Swarm and Evolutionary Computation 62 (2021) 100849

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Differential evolution using improved crowding distance for multimodal multiobjective optimization

Caitong Yue [a], P.N. Suganthan [b,*], Jing Liang [a,*], Boyang Qu [c], Kunjie Yu [a], Yongsheng Zhu [c], Li Yan [c]

[a] School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[c] School of Electronic and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China

ARTICLE INFO

Keywords:
Multimodal multiobjective optimization
Differential evolution
Crowding distance

ABSTRACT

In multiobjective optimization, the relationship between decision space and objective space is generally assumed to be a one-to-one mapping, but it is not always the case. In some problems, different variables have the same or similar objective value, which means a many-to-one mapping. In this situation, there is more than one Pareto Set (PS) mapping to the same Pareto Front (PF) and these problems are called multimodal multiobjective problems. This paper proposes a multimodal multiobjective differential evolution algorithm to solve these problems. In the proposed method, the difference vector is generated taking the diversity in both decision and objective space into account. The way to calculate crowding distance is quite different from the others. In the crowding distance calculation process, all the selected individuals are taken into account instead of considering each Pareto rank separately. The crowding distance in decision space is replaced with the weighted sum of Euclidean distances to its neighbors. In the environmental selection process, not all the individuals in top ranks are selected, because some of them may be very crowded. Instead, the potential solutions in the bottom rank are given a chance to evolve. With these operations, the proposed algorithm can maintain multiple PSs of multimodal multiobjective optimization problems and improve the diversity in both decision and objective space. Experimental results show that the proposed method can achieve high comprehensive performance.

1. Introduction

Multiobjective optimization [1–3] refers to problems with more than one objective to be optimized. In most cases, improving one objective value will deteriorate others. Therefore, it is infeasible to optimize these objectives separately. Without loss of generality, a minimization-based multiobjective optimization problem can be defined as

$$\min \ \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ \dots\ ,\ f_m(\mathbf{x}) \right] \tag{1}$$

subjected to: $\mathbf{g}(\mathbf{x}) \le 0, \mathbf{h}(\mathbf{x}) = 0$ where $\mathbf{f} \in R^m, \mathbf{x} \in R^n$, $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are inequality and equality constraints, respectively. For two feasible solutions $\mathbf{x}^1$ and $\mathbf{x}^2$, $\mathbf{x}^1$ dominates $\mathbf{x}^2$ on the condition that $\mathbf{f}(\mathbf{x}^1) \le \mathbf{f}(\mathbf{x}^2)$ and $\exists j \in [1, \dots, m], f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$. If a certain solution is not dominated by any other solutions, it is called nondominated solution [4]. All nondominated solutions consist of Pareto optimal Set (PS). The image of PS in objective space is called Pareto Front (PF). The study on algorithms [5] and application [6] of multiobjective optimization has always been a popular research field.

Similar to multimodal single objective optimization [7–11], the multimodal situation also occurs in multiobjective optimization. In multiobjective optimization, the mapping from decision space to objective space is not always one-to-one mapping. Therefore, the PS is not unique in some situations. As described in Ref. [12], there may be multiple PSs mapping to the same PF. This kind of problem is called Multimodal Multiobjective Optimization Problem (MMOP).

It is of great theoretical and practical significance to find multiple PSs of MMOPs [13–15]. On one hand, multiple PSs can reveal the characteristics of the problem more comprehensively. If only one of these PSs is found, many characteristics of the problem cannot be shown. It is not conducive to the study of the problem. On the other hand, finding multiple PSs can offer decision-makers more feasible choices. In practical applications, some Pareto optimal solutions may become infeasible due to environmental changes or practical constraints. Therefore, it is necessary to find multiple PSs of MMOPs.

A multimodal multiobjective problem in feature selection [13] is given in Fig. 1 to illustrate the significance of solving MMOPs intuitively.

* Corresponding authors.
E-mail addresses: epnsugan@ntu.edu.sg (P.N. Suganthan), liangjing@zzu.edu.cn (J. Liang), quboyang@zut.edu.cn (B. Qu), yukunjie@zzu.edu.cn (K. Yu), yanli@zut.edu.cn (L. Yan).

https://doi.org/10.1016/j.swevo.2021.100849
Received 15 July 2020; Received in revised form 24 November 2020; Accepted 1 February 2021
Available online 9 February 2021

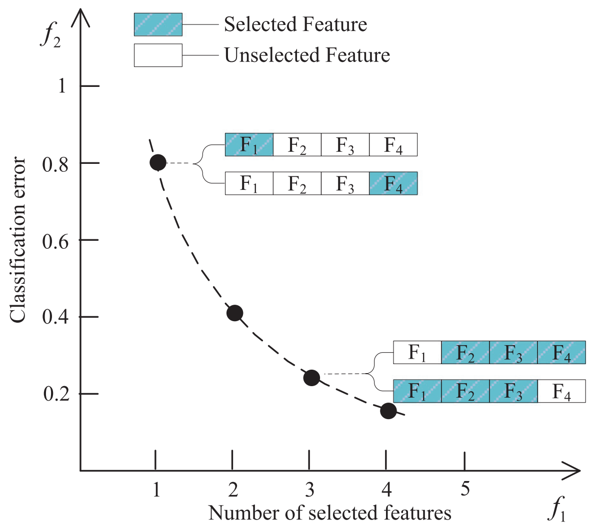


Fig. 1. Multimodal multiobjective optimization problem in feature selection.

In Fig. 1, the number of selected features and classification error are two objectives to be minimized. F_1-F_4 are four alternative features. The white rectangles represent unselected features, while the shaded ones represent selected features. The black dots lie on the PF of this problem. If only one feature is allowed to be selected, either {F_1} or {F_4} will achieve the same classification error 0.8. If three features are allowed to be selected, either {F_2, F_3, F_4} or {F_1, F_2, F_3} will achieve the same classification error 0.2. There are at least four equivalent feature combinations '{F_1} {F_2, F_3, F_4}'; '{F_1} {F_1, F_2, F_3}'; '{F_4} {F_2, F_3, F_4}'; '{F_4} {F_1, F_2, F_3}' in the PS of this problem. Single-modal multiobjective feature selection methods only provide one of these combinations. However, providing all of them for decision-makers is of great significance. On one hand, these four equivalent feature combinations can reveal that either F_1 or F_4 is redundant, because selecting or neglecting one of them can achieve the same classification error. On the other hand, these four equivalent feature combinations can provide more choices for decision-makers. If F_1 is quite difficult to extract and all of these four equivalent feature combinations are provided, the decision-maker can select '{F_4} {F_2, F_3, F_4}' to avoid extracting F_1. By contrast, if single-modal multiobjective feature selection methods provide only '{F_1} {F_1, F_2, F_3}' for the decision-maker, then he/she has to waste lots of time or money to extract F_1.

Many researchers have proposed different multimodal multiobjective optimization algorithms. Deb and Tiwari [16] proposed Omni-optimizer to solve multiple optimization problems including uni- and multi- modal, single- and multi- objective optimization problems. In Omni-optimizer, crowding distances in both decision and objective space are considered. However, MMOPs are not specifically studied in depth. Liang et al. [17] focused on MMOPs and analyzed the challenges to solve this kind of problem. Subsequently, Yue et al. [12] proposed a multimodal multiobjective particle swarm optimizer using ring topology and special crowding distance (MO_Ring_PSO_SCD). Liu et al. [18] also designed a multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. Many algorithms were designed to solve these test functions and an MMO competition was organized to compare different MMO algorithms in CEC 2019 [19]. Although there are many different MMO algorithms, they have a common disadvantage. They sacrifice the performance in objective space to improve the performance in decision space. In other words, though they find more Pareto optimal solutions, the diversity and convergence in objective space deteriorate. To deal with this issue, a Multimodal Multiobjective Differential Evolution Algorithm Using Improved Crowding Distance (MMODE_ICD) is proposed.

The main contributions of this paper include:

(1) A novel differential vectors generation method which balances the exploration and exploitation abilities and improves the diversity in both decision and objective space is proposed.
(2) A crowding distance calculation method which can measure the true crowding degree and balance the diversity and convergence is designed. On one hand, all the selected solutions are taken into account instead of only considering the ones in the same Pareto rank to reflect the true crowding degree. On the other hand, the Pareto rank value is embedded into the special crowding distance (*SCD*) to balance the diversity and convergence.
(3) An extraordinary environmental selection scheme which gives chance to the potential solutions in low rank is proposed. In the proposed selection scheme, not all the members in the top ranks are selected but a certain percentage with large crowding distance is selected adaptively.

The rest of this paper is organized as follows. Section II presents the related definitions and difficulties to be deal with. Section IV describes the proposed method in detail. The effectiveness of the algorithm mechanism is analyzed in Section V. Experimental results and discussions are given in Section VI. At last, this paper is concluded in Section VII.

