Two-Stage Double Niched Evolution Strategy for Multimodal Multiobjective Optimization

Kai Zhang, Chaonan Shen, Gary G. Yen, *Fellow, IEEE*, Zhiwei Xu, *Graduate Student Member, IEEE*, and Juanjuan He, *Member, IEEE*

***Abstract*—In recent years, numerous efficient and effective multimodal multiobjective evolutionary algorithms (MMOEAs) have been developed to search for multiple equivalent sets of Pareto optimal solutions simultaneously. However, some of the MMOEAs prefer convergent individuals over diversified individuals to construct the mating pool, and the individuals with slightly better decision space distribution may be replaced by significantly better objective space distribution. Therefore, the diversity in the decision space may become deteriorated, in spite of the decision and objective diversities have been taken into account simultaneously in most MMOEAs. Because the Pareto optimal subsets may have various shapes and locations in the decision space, it is very difficult to drive the individuals converged to every Pareto subregion with a uniform density. Some of the Pareto subregions may be overly crowded, while others are rather sparsely distributed. Consequently, many existing MMOEAs obtain Pareto subregions with imbalanced density. In this article, we present a two-stage double niched evolution strategy, namely DN-MMOES, to search for the equivalent global Pareto optimal solutions which can address the above challenges effectively and efficiently. The proposed DN-MMOES solves the multimodal multiobjective optimization problem (MMOP) in two stages. The first stage adopts the niching strategy in the decision space, while the second stage adapts double niching strategy in both spaces. Moreover, an effective decision density self-adaptive strategy is designed for improving the imbalanced decision space density. The proposed algorithm is compared against eight state-of-the-art MMOEAs. The inverted generational distance union (IGDunion) performance indicator is proposed to fairly compare two competing MMOEAs as a whole. The experimental results show that DN-MMOES provides a better performance to search for the complete Pareto Subsets and Pareto Front on IDMP and CEC 2019 MMOPs test suite.**

***Index Terms*—Evolution strategy, multimodal multiobjective evolutionary algorithm (MMOEA), multimodal multiobjective optimization problem (MMOP).**

Manuscript received August 24, 2020; revised November 28, 2020 and January 17, 2021; accepted March 3, 2021. Date of publication March 8, 2021; date of current version July 30, 2021. This work was supported by the National Natural Science Foundation of China under Grant 61702383, Grant U1803262, and Grant 61602350. *(Corresponding author: Gary G. Yen.)*

Kai Zhang and Juanjuan He are with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China (e-mail: zhangkai@wust.edu.cn; hejuanjuan@wust.edu.cn).

Chaonan Shen and Zhiwei Xu are with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-Time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China (e-mail: 449777215@qq.com; kenxucn95@gmail.com).

Gary G. Yen is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: gyen@okstate.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2021.3064508.

Digital Object Identifier 10.1109/TEVC.2021.3064508

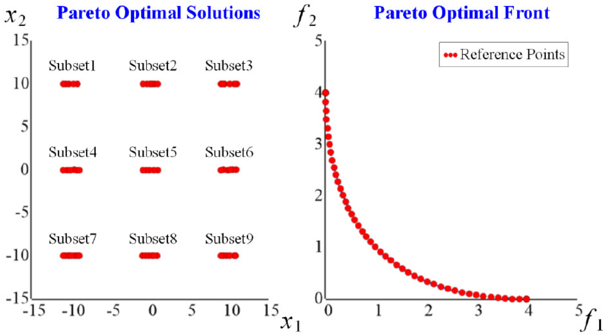


Fig. 1. Nine different Pareto subsets (i.e., Subset1–Subset9) map to the same Pareto front. The Pareto optimal solutions are the union of the subsets from Subset1 to Subset9.

I. Introduction

In RECENT years, many multiobjectiv evolutionary algorithms (MOEAs) have been adapted to tackle multimodal multiobjective optimization problems (MMOPs) involving multiple groups of optimal solutions to be found simultaneously [1]–[6]. Additional multimodal MOEAs (MMOEAs) have been carefully crafted to solve real-world applications [7]–[9], such as hybrid rocket engine design [10], and spatiotemporal patterns mining [11], to name a few.

The mathematical model of a multiobjective optimization problem (MOP) can be formulated as follows:

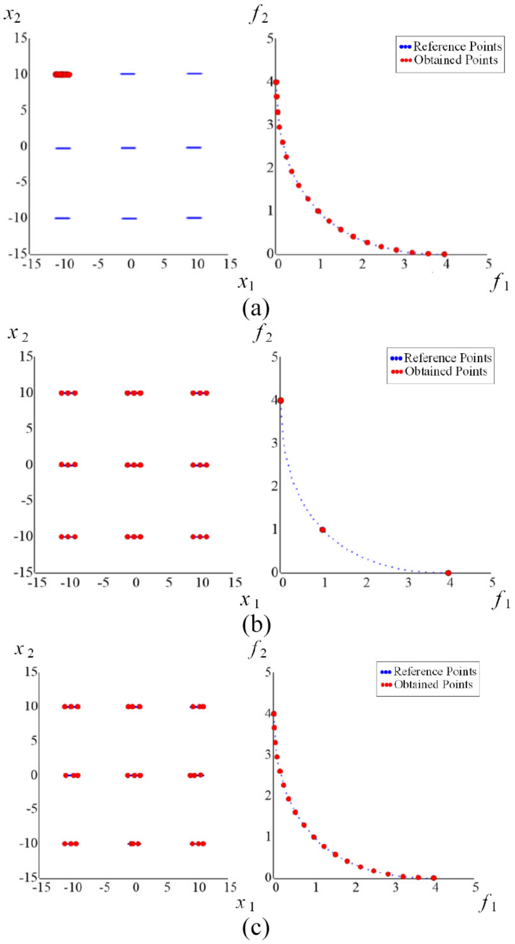
$$\min f(x) = \min\left[f_1(x), f_2(x), \ldots, f_M(x)\right]^T \tag{1}$$

where $x = [x_1, x_2, \ldots, x_N]^T \in \Omega$, x consists of N decision variables. $f(x)$ consists of M objective functions, $f_i(x) \in R^M$, $i = 1, \ldots, M$. R^M denotes the objective space.

The MOPs mandate a search for a set of tradeoff optimal solutions called the Pareto set (PS). The corresponding objective vectors of PS is called the Pareto optimal front (PF). The MMOP is a special class of MOPs which possesses more than one equivalent set of Pareto optimal solutions or at least more than one local Pareto optimal solution for any point on the PF.

Fig. 1 shows an example of MMOP given nine distinct Pareto Subsets in the decision space (i.e., Subset1 to Subset9 of PS). All the Pareto subsets are corresponding to the same Pareto front in the objective space (i.e., PF).

Finding the Pareto optimal solutions for a given MMOP is quite different from that of the traditional MOPs, which dictates to satisfy three conditions simultaneously [12]:



(a)

(b)

(c)

Fig. 2. Obtained three different results on SYM-PART1 instance with 27 individuals. (a) Best objective space distribution, but poor decision space distribution. (b) Best decision space distribution, but poor objective space distribution. (c) Balance decision and objective spaces distribution.

1) well-converged; 2) well-distributed in the objective space; and 3) well-distributed in the decision space.

For example, let the population size be 27 for SYM-PART1 test instance, which has nine equivalent groups of Pareto optimal solutions [13]. If the deployed MOEA satisfies conditions 1) and 2), the resulted PS and Pareto front (PF) could appear like Fig. 2(a). Although the obtained 27 solutions are well-distributed and well-converged in the objective space, only single Pareto subset is identified in the decision space. If the underlying MOEA focuses solely on conditions 1) and 3), the obtained PS/PF could appear as Fig. 2(b). The obtained 27 solutions could have located in all nine Pareto subsets, however, the corresponding Pareto front comprises of only three distinct solutions in the objective space. Actually, the ideal MMOEA should satisfy all three conditions simultaneously, and the resulted Pareto optimal solutions would be well-distributed and well-converged in both the objective and decision spaces, as can be seen from Fig. 2(c).

Existing MMOEAs could be broadly classified into three different categories, including Pareto-based approaches [13]–[20], decomposition-based approaches [21], [22], and indicator-based approaches [23]–[25]. The Pareto-based MMOEAs select converged solutions with Pareto dominance principle, and then enhance the decision space diversity for solving MMOPs. On the other hand, the decomposition-based approaches adopt the notion to decompose a given MMOPs into a set of single-objective subproblems, which integrates decision space diversity for finding multimodal optimal solutions. Finally, the indicator-based approaches adopt performance indicators to guide the search process in both the decision and objective spaces.

Unfortunately, most of the existing MMOEAs often face a number of challenges to handle three conditions simultaneously: 1) well-converged; 2) well-distributed in the objective space; and 3) well-distributed in the decision space. Some MMOEAs select high-quality individuals into the mating pool of the offspring, and "convergent individuals" are treated a higher priority than that of "diversity individuals." However, when too many well-converged, but poor diversity individuals are selected into the mating pool, the diversity will be inadvertently deteriorated in the decision space. In addition, many existing MMOEAs tend to obtain Pareto subregions with *imbalanced density*, i.e., some of the Pareto subregions are overly crowded, while others are rather sparsely occupied. Moreover, although diversities in the decision and objective spaces have been taken into account simultaneously in most MMOEAs, some individuals with significantly better objective space distribution may replace the individuals with slightly better decision space distribution. Consequently, it is impossible to improve the local decision space diversity anymore without neighboring individuals. Since the MMOP involves a many-to-one unidirectional mapping from the decision space to the objective space, traditional diversity strategies may lead to decision space diversity deterioration, including those designs exploiting diversity in the objective space first and then diversity in the decision space, or diversity in both spaces at the same time from the beginning of the evolutionary process.

In this article, a novel double niched evolution strategy is proposed for solving MMOPs, namely DN-MMOES. The main contributions of this study are threefold.

First, the proposed algorithm solves the MMOP in two stages. The first stage adopts the niching strategy in the decision space, whose goal is to find well-distributed and well-converged solutions in the decision space. On the other hand, the second stage invokes double niching strategy in both the objective and decision spaces, whose goal is to improve the diversity in the objective space.

Second, the evolution strategy is adapted in the proposed algorithm. Because new mutated candidate solution needs only to compare with the original solution, no mating pool is required for offspring generation, which can avoid the negative influence by choosing more convergent individuals.

Third, an effective decision density self-adaptive strategy is designed herein to improve the imbalanced density of the different Pareto subregions in the decision space.

The remaining parts of this article are presented as follows. A comprehensive review of related works is analyzed to inspire the motivation of the proposed work in Section II. Section III provides the details of the proposed DN-MMOES thoroughly. The experimental results and relevant observations on IDMP and CEC 2019 MMOP competition benchmark functions are given in Section IV. Finally, conclusions and future research directions are outlined in Section V.

II. Related Works and Motivation

The design objective of MMOEAs is quite different from that of the generic MOEAs, which demands the obtained solutions satisfy three conditions simultaneously, including well-converged, well-distributed in both the objective and decision spaces. In recent years, numerous MMOEAs have been proposed for solving MMOPs with different strategies, which can be roughly classified into three different categories, including Pareto-based approaches, decomposition-based approaches, and indicator-based approaches.

A. Multimodal Multiobjective Optimization Approaches

1) Pareto-Based Approaches: Deb and Tiwari [14] proposed the Omnioptimizer that utilized crowding distances in both the decision and objective spaces to search for multiple sets of Pareto optimal solutions. Liu *et al.* [15] suggested a double-niched evolutionary algorithm (DNEA), which adopted the sharing functions to maintain good individual diversity. Liang *et al.* [16] replaced the crowding distance in the objective space by the crowding distance in the decision space, and presented an innovative decision space-based niching NSGAII (DN-NSGA-II) for solving MMOPs.

Kim *et al.* [17] incorporated two archives to improve the performance of the strength Pareto evolutionary algorithm 2 (SPEA2+), which updated the archives with density qualities. Liu *et al.* [18], on the other hand, designed an MMOEA with two-archive and recombination strategies (called TriMOEA-TA&R), which achieved multiple equivalent diversified and converged Pareto optimal solutions at the same time.

