Optimizing Niche Center for Multimodal Optimization Problems

Yi Jiang, *Student Member, IEEE*, Zhi-Hui Zhan, *Senior Member, IEEE*, Kay Chen Tan, *Fellow, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Many real-world optimization problems require searching for multiple optimal solutions simultaneously, which are called multimodal optimization problems (MMOPs). For MMOPs, the algorithm is required both to enlarge population diversity for locating more global optima and to enhance refine ability for increasing the accuracy of the obtained solutions. Thus, numerous niching techniques have been proposed to divide the population into different niches, and each niche is responsible for searching on one or more peaks. However, it is often a challenge to distinguish proper individuals as niche centers in existing niching approaches, which has become a key issue for efficiently solving MMOPs. In this article, the niche center distinguish (NCD) problem is treated as an optimization problem and an NCD-based differential evolution (NCD-DE) algorithm is proposed. In NCD-DE, the niches are formed by using an internal genetic algorithm (GA) to online solve the NCD optimization problem. In the internal GA, a fitness-entropy measurement objective function is designed to evaluate whether a group of niche centers (i.e., encoded by a chromosome in the internal GA) is promising. Moreover, to enhance the exploration and exploitation abilities of NCD-DE in solving the MMOPs, a niching and global cooperative mutation strategy that uses both niche and population information is proposed to generate new individuals. The proposed NCD-DE is compared with some state-of-the-art and recent well-performing algorithms. The experimental results show that NCD-DE achieves better or competitive performance on both the accuracy and completeness of the solutions than the compared algorithms.**

Manuscript received June 1, 2021; revised September 28, 2021; accepted October 23, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102102; in part by the National Natural Science Foundations of China (NSFC) under Grant 62176094, Grant 61822602, Grant 61772207, and Grant 61873097; in part by the Key-Area Research and Development of Guangdong Province under Grant 2020B010166002; in part by the Guangdong Natural Science Foundation Research Team under Grant 2018B030312003; and in part by the National Research Foundation of Korea under Grant NRF-2021H1D3A2A01082705. This article was recommended by Associate Editor S. Das. *(Corresponding authors: Zhi-Hui Zhan; Jun Zhang.)*

Yi Jiang and Zhi-Hui Zhan are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Pazhou Laboratory, Guangzhou 510330, China, and also with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China (e-mail: zhanapollo@163.com).

Kay Chen Tan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: kctan@polyu.edu.hk).

Jun Zhang is with the Hanyang University (ERICA Campus), Ansan 15588, South Korea, and also with the Chaoyang University of Technology, Taichung 41349, Taiwan (e-mail: junzhang@ieee.org).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3125362.

Digital Object Identifier 10.1109/TCYB.2021.3125362

***Index Terms*—Archive technique, differential evolution (DE), dual-scale local search strategy, fitness-entropy measurement (FEM), multimodal optimization problems (MMOPs), multiple peaks, niche center distinguish (NCD), niching and global cooperative mutation (NGCM) strategy.**

I. Introduction

SEARCHING for multiple optima is preferred in many real-world applications, such as machine learning [1], [2], pattern classification [3]–[5], protein structure prediction [6], and electromagnetic design [7]. These problems are called multimodal optimization problems (MMOPs), which require finding multiple optimal solutions. However, classical evolutionary computation (EC) algorithms, including genetic algorithm (GA) [8]–[10], differential evolution (DE) [11]–[14], and particle swarm optimization (PSO) [15]–[17], are originally proposed for convergence to a single global optimum and are thus not suitable for directly solving MMOPs. Therefore, it is necessary to modify and extend classical EC algorithms to locate as many optima as possible when solving MMOPs [18].

To extend EC algorithms for solving MMOPs, the niching technique is commonly adopted to preserve the diversity of individuals in many studies [19]–[24]. For instance, the niching methods include crowding [22], fitness sharing [25], speciation [26]–[28], clustering [29], [30], hill-valley [31], and recursive middling [32]. The niching technique partitions the entire population into many niches, and each niche deals with one or a few global optima in a small region. A fundamental problem in niching techniques is how to create niches properly. Generally speaking, the difficulties of creating proper niches include how many niches should be partitioned and where the niche centers should be. First, the proper number of niches enables the algorithm to locate all the optima efficiently. Second, the proper location of the niche center can help the individuals belonging to this niche center to speed up converging to the corresponding optimum. However, the number of niches and the locations of the niche centers are usually different in different problems and should be accommodated with the distribution of the population during the different evolutionary states. This difference is attributed to the fact that niches depend on not only the landscape of the problem but also the current population distribution. Therefore, it is a challenge to design a method to distinguish the optimal niche centers (including the number of niches and the location of the niche centers) according to not only the MMOP's landscape but also the distribution of the current population.

To address this issue, we propose an interesting and effective idea by transforming the niche center distinguish (NCD) problem into an optimization problem. That is, to distinguish the optimal niche centers to partition the population into a set of promising niches, all the individuals are first regarded as the candidates to be the niche centers. The NCD problem is transformed to a 0/1 binary optimization problem to determine which individuals are selected as the niche centers (i.e., encoded as 1). Since GA can efficiently solve the 0/1 binary optimization problem, an internal GA is designed to optimize the NCD optimization problem. Note that in this transformed optimization problem, both the number of niches (i.e., how many individuals are selected) and the locations of the niches (i.e., which individuals are selected) are not fixed and are determined by the internal GA.

Therefore, this study proposes a ground-breaking and easy-extended idea for dealing with MMOPs. Transforming the niche center selecting problem into an optimization problem is a novel method for niching. We do not need to design ad hoc niching approaches for different MMOPs but only design the objective function of the NCD optimization to determine the niche centers, that is, to find out the niche centers with good quality and diversity.