2. Related works

In this section, the related definitions of MMO are introduced. Then the basic framework of the differential evolution (DE) algorithm is presented. The difficulties of solving MMOP and why DE is chosen are analyzed.

2.1. Related definitions of MMO

Generally, PS refers to global PS, which can be defined as: A solution set is called *global PS* [20] if none of its elements are dominated by any other solutions in the whole feasible region. Similar to local peaks in multimodal single-objective optimization, there are local PSs in MMO. A solution set is called *local PS* if none of its elements are dominated by their local neighbors.

Given a multiobjective optimization problem, it belongs to MMOP if it meets one of the following conditions [13]:

(1) It has at least one local PS;
(2) It has more than one global PS.

Note that global PS doesn't belong to local PS in the above definition. Therefore, if a problem has a local PS, it must have at least one global PS. In extreme cases, the local PS may have only one solution.

In general, there are two types of MMOPs. The first type has only global PSs and the second type has both global and local PSs[21]. In this paper, only the first type of MMOP is studied.

2.2. Differential evolution

The DE algorithm [22–24] is a simple but quite effective evolution algorithm. Similar to other population-based stochastic algorithms, the procedure of DE mainly consists of four steps: population initialization, variation, recombination, and environmental selection. In the variation step, there are several different DE mutation strategies, like DE/rand/1, DE/best/1, DE/best/2, and DE/rand/2. Since multiple solution sets need to be obtained, strong exploration ability is preferred. Therefore, DE/rand/2 is used in this paper. The difference vector is generated as:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \left[\left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3} \right) + \left(\mathbf{x}_{r_4} - \mathbf{x}_{r_5} \right) \right] \tag{2}$$

where $\mathbf{v}_i$ represents the difference vector, r_1, r_2, r_3, r_4, and r_5 are mutually unequal integers. F is the scale factor that is used to scale the difference vectors. In this paper, a Euclidean distance-based niching method [7,25,26] is embedded into the DE algorithm, r_1, r_2, r_3, r_4, and r_5 are selected in the neighbors of the corresponding individual according to their crowding distances in the decision or objective space. The detailed selection method is introduced in Section 3. The reason why DE is chosen is that it is simple but quite effective in enhancing population diversity [7,27,28]. Its outstanding performance in solving multimodal single-objective problems has proved its ability in multimodal optimization [7,29]. In MMO, improving population diversity is of great significance. Therefore, DE with modified DE/rand/2 operation is employed to solve MMOPs.

There are several challenges in solving MMOPs [13,16]. First, the convergence strength will decrease significantly once one of the PSs is found. It is because when one of the PSs is obtained, the whole PF is found. The other solutions except solutions just on other PSs are all dominated by the obtained PS. Therefore, the convergence strength will decrease. Second, multiple equivalent Pareto optimal solutions are difficult to survive simultaneously in environmental selection. These equivalent Pareto optimal solutions map to the same point in the objective space. If the crowding distance in objective space is employed as the environmental selection criteria, they are too crowded to survive.

Several different algorithms have been proposed to solve MMOPs. In 2016, Liang et al. [17] draw researchers' attention to MMO by designing comprehensible MMOPs and a modified NSGAII is used to solve these problems. Subsequently, Yue et al. [12] proposed benchmark functions, performance indicator and novel algorithm at the same time. Most related works test their algorithms on this benchmark suite. Liu et al. [18] proposed TriMOEA-TA&R to solve MMOP. Two archive and recombination strategies are used to balance the diversity and convergence of the population. Li et al. [30] used reinforcement learning to help the evolution algorithm find multiple PSs. Zhang et al. [31] proposed a cluster-based particle swarm optimization algorithm to solve MMOPs. Fan and Yan [32] divided decision variables into several segments to promote population diversity and find multiple PSs. Tanabe and Ishibuchi [33] proposed a decomposition-based evolutionary algorithm framework to handle multimodal multiobjective optimization. Liang et al. [34] proposed MMODE for MMOPs. The above algorithms can enhance the diversity in the decision space. However, they will sacrifice the performance in the objective space more or less. Many other MMO algorithms like MO_PSO_MM [35], MMOPIO [36] also share the same disadvantages. They cannot balance the diversity between decision space and objective space. Most of them can improve diversity in one space but deteriorate the diversity in the other space. To deal with these challenges and disadvantages, a novel MMO algorithm is proposed in this paper. Its details are introduced in the following section.

3. Proposed method

This section introduces the framework and details of the proposed algorithm.

3.1. Framework of MMODE_ICD

The overall framework of MMODE_ICD is presented in Algorithm 1. First, the population is initialized in the search space. Then, individuals to generate difference vector are selected adaptively and the population is mutated according to Eq. (2). Third, the offspring and parents are combined into ***P***_combination. The population in the next generation is selected from ***P***_combination according to nondominated rank values and the improved crowding distances (*ICD*) and special crowding distance (*SCD*). The individuals with small nondominated rank value and large *SCD* will be selected with high probability. In the selection process, only certain ratio elements are selected in the top Pareto ranks thus giving chance to the other potential solutions. The details of the adaptive individual selection, the improved crowding distance, and the ratio selection are described in the following texts.

3.2. Adaptive individuals selection to generate difference vector

In order to improve the diversity in both decision and objective space, the individuals to generate difference vector are selected using the following three methods adaptively.

(1) Randomly select five neighbors in the whole population as $\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \mathbf{x}_{r_4}, \mathbf{x}_{r_5}$;
(2) Select a certain number of neighbors according to the Euclidean distance in the decision space between the current individual and the remaining solutions in the whole population. Then, randomly select five neighbors among the certain number of neighbors and choose the one with the largest crowding distance in the decision space as $\mathbf{x}_{r_1}$. The left four act as $\mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \mathbf{x}_{r_4}, \mathbf{x}_{r_5}$;
(3) Select a certain number of neighbors according to the Euclidean distance in the objective space between the current individual and the remaining solutions in the whole population. Then, randomly select five neighbors among the certain number of neighbors and choose the one with the largest crowding distance in objective space as $\mathbf{x}_{r_1}$. The left four act as $\mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \mathbf{x}_{r_4}, \mathbf{x}_{r_5}$.

The individuals $\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \mathbf{x}_{r_4}, \mathbf{x}_{r_5}$ in Eq. (2) are selected using the first method with probability p_1, using the second method with probability p_2, and using the third method with probability p_3, where $p_1 = 1 - (G_c - 1)/Max\_gen$, $p_2 = (1 - p_1)/2$, $p_3 = 1 - p_1 - p_2$. G_c represents the current generation number and *Max_gen* represents the maximum generation number. Therefore, p_1 decreases from 1 to 0, while p_2 and p_3 increase from 0 to 0.5 as the current generation number G_c increases from 1 to *Max_gen*.

The reason why these individuals are selected in the above ways is that the exploration and exploitation abilities are balanced adaptively and the diversities in decision and objective space are all improved. On one hand, randomly selecting individuals in the whole population with high probability at the beginning of evolution can enhance the exploration ability and avoid falling into the local optimal area. In the later stage of evolution, selecting neighbors of the current individual can enhance the exploitation ability. On the other hand, selecting neighbors in decision space and choose the one with the largest crowding distance in decision space as the base vector can improve the diversity in decision space. On the other hand, selecting neighbors in objective space and choose the one with the largest crowding distance in objective space as the base vector can improve the diversity in objective space. The effectiveness of the proposed adaptive individual selection method is verified in Section 4.B.

3.3. Considering all the selected individuals in ICD

To improve the diversity of the population, crowding distances are used to measure the crowding degree in the population. The traditional crowding distance calculation procedure [4,12,17] is described as follows. First, the whole population is ranked into different Pareto ranks according to the nondominated relationship. Second, individuals in each Pareto rank are sorted according to one objective/variable value. Third, the objective/variable value difference between the next and previous individual is calculated. Fourth, the second and third steps are repeated until all the objectives are considered. Then the differences of each objective/variable are summed.

The traditional crowding distance can reflect the crowding degree of the individuals to some extent. However, it has an obvious disadvantage. Calculating crowding distance in each Pareto rank separately cannot reflect the true crowding degree. The individuals which have already been selected should be taken into account.