Liu *et al.* [19] transformed the decision space density as the selection criterion and presented a convergence-penalized density method (CPDEA) for solving MMOPs. Furthermore, Yue *et al.* [20] integrated special crowding distance into multiobjective particle swarm optimization algorithm (MO-Ring-PSO-SCD) to search for multiple equivalent Pareto optimal solutions at once.

2) Decomposition-Based Approaches: The MMOEAs in this category adopt a group of uniformly distributed reference points or vectors to decompose the MMOP into a group of subproblems, such as MOEA based on decomposition (MOEA/D) [21]. Hu and Ishibuchi [22] presented a decomposition-based method, which integrated the diversity maintenance mechanism in the decision space. In addition, Tanabe and Ishibuchi [23] proposed a decomposition-based algorithm, which adopted dynamic population size and integrated addition and deletion operators within MOEA/D (MOEA/D-AD).

3) Indicator-Based Approaches: The indicator-based MMOEAs prefer to guide the evolutionary progress along reference vectors with chosen performance indicators. Ulrich *et al.* [24] presented a hypervolume indicator-based MMOEA, which adopted Solow–Polasky diversity technique in both the decision and objective spaces. Alternatively, Ishibuchi *et al.* [25] designed two-objective solution set to optimize hypervolume indicator and decision diversity. Tanabe and Ishibuchi [26] presented a niching indicator-based algorithm for multimodal many-objective optimizer (called NIMMO) for many-objective optimization problems.

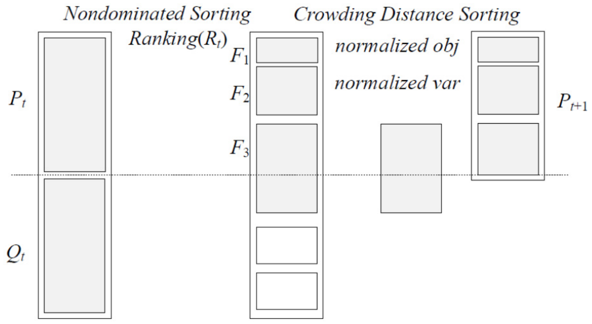


Fig. 3. Procedure of Omnioptimizer algorithm [14].

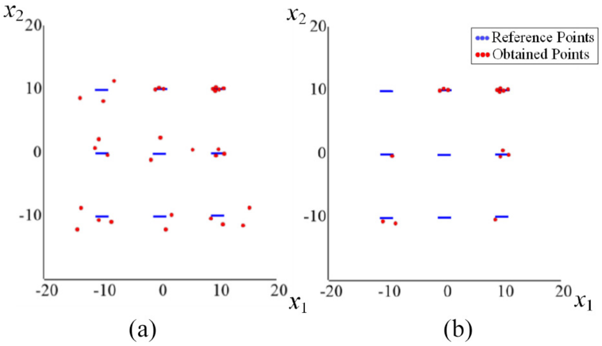


Fig. 4. (a) Thirty candidate individuals distributed near every Pareto subregion. (b) Selected fifteen individuals with better convergent quality.

B. Motivation

Even appreciable progress has been made in the state-of-the-art designs, unfortunately, existing MMOEAs still face the following challenges [12], [19].

1) Diversity Deterioration in the Decision Space: In order to generate high-quality offspring, some MMOEAs choose the better convergent solutions at first. Then, the diversity quality in the decision and objective spaces would be taken into account next. During the selection process, those well-converged, but poorly diversified solutions are preferable to those poorly converged, but well-diversified solutions. As can be seen from Fig. 3, Omnioptimizer, as a representative, places a high priority in selecting convergent solutions to the next generation in the nondominated sorting step.

However, some of the convergent solutions may have poor diversity quality. When too many overly crowded convergent solutions are chosen, the decision space diversity would become deteriorated. For example, Fig. 4(a) shows 30 individuals for SYM-PART1 instance, which are distributed near every Pareto subregion. On the other hand, Fig. 4(b), produced by Omnioptimizer [14], shows the chosen 15 individuals with better convergent priority criterion. Obviously, many potential Pareto subregions are missing out. As a result, the distribution in the decision space become worsen.

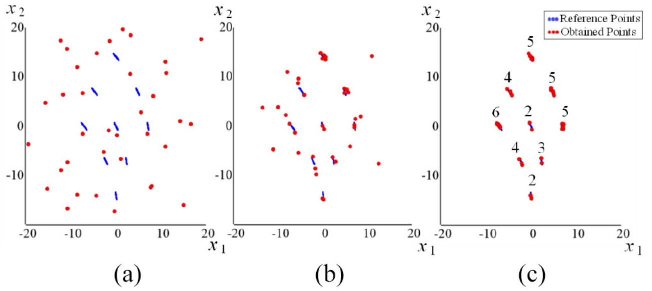


Fig. 5. (a) Population initialization on SYM-PART3 test instance. (b) Converging to the Pareto subregions. (c) Obtained Pareto optimal solutions with imbalanced densities.

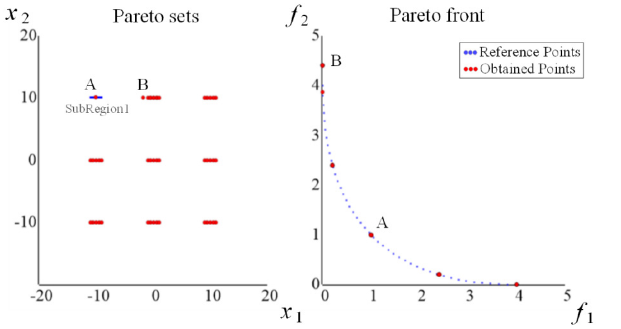


Fig. 6. Example of objective space distribution versus decision space distribution.

In order to solve the degradation of diversity in the decision space, many techniques have been proposed. The MO-Ring-PSO-SCD algorithm uses ring structure, which applies a local Pareto sorting for selection operation. CPDEA algorithm takes into account both the deterioration of diversity of solution set in the decision space and the imbalanced density, so it has achieved a competitive performance.

2) Imbalanced Density in the Pareto Subregions: Because the Pareto optimal subsets may have various shapes and locations in the decision space, it is very difficult to drive the individuals converged to every Pareto subregion with a uniform density. Some of the Pareto subregions may be overly crowded, while others are rather sparsely distributed. It is very important to improve the imbalanced densities of Pareto subregions in the decision space. As can be seen from Fig. 5, the SYM-PART3 has nine rotated and transformed Pareto-optimal subsets [13]. Although the initialized 36 individuals are randomly distributed uniformly as displayed in Fig. 5(a), some MMOEAs are very easy to obtain imbalanced Pareto subregions, as exemplified from Fig. 5(b) and (c). As shown in Fig. 5(c), some of the subregions have 5 or 6 individuals, while other subregions have only 2 or 3 individuals. The experimental results are produced by the first stage of the proposed DN-MMOES.

3) Distribution Tradeoff Between Decision and Objective Spaces: In spite of the decision and objective diversities have been taken into account simultaneously in most MMOEAs, the decision space diversity should not be treated equally as objective space diversity. During the lengthy evolutionary process, some individuals with significantly better objective space distribution may replace the individuals with slightly better decision space distribution. However, it is extremely difficult to improve the diversity anymore for some local decision regions where few or none neighboring individuals remain.

For example, Fig. 6 shows 42 individuals for SYM-PART1 instance, which are distributed near every Pareto subregion. Individual A is the only candidate solution in subregion1, which has good decision space quality, but poor objective space quality. On the contrary, individual B located near subregion2, which has poor decision space quality, but good objective space quality. Because individual B is a boundary solution, the normalized objective value will be infinite. Most of the algorithms would choose individual B with a higher priority over individual A, however, the subregion1 would be inadvertently missing out without maintaining individual A. The experimental results are produced by the proposed DN-MMOES with grouping individuals.

III. Proposed Algorithm

In this article, we present a two-stage double niched evolution strategy for solving MMOPs, named DN-MMOES, which can address these challenges listed in Section II-B efficiently and effectively. For the diversity deterioration challenge, in the proposed evolution strategy, every new mutated candidate solution needs only to compare with the corresponding original solution, which can avoid the negative influence from mating pool. For the distribution tradeoff challenge, the proposed DN-MMOES solves the MMOP in two stages. The first stage adopts the niching strategy in the decision space, while the second stage incorporates the double niching strategy in both the objective and decision spaces. The seamlessly integrated two-stage strategy can improve the decision and objective distributions simultaneously. In addition, an effective decision density self-adaptive strategy is designed to move the overcrowded individuals to the sparse region, which is helpful to tackle the challenge of imbalanced density in the Pareto subregions.

A. Decision Density Self-Adaptive Strategy

Given an MMOP, the Pareto optimal subsets may have uneven distribution, and as a result, a uniform initialization of population may find imbalanced Pareto subsets. Some of the Pareto subregions are overly crowded, while others could be rather sparse. The decision density self-adaptive strategy is designed to improve the decision densities in imbalanced Pareto subregions.

Our algorithm adopts the density estimation method of DBSCAN [27] and OPTICS [28], which count the number of nearest neighbors in a given radius. Let $P_t^{(i)}$ be the ith individual in Population P_t at the tth generation, and $x_n^{(i)}$ is the nth decision variable of the individual $P_t^{(i)}$. Let $x_{\max}^n$ and $x_{\min}^n$ be the maximum and minimum decision variables in the nth dimension. The radius value is the δ percent of the Euclidean distance between the maximum and minimum decision vectors, as shown in lines 1–7 of Algorithm 1.

Let variable *dist* be the distance between two individuals, and variable NeighborCount[i] be the number of nearest

Algorithm 1 Decision Density Self-Adaptive Strategy

Input: Original Population

Output: New Population with Density Self-Adaptive Strategy

```
1:  $Radius = 0;$
2:  for $n = 1$ to $N$
3:      $x^n_{max} = \max_i(x_n^{(i)})\ ; i \in [1, P]$
4:      $x^n_{min} = \min_i(x_n^{(i)})\ ; i \in [1, P]$
5:      $Radius = Radius + (x^n_{max} - x^n_{min})^2;$
6:  end for
7:  $Radius = sqrt(Radius)/\delta;$
8:  for $i = 1$ to $P$
9:      $NeighborCount[i] = 0;$
10:     for $j = 1$ to $P$ do
11:         $dist = sqrt(\sum_{n=1}^{N} (x_n^{(i)} - x_n^{(j)})^2);$
12:         if $dist < Radius$ then
13:             $NeighborCount[i] = NeighborCount[i] + 1;$
14:         end if
15:     end for
16: end for
17: $MaxDenIdx = \max_i(NeighborCount[i]); i \in [1, P]$
18: $MinDenIdx = \min_i(NeighborCount[i]); i \in PS$
19: $NearDist_{MaxDenIdx} = \min_{j \in P}(\sqrt{\sum_{n=1}^{N} (x_n^{(MaxDenIdx)} - x_n^{(j)})^2});$
20: $NearDist_{MinDenIdx} = \min_{j \in P}(\sqrt{\sum_{n=1}^{N} (x_n^{(MinDenIdx)} - x_n^{(j)})^2});$
21: if $\frac{NeighborCount[MaxDenIdx]}{NeighborCount[MinDenIdx]} > 1$ and $\frac{NearDist_{MaxDenIdx}}{NearDist_{MinDenIdx}} < 1$
22:     $P_t^{(MaxDenIdx)} = GaussMutation\left(P_t^{(MinDenIdx)}\right)$
23: end if
```

neighbors of the ith individual. For each individual, if the *dist* is smaller than the *radius*, then accumulate the variable NeighborCount[i], as shown in lines 8–16 of Algorithm 1.