Based on the above idea and method, the NCD problem is solved to provide suitable niche centers and is integrated with the DE algorithm to efficiently solve MMOPs. Therefore, we propose a novel and efficient NCD-based DE (called NCD-DE) algorithm. The characteristics and advantages of NCD-DE are described as follows.

1) During the evolution, the NCD problem is treated as an optimization problem. To efficiently solve the NCD optimization problem, an internal GA is incorporated to select suitable individuals with both good quality and diversity as the niche centers automatically. In the internal GA, a novel fitness-entropy measurement (FEM) that both considers the quality and diversity is proposed to evaluate the selected niche centers. Since the population distribution is different in different problems and different evolutionary stages, dynamically selecting promising individuals as niche centers can satisfy the search requirements of different problems and different evolutionary stages. Note that the chromosomes in internal GA are evaluated via the FEM objective function rather than the objective function of the MMOP, thus the internal GA does not consume the fitness evaluations of the MMOP.
2) A niching and global cooperative mutation (NGCM) strategy that uses both niche-based and population-based information is designed to improve the abilities of exploration and exploitation. On the one hand, a niching mutation strategy with dual-scale local search is used in NGCM to enhance the capability of exploitation to speed up convergence. On the other hand, a global mutation strategy is used in NGCM to enable individuals to jump out from the current stagnate to another potential peak to better explore the search space.
3) During the evolution process, individuals may jump from one niche to another peak, leading to the extinction of some optimal solutions. Therefore, the archive strategy is applied to retain the elite individuals from extinction.

The remainder of this article is organized as follows. In Section II, we introduce several relative works on MMOPs and the framework of basic DE are also presented. In Section III, we describe the NCD-DE and analyze the time complexity of NCD-DE. The experimental studies are provided in Section IV. Finally, we conclude and summarize this article in Section V.

II. Background

A. Multimodal Optimization

To efficiently solve the MMOPs, multiple studies have been devoted in recent years. On the one hand, to locate as many peaks as possible, these proposed algorithms have to maintain good population diversity. On the other hand, since the number of function evaluations (FEs) is limited, convergence speed must also be considered to increase solutions' accuracy. Therefore, the niching method, in which the population is partitioned to form several niches and each niche evolves independently, has been widely incorporated with EC algorithms to search for multiple optima. The famous crowding [22] and speciation [26]–[28] techniques serve as examples. The crowding technique allows the offspring to compete with the most similar individual among the randomly selected C parental individuals (C is the crowd size). If the offspring is fitter, the parent is replaced; otherwise, the offspring is discarded. In the speciation approach, each subpopulation (specie) is formed based on a predefined radius r and each offspring is generated via evolutionary operations within its species.

Although the effectiveness and efficiency of the two niching techniques have been demonstrated in crowding-based DE (CDE) [22] and speciation-based DE (SDE) [26], two limitations remain. First, the results of the two techniques highly depend on the parameter settings. Second, experimental results show that they are not feasible for the problem whose search space is complex [33], [34], such as the high-dimensional problem or the problem that contains many local optima. Therefore, some enhanced versions of these two niching techniques have been proposed. For instance, some studies have combined crowding and speciation techniques with the clustering technique to propose the neighborhood-based CDE/SDE (NCDE and NSDE) [29], as well as the self-adaptive clustering-based CDE/SDE (Self_CCDE and Self_CSDE) [30]. However, in these algorithms, an additional parameter (i.e., clustering size) is also introduced. In addition, the clustering technique is too deterministic to distinguish promising niches in some situations [33]. Even if ignoring the disadvantages of the above algorithms, the effects of the "no free lunch" theorem still encourage researchers to design suitable algorithms for particular problems.

To enhance the performance, some niche parameter-insensitive approaches have drawn considerable attention. For example, an improved information sharing method is incorporated with niching DE (LoINDE) to induce efficient niching behavior in [35]. Yang *et al.* [33] took advantage of the estimation of distribution algorithm (EDA) and incorporated it with a clustering strategy, forming the LMCEDA and LMSEDA algorithms. Wang *et al.* [34] proposed an automatic niching DE (ANDE) to form the niches via affinity propagation clustering. Lin *et al.* [36] developed a DE-based algorithm that adopted nearest-better clustering to partition the population and species balance strategy to fine-tune the species' size. Chen *et al.* [37] proposed a distributed individuals DE (DIDE) algorithm. In DIDE, each individual is assigned with a virtual subpopulation to search for the optimal solution in a small region. Luo *et al.* [38] developed a nearest-better-neighbor clustering strategy and incorporated it with the PSO and the evolutionary strategy to better explore and exploit the search space. Huang *et al.* [39] designed a probabilistic niching framework for MMOPs, in which the historical information is stored in an enhanced binary space partition tree to help enhance the exploration and exploitation abilities.

In addition, some studies have focused on tackling MMOPs via multiobjective techniques. Yao *et al.* [40] and Deb and Saha [41] proposed to transform the MMOPs into multiobjective problems, in which the objectives are generated according to the fitness and the gradient. To maintain the population's elitism and diversity, Wang *et al.* [42] proposed a multiobjective optimization for MMOPs (MOMMOPs), in which two conflicting objectives were designed to address MMOPs. Cheng *et al.* [43] proposed to transform the MMOP into a biobjective optimization problem, whose objective functions are the fitness for MMOP and the grid-based diversity indicator.

Although many MMOP algorithms have been proposed to date, there are still some deficiencies, such as confronting difficulties in tackling high-dimensional problems or many-optima problems. To overcome these deficiencies, enhancing the exploitation (to reach the elitism of the accuracy of the solutions) ability and improving the exploration (to locate all global optima regions) ability of MMOP algorithms are essential in solving MMOPs.

B. Differential Evolution