To describe this issue clearly, Fig. 2 presents an example to compare the crowding distance considering selected individuals or not. In Fig. 2(a), f_1 and f_2 are two objectives to be minimized. Filled circles and hollow circles represent the first Pareto rank and the second Pareto

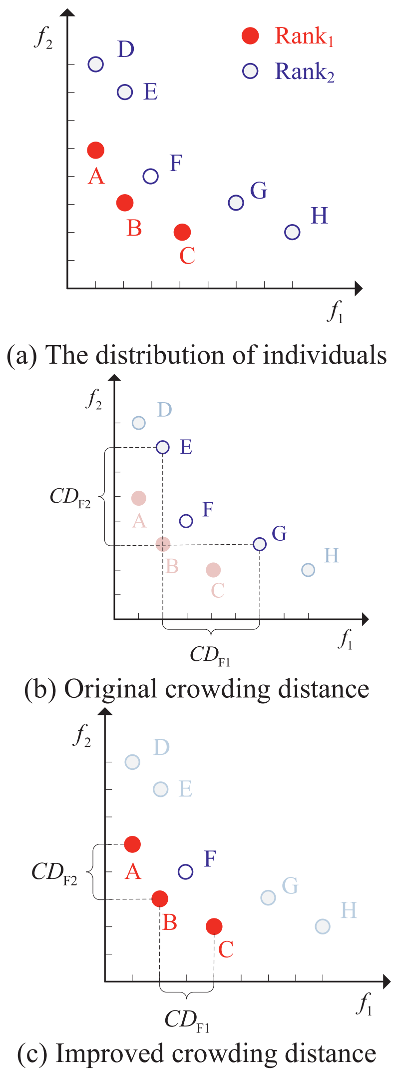


Fig. 2. Comparison of the original and improved crowding distance.

rank, respectively. The crowding distance calculation of individual F is taken as an example.

Fig. 2(b) shows the traditional crowding distance calculation for individual F. Only the hollow circles are considered because F belongs to the second Pareto rank. The previous and next individuals of F are E and G according to f_1. Thus, the crowding distance of F in f_1 is $CD_{F1} = f_{G1} - f_{E1} = 6 - 2$. According to f_2, the previous and next individuals of F are G and E. Therefore, the crowding distance of F in f_2 is $CD_{F2} = f_{E2} - f_{G2} = 7 - 3$. The crowding distance of F is $CD_F = CD_{F1} + CD_{F2} = (6 - 2) + (7 - 3) = 8$.

Fig. 2(c) shows the modified crowding distance calculation for individual F. Both the hollow circles and the filled circles are considered because the filled circles have already been selected into the next generation. The previous and next individuals of F are B and C according to f_1. Thus, the crowding distance of F in f_1 is $CD_{F1} = f_{C1} - f_{B1} = 4 - 2$. According to f_2, the previous and next individuals of F are A and B. Therefore, the crowding distance of F in f_2 is $CD_{F2} = f_{A1} - f_{B1} = 5 - 3$. The crowding distance of F is $CD_F = CD_{F1} + CD_{F2} = (4 - 2) + (5 - 3) = 4$.

The crowding distance of F calculated by traditional crowding distance calculation is twice of the crowding distance calculated by the modified method. In fact, the true neighbors of F are {A, B, C} not {E, G} as shown in Fig. 2(a). Therefore, the crowding distance computed independently in each Pareto rank cannot reflect the true crowding degree in the population. If the traditional crowding distance is assigned to F, it survives with a high probability because it has a relatively large crowding distance. However, its actual crowding distance is much smaller. To deal with this issue, the already selected individuals should also be considered.

In objective space, the crowding distance is calculated using the previous and next neighbor in each dimension, while in decision space the crowding distance is calculated by averaging the weighted Euclidean distance to neighbors. Let $CD_{i,x}$ denote the crowding distance of the i^{th} individual in decision space. Then $CD_{i,x}$ is calculated as follows.

$$CD_{i,x} = \sum_{j=1}^{k_1} \left(k_1 - j + 1\right) d_{i,j} \tag{3}$$

where $d_{i,j}$ represents the Euclidean distance from the i^{th} individual in the whole population to the j^{th} individual in its neighborhood. There are k_1 neighbors in its neighborhood and k_1 is set to 0.02**popsize* as suggested in [37].

3.4. Embed the nondominated rank into SCD

In Ref [12], the special crowding distance (*SCD*) was proposed. The *SCD* of the i^{th} individual is defined as:

$$SCD_i = \begin{cases} \max\left(CD_{i,x},\ CD_{i,f}\right) & \text{if } CD_{i,x} > CD_{avg,x} \text{ or } CD_{i,f} > CD_{avg,f} \\ \min\left(CD_{i,x},\ CD_{i,f}\right) & \text{otherwise} \end{cases} \tag{4}$$

where $CD_{i,x}$ denotes the crowding distance in decision space, $CD_{i,f}$ is the crowding distance in objective space, $CD_{avg,x}$ means the average crowding distance in decision space, and $CD_{avg,f}$ means the average crowding distance in objective space. *SCD* selects crowding distance in decision or objective space depending on the distribution in these two spaces. If the crowding distance in one space is larger than the average level, the larger one is selected as *SCD* to increase its probability of survival in environmental selection. However, this *SCD* calculation method deals with all individuals equally. In fact, improving the diversity of the last several nondominated ranks in the decision space is meaningless. The individuals in the last several nondominated ranks will not be chosen even though they have good diversity in decision space, because they are far away from PF. The diversity in decision space is more important than that in objective space for these individuals. The reason is that improving the diversity in decision space can enlarge the search area and increase the probability of finding more Pareto optimal solutions. To achieve this goal, the *SCD* of the i^{th} individual is modified as:

$$SCD_i = \begin{cases} \max\left(CD_{i,x},\ \frac{CD_{i,f}}{Rank}\right) & \text{if } CD_{i,x} > CD_{avg,x} \text{ or } CD_{i,f} > CD_{avg,f} \\ \min\left(CD_{i,x},\ CD_{i,f}\right) & \text{otherwise} \end{cases} \tag{5}$$

where *Rank* means the nondominated rank value of the i^{th} individual. In the modified method, the individuals in the first nondominated rank are not affected because *Rank* = 1 for them. For individuals whose *Rank* > 1, their crowding distances in objective space are reduced after dividing by *Rank*. The larger the *Rank* is, the more the crowding distance in objective space is reduced. Therefore, the crowding distance in the decision space is selected as *SCD* with high probability. Then the aim of improving the diversity in decision space for individuals in the last several ranks can be achieved.

3.5. Ratio selection

In multiobjective optimization, the common environmental selection method is described as follows. First, the population is sorted into different nondominated ranks. Second, individuals in the first several ranks are all selected until no more ranks can be accommodated. Third, the individuals in the last rank are sorted according to crowding distance and only the ones with small crowding distances are selected. However, some individuals in the front ranks are undesirable, while some in the last ranks are desirable. At the prophase of evolution, the obtained PF

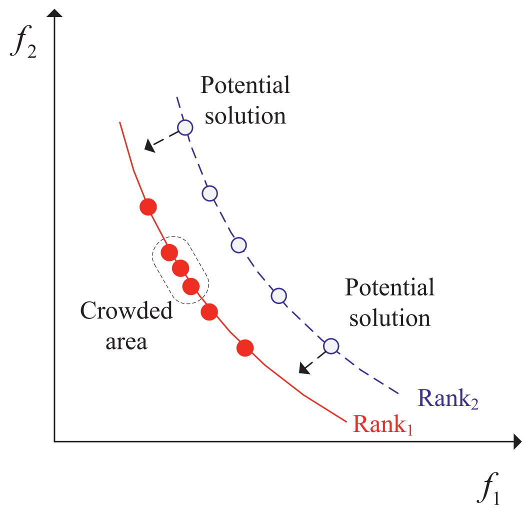


Fig. 3. The disadvantages in traditional environmental selection.

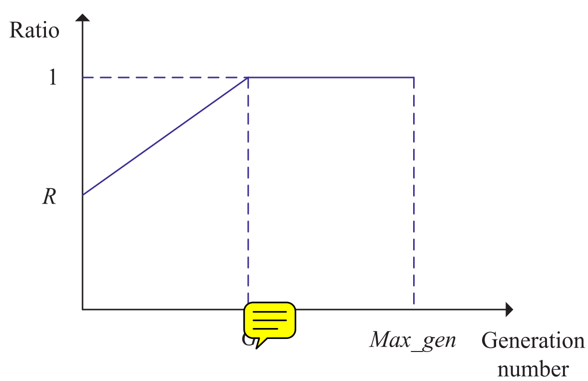


Fig. 4. The relationship between selected ratio and generation number.

is incomplete. Some elements in the last few ranks can fill the gap. As shown in Fig. 3, some individuals in the crowded area of Rank$_1$ are undesirable, while the boundary individuals in Rank$_2$ are potential solutions. These potential solutions are likely to evolve into the sparse area of Rank$_1$.

