henan, China, in 2018. And received the Ph.D. degree from Zhejiang University of Technology, Hangzhou, China, in 2022. She is CCF student member and No.97732G. Her main research direction is computational intelligence, decision optimization.

Qicang Qiu received the B.S. degree from Zhejiang University, Zhejiang, China, in 2015, the M.S. degree from San Jose State University, California, United States of America, in 2018. He is currently an engineer in Zhejiang Lab, China. His main research direction is computational intelligence, network security.