Let variable *MaxDenIdx* be the index of the maximum density individual, which has the largest number of nearest neighbors NeighborCount[i], and variable MinDenIdx be the index of the minimum density individual, which has the smallest number of nearest neighbors NeighborCount[i]. Then, the maximum density index *MaxDenIdx* would be selected from the whole population, and the minimum density index MinDenIdx would be chosen from the nondominated solutions. Furthermore, the maximum density index, MaxDenIdx, and the minimum density index, MinDenIdx, would be chosen for further self-adaptive adjusting. In the whole population, the individual with minimum density may have poor convergent quality. To avoid convergence deterioration, the MaxDenIdx individuals would be selected from the whole population, while the MinDenIdx individuals would be chosen from the nondominated solutions. Finally, the maximum density individual should be taken place with mutated minimum density individuals, if and only if the NeighborCount is greater and the NearDist is smaller than those of the minimum density individual, as shown in lines 17–23 of Algorithm 1. Moreover, the quality of the new mutated individual would be evaluated in the next generation.

For example, in Fig. 7, individual p has the maximum number of neighbors, while individual q has the minimum number of neighbors within a given radius. The NearDist(p) refers to the distance between p and its nearest neighbor. In Fig. 7(a), the value of NearDist(p) is smaller than the value of NearDist(q), which means the density of p is higher than that of q. Clearly, the distribution quality of individual p needs to be improved. However, in Fig. 7(b), the value of NearDist(p) is greater than the value of NearDist(q), which implies the distribution of p maybe sparser than that of q. However, the density of p is not higher than that of q, and the individual p does not need to be changed.

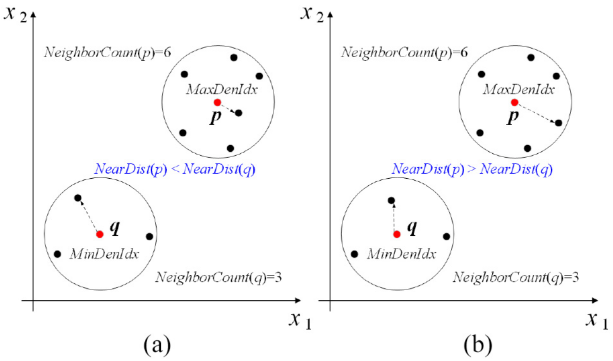


Fig. 7. (a) Individual p has the maximum number of neighbors, and smaller distance from nearest neighbor than that of individual q. The distribution quality of individual p needs to be improved. (b) Individual p has the maximum number of neighbors, but with a greater distance from nearest neighbor than that of individual q. The individual p does not need to be changed.

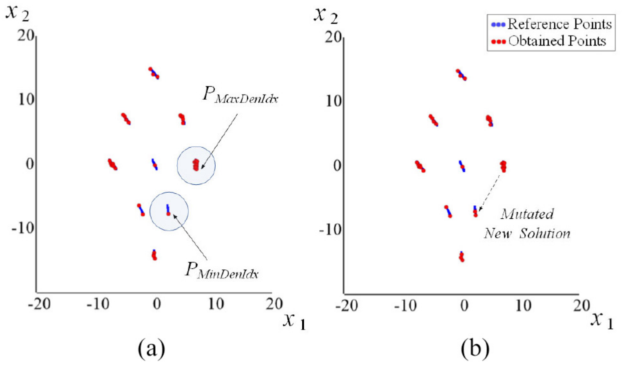


Fig. 8. Example of the decision density self-adaptive strategy.

The computational complexity of the decision density self-adaptive strategy is $O(NP^2)$, where P is the population size and N is the number of decision variables. The pseudocode of the proposed decision density self-adaptive strategy is given in Algorithm 1.

Fig. 8 shows an example of the decision density self-adaptive strategy. As can be seen from Fig. 8(a), our algorithm first identifies the individuals $P_{\text{MaxDenIdx}}$ and $P_{\text{MinDenIdx}}$, which are located in the most overcrowded and sparsest regions, respectively. Because the NeighborCount and NearDist values of the maximum density individual are greater than those of the minimum density individual, our algorithm would replace the most overcrowded individual $P_{\text{MaxDenIdx}}$ with the sparsest mutated individual $P_{\text{MinDenIdx}}$. As shown in Fig. 8(b), the decision density has been improved appreciably and efficiently.

B. Simulated Isotropic Magnetic Particles Niching Strategy

The magnetic force is one of the fundamental forces of nature. When isotropic magnetic particles are close to each

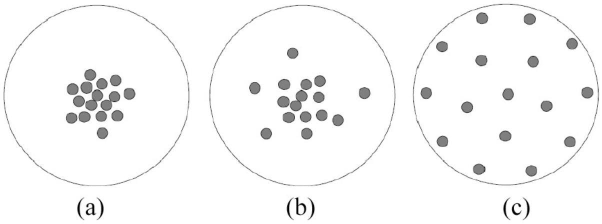


Fig. 9. Repulsive force drives isotropic magnetic particles moving in opposite directions until the point of equilibrium. (a) Initialization. (b) Moves under repulsion. (c) Keeps equilibrium.

Algorithm 2 MaximumExtensionDistanceX

```
Input: Individual $P_t^{(i)}$
Output: MEDx value of $P_t^{(i)}$
TotalDistX = 0;
NearDistX = +∞;
for $j = 1$ to $P$
    if $j = i$
        continue;
    end if
    Distance = 0;
    for $n = 1$ to $N$
        $Distance = Distance + \left(x_n^{(i)} - x_n^{(j)}\right)^2$;
    end for
    Distance = sqrt(Distance)
    TotalDistX = TotalDistX + Distance;
    if Distance < NearDistX then NearDistX = Distance;
end for
MEDx = TotalDistX × NearDistX;
```

other, the repulsive force will push them moving in opposite directions. Given a limited boundary, isotropic magnetic particles will reach a point of equilibrium [29], [30], that all the particles maintain similar distances from each other, as shown in Fig. 9.

In [31] and [32], a novel niching technology was designed to simulate the movements of isotropic magnetic particles, which drive individuals to preserve uniform distance from each other and spread to the whole Pareto front. The niching technology not only does not need any niching parameter, but also can obtain good distribution with much less population size.

Let $f_m^{(i)}$ be the mth objective value of the individual $P_t^{(i)}$. The maximum extension distance, MED, in the objective space is defined in (2).

$$\mathrm{MED}\left(P_t^{(i)}\right) = \mathrm{TotalDist}\left(P_t^{(i)}\right) \times \mathrm{NearDist}\left(P_t^{(i)}\right) \tag{2}$$

where

$$\mathrm{TotalDist}\left(P_t^{(i)}\right) = \sum_{j=1}^{P}\left(\sqrt{\sum_{m=1}^{M}\left(f_m^{(i)} - f_m^{(j)}\right)^2}\right)$$

$$\mathrm{NearDist}\left(P_t^{(i)}\right) = \min_{j,j\neq i}\sqrt{\sum_{m=1}^{M}\left(f_m^{(i)} - f_m^{(j)}\right)^2}.$$

In this algorithm, the niching technique is further extended to the decision space, and the maximum extension distance in the decision space (MEDx) is defined according to (3).

$$\mathrm{MEDx}\left(P_t^{(i)}\right) = \mathrm{TotalDistX}\left(P_t^{(i)}\right) \times \mathrm{NearDistX}\left(P_t^{(i)}\right) \tag{3}$$

where

$$\mathrm{TotalDistX}\left(P_t^{(i)}\right) = \sum_{j=1}^{P}\left(\sqrt{\sum_{n=1}^{N}\left(x_n^{(i)} - x_n^{(j)}\right)^2}\right)$$

$$\mathrm{NearDistX}\left(P_t^{(i)}\right) = \min_{j,j\neq i}\sqrt{\sum_{n=1}^{N}\left(x_n^{(i)} - x_n^{(j)}\right)^2}.$$

In (3), $P_t^{(i)}$ refers to the ith individual in Population P_t at the tth generation, and $x_n^{(i)}$ is the nth decision variable of $P_t^{(i)}$. TotalDistX($P_t^{(i)}$) calculates the summation of Euclidean distances between $P_t^{(i)}$ and $P_t^{(j)}$in the decision space. In order to explore the entire decision space, our algorithm prefers a greater value of TotalDistX($P_t^{(i)}$), which implies the individual $P_t^{(i)}$ moving away from the other individuals in the decision space. The NearDistX($P_t^{(i)}$) calculates the minimum Euclidean distance between $P_t^{(i)}$ and $P_t^{(j)}$. A greater value of NearDistX($P_t^{(i)}$) refers to a better individual distance.

The value of MEDx($P_t^{(i)}$) is the result obtained by multiplying TotalDistX($P_t^{(i)}$) and NearDistX($P_t^{(i)}$). The greater MEDx($P_t^{(i)}$) means an individual become farther away from the other individuals which may explore the new decision region.

The computational complexity of the *MEDx* is $O(NP)$, where N is the number of decision variables and P is the population size. The pseudocode of the *MEDx* is given in Algorithm 2.

C. Overall Algorithm

To reduce the negative influence of excessive objective space diversity on decision space diversity, the proposed DN-MMOES solves the MMOP in two stages. The first stage adopts the niching strategy only in the decision space, which aims at finding well-converged solutions which maintain good distribution in the decision space. In addition, the decision density self-adaptive strategy is integrated to improve the imbalanced densities within different Pareto subregions. The second stage of the algorithm adopts double niching strategy in both the decision and objective spaces, whose goal is to improve the objective and decision distributions. The computational complexity of the proposed DN-MMOES is $O(NP^2)$, where N is the number of decision variables and P is the population size. The pseudocode of DN-MMOES is given in Algorithm 3.

For each generation, the proposed evolution strategy uses the Gaussian mutation [33] to generate new candidate individuals, which create new offspring x_i' by adding a random value from a Gaussian distribution, as shown in

$$x_i' = x_i + N(0, \sigma). \tag{4}$$

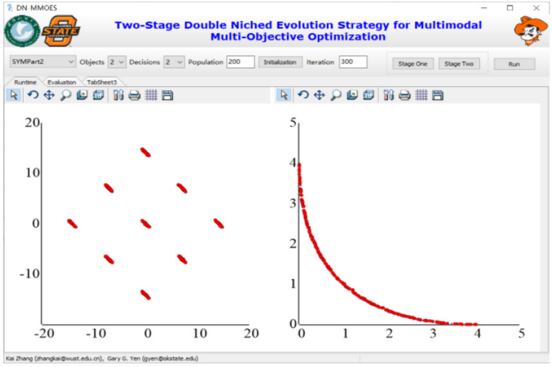
Let $P_t^{(i)}$ be the ith original solution in population P_t, and New$P_t^{(i)}$ be the newly mutated candidate solution, as shown in lines 4 and 5 of Algorithm 3. If New$P_t^{(i)}$ dominates $P_t^{(i)}$, the New$P_t^{(i)}$ would replace the $P_t^{(i)}$, as shown in lines 6 and 7 of Algorithm 3.