In order to delete undesirable solutions in front ranks and select potential solutions in the following ranks, a novel environmental selection method is proposed. In the proposed environmental selection method, only a certain percent of elements in front ranks are selected. The selected ratio changes with the generation number. Their relationship is shown in Fig. 4 and Eq. (6).

$$Ratio = \begin{cases} R + \frac{1-R}{G}(G_c - 1) & \text{if } 1 < G_c < G \\ 1 & \text{if } G \le G_c \le Max\_gen \end{cases} \quad (6)$$

where G_c is the current generation number and Max_gen is the maximum generation number.

When the generation number is 1, the selected ratio in front ranks is R ($0 < R < 1$). When the generation number is G ($1 < G < Max\_gen$), the selected ratio is equal to 1. The settings of R and G are discussed in Section 4.A.

The reason why the selected ratio need to vary over generation number is that the obtained PF becomes complete generation by generation. In the beginning, the obtained PF is incomplete, but there are many potential individuals in the following few ranks which can make the PF complete. Then the obtained PF becomes more and more complete with evolution. In the later stages of evolution, the obtained PF is relatively complete. It is unnecessary to select individuals in the last ranks. In the later stages, the primary task is to improve the diversity in the top ranks.

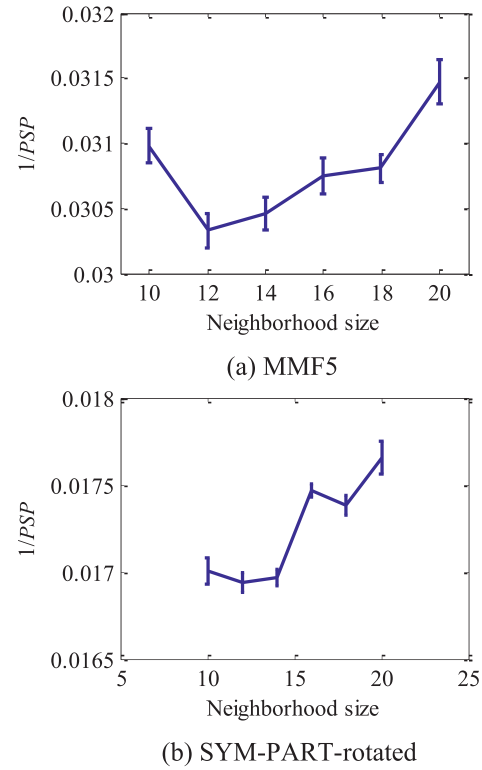


FIg. 5. The curve of $1/PSP$ changing with neighborhood size.

4. Experimental results

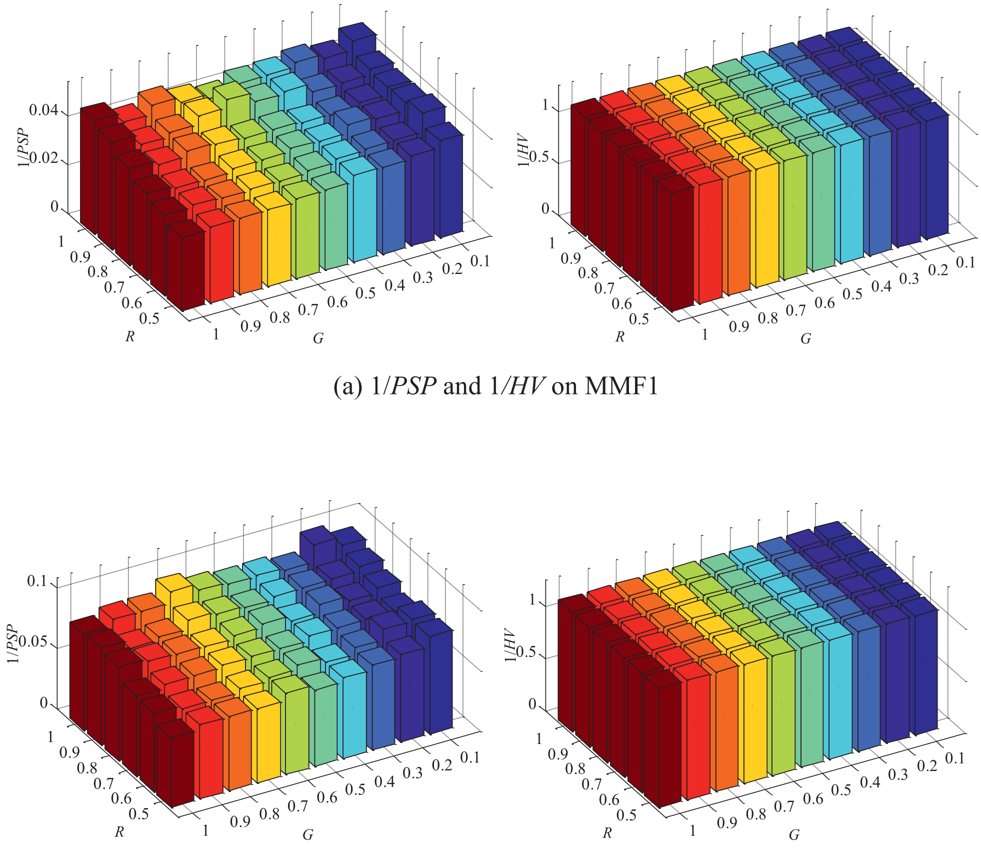
4.1. Experimental settings

4.1.1. Neighborhood size

In DE/rand/2 strategy, five individuals are needed to generate a difference vector. In the proposed algorithm, these five individuals are selected from neighbors instead of from the whole population. To determine the proper neighbor size, two to four times i.e. 10~20 neighbors are selected then these five individuals are selected according to the methods in Section 3.B. The proposed algorithm with the neighbor size of [10,12,14,16,18,20] is tested on MMF5 and SYM-PART-rotated. The reason why these two test functions are selected is that they are relatively sensitive to the neighbor size. The proper neighbor size for them is also suitable for the others. The curve of $1/PSP$ [13] changing with neighborhood size is shown in Fig. 5. Pareto Sets Proximity (*PSP*) is a performance indicator that reflects both the overlap ratio and distance between the true and the obtained PSs. $1/PSP$ can measure the performance in decision space. The smaller $1/PSP$ value means the better performance in decision space. According to Fig. 5, the suggested range of neighborhood size is from 12 to 14. In this paper, the neighborhood size is set to 12.

4.1.2. Analysis *of the influence of parameters R and G*

R (the starting selected ratio in front ranks) and G (the end generation number when the selected ratio is 1) are two important parameters in the environmental selection process. To get the best settings, the proposed algorithm with different combinations of R∈[0.5, 0.6, 0.7, 0.8, 0.9, 1.0] and G ∈[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] is tested on MMF1 and MMF5. These two test functions have the representative performance on different R and G. The mean $1/PSP$ and $1/HV$ values are shown in Fig. 6. Hypervolume (*HV*) [38] is used to measure the performance in objective space. The small value means good performance for both $1/PSP$ and $1/HV$. The following conclusions can be drawn from these figures. Judging from $1/PSP$, the performance deteriorates with



(a) $1/PSP$ and $1/HV$ on MMF1

(b) $1/PSP$ and $1/HV$ on MMF5

Fig. 6. $1/PSP$ and $1/HV$ values changing with R and G.

the increase of R while is improved with the increase of G. $1/HV$ doesn't change a lot with R and G. Considering both of these two indicators, R is set to 0.5 and G is set to 1.0 in this paper.

4.2. Analysis of the effectiveness of the proposed algorithm

There are three main effective operations in the proposed algorithm: (1) the adaptive selection mechanism to generate difference vector; (2) modification of crowding distance calculation; (3) improvement of environmental selection scheme.

(1) The proposed selection mechanism to generate difference vector can improve the convergence and diversity in both decision space and objective space. In the proposed selection mechanism, individuals to generate difference vectors are selected in three methods adaptively. The first selection method enhances exploration ability and avoids falling into local optima, the second method helps improve the convergence and diversity in decision space and the third method can improve the convergence and diversity in objective space.
(2) Modified crowding distance helps find more Pareto optimal solutions. The original crowding distance calculation method ignores the interactions between different nondominated ranks. Calculating crowding distance in each rank separately cannot reflect the real crowding degree. In the modified method, all the selected individuals are considered. In addition, it inherits the advantages of *SCD*. i.e. reflecting the crowding degree in both decision and objective space.
(3) The improved environmental selection scheme gives chance to individuals in the low ranks in the beginning, and gives priority to the nondominated front as generation number increases. It gives the individuals in the low rank a chance to evolve towards the sparse parts of the front rank.

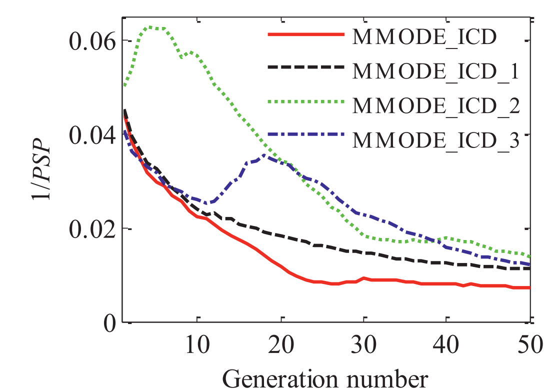


Fig. 7. Convergence behaviors of MMODE_ICD with and without effective operations.

To verify the effectiveness of the above three operations, MMODE_ICD with and without these three operations are compared. The results on MMF3 are taken as an example and they are similar to the other test functions. The experiment is carried out 21 times and the mean $1/PSP$ with generation number ranging from 1 to 50 is plotted in Fig. 7. The obtained PSs with median $1/PSP$ in the 21 times of these algorithms are shown in Fig. 8. MMODE_ICD denotes the proposed algorithm with all the above operations. MMODE_ICD_1 denotes MMODE_ICD without operation 1. It randomly selects individuals in the whole population instead of using the proposed selection method. MMODE_ICD_2 denotes MMODE_ICD without operation 2. It calculates crowding distance in each rank separately instead of considering the already selected ones. In addition, its crowding distances in the decision and objective space are

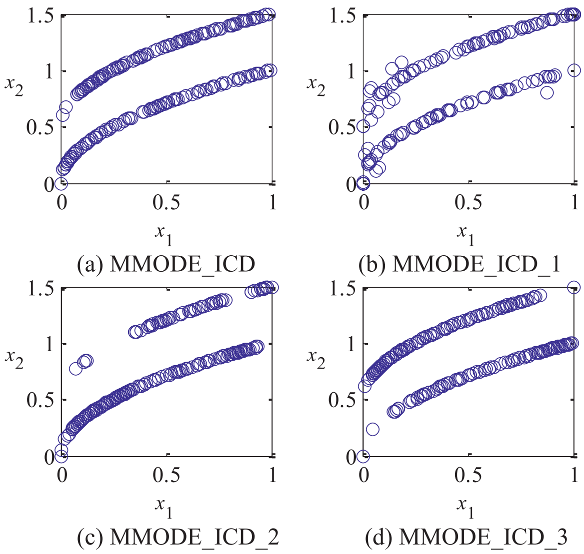


Fig. 8. The obtained PSs of MMODE_ICD with and without effective operations.

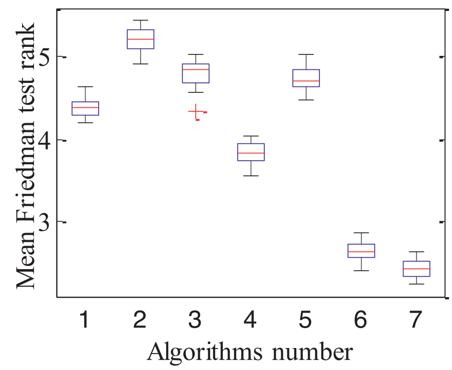


Fig. 9. Box-plots of the final Friedman test rank value.
The algorithm number on the horizontal axis indicates the following algorithms:
1 = MO_Ring_PSO_SCD, 2 = DN-NSGAII,
3 = Omni-optimizer, 4 = MMODE, 5 = TriMOEA-TA&R,
6 = MMO-Clustering PSO, and 7 = MMODE_ICD.

calculated in the same way as that of NSGAII (Fig. 2(b)) instead of using weighted sum Euclidean distance in decision space. MMODE_ICD_3 denotes MMODE_ICD without operation 3. It selects all the particles in the front ranks without giving chance to individuals in the lower ranks (like the potential solutions in Fig. 3).

Judging from Figs. 7 and 8, MMODE_ICD without any of these three operations makes it perform worse. As shown in Fig. 7, MMODE_ICD_1 converges with generation but its performance is worse than MMODE_ICD. As shown in Fig. 8(b), the population doesn't converge to the true PSs. In Fig. 7, MMODE_ICD_2 diverges in the beginning because it ignores the selected individuals and calculates the crowding distance in each rank separately. In Fig. 8(c), the obtained PSs by MMODE_ICD_2 miss several parts of the true PSs. As for MMODE_ICD_3, it diverges during the evolution as shown in Fig. 7. The obtained PSs of MMODE_ICD_3 are incomplete as shown in Fig. 8(d) because it doesn't give chance to individuals in the following ranks. Overall, MMODE_ICD_3, MMODE_ICD_2, and MMODE_ICD_3 are all worse than MMODE_ICD which can verify that all three operations are effective.

4.3. Comprehensive comparison

To verify the effectiveness of the proposed algorithm, a comprehensive comparison is given. It is compared with seven state-of-the-art MMO algorithms: MO_Ring_PSO_SCD [12], DN-NSGAII [17], Omni-optimizer [16], MMODE [34], TriMOEA-TA&R [18], and MMO-Clustering PSO.[1] Among the compared algorithms, MO_Ring_PSO_SCD and TriMOEA-TA&R are popular MMO algorithms. MMODE is another type of multimodal multiobjective DE algorithm. It uses pre-selection and boundary processing mechanisms to solve MMOPs which is significantly different from the proposed algorithm. Omni-optimizer and DN-NSGAII are two classical MMO algorithms. MMO-Clustering PSO is the champion in the CEC2019 MMO competition. Therefore, the proposed algorithm is compared with these seven MMO algorithms.

All the algorithms are tested on the test problems in CEC2019. According to the technical reports [19], the population size is set to 100^*N_{var} and maximal fitness evaluations are set to 5000^*N_{var} where N_{var} denotes the number of decision variables. In MMODE_ICD, F in Eq. (2) is set to 0.5 and the Crossover Rate (CR) is set to 0.5. All the experiments are carried out 21 times. Two performance indicators $1/PSP$ [13] and $1/HV$ [38] are used to evaluate their performances. Among them, $1/PSP$ can measure the performance in the decision space, and $1/HV$ is used to measure the performance in the objective space. For both of the indicators, the smaller value means the better performance.

The detailed results of the indicator values are shown in Tables 1 and 2. Wilcoxon's rank-sum test results are given in the last column of each table. The significant level is set to 0.05. '+' indicates that the proposed algorithm is significantly better than the algorithm in the current column, '≈' indicates the proposed algorithm is statistically similar to the algorithm in the current column, and '-' indicates the proposed algorithm is significantly worse than the algorithm in the current column. For example in the last row of column 2 of Table 1, '21/1/0' indicates the proposed algorithm is significantly better than MO_Ring_PSO_SCD on 21 test problems, similar to the MO_Ring_PSO_SCD on 1 test problem and worse than MO_Ring_PSO_SCD on 0 test problems. Judging from Table 1, the proposed algorithm is significantly better than the compared algorithms on most of the test functions. Therefore, the proposed algorithm performs best in objective space. Judging from Table 2, the proposed algorithm is significantly better than the compared algorithms except MMO-Cluster PSO. Note that MMF10–13, MMF15 and MMF15a in CEC2019 have both local and global PSs. The existence of local PSs will disturb the convergence of MMO algorithms. As shown in Table 2, the values of $1/PSP$ of these functions increase significantly. Therefore, the existing methods cannot deal with local PS well. How to avoid the interference of local PS is a potential research direction in MMO. In the following paragraph, their comprehensive performance in decision and objective space are compared.

To compare the comprehensive performance of different algorithms in both decision and objective space, the Friedman test [39] is used to rank their $1/HV$ and $1/PSP$ values. Friedman test is suitable to compute the ranking values of k algorithms over N case problems. In this paper, the ranking values of seven algorithms over 22 test problems are obtained. Since two indicators ($1/HV$ and $1/PSP$) are used, two ranking values are obtained each time. Then the mean value of these two Friedman ranking values acts as the final rank of each algorithm. Therefore, the final ranking values can reflect the performance in both decision and objective space over all the 22 test problems. The experiment is carried out 21 times. The box-plots of the final Friedman test rank value are shown in Fig. 9.

In Fig. 9, the smaller ranking value means the better comprehensive performance. The algorithm numbers 1–7 represent the seven algorithms MO_Ring_PSO_SCD, DN-NSGAII, Omni-optimizer, MMODE, TriMOEA-TA&R, MMO-Clustering PSO, and MMODE_ICD. As shown, the proposed algorithm ranks first and the competition champion in CEC

[1] MMO-Clustering PSO has not been publised. The results are obtained from the CEC2019 competition.

Table 1
1/HV values of different algorithms.