Algorithm 3 Proposed *DN-MMOES* Algorithm

```
1:  Initialization $P_t$, $t = 0$
    // the first stage
2:  while ($t < 1/2$ maximum generation) {
3:      for $i = 1$ to $P$ {
4:          $NewP_t^{(i)} = GaussMutation\left(P_t^{(i)}\right)$
5:          Objective Functions Evaluation ($NewP_t^{(i)}$)
6:          if ($NewP_t^{(i)} \prec P_t^{(i)}$) {
7:              $P_t^{(i)} = NewP_t^{(i)}$
8:          else if ($NewP_t^{(i)} \nprec P_t^{(i)}$) and ($P_t^{(i)} \nprec NewP_t^{(i)}$) {
9:              if DomCount$\left(NewP_t^{(i)}\right)$ < DomCount$\left(P_t^{(i)}\right)${
10:                 $P_t^{(i)} = NewP_t^{(i)}$
11:             else if DomCount$\left(NewP_t^{(i)}\right)$ = DomCount$\left(P_t^{(i)}\right)$
12:                 if $\frac{\mathbf{MED_x}(NewP_t^{(i)})}{\mathbf{MED_x}(P_t^{(i)})} > 1${
13:                     $P_t^{(i)} = NewP_t^{(i)}$
14:                 end if
15:             end if
16:         end if
17:     end for
18:     Decision Density Self-Adaptive Strategy ();
19: end while
    // the second stage
20: while ($t <$ maximum generation)
21:     for $i = 1$ to $P$
22:         $NewP_t^{(i)} = GaussMutation(P_t^{(i)})$
23:         Objective Functions Evaluation ($NewP_t^{(i)}$)
24:         if ($NewP_t^{(i)} \prec P_t^{(i)}$)
25:             $P_t^{(i)} = NewP_t^{(i)}$
26:         else if ($NewP_t^{(i)} \nprec P_t^{(i)}$) and ($P_t^{(i)} \nprec NewP_t^{(i)}$)
27:             if DomCount$\left(NewP_t^{(i)}\right)$ < DomCount$\left(P_t^{(i)}\right)$
28:                 $P_t^{(i)} = NewP_t^{(i)}$
29:             else if DomCount$\left(NewP_t^{(i)}\right)$ = DomCount$\left(P_t^{(i)}\right)$
30:                 if $\frac{\mathbf{MED_x}\left(NewP_t^{(i)}\right)}{\mathbf{MED_x}\left(P_t^{(i)}\right)} > 1$ and $\frac{\mathbf{MED}\left(NewP_t^{(i)}\right)}{\mathbf{MED}\left(P_t^{(i)}\right)} > 1$
31:                     $P_t^{(i)} = NewP_t^{(i)}$
32:                 end if
33:             end if
34:         end if
35:     end for
36: end while
```

If New$P_t^{(i)}$ and $P_t^{(i)}$ are nondominated with respect to each other, the values of their *DomCount* would be compared. DomCount($P_t^{(i)}$) is the function to calculate the number of other solutions that dominate $P_t^{(i)}$. If the value of DomCount(New$P_t^{(i)}$) is smaller than the value of DomCount($P_t^{(i)}$), which implies fewer solutions can dominate New$P_t^{(i)}$. The New$P_t^{(i)}$ would be accepted and replace the $P_t^{(i)}$. If the value of DomCount(New$P_t^{(i)}$) is equal to the value of DomCount($P_t^{(i)}$), the value of maximum extension distance would then be compared, as shown in lines 8–16 of Algorithm 3.

In the first stage, if the MEDx(New$P_t^{(i)}$) is greater than MEDx($P_t^{(i)}$), the New$P_t^{(i)}$ would replace $P_t^{(i)}$ which implies the mutated new candidate solution New$P_t^{(i)}$ is superior to the original $P_t^{(i)}$. On the contrary, if the MEDx(New$P_t^{(i)}$) is not greater than MEDx($P_t^{(i)}$), we should neglect New$P_t^{(i)}$, as can be seen lines 12–14. Finally, the decision density self-adaptive strategy is integrated to improve the decision densities in imbalanced Pareto subregions.


DN-MMOES

Two-Stage Double Niched Evolution Strategy for Multimodal Multi-Objective Optimization

SYMPart2 Objects 2 Decisions 2 Population 200 Iteration 300

Kai Zhang (zhangkai@wust.edu.cn), Gary G. Yen (gyen@okstate.edu)

Fig. 10. Developed GUI program for the proposed DN-MMOES algorithm.

In the second stage, if the MED(New$P_t^{(i)}$) and MEDx(New$P_t^{(i)}$) are greater than those of the original MED($P_t^{(i)}$) and MEDx($P_t^{(i)}$), that means the mutated new candidate solution New$P_t^{(i)}$ is considered superior to the original $P_t^{(i)}$. The New$P_t^{(i)}$ would replace $P_t^{(i)}$, as shown in lines 30–32.

IV. Experimental Results

A. Competing State-of-the-Art MMOEAs and Performance Indicators

To validate the performance, DN-MMOES is compared with eight state-of-the-art MMOEAs designed specifically for MMOPs, including Omnioptimizer [14], CPDEA [19], TriMOEA-TA&R [18], MO_Ring_PSO_SCD [20], MOEA/D-AD [22], NIMMO [25], MMOCLRPSO [34], and NMOHSA [35]. Four of these competing approaches are Pareto-based MMOEAs with different decision niching strategies. MOEA/D-AD is chosen from the decomposition-based group, while NIMMO is the representative of the indicator-based MMOEA. MMOCLRPSO and NMOHSA are among the top three MMOEAs in CEC'2019 MMOP competition session.

These competing algorithms are evaluated on 23 CEC 2019 competition MMOPs [36], twelve IDMP test MMOPs [19], and a real-world map-based test problem [37]. These MMOPs are characterized by concave, convex, and sphere Pareto fronts, and the number of equivalent Pareto subsets varies from 2, 4, 9–27.

The inverted generational distance in the objective space (IGD) [38] and in the decision space (IGDx) [39] are adopted to evaluate the performance of these competing algorithms. For each test functions, 400 true Pareto optimal solutions are sampled. The IGD indicator is defined in (4), which calculates the average Euclidean distance from these reference true PF to the obtained objective values. The smaller IGD value means the obtained solutions have better diversity and convergence in the objective space

$$\mathrm{IGD} = \frac{\left(\sum_{i=1}^{|\mathrm{PF}|} d_i^2\right)^{1/2}}{|\mathrm{PF}|} \tag{5}$$

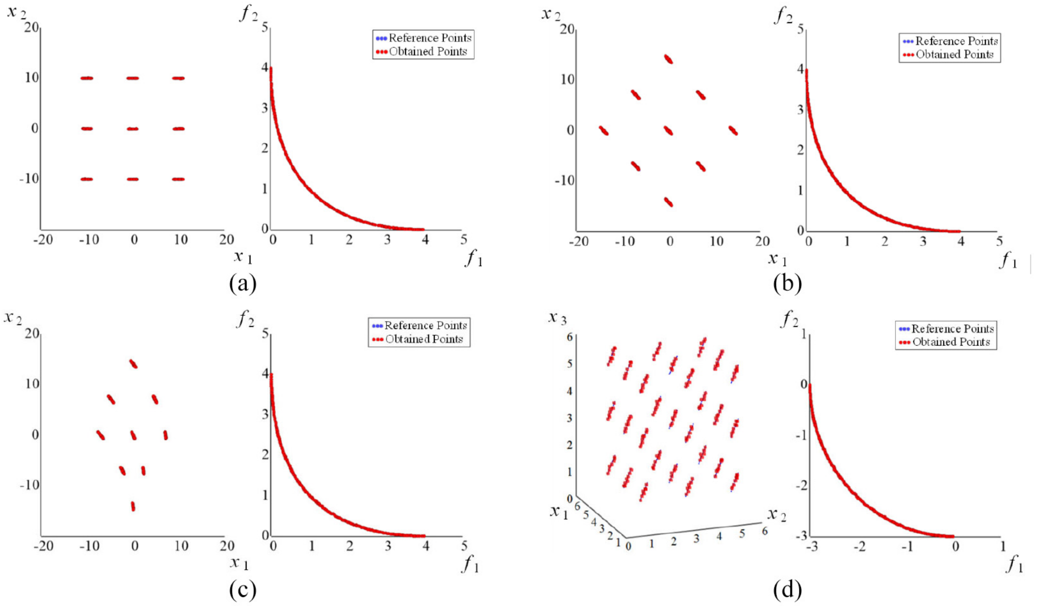


Fig. 11. Experimental results on SYM-PART and Omni test problem by DN-MMOES. (a) SYM-PART 1. (b) SYM-PART 2. (c) SYM-PART 3. (d) Omni Test Problem.

where d_i represents the Euclidean distance between ith reference objective vector and obtained nearest objective values.

On the other hand, the IGDx is defined in (5), which calculate the average Euclidean distance from the true PS to the obtained solutions. The smaller IGDx value implies the obtained solutions have better diversity and convergence in the decision space

$$\text{IGDX} = \frac{\left(\sum_{j=1}^{|\text{PS}|} d_j^2\right)^{1/2}}{|\text{PS}|} \tag{6}$$

where d_j represents the Euclidean distance between the jth reference decision vector and obtained nearest decision variables.

The DN-MMOES algorithm and a GUI program have been implemented to facilitate the experiments, as shown in Fig. 10. The program and source code are available on the GitHub https://github.com/MaOEA/DN-MMOES. For the sake of convenience, interested readers could verify the performance and obtain the source code.

In our experiment, the population size, P, is set to 200, while the maximum iteration is set to 1000 as well as 200 000 maximum function evaluation. For the Gaussian mutation parameters of DN-MMOES, distribution mean, μ, is set to 0, and standard deviation, σ, is set to 0.2. Given an MMOP, different Pareto subregions may be expanding or contracting. During the long evolutionary process, a fixed value of radius parameter δ could lead to inappropriate density estimate. In our algorithm, the radius percent, δ, is set to a random number between 1 and 10. The experimental results show a satisfactory performance. For all competing algorithms, the SBX distribution index [40] and the Polynomial mutation distribution index [41] are set to 20, respectively. We set the other parameters the same as the original articles. In order to make a fair comparison, all the competing algorithms are run 30 times independently for every MMOP. The parameter settings of other compared algorithms are provided in the Supplemental Material.

B. Performance on Test MMOPs

Fig. 11 shows the experimental results on MMOP test suite of CEC2019 competition by DN-MMOES. As shown in Fig. 11(a)–(c), although SYM-PART test problems contain rotated and transformed Pareto-optimal sets, DN-MMOES can find all the nine Pareto subregions for the SYM-PART problems, and obtain the quality converged and distributed PF. In each subfigure, the figure on the left refers to the PS distribution in the decision space, while the one on the right refers to the PF distribution in the objective space.

As can be seen from Fig. 11(d), DN-MMOES find 27 multiple subgroups of optimal solutions on Omni test problem simultaneously and obtains the well-distributed and well-converged solutions in both the objective and decision spaces. In spite of the population size being 200, all the Pareto subregions contain similar number of solutions and similar densities.

As shown in Fig. 12, for most of the MMF MMOPs, the DN-MMOES can consistently find complete sets of global Pareto optimal solutions simultaneously, and the optimal solutions is well-distributed and well-converged in both the objective and decision spaces. However, the proposed DN-MMOES cannot identify the local optimal solutions for six MMOPs, including MMF10-MMF13, MMF15 and MMF15a. Because some local optimal solutions may be dominated by global optimal solutions, it is much complicated to obtain local PS for these MMOPs.

C. Comparison With Competitive MMOEAs

Table I shows the IGD measures that obtained by the nine competing algorithms. The proposed DN-MMOES wins in 21