	MO_Ring_PSO_SCD [12]	DN-NSGAII [17]	Omni-optimizer [16]	MMODE [34]	TriMOEA-TA&R [18]	MMO-Clustering PSO[2]	MMODE_ICD
MMF1	1.1484 ± 0.0005 (+)	1.1498 ± 0.0017 (+)	1.1481 ± 0.0011 (+)	1.1473 ± 0.0004 (+)	0.4916 ± 1.8164 (-)	1.1453 ± 0.0004 (-)	1.1458 ± 0.0003
MMF2	1.1855 ± 0.0125 (+)	1.1947 ± 0.0292 (≈)	1.1855 ± 0.0405 (≈)	1.1759 ± 0.0135 (-)	1.1835 ± 0.0130 (≈)	1.1801 ± 0.0092 (+)	1.1765 ± 0.0062
MMF3	1.1740 ± 0.0060 (+)	1.1782 ± 0.0203 (≈)	1.1808 ± 0.0355 (≈)	1.1702 ± 0.0098 (≈)	1.1865 ± 0.0146 (+)	1.1711 ± 0.0048 (+)	1.1672 ± 0.0048
MMF4	1.8614 ± 0.0021 (+)	1.8575 ± 0.001 (+)	1.8552 ± 0.0008 (+)	1.8604 ± 0.0035 (+)	0.97,608 ± 1.6436 (-)	1.8501 ± 0.0017 (-)	1.8522 ± 0.0005
MMF5	1.1483 ± 0.0004 (+)	1.1487 ± 0.0011 (+)	1.1472 ± 0.0008 (+)	1.1474 ± 0.0006 (+)	1.1502 ± 0.0019 (+)	1.1453 ± 0.0004 (-)	1.1461 ± 0.0003
MMF6	1.1491 ± 0.0014 (+)	1.1502 ± 0.004 (+)	1.1473 ± 0.0012 (+)	1.1471 ± 0.0005 (+)	1.1502 ± 0.0034 (+)	1.1448 ± 0.0004 (-)	1.1456 ± 0.0002
MMF7	1.1485 ± 0.0008 (+)	1.1493 ± 0.0014 (+)	1.1473 ± 0.0006 (+)	1.1463 ± 0.0003 (+)	1.1905 ± 0.0838 (+)	1.1435 ± 0.0001 (-)	1.1453 ± 0.0002
MMF8	2.4050 ± 0.0159 (+)	2.3819 ± 0.0053 (+)	2.3745 ± 0.0010 (≈)	2.3904 ± 0.0176 (+)	2.3806 ± 0.0020 (+)	2.3918 ± 0.0144 (+)	2.3764 ± 0.0038
MMF9	0.1034 ± 2.6409e-05 (+)	0.1034 ± 2.7824e-05 (+)	0.1033 ± 3.1523e-05 (+)	0.10,328 ± 7.9703e-05 (+)	0.10,468 ± 9.7715e-05	0.1032 ± 1.8920e-05 (+)	0.1032 ± 1.4122e-05
MMF10	0.0797 ± 0.0005 (+)	0.0817 ± 0.0027 (+)	0.0807 ± 0.0031 (+)	0.0797 ± 0.0034 (+)	0.0781 ± 0.0001 (-)	0.0802 ± 0.0008 (+)	0.0784 ± 0.0023
MMF11	0.0690 ± 1.8815e-05 (+)	0.0689 ± 1.5362e-05 (+)	0.0689 ± 1.1921e-05 (-)	0.0689 ± 1.9638e-05 (+)	0.0696 ± 3.9369e-05 (+)	0.0689 ± 1.7768e-05 (≈)	0.0689 ± 1.2150e-05
MMF12	0.64,048 ± 0.0025 (+)	0.65,914 ± 0.0592 (+)	0.65,125 ± 0.0495 (+)	0.63,572 ± 0.0002 (+)	0.63,607 ± 0.0001 (+)	0.6381 ± 0.0012 (+)	0.6355 ± 0.0001
MMF13	0.0544 ± 2.6287e-05 (+)	0.0542 ± 1.2963e-05 (-)	0.0542 ± 5.5381e-06 (≈)	0.0542 ± 4.2456e-06 (-)	0.0550 ± 3.8073e-05 (+)	0.0543 ± 2.2184e-05 (+)	0.0542 ± 4.8097e-05
MMF14	0.34,407 ± 0.0276 (+)	0.32,675 ± 0.0124 (+)	0.33,923 ± 0.011 (+)	0.35,877 ± 0.0238 (+)	0.30,853 ± 0.0147 (≈)	0.3138 ± 0.024 (+)	0.3131 ± 0.0081
MMF15	0.24,149 ± 0.0107 (+)	0.23,385 ± 0.0131 (+)	0.23,488 ± 0.0086 (+)	0.24,445 ± 0.0081 (+)	0.21,487 ± 0.0082 (-)	0.2238 ± 0.0064 (+)	0.2211 ± 0.0056
MMF1_z	1.1482 ± 0.0004 (+)	1.1481 ± 0.0009 (+)	1.1471 ± 0.0009 (+)	1.1474 ± 0.0005 (+)	1.1495 ± 0.0017 (+)	1.1448 ± 0.0002 (-)	1.1455 ± 0.0002
MMF1_e	1.208 ± 0.0637 (+)	1.2632 ± 0.1606 (+)	1.1915 ± 0.0478 (+)	1.1508 ± 0.002 (+)	1.0601 ± 0.3229 (-)	1.1811 ± 0.0284 (+)	1.1482 ± 0.0012
MMF14_a	0.32,957 ± 0.0283 (≈)	0.31,994 ± 0.0102 (≈)	0.3283 ± 0.0079 (+)	0.36,252 ± 0.0234 (+)	0.34,163 ± 0.0185 (+)	0.3276 ± 0.021 (+)	0.3181 ± 0.0100
MMF15_a	0.24,252 ± 0.0096 (+)	0.23,175 ± 0.0109 (+)	0.23,277 ± 0.0072 (+)	0.24,087 ± 0.0055 (+)	0.21,803 ± 0.0075 (-)	0.2269 ± 0.0058 (+)	0.2242 ± 0.0054
SYM-PART simple	0.0605 ± 5.6491e-05 (+)	0.0601 ± 1.1505e-05 (+)	0.0601 ± 6.5325e-06 (+)	0.0601 ± 4.1602e-06 (+)	0.0601 ± 1.5433e-05 (+)	0.0603 ± 3.5833e-05 (+)	0.0600 ± 6.5117e06
SYM-PART rotated	0.0606 ± 8.8876e-05 (+)	0.0601 ± 1.2144e-05 (+)	0.0601 ± 5.3381e-06 (-)	0.0601 ± 7.1349e-06 (+)	0.0602 ± 1.4502e-05 (+)	0.0603 ± 4.3681e-05 (+)	0.0601 ± 5.6271e-06
Omni-test	0.0190 ± 1.7289e-05 (+)	0.0189 ± 4.1602e-07 (-)	0.0188 ± 4.7385e-07 (≈)	0.0189 ± 3.7250e-06 (+)	0.0190 ± 1.6108e-05 (+)	0.0190 ± 1.2425e-05 (+)	0.0189 ± 2.9395e-06
+/≈/-	21/1/0	17/3/2	15/5/2	19/1/2	14/2/6	15/1/6	

[2] MMO-Clustering PSO has not been publised. The results are obtained from the CEC2019 competition.

Table 2
1/PSP values of different algorithms.

	MO_Ring_PSO_SCD [12]	DN-NSGAII [17]	Omni-optimizer [16]	MMODE [34]	TriMOEA-TA&R [18]	MMO-Clustering PSO[2]	MMODE_ICD
MMF1	0.0488 ± 0.0019 (≈)	0.0969 ± 0.0145 (+)	0.0975 ± 0.0130 (+)	0.0492 ± 0.0028 (≈)	0.0735 ± 0.0108 (+)	0.0328 ± 0.0032 (-)	0.0493 ± 0.0030
MMF2	0.0465 ± 0.0147 (+)	0.1742 ± 0.1368 (+)	0.1643 ± 0.1305 (+)	0.0362 ± 0.0154 (+)	0.0931 ± 0.0574 (+)	0.0518 ± 0.0318 (+)	0.0247 ± 0.0054
MMF3	0.0335 ± 0.0099 (+)	0.1172 ± 0.0759 (+)	0.1351 ± 0.0994 (+)	0.0324 ± 0.0181 (+)	0.0871 ± 0.0275 (+)	0.0286 ± 0.0076 (+)	0.0209 ± 0.0037
MMF4	0.0273 ± 0.002 (+)	0.0771 ± 0.0137 (+)	0.0840 ± 0.0238 (+)	0.0303 ± 0.0028 (+)	0.1537 ± 0.2262 (+)	0.0129 ± 0.0006 (-)	0.0257 ± 0.0031
MMF5	0.0869 ± 0.0060 (≈)	0.1773 ± 0.0217 (+)	0.1789 ± 0.0245 (+)	0.0867 ± 0.0075 (≈)	0.1132 ± 0.0126 (+)	0.0566 ± 0.0027 (-)	0.0853 ± 0.0039
MMF6	0.0733 ± 0.0043 (≈)	0.1427 ± 0.0150 (+)	0.1523 ± 0.0185 (+)	0.0773 ± 0.0067 (+)	0.0958 ± 0.0123 (+)	0.0441 ± 0.0023 (-)	0.0713 ± 0.0043
MMF7	0.0267 ± 0.0015 (≈)	0.0553 ± 0.0151 (+)	0.0511 ± 0.0127 (+)	0.0314 ± 0.0037 (+)	0.0672 ± 0.0520 (+)	0.0127 ± 0.0010 (-)	0.0263 ± 0.0046
MMF8	0.0678 ± 0.0042 (-)	0.2799 ± 0.0911 (+)	0.3149 ± 0.1326 (+)	0.0754 ± 0.016 (-)	0.3974 ± 0.1572 (+)	0.0522 ± 0.0065 (-)	0.1303 ± 0.0352
MMF9	0.0082 ± 0.0008 (+)	0.0219 ± 0.0078 (+)	0.0316 ± 0.0269 (+)	0.0066 ± 0.0005 (+)	0.0031 ± 0.0001 (-)	0.0041 ± 0.0002 (-)	0.0047 ± 0.0003
MMF10	0.1708 ± 0.0231 (-)	1.3420 ± 2.3280 (≈)	3.1260 ± 3.3420 (≈)	4.2822 ± 2.6259 (+)	0.2014 ± 0.0001 (+)	0.1657 ± 0.0123 (-)	0.2011 ± 0.0010
MMF11	0.5092 ± 0.4740 (≈)	1.7528 ± 0.1724 (+)	1.8007 ± 0.1351 (+)	1.5996 ± 0.4411 (+)	0.2524 ± 0.0001 (+)	0.6055 ± 0.5503 (+)	0.2521 ± 0.0003
MMF12	0.4847 ± 0.3989 (≈)	1.9959 ± 0.7573 (+)	2.0419 ± 0.5808 (+)	2.0756 ± 0.1402 (+)	0.2476 ± 0.0009 (+)	0.7245 ± 0.5800 (+)	0.2474 ± 0.0003
MMF13	0.3365 ± 0.0890 (-)	0.6213 ± 0.0377 (+)	0.6195 ± 0.0526 (+)	0.5672 ± 0.0129 (+)	0.6438 ± 0.0625 (+)	0.3640 ± 0.1185 (-)	0.5370 ± 0.0025
MMF14	0.0533 ± 0.0016 (+)	0.0965 ± 0.0077 (+)	0.0890 ± 0.0066 (+)	0.0550 ± 0.0024 (≈)	0.0393 ± 0.0006 (-)	0.0265 ± 0.0005 (-)	0.0422 ± 0.0010
MMF15	0.1523 ± 0.0141 (-)	0.2434 ± 0.0875 (-)	0.3224 ± 0.1430 (≈)	0.2305 ± 0.0328 (-)	0.2725 ± 0.0004 (+)	0.1651 ± 0.0171 (-)	0.2664 ± 0.0006
MMF1_z	0.0363 ± 0.0021 (-)	0.0813 ± 0.0248	0.0754 ± 0.0155 (+)	0.0376 ± 0.0030 (-)	0.0708 ± 0.0131 (+)	0.0228 ± 0.0024 (-)	0.0387 ± 0.0039
MMF1_e	0.6038 ± 0.2020 (-)	2.0866 ± 1.7112 (-)	2.4813 ± 1.3708 (≈)	4.1128 ± 3.4048 (≈)	4.6340 ± 4.5938 (+)	0.6526 ± 0.264 (-)	2.6551 ± 1.1333
MMF14_a	0.0617 ± 0.0022 (+)	0.1164 ± 0.0084 (+)	0.1128 ± 0.0078 (+)	0.0658 ± 0.0018 (+)	0.0966 ± 0.0064 (+)	0.0313 ± 0.0007 (-)	0.0580 ± 0.0013
MMF15_a	0.1664 ± 0.0166 (-)	0.2286 ± 0.0356 (-)	0.2345 ± 0.0334 (-)	0.1999 ± 0.0162 (-)	0.2806 ± 0.0031 (+)	0.1597 ± 0.0177 (-)	0.2674 ± 0.0026
SYM-PART simple	0.1776 ± 0.0226 (+)	5.45,110 ± 2.5760 (+)	6.4608 ± 3.0139 (+)	0.0660 ± 0.0064 (+)	0.0210 ± 0.0021 (-)	0.1291 ± 0.0207 (+)	0.0427 ± 0.0055
SYM-PART rotated	0.2784 ± 0.2500 (+)	5.2742 ± 2.7404 (+)	6.3357 ± 3.7485 (+)	0.0759 ± 0.0078 (+)	2.0975 ± 1.4432	0.2441 ± 0.2737 (+)	0.0892 ± 0.0153
Omni-test	0.4279 ± 0.0954 (+)	1.5563 ± 0.2878 (+)	1.7939 ± 0.6207 (+)	0.0880 ± 0.0243 (+)	0.7547 ± 0.2166	0.3567 ± 0.0987 (+)	0.0512 ± 0.0036
+/≈/-	9/6/7	18/1/3	18/3/1	14/4/4	19/0/3	7/0/15	

Algorithm 1
MMODE_ICD.

```
//Initialize the population P
  Evaluation(P)
while the stopping criterion is not met do
  for i = 1: population_size
  //Select individuals to generate difference vector
  Adaptively select individuals considering diversity in decision and objective space
//Generate offspring
  Generate offspring O according to Eq. (2)
  Evaluation(O)
//Environmental selection
  P_combination = [P, O]
  Sort P_combination with nondominated sorting
Calculate the improved crowding distance (ICD) and special crowding distance (SCD)
P = Ratio selection (P_combination)
end for
end while
Output the final population P
```

2019 ranks second. These two algorithms are much better than the other ones.

It is not an easy task to improve the performance in both decision and objective space. Although all the compared algorithms consider the crowding degree in both spaces, the performance in one space will be deteriorated by that in another. For example, MO_Ring_PSO_SCD, MMODE, and MMO-Clustering PSO perform relatively good in decision space as shown in Table 2, but their performances in objective space turn bad. The proposed algorithm considers the convergence and diversity in decision and objective space not only in environmental selection but also in parent selection. Therefore, it has relatively good performance judging from the comprehensive ranking. The results are consistent with the mechanism analysis, which further verifies the effectiveness of the proposed algorithm.

4.4. Computational complexity analysis

In this subsection, the computation complexity of the improved crowding distance is analyzed and compared with that of the traditional one. Then, the total computation complexity of MMODE_ICD is discussed.

The differences in computation complexity between the improved crowding distance method and the traditional one lie in the individual sorting. To determine the left and right neighbors of the current individual, the considered individuals need to be sorted. Suppose that the number of individuals in Rank_i is n_i, and the number of selected individuals till Rank_i is ns_i. It is known that the computation complexity of sorting n point is $n\log(n)$. Then, the computation complexity of improved crowding distance method is $\sum_{i=1} ns_i \log(ns_i)$, and that of the traditional one is $\sum_{i=1} n_i \log(n_i)$. Since ns_i and n_i are of the same magnitude, they are of similar computation complexity. The effectiveness of the improved crowding distance is verified in Section 4.B. It can be concluded that the improved crowding distance helps achieve better performance without adding too much computation complexity.

DE is popular because of its simplicity. Although the random neighbor selection and the improved crowding distance calculation increase the complexity of DE, the total computational complexity is still accessible. The complexity of the proposed algorithm mainly lies in three aspects: the main iteration loop of DE, the selection of neighbors, and the calculation of crowding distance. Let N_{pop} denote the population size. The iteration time of the main loop is N_{pop}. In the selection of neighbors, NS neighbors are needed to generate difference vectors for each individual so the iteration time in this process is $N_{pop}(N_{pop} \cdot NS)$ where NS is the neighborhood size. In the crowding distance calculation, since the difference in each dimension of objective and decision space need to be calculated, $N_{pop} \cdot N_{var} + N_{pop} \cdot N_{obj}$ times calculation is needed. Therefore, the total computational complexity of the proposed algorithm is $N_{pop}(N_{pop}(N_{pop} \cdot NS) + N_{pop} \cdot N_{var} + N_{pop} \cdot N_{obj})$. In this paper, NS is set to 12 which is much smaller than N_{pop}. In addition, N_{var} and N_{obj} are $\frac{1}{100}$ of N_{pop}. Therefore, the total computational complexity is between $O({N_{pop}}^2)$ and $O({N_{pop}}^3)$.