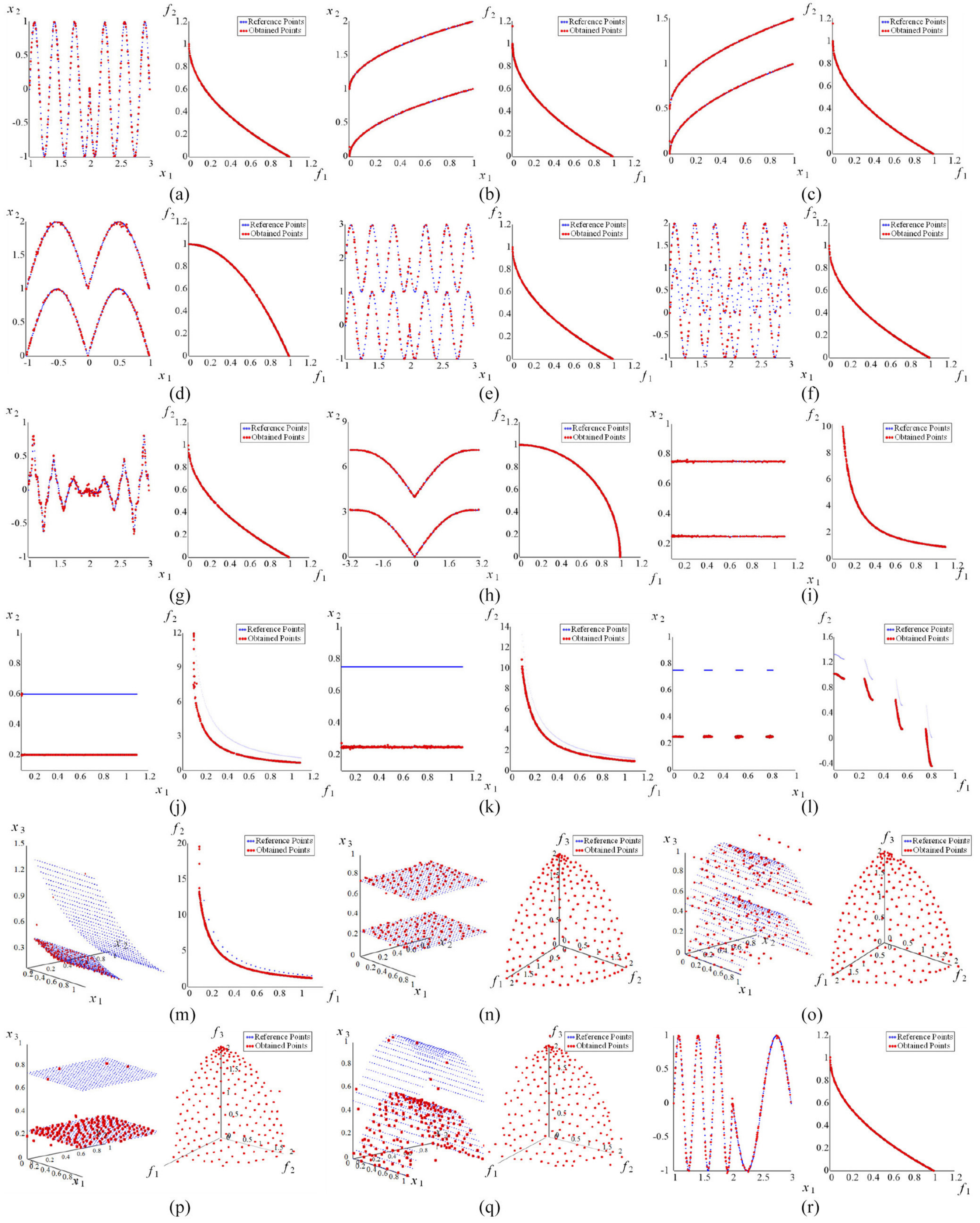


Fig. 12. Experimental results on CEC2019 competition MMOPs by DN-MMOES. (a) MMF1. (b) MMF2. (c) MMF3. (d) MMF4 (e) MMF5. (f) MMF6. (g) MMF7. (h) MMF8. (i) MMF9. (j) MMF10. (k) MMF11. (l) MMF12. (m) MMF13. (n) MMF14. (o) MMF14a. (p) MMF15. (q) MMF15a. (r) MMF1z.

test instances, including two SYM-PART problems, 14 MMF test problems, and five IDMP problems. These MMOPs characterized by concave, convex, and sphere Pareto fronts. The results indicate the ability of the proposed DN-MMOES to find well-diversified and well-converged Pareto fronts. For the remaining MMOPs, CPDEA and NMOHSA wins eight

TABLE I
AVERAGE IGD VALUES OVER 30 RUNS ON BENCHMARK INSTANCES (POPULATION SIZE 200), WHERE THE BEST MEAN IS SHOWN IN A GRAY BACKGROUND

Problem	DN-MMOES	MO_Ring_PSO_SCD	Omni-optimizer	MOEA/D-AD	NIMMO	TriMOEA-TA&R	CPDEA	MMOCLR PSO	NMOHSA
OMNI TEST	0.009105	0.025769(+)	0.008339(–)	0.110350(+)	0.021287(+)	0.015566(+)	0.008998(–)	0.017688(+)	0.010169(+)
SYM PART1	0.008510	0.016349(+)	0.011117(+)	0.263950(+)	0.030711(+)	0.045022(+)	0.008752(+)	0.013027(+)	0.012987(+)
SYM PART2	0.009624	0.023341(+)	0.011612(+)	0.187580(+)	0.036490(+)	0.044581(+)	0.012317(+)	0.015458(+)	0.011114(+)
SYM PART3	0.009783	0.024818(+)	0.012244(+)	0.301090(+)	0.029373(+)	0.467410(+)	0.011530(+)	0.021691(+)	0.009408(–)
MMF1	0.002114	0.005715(+)	0.002880(+)	0.015951(+)	0.004754(+)	0.003403(+)	0.002423(+)	0.002918(+)	0.002408(+)
MMF2	0.002308	0.004563(+)	0.003030(+)	0.073816(+)	0.007771(+)	0.003562(+)	0.002732(+)	0.004133(+)	0.002368(+)
MMF3	0.002369	0.007633(+)	0.003238(+)	0.042558(+)	0.008205(+)	0.003718(+)	0.002599(+)	0.003837(+)	0.002112(–)
MMF4	0.002025	0.004917(+)	0.002353(+)	0.007932(+)	0.005211(+)	0.003642(+)	0.002508(+)	0.002986(+)	0.002531(+)
MMF5	0.002053	0.004694(+)	0.002766(+)	0.008205(+)	0.003597(+)	0.003404(+)	0.002323(+)	0.003041(+)	0.002603(+)
MMF6	0.002138	0.004311(+)	0.002829(+)	0.009771(+)	0.004654(+)	0.003375(+)	0.002322(+)	0.003043(+)	0.002398(+)
MMF7	0.002358	0.004767(+)	0.002900(+)	0.010309(+)	0.002924(+)	0.007582(+)	0.002564(+)	0.003182(+)	0.002633(+)
MMF8	0.002113	0.004852(+)	0.002887(+)	0.011212(+)	0.008646(+)	0.029627(+)	0.002568(+)	0.003624(+)	0.003316(+)
MMF9	0.009170	0.050286(+)	0.010253(+)	0.056719(+)	0.013081(+)	0.074608(+)	0.010711(+)	0.016818(+)	0.011035(+)
MMF10	0.140666	0.181183(+)	0.181961(+)	0.209443(+)	0.169068(+)	0.232737(+)	0.194138(+)	0.159589(+)	0.160631(+)
MMF11	0.087242	0.136838(+)	0.093380(+)	0.117820(+)	0.087900(+)	0.168572(+)	0.093111(+)	0.085036(–)	0.090335(+)
MMF12	0.068353	0.059598(–)	0.082713(+)	0.083787(+)	0.093997(+)	0.084769(+)	0.082631(+)	0.069339(+)	0.062030(–)
MMF13	0.073831	0.155101(+)	0.147841(+)	0.178614(+)	0.147986(+)	0.250556(+)	0.148200(+)	0.090514(+)	0.126986(+)
MMF14	0.068968	0.109030(+)	0.093230(+)	0.085708(+)	0.110200(+)	0.089296(+)	0.071087(+)	0.097527(+)	0.097457(+)
MMF14A	0.070796	0.100990(+)	0.093639(+)	0.111940(+)	0.129440(+)	0.100940(+)	0.072540(+)	0.095480(+)	0.090238(+)
MMF15	0.186857	0.200963(+)	0.193996(+)	0.194014(+)	0.202103(+)	0.208992(+)	0.178177(–)	0.188745(+)	0.172610(–)
MMF15A	0.175620	0.193140(+)	0.195134(+)	0.184590(+)	0.217929(+)	0.200170(+)	0.175418(–)	0.192696(+)	0.184590(+)
MMF1Z	0.001965	0.005436(+)	0.002833(+)	0.008850(+)	0.004176(+)	0.003485(+)	0.002394(+)	0.002759(+)	0.002391(+)
MMF1E	0.002433	0.005677(+)	0.003485(+)	0.108050(+)	0.004667(+)	0.003608(+)	0.002654(+)	0.003872(+)	0.002784(+)
IDMP-M2-T1	0.000359	0.001866(+)	0.000461(+)	0.002219(+)	0.000475(+)	0.000713(+)	0.000457(+)	0.000947(+)	0.000525(+)
IDMP-M2-T2	0.000360	0.001034(+)	0.000500(+)	0.000486(+)	0.000464(+)	0.000712(+)	0.000414(+)	0.001764(+)	0.000491(+)
IDMP-M2-T3	0.000377	0.000864(+)	0.000511(+)	0.000534(+)	0.000484(+)	0.000713(+)	0.000461(+)	0.003420(+)	0.000561(+)
IDMP-M2-T4	0.000363	0.000958(+)	0.000478(+)	0.000542(+)	0.000466(+)	0.000713(+)	0.000411(+)	0.005548(+)	0.000493(+)
IDMP-M3-T1	0.004901	0.008816(+)	0.005893(+)	0.006831(+)	0.005025(+)	0.013178(+)	0.004839(–)	0.007739(+)	0.007031(+)
IDMP-M3-T2	0.004887	0.007402(+)	0.005641(+)	0.005883(+)	0.004887(+)	0.011617(+)	0.004766(–)	0.010305(+)	0.006453(+)
IDMP-M3-T3	0.005100	0.007181(+)	0.005938(+)	0.006163(+)	0.005045(–)	0.011199(+)	0.005143(+)	0.009599(+)	0.006457(+)
IDMP-M3-T4	0.004939	0.007637(+)	0.005720(+)	0.007229(+)	0.004959(+)	0.012484(+)	0.004853(–)	0.025522(+)	0.006300(+)
IDMP-M4-T1	0.006821	0.020644(+)	0.009140(+)	0.017407(+)	0.006892(+)	0.029332(+)	0.006695(–)	0.022628(+)	0.011413(+)
IDMP-M4-T2	0.006762	0.011883(+)	0.008645(+)	0.014066(+)	0.006821(+)	0.025849(+)	0.006476(–)	0.215624(+)	0.010185(+)
IDMP-M4-T3	0.006934	0.010867(+)	0.008450(+)	0.019521(+)	0.007083(+)	0.029126(+)	0.006979(+)	0.045734(+)	0.009869(+)
IDMP-M4-T4	0.006677	0.012506(+)	0.008255(+)	0.022409(+)	0.007165(+)	0.028762(+)	0.006539(–)	0.523465(+)	0.009370(+)
REAL MAP	1.041844	1.329818(+)	1.206332(+)	2.698940(+)	1.468723(+)	2.725678(+)	1.003779(–)	1.209730(+)	1.629695(+)
		+35/–1/=0	+35/–1/=0	+36/–0/=0	+35/–1/=0	+36/–0/=0	+28/–8/=1	+35/–1/=0	+33/–3/=0

Wilcoxon's rank sum test at a 0.05 significance level is performed between DN-MMOES and compared MMOEAs. "+" means DN-MMOES better than compared algorithm, "–" means compared algorithm better than DN-MMOES, "=" means not comparable)

and three test instances. MO_RING_PSO_SCD, Omni optimizer, NIMMO and MMOCLRPSO wins one test instance, respectively.

Table II shows the IGDx values that obtained by the nine competing algorithms on 36 MMOPs. Compared with the other competing MMOEAs, DN-MMOES obtains significantly better IGDx performance on eleven MMOPs. From these MMOP instances, some of the PSs are contracting, while others spread all over the decision space. The results clearly demonstrate its ability to find multiple groups of well-distributed and well-converged optimal solutions with various problem characterizes. The NIMMO and CPDEA win eight test instances. TriMOEA-TA&R and NMOHSA also win in three test instances, respectively. MOEA/D-AD wins two MMOP instances.

As can be seen from Tables I and II, the proposed DN-MMOES obtain better performance on most of the CEC2019 competition MMOPs. The NIMMO and NMOHSA shows better IGDx performance on MMOPs with local Pareto optimal solutions. CPDEA and NIMMO obtained better performance on IDMP test suite, while DN-MMOES is not far behind.

D. IGD Union Indicator for MMOPs

For any given test instance, two competing algorithms $A1$ and $A2$ could obtain two groups of IGD and IGDx values. At times, one algorithm could obtain a better IGD value in the objective space than another, and at the same time, it could obtain a worse IGDx value in the decision spaces or vice versa. In order to compare the performance of two MMOEAs on the same test instance, we propose an IGD indicator, $\mathrm{IGD}_{\mathrm{Union}}(A1, A2)$, which calculates the difference of the summation of ratios of IGD and IGDx, as shown in

$$\mathrm{IGD}_{\mathrm{Union}}(A1, A2) = \left(\frac{\mathrm{IGD}(A1)}{\mathrm{IGD}(A2)} + \frac{\mathrm{IGDx}(A1)}{\mathrm{IGDx}(A2)}\right) - \left(\frac{\mathrm{IGD}(A2)}{\mathrm{IGD}(A1)} + \frac{\mathrm{IGDx}(A2)}{\mathrm{IGDx}(A1)}\right) \quad (7)$$

TABLE II
AVERAGE IGDX VALUES OVER 30 RUNS ON BENCHMARK INSTANCES (POPULATION SIZE 200), WHERE THE BEST MEAN IS SHOWN IN A GRAY BACKGROUND

Problem	DN-MMOES	MO_Ring_PSO_SCD	Omni-optimizer	MOEA/D-AD	NIMMO	TriMOEA-TA&R	CPDEA	MMOCLR PSO	NMOHSA
Omni Test	0.072417	0.285230(+)	0.311740(+)	0.941190(+)	0.331240(+)	1.424200(+)	0.077460(+)	0.170941(+)	0.166548(+)
SYM Part1	0.029198	0.090250(+)	1.296200(+)	1.563000(+)	0.042239(+)	0.023767(–)	0.031498(+)	0.061511(+)	0.122423(+)
SYM Part2	0.054387	0.090766(+)	0.130420(+)	4.044600(+)	0.069384(+)	2.194400(+)	0.055375(+)	0.082943(+)	1.005996(+)
SYM Part3	0.039068	0.121330(+)	0.081585(+)	4.156000(+)	0.054507(+)	0.393110(+)	0.043012(+)	0.162156(+)	0.108492(+)
MMF1	0.033204	0.071021(+)	0.046441(+)	0.149380(+)	0.055408(+)	0.039788(+)	0.032407(–)	0.041366(+)	0.039847(+)
MMF2	0.005604	0.014250(+)	0.046738(+)	0.133350(+)	0.049071(+)	0.025958(+)	0.005085(–)	0.009530(+)	0.005899(+)
MMF3	0.004939	0.013989(+)	0.008788(+)	0.083116(+)	0.086508(+)	0.014937(+)	0.004547(–)	0.008122(+)	0.005170(+)
MMF4	0.015897	0.034782(+)	0.027968(+)	0.120470(+)	0.044792(+)	0.027028(+)	0.018713(+)	0.027623(+)	0.034819(+)
MMF5	0.058900	0.107780(+)	0.087679(+)	0.133590(+)	0.070787(+)	0.080222(+)	0.059955(+)	0.074428(+)	0.092718(+)
MMF6	0.053408	0.082584(+)	0.078845(+)	0.139040(+)	0.103760(+)	0.077638(+)	0.054425(+)	0.065435(+)	0.082589(+)
MMF7	0.016929	0.031279(+)	0.022937(+)	0.070639(+)	0.016530(–)	0.088707(+)	0.019965(+)	0.021243(+)	0.020136(+)
MMF8	0.028955	0.082511(+)	0.074866(+)	0.238730(+)	0.184470(+)	0.524290(+)	0.037134(+)	0.048142(+)	0.091301(+)
MMF9	0.003956	0.012104(+)	0.010237(+)	0.034204(+)	0.003461(–)	0.003379(–)	0.006824(+)	0.006199(+)	0.006210(+)
MMF10	0.160040	0.137483(–)	0.200702(+)	0.204469(+)	0.020972(–)	0.201521(+)	0.200895(+)	0.200960(+)	0.160394(+)
MMF11	0.236000	0.165705(–)	0.249459(+)	0.248313(+)	0.005037(–)	0.252486(+)	0.248926(+)	0.245618(+)	0.194027(–)
MMF12	0.204152	0.145677(–)	0.245399(+)	0.055540(–)	0.003358(–)	0.248119(+)	0.245283(+)	0.229950(+)	0.244909(+)
MMF13	0.086407	0.205417(+)	0.260787(+)	0.286979(+)	0.138750	0.262970(+)	0.254850(+)	0.226840(+)	0.260640(+)
MMF14	0.045515	0.074407(+)	0.062174(+)	0.055816(+)	0.069368(+)	0.037049(–)	0.046433(+)	0.066111(+)	0.068115(+)
MMF14a	0.060320	0.080832(+)	0.079091(+)	0.090578(+)	0.119100(+)	0.067521(+)	0.063205(+)	0.080177(+)	0.076734(+)
MMF15	0.236856	0.209278(–)	0.258321(+)	0.265605(+)	0.183492(–)	0.271241(+)	0.206743(–)	0.177063(–)	0.136163(–)
MMF15A	0.191381	0.184394(–)	0.222602(+)	0.196662(+)	0.218297(+)	0.224745(+)	0.210192(+)	0.186619(–)	0.183200(–)
MMF1z	0.021766	0.047795(+)	0.034740(+)	0.085130(+)	0.035517(+)	0.030980(+)	0.024931(+)	0.029995(+)	0.028231(+)
MMF1e	0.130272	0.223270(+)	0.728450(+)	1.908400(+)	2.033100(+)	1.019200(+)	0.140560(+)	0.188774(+)	0.121428(–)
IDMP-M2-T1	0.000560	0.003953(+)	0.000669(+)	0.003192(+)	0.000607(+)	0.673275(+)	0.000563(+)	0.002605(+)	0.000620(+)
IDMP-M2-T2	0.000551	0.002773(+)	0.673373(+)	0.000568(+)	0.000576(+)	0.673275(+)	0.000597(+)	0.002807(+)	0.000730(+)
IDMP-M2-T3	0.000684	0.001708(+)	0.002238(+)	0.000906(+)	0.000625(–)	0.673572(+)	0.000832(+)	0.005134(+)	0.000999(+)
IDMP-M2-T4	0.000584	0.001743(+)	0.673212(+)	0.000549(–)	0.000609(+)	0.673276(+)	0.000570(–)	0.005482(+)	0.000609(+)
IDMP-M3-T1	0.008850	0.019548(+)	0.267540(+)	0.008003(–)	0.007737(–)	1.014524(+)	0.008277(–)	0.256796(+)	0.010860(+)
IDMP-M3-T2	0.008572	0.014534(+)	0.505420(+)	0.008491(–)	0.008158(–)	0.628167(+)	0.008184(–)	0.269431(+)	0.010455(+)
IDMP-M3-T3	0.009109	0.011793(+)	0.011164(+)	0.008772(–)	0.007719(–)	0.267899(+)	0.008910(–)	0.016488(+)	0.011227(+)
IDMP-M3-T4	0.009500	0.018678(+)	0.626722(+)	0.007840(–)	0.009335(–)	0.627941(+)	0.008282(–)	0.132826(+)	0.010411(+)
IDMP-M4-T1	0.012701	0.298573(+)	1.185755(+)	0.279750(+)	0.010876(–)	1.193154(+)	0.009382(–)	0.540619(+)	0.011791(–)
IDMP-M4-T2	0.011720	0.023368(+)	1.187074(+)	0.279277(+)	0.278490(+)	1.192950(+)	0.009344(–)	0.390685(+)	0.012578(+)
IDMP-M4-T3	0.010507	0.015443(+)	1.186990(+)	0.279321(+)	0.010073(–)	1.192883(+)	0.010070(–)	0.054319(+)	0.012831(+)
IDMP-M4-T4	0.010095	0.015460(+)	1.187437(+)	0.548193(+)	0.272988(+)	1.193561(+)	0.009464(–)	0.243144(+)	0.012543(+)
Real MAP	0.842058	1.034292(+)	0.950114(+)	2.138150(+)	1.272156(+)	2.198555(+)	0.785526(–)	0.952703(+)	1.325232(+)
		+31/–5/=0	+36/–0/=0	+30/–6/=0	+23/–13/=0	+33/–3/=0	+22/–14/=1	+34/–2/=0	+31/–5/=0

Wilcoxon's rank sum test at a 0.05 significance level is performed between DN-MMOES and compared MMOEAs. "+" means DN-MMOES better than compared algorithm, "–" means compared algorithm better than DN-MMOES, "=" means not comparable)

where IGD(A) and IGDx(A) denote the IGD and IGDx values of algorithm A.

The proposed $\text{IGD}_{\text{Union}}(A1, A2)$ consider IGD and IGDx values as a whole, as shown in (7). When the value of $\text{IGD}_{\text{Union}}(A1, A2)$ is smaller than zero, which means A1 obtain a better overall performance than A2 on the given test instance. When the $\text{IGD}_{\text{Union}}(A1, A2)$ is equal to zero, the performance of the two algorithms is approximately the same. If the $\text{IGD}_{\text{Union}}(A1, A2)$ is greater than zero, this implies A1 obtains a poor overall performance than A2. The absolute value of $\text{IGD}_{\text{Union}}$ indicates the degree of that algorithm A1 more superior to or much worse than algorithm A2

$$\text{IGD}_{\text{Union}}(A1, A2) = \begin{cases} < 0, & A1 \text{ is better than } A2 \\ = 0, & A1 \text{ is equal to } A2 \\ > 0, & A1 \text{ is worse than } A2. \end{cases} \tag{8}$$

If the IGD value of A1 is smaller than that of A2, which means A1 obtains a better performance than A2 in the objective space, and the IGD ratio of $A1$ and $A2$ should be smaller than one. When the IGD values of $A1$ and $A2$ are the same, the performance of the two algorithms is approximately similar, and the ratio of IGD values is one. If the IGD value of A1 is greater than that of A2, which means A1 obtains a worse performance than A2 in the objective space, and the IGD ratio of $A1$ and $A2$ should be greater than one. The ratio of IGDx values can be interpreted in a similar way with IGD values. It is worthy to note that value of $\text{IGD}_{\text{Union}}(A2, A1)$ is the negative value of $\text{IGD}_{\text{Union}}(A1, A2)$, providing the same and consistent conclusion, as shown in the following equation:

$$\text{IGD}_{\text{Union}}(A2, A1) = -\text{IGD}_{\text{Union}}(A1, A2). \tag{9}$$

Table III shows the $\text{IGD}_{\text{Union}}$ values of DN-MMOES against eight competing MMOEAs on 36 MMOP instances. For example, DN-MMOES obtains a better IGDx value (i.e., 0.039068) than that of NMOHSA (i.e., 0.108492) on SYM-PART3 instance. On the contrary, the DN-MMOES obtain a worse IGD value (i.e., 0.009783) than that of NMOHSA (i.e., 0.009408) on SYM-PART3 instance. The IGD union value

TABLE III
IGD Union Indicator Values of DN-MMOES Against Eight Competing MMOEAs on Benchmark Instances

Problem	MO_Ring_PSO _SCD	Omni-optimizer	MOEA/D-AD	NIMMO	TriMOEA- TA&R	CPDEA	MMOCLR PSO	NMOHSA
Omni Test	-6.161698	-3.896504	-24.95707	-6.265660	-20.740487	-0.111100	-3.364785	-2.086526
SYM Part1	-4.168072	-44.91177	-84.49659	-4.087097	-4.686947	-0.207881	-2.509535	-4.825171
SYM Part2	-3.082661	-2.358754	-93.79314	-4.019710	-44.739490	-0.534470	-1.852939	-18.73181
SYM Part3	-4.926273	-2.061975	-137.1135	-3.347823	-57.719665	-0.522740	-5.675878	-2.338713
MMF1	-4.004907	-1.312004	-11.68945	-2.873590	-1.352294	-0.225099	-1.098978	-0.627945
MMF2	-3.620792	-8.771321	-55.70488	-11.71220	-5.311538	-0.144229	-2.344829	-0.153984
MMF3	-5.390963	-1.852488	-34.67796	-20.63295	-3.625909	-0.020004	-2.038622	**0.138719**
MMF4	-3.747225	-1.492298	-11.10794	-4.647464	-2.354533	-0.758726	-1.958528	-2.183524
MMF5	-3.132441	-1.421909	-5.573557	-1.551060	-1.682738	-0.283252	-1.278405	-1.418095
MMF6	-2.420004	-1.366352	-6.570584	-3.145465	-1.710863	-0.203032	-1.129688	-1.129739
MMF7	-2.833410	-1.033581	-8.076200	-0.385897	-7.953539	-0.499109	-1.066313	-0.569773
MMF8	-4.359477	-2.833243	-13.24131	-10.06137	-32.00181	-0.895244	-2.193239	-3.768185
MMF9	-8.034217	-2.425004	-14.55405	-0.457334	-7.696572	-1.457175	-2.217594	-1.305119
MMF10	-0.206648	-0.977185	-1.312220	**7.130182**	-1.515168	-1.114212	-0.712406	-0.270642
MMF11	-0.208818	-0.247070	-0.711788	**46.81691**	-1.549850	-0.237003	-0.028685	**0.324485**
MMF12	**0.962789**	-0.753820	**2.993709**	**60.13127**	-0.826386	-0.750839	-0.267201	-0.171618
MMF13	-3.581429	-4.189834	-5.026030	-2.488503	-5.813780	-4.119474	-2.654610	-3.823448
MMF14	-1.971395	-1.245977	-0.848909	-1.839929	-0.107880	-0.100473	-1.470969	-1.533731
MMF14a	-1.319287	-1.115132	-1.784396	-2.749416	-0.950448	-0.142149	-1.184048	-0.976096
MMF15	**0.102518**	-0.248733	-0.304808	**0.359098**	-0.496314	**0.367959**	**0.570031**	**1.323411**
MMF15A	-0.116079	-0.514512	-0.154118	-0.698994	-0.585222	-0.185483	-0.135449	-0.012267
MMF1z	-4.145386	-1.717647	-7.937249	-2.673577	-1.930430	-0.669879	-1.344269	-0.920988
MMF1e	-3.035165	-6.147180	-58.96876	-16.93939	-8.504421	-0.326271	-1.722074	-0.129622
IDMP-M2-T1	-11.92264	-0.862953	-11.54383	-0.728689	-1203.758	-0.498108	-6.695606	-0.982503
IDMP-M2-T2	-7.358026	-1222.760	-0.670042	-0.601802	-1223.386	-0.440971	-9.593997	-1.200761
IDMP-M2-T3	-3.952042	-3.583968	-1.280048	-0.324237	-986.1158	-0.799283	-16.33403	-1.591891
IDMP-M2-T4	-4.909739	-1153.316	-0.699687	-0.588636	-1154.324	-0.200486	-24.49877	-0.705678
IDMP-M3-T1	-2.998977	-30.56817	-0.474791	**0.219639**	-116.9436	**0.159437**	-29.92781	-1.149751
IDMP-M3-T2	-1.960135	-59.23272	-0.354119	**0.099045**	-75.22406	**0.142821**	-33.03412	-0.962894
IDMP-M3-T3	-1.220078	-0.715114	-0.305498	**0.354358**	-31.11684	**0.027389**	-2.608467	-0.897408
IDMP-M3-T4	-2.357030	-66.25024	-0.393965	**0.026961**	-68.21593	**0.310410**	-18.88408	-0.674992
IDMP-M4-T1	-26.16142	-93.94217	-24.14054	**0.290779**	-97.99879	**0.652373**	-45.55754	-0.926738
IDMP-M4-T2	-2.680588	-101.7725	-25.38655	-23.73723	-105.3388	**0.543469**	-65.16114	-0.983716
IDMP-M4-T3	-1.718536	-113.3605	-29.00671	**0.041867**	-117.4858	**0.072050**	-11.42036	-1.122982
IDMP-M4-T4	-2.217569	-118.0452	-57.34319	-27.14611	-122.2999	**0.170952**	-102.4295	-1.128396
Real MAP	-0.907111	-0.536289	-4.349892	-1.549236	-4.461898	**0.213560**	-0.547459	-1.863350
	+34/−2/=0	+36/−0/=0	+35/−1/=0	+26/−10/=0	+36/−0/=0	+26/−10/=1	+35/−1/=0	+33/−3/=0

The IGD union values are produced by $IGD_{Union}(\mathrm{DN-MMOES, Competing\ MMOEA})$

is calculated according to (6). Although the IGD value of DN-MMOES is slightly greater than that of NMOHSA, the IGDx value of DN-MMOES is much smaller than that of NMOHSA. The overall IGD union value is −2.338713, which implies DN-MMOES performs better than NMOHSA on SYM-PART3 instance, indeed significantly better due to the absolute value of $\mathrm{IGD}_{\mathrm{Union}}(\mathrm{DN-MMOES, NMOHSA})$

$$
\begin{aligned}
&\mathrm{IGD}_{\mathrm{Union}}(\mathrm{DN-MMOES, NMOHSA})\\
&\quad=\left(\frac{\mathrm{IGD(DN-MMOES)}}{\mathrm{IGD(NMOHSA)}}+\frac{\mathrm{IGDx(DN-MMOES)}}{\mathrm{IGDx(NMOHSA)}}\right)\\
&\qquad-\left(\frac{\mathrm{IGD(NMOHSA)}}{\mathrm{IGD(DN-MMOES)}}+\frac{\mathrm{IGDx(NMOHSA)}}{\mathrm{IGDx(DN-MMOES)}}\right)\\
&\quad=\left(\frac{0.009783}{0.009408}+\frac{0.039068}{0.108492}\right)-\left(\frac{0.009408}{0.009783}+\frac{0.108492}{0.039068}\right)\\
&\quad=(1.039859+0.360100)-(0.961668+2.777004)\\
&\quad=-2.338713.
\end{aligned}
$$

For some test problems, the difference between their IGD values is much larger or much smaller than the difference between their IGDx values. The proposed $\mathrm{IGD}_{\mathrm{Union}}$ indicator factors the respective ratios to evaluate the difference between the two competing algorithms. For example, it is assumed that the IGDx values obtained by DN-MMOES and NMOHSA are 3.9068 and 10.8492, respectively, which are 100 times greater than the original IGDx values. Although the magnitudes of IGD and IGDx are quite different, the $\mathrm{IGD}_{\mathrm{Union}}$ indicator can reflect the true performance in both spaces faithfully. The value of the $\mathrm{IGD}_{\mathrm{Union}}(\mathrm{DN-MMOES, NMOHSA})$ is shown as

$$
\begin{aligned}
&\mathrm{IGD}_{\mathrm{Union}}(\mathrm{DN-MMOES, NMOHSA})\\
&\quad=\left(\frac{\mathrm{IGD(DN-MMOES)}}{\mathrm{IGD(NMOHSA)}}+\frac{\mathrm{IGDx(DN-MMOES)}}{\mathrm{IGDx(NMOHSA)}}\right)\\
&\qquad-\left(\frac{\mathrm{IGD(NMOHSA)}}{\mathrm{IGD(DN-MMOES)}}+\frac{\mathrm{IGDx(NMOHSA)}}{\mathrm{IGDx(DN-MMOES)}}\right)\\
&\quad=\left(\frac{0.009783}{0.009408}+\frac{3.9068}{10.8492}\right)-\left(\frac{0.009408}{0.009783}+\frac{10.8492}{3.9068}\right)\\
&\quad=(1.039859+0.360100)-(0.961668+2.777004)\\
&\quad=-2.338713.
\end{aligned}
$$

As can be seen from Table III, most of the $\mathrm{IGD}_{\mathrm{Union}}$ values are smaller than zero, except in few cases which are

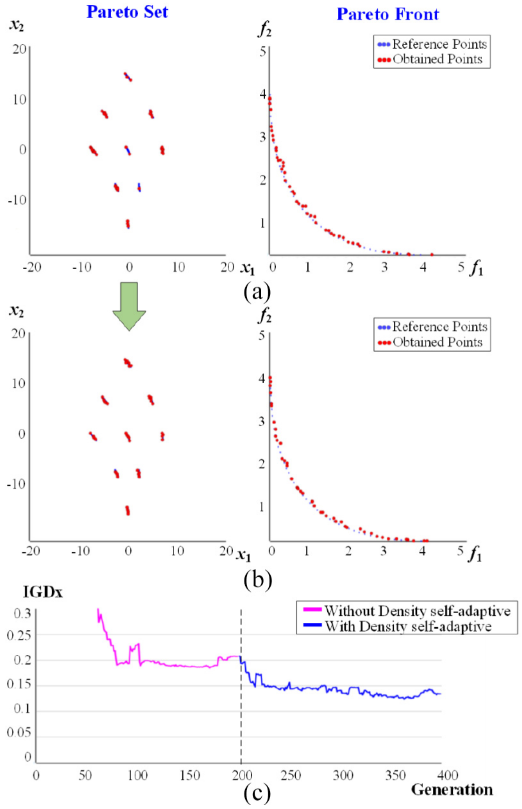


Fig. 13. Experimental results on SYM-PART3 by DN-MMOES without and with density self-adaptive strategy.

highlighted in bold face. The experimental results undoubtedly demonstrate that DN-MMOES performs significantly better than eight competing MMOEAs. Compared with MO_Ring_PSO_SCD, TriMOEA-TA&R, MMOCLRPSO, NMOHSA, MOEA/D-AD, and Omnioptimizer, DN-MMOES wins more than 33 out of 36 test instances on $\text{IGD}_{\text{Union}}$ indicator. Although CPDEA and NIMMO obtain good performance on IDMP test problem, DN-MMOES wins 26 out of 26 test instances on $\text{IGD}_{\text{Union}}$ indicator. Obviously, when we consider IGD and IGDx values as a whole, DN-MMOES obtains significant better $\text{IGD}_{\text{Union}}$ performance than other MMOEAs.

E. Discussion on Density Self-Adapted Strategy

For addressing the MMOPs, a novel decision density self-adaptive strategy is proposed to improve the distribution in the decision space. In order to study the role of the density self-adaptive strategy, we compare the performance with and without the density self-adaptive strategy on SYM-PART3. The SYM-PART3 test problem has nine rotated and transformed Pareto-optimal subsets.

In this experiment, our algorithm only adopts MEDx to improve the decision space distribution. In order to observe the improvement in the decision space more clearly, the population size is set to 45. We do not use density self-adaptive strategy in the first 200 generations, and the obtained optimal solutions are shown in Fig. 13(a). Obviously, some of the Pareto subregions may be overly crowded, while others are rather sparse. In the following 200 generations, the density self-adaptive strategy is activated to improve the imbalanced distribution. As shown in Fig. 13(b), the distribution of the obtained optimal solutions has been improved significantly. Fig. 13(c) shows the obtained IGDx curve on SYM-PART3, as the algorithm with density self-adaptive strategy has achieved a better performance than that without adopting the strategy.

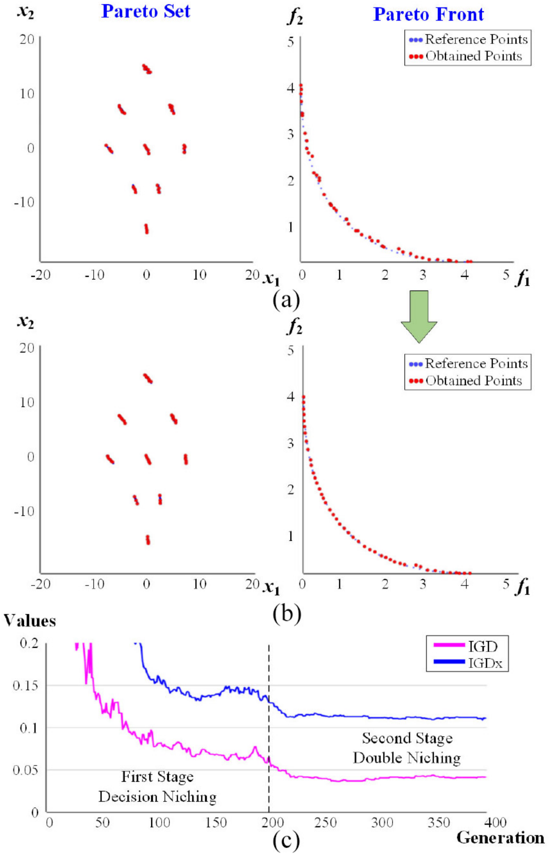


Fig. 14. Experimental results on SYM-PART3 by decision niching and double niching.

F. Discussion on Two-Stage Double Niched Strategy

To validate the proposed two-stage double niched strategy, we compare the performances of DN-MMOES by replacing decision niching $([\boldsymbol{MED}_{\boldsymbol{x}}(\text{New}P_t^{(i)})]/[\boldsymbol{MED}_{\boldsymbol{x}}(P_t^{(i)})]) > 1$ with double niching $([\boldsymbol{MED}_{\boldsymbol{x}}(\text{New}P_t^{(i)})]/[\boldsymbol{MED}_{\boldsymbol{x}}(P_t^{(i)})]) > 1$ and $([\boldsymbol{MED}(\text{New}P_t^{(i)})]/[\boldsymbol{MED}(P_t^{(i)})]) > 1$ in both the objective and decision spaces. In the preceding 200 generations, only MEDx is incorporated to improve the decision space distribution, and the obtained optimal solutions are shown in Fig. 14(a). In spite of the obtained optimal solutions are well-distributed and well-converged in Pareto subregions, the Pareto front is not well distributed in the objective space.

In the following 200 generations, both MED and MEDx are applied to improve the distributions in the decision and objective spaces. As shown in Fig. 14(b), the distribution of Pareto front has been improved appreciably. Fig. 14(c) shows the obtained IGD and IGDx curves, as the double niched strategy has achieved a better performance than that of niched only in the decision space.

V. Conclusion

In this article, a two-stage double niched evolutionary algorithm is proposed for solving MMOPs based on evolution strategy. First, new mutated candidate solution need only to be compared with the original solution, which can avoid the negative influence on the mating pool. Second, the proposed algorithm solves the MMOPs in two stages. The first stage adopts the niching strategy in the decision space, while the second stage adapts double niching strategy in both the objective and decision spaces. In contrast to traditional MMOEAs, the two stages strategy can reduce the phenomenon that some individuals with significantly better objective space distribution may replace the individuals with slightly better decision space distribution. Third, an effective decision density self-adaptive strategy is designed for improving the imbalanced density in the decision space. Fourth, the IGD union indicator $\mathrm{IGD}_{\mathrm{Union}}$ is proposed to evaluate IGDx and IGD values as a whole. The performance of DN-MMOES is compared against eight state-of-the-art MMOEAs, including MO_Ring_PSO_SCD, Omnioptimizer, MOEA/D-AD, NIMMO, TriMOEA-TA&R, CPDEA, MMOCLRPSO, and NMOHSA. The experimental results indicate the DN-MMOES provides better performance than competing MMOEAs on 36 test MMOPs with IGD, IGDx, $\mathrm{IGD}_{\mathrm{Union}}$ indicators.

Although the proposed DN-MMOES can find multiple groups of equivalent PS efficiently, finding the local Pareto optimal solutions of MMOPs still leaves room for improvement. Because some local optimal solutions may be dominated by global optimal solutions, it is much complicated to obtain local PS for MMOPs. We plan to design and incorporate some strategies to improve the performances of local searching.

References

[1] O. M. Shir, M. Preuss, B. Naujoks, and M. T. Emmerich, "Enhancing decision space diversity in evolutionary multiobjective algorithms," in *Proc. Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Nantes, France, 2009, pp. 95–109.

[2] E. M. Zechman, M. H. Giacomoni, and M. E. Shafiee, "An evolutionary algorithm approach to generate distinct sets of non-dominated solutions for wicked problems," *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1442–1457, May/Jun. 2013.

[3] Y. Wang, H.-X. Li, G. G. Yen, and W. Song, "MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 830–843, Apr. 2015.

[4] K. M. Woldemariam and G. G. Yen, "Vaccine-enhanced artificial immune system for multimodal function optimization," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 40, no. 1, pp. 218–228, Feb. 2010.

[5] K. Zhang, C. N. Shen, J. J. He, and G. G. Yen, "Knee based multimodal multi-objective evolutionary algorithm for decision making," *Inf. Sci.*, vol. 544, no. 12, pp. 39–55, Jan. 2021.

[6] K. Zhang, G. G. Yen, and Z. N. He., "Evolutionary algorithm for knee-based multiple criteria decision making," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 722–735, Feb. 2021.

[7] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—A comparative experiment," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 402–412, Aug. 2002.

[8] Y. Han, D. Gong, Y. Jin, and Q. Pan, "Evolutionary multiobjective blocking lot-streaming flow shop scheduling with machine breakdowns," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 184–197, Jan. 2019.

[9] O. Schütze, M. Vasile, and C. A. C. Coello, "Computing the set of epsilon-efficient solutions in multiobjective space mission design," *J. Aerosp. Comput. Inf. Commun.*, vol. 8, no. 3, pp. 53–70, Mar. 2011.

[10] F. Kudo, T. Yoshikawa, and T. Furuhashi, "A study on analysis of design variables in Pareto solutions for conceptual design optimization problem of hybrid rocket engine," in *Proc. Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, 2011, pp. 2558–2562.

[11] M. Sebag, N. Tarrisson, O. Teytaud, J. Lefevre, and S. Baillet, "A multi-objective multi-modal optimization approach for mining stable spatio-temporal patterns," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, Edinburgh, Scotland, 2005, pp. 859–864.

[12] R. Tanabe and H. Ishibuchi, "A review of evolutionary multimodal multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 193–200, Feb. 2020.

[13] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of EMOA to detect and preserve equivalent Pareto subsets," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Matsushima, Japan, 2007, pp. 36–50.

[14] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, vol. 3410. Guanajuato, Mexico, 2005, pp. 41–65.

[15] Y. Liu, H. Ishibuchi, Y. Nojima, N. Masuyama, and K. Shang, "A double-niched evolutionary algorithm and its behavior on polygon-based problems," in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, Coimbra, Portugal, 2018, pp. 262–273.

[16] J. J. Liang, C. T. Yue, and B. Y. Qu, "Multimodal multi-objective optimization: A preliminary study," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, 2016, pp. 2454–2461.

[17] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2," in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, Birmingham, U.K., 2004, pp. 742–751.

[18] Y. Liu, G. G. Yen, and D. Gong, "A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 660–674, Aug. 2019.

[19] Y. Liu, H. Ishibuchi, G. G. Yen, Y. Nojima, and N. Masuyama, "Handling imbalance between convergence and diversity in the decision space in evolutionary multimodal multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 551–565, Jun. 2020.

[20] C. Yue, B. Qu, and J. Liang, "A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 805–817, Oct. 2018.

[21] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[22] C. Hu and H. Ishibuchi, "Incorporation of a decision space diversity maintenance mechanism into MOEA/D for multi-modal multi-objective optimization," in *Proc. Genet. Evol. Comput. Conf. Companion (GECCO)*, Kyoto, Japan, 2018, pp. 1898–1901.

[23] R. Tanabe and H. Ishibuchi, "A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization," in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, Coimbra, Portugal, 2018, pp. 249–261.

[24] T. Ulrich, J. Bader, and L. Thiele, "Defining and optimizing indicator-based diversity measures in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, Kraków, Poland, 2010, pp. 707–717.

[25] H. Ishibuchi, M. Yamane, N. Akedo, and Y. Nojima, "Two-objective solution set optimization to maximize hypervolume and decision space diversity in multiobjective optimization," in *Proc. 6th Int. Conf. Soft Comput. Intell. Syst. 13th Int. Symp. Adv. Intell. Syst. (SCIS)*, Kobe, Japan, 2012, pp. 1871–1876.

[26] R. Tanabe and H. Ishibuchi, "A niching indicator-based multi-modal many-objective optimizer," *Swarm Evol. Comput.*, vol. 49, pp. 134–146, Sep. 2019.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery. Data Min. (KDD)*, Portland, OR, USA, 1996, pp. 226–231.

[28] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 1999, pp. 49–60

[29] R. Fuller, *Synergetics—Explorations in the Geometry of Thinking*, vol. 1. New York, NY, USA: Macmillan, 1975.

[30] R. Fuller, *Synergetics—Explorations in the Geometry of Thinking*, vol. 2. New York, NY, USA: Macmillan, 1979.

[31] K. Zhang, Z. W. Xu, S. L. Xie, and G. G. Yen, "Evolution strategy-based many-objective evolutionary algorithm through vector equilibrium," *IEEE Trans. Cybern.*, early access, Jan. 10, 2020, doi: 10.1109/TCYB.2019.2960039.

[32] K. Zhang, C. N. Shen, X. M. Liu, and G. G. Yen, "Multiobjective evolution strategy for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 974–988, Oct. 2020.

[33] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Nat. Comput.*, vol. 1, no. 1, pp. 3–52, Mar. 2002.

[34] W. Zhang, G. Li, W. Zhang, J. Liang, and G. G. Yen, "A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization," *Swarm Evol. Comput.*, vol. 50, no. 2, 2019, Art. no. 100569.

[35] J. Liang, M. Gong, H. Li, T. Y. Cai, and B. Y. Qu, "Problem definitions and evaluation criteria for the CEC special session on evolutionary algorithms for sparse optimization," Dept. Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Rep. 2018001, 2018.

[36] J. J. Liang, B. Y. Qu, D. W. Gong, and C. T. Yue, "Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization," Zhengzhou Univ., Zhengzhou, China, Rep. 201912, 2019.

[37] H. Ishibuchi, N. Akedo, and Y. Nojima, "A many-objective test problem for visually examining diversity maintenance behavior in a decision space," in *Proc. 13th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Dublin, Ireland, 2011, pp. 649–656.

[38] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

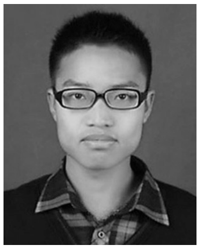
[39] A. Zhou, Q. Zhang, and Y. Jin, "Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1167–1189, Oct. 2009.

[40] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.

[41] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1062–1087, Mar. 2008.

Kai Zhang received the Ph.D. degree in system analyses and integrate from the Huazhong University of Science and Technology, Wuhan, China, in 2008.

He was a Postdoctoral Research Fellow with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China, from 2008 to 2010. He is currently a Professor with the School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan. His research focuses on evolutionary computation and multicriteria decision making.



Chaonan Shen received the master's degree in control engineering from Zhengzhou University of Light Industry, Zhengzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-time IndustrialSystem, Wuhan University of Science and Technology, Wuhan, China.

His research interests include evolutionary computation, and many-objective optimization.

Gary G. Yen (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1992.

He was with the Structure Control Division, U.S. Air Force Research Laboratory, Albuquerque, NM, USA. In 1997, he joined Oklahoma State University, Stillwater, OK, USA, where he is currently a Regents Professor with the School of Electrical and Computer Engineering. His research interest includes intelligent control, computational intelligence, conditional health monitoring, signal processing and their industrial/defense applications.

Dr. Yen received the Andrew P Sage Best Transactions Paper award from IEEE Systems, Man and Cybernetics Society, in 2011, and the Meritorious Service award from IEEE Computational Intelligence Society in 2014. He was an Associate Editor of the *IEEE Control Systems Magazine*, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, *Automatica*, *Mechantronics*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON CYBERNETICS. He served as the General Chair for the 2003 IEEE International Symposium on Intelligent Control held in Houston, TX, and 2006 IEEE World Congress on Computational Intelligence held in Vancouver, Canada. He served as the Vice President for the Technical Activities in 2005 and 2006 and President in 2010 and 2011 of the IEEE Computational intelligence Society and is the Founding Editor-in-Chief of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE from 2006 to 2009. He is a Fellow of IET.

Zhiwei Xu (Graduate Student Member, IEEE) received the B.S. degree in information security from the Wuhan University of Science and Technology, Wuhan, China, in 2017. He is currently pursuing the Ph.D. degree with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-time IndustrialSystem, Wuhan University of Science and Technology, Wuhan, China.

His research interests include evolutionary computation, and many-objective optimization.

Juanjuan He (Member, IEEE) received the Ph.D. degree in engineering from the School of Automation, Huazhong University of Science and Technology, Wuhan, China, in 2014.

She is an Associate Professor of Computer Science with the Wuhan University of Science and Technology, Wuhan. She has been invited to the Department of Computer Science, Western University, London, ON, Canada, as a Visiting Professor for 24 months. Her primary research interests lie in the fields of computational intelligence, machine learning, membrane computing, and various application domains.