5. Conclusion

This paper proposes a modified DE to solve multimodal multiobjective optimization problems. In the improved method, the parents to generate difference vectors are selected adaptively considering crowding distance in both decision and objective space. Moreover, an improved crowding distance calculation method is designed to balance the distribution between decision and objective space. The improved crowding distance can better reflect the real crowding degree. In addition, the nondominated ranking values are embedded into the crowding distance calculation so that the crowding degree in decision space is given priority in the following ranks. In environmental selection, only a certain ratio of individuals is selected adaptively thus giving chance to the potential dominated individuals. Experimental results show that the proposed method achieves best comprehensive performance.

The proposed algorithm has some disadvantages. First, its performance in decision space needs to be further improved. Second, its computational complexity is relatively high. Third, the parameters are not easy to set for different kinds of problems. These disadvantages will be overcome in the future.

There are many future works to do in multimodal multiobjective optimization. First, we will study MMOPs with both local and global PSs [21]. Second, the parameter adaptive mechanism depending on the current status of the population needs to be studied. Third, the improved crowding distance and dominance [40] will be combined to solve multimodal many-objective optimization problems.

Declaration of Competing Interest

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

Acknowledgment

We acknowledge financial support by the National Natural Science Foundation of China (61922072, 61876169, 61976237, 61806179 and 61873292), Key R&D and Promotion Projects in Henan Province (212102210510), the Key Scientific Research Projects in Colleges and Universities of Henan Province (Grant No.19A120014) and Scholarship under the State Scholarship Fund by the China Scholarship Council.

Author statement

Caitong Yue programmed the codes and wrote the manuscript.

P. N. Suganthan proposed the idea of the algorithm supervised programming and writing.

Jing Liang developed the idea of multimodal multiobjective optimization and supervised programming and writing.

Boyang Qu helped to program the algorithms.

Kunjie Yu helped to edit and proofread the manuscript.

Yongsheng Zhu discussed and helped to proofread the manuscript.

Li Yan discussed and helped to proofread the manuscript.

All authors read and contributed to the manuscript.

References

[1] A. Konak, D.W. Coit, A.E. Smith, Multi-objective optimization using genetic algorithms: a tutorial, Reliab. Eng. Syst. Saf. 91 (9) (2006) 992–1007.

[2] J.J. Durillo, J.A.N. Garc, A.J. Nebro, C.A. Coello, F. Luna, E. Alba, Multi-objective particle swarm optimizers: an experimental comparison, in: Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, 2009, pp. 495–509.

[3] A. Zhou, B. Qu, H. Li, S. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evol. Comput. 1 (1) (2011) 32–49.

[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[5] Y. Guo, H. Yang, M. Chen, J. Cheng, D. Gong, Ensemble prediction-based dynamic robust multi-objective optimization methods, Swarm Evol. Comput. 48 (2019) (2019) 156–171.

[6] Y. Guo, X. Zhang, D. Gong, Z. Zhang, J. Yang, Novel interactive preference-based multi-objective evolutionary optimization for bolt supporting networks, IEEE Trans. Evol. Comput. 24 (4) (2019) 750–764.

[7] B. Qu, P.N. Suganthan, J.J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.

[8] X. Li, M.G. Epitropakis, K. Deb, A. Engelbrecht, Seeking multiple solutions: an updated survey on niching methods and their applications, IEEE Trans. Evol. Comput. 21 (4) (2017) 518–538.

[9] R.Z. Shi, W. Lin, Q.Z. Lin, Z.X. Zhu, J.Y. Chen, Multimodal multi-objective optimization using a density-based one-by-one update strategy, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2019, pp. 295–301.

[10] K. Maity, R. Sengupta, S. Sha, MM-NAEMO: multimodal neighborhood-sensitive archived evolutionary many-objective optimization algorithm, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2019, pp. 286–294.

[11] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput. 14 (1) (2010) 150–169.

[12] C. Yue, B. Qu, J.J. Liang, A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems, IEEE Trans. Evol. Comput. 22 (5) (2018) 805–817.

[13] C. Yue, B. Qu, K. Yu, J.J. Liang, X. Li, A novel scalable test problem suite for multimodal multiobjective optimization, Swarm Evol. Comput. 48 (2019) (2019) 62–71.

[14] Y. Liu, H. Ishibuchi, G.G. Yen, Y. Nojima, N. Masuyama, Handling imbalance between convergence and diversity in the decision space in evolutionary multi-modal multi-objective optimization, IEEE Trans. Evol. Comput. 24 (3) (2019) 551–565.

[15] C. Yue, J.J. Liang, B. Qu, K. Yu, H. Song, Multimodal multiobjective optimization in feature selection, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2019, pp. 302–309.

[16] K. Deb, S. Tiwari, Omni-optimizer: a procedure for single and multi-objective optimization, in: Proceedings of the International Conference on Evolutionary Multi--Criterion Optimization, 2005, pp. 47–61.

[17] J.J. Liang, C. Yue, B. Qu, Multimodal multi-objective optimization: a preliminary study, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2016, pp. 2454–2461.

[18] Y. Liu, G.G. Yen, D. Gong, A multi-modal multi-objective evolutionary algorithm using two-archive and recombination strategies, IEEE Trans. Evol. Comput. 23 (4) (2018) 660–674.

[19] J.J. Liang, B. Qu, D. Gong, C. Yue, Problem Definitions and Evaluation Criteria For the CEC 2019 Special Session On Multimodal Multiobjective Optimization, Zhengzhou University, 2019.

[20] K. Deb, Multi-objective genetic algorithms: problem difficulties and construction of test problems, Evol. Comput. 7 (3) (1999) 205–230.

[21] J.J. Liang, P.N. Suganthan, B.Y. Qu, D.W. Gong, C.T. Yue, Definitions and Evaluation Criteria for the CEC 2020 Special Session on Multimodal Multiobjective Optimization, Zhengzhou University and Nanyang Technological University, 2020.

[22] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2010) 4–31.

[23] K.A. Qin, L.V. Huang, Suganthan, N. P., Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.

[24] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (1) (2009) 108–132.

[25] P.N. Suganthan, Particle swarm optimiser with neighbourhood operator, IEEE Congr. Evol. Comput. (1999) 1958–1962.

[26] B. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (3) (2013) 387–402.

[27] Y. Wang, Z.L. Yang, Y.J. Guo, J.C. Zhu, X.D. Zhu, A novel multi-objective competitive swarm optimization algorithm for multi-modal multiobjective problems, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2019, pp. 271–278.

[28] A. Basak, S. Das, K.C. Tan, Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection, IEEE Trans. Evol. Comput. 17 (5) (2013) 666–685.

[29] S. Hui, P.N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, IEEE Trans. Cybern. 46 (1) (2015) 64–74.

[30] Z. Li, L. Shi, C. Yue, Z. Shang, B. Qu, Differential evolution based on reinforcement learning with fitness ranking for solving multimodal multiobjective problems, Swarm Evol. Comput. 49 (2019) (2019) 234–244.

[31] W. Zhang, G. Li, W. Zhang, J. Liang, G.G. Yen, A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization, Swarm Evol. Comput. 50 (2019) (2019) 100569 1-13.

[32] Q. Fan, X. Yan, Solving multimodal multiobjective problems through zoning search, IEEE Trans. Syst. Man Cybern. Syst. (2019) 1–12, doi:10.1109/TSMC.2019.2944338.

[33] R. Tanabe, H. Ishibuchi, A framework to handle multi-modal multi-objective optimization in decomposition-based evolutionary algorithms, IEEE Trans. Evol. Comput. 24 (4) (2019) 720–734.

[34] J.J. Liang, W.W. Xu, C.T. Yue, K.J. Yu, H. Song, O.D. Crisalle, B.Y. Qu, Multimodal multiobjective optimization with diffierential evolution, Swarm Evol. Comput. 44 (2019) (2018) 1028–1059.

[35] J.J. Liang, Q. Guo, C. Yue, B. Qu, K. Yu, A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems, in: Proceedings of the International Conference on Swarm Intelligence, 2018, pp. 550–560.

[36] Y. Hu, J. Wang, J.J. Liang, K. Yu, H. Song, Q. Guo, C. Yue, Y. Wang, A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm, Sci. China Inf. Sci. 62 (070206) (2019) 1–17.

[37] H. Xia, J. Zhuang, D. Yu, Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization, IEEE Trans. Cybern. 44 (3) (2013) 378–393.

[38] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G.D. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.

[39] M. Pal, S. Bandyopadhyay, Differential evolution for multi-modal multi-objective problems, Genet. Evol. Comput. Conf. Companion (2019) 1399–1406.

[40] W. Qiu, J. Zhu, G. Wu, M. Fan, P.N. Suganthan, Evolutionary many-objective algorithm based on fractional dominance relation and improved objective space decomposition strategy, Swarm Evol. Comput. 60 (2021) 100776 1-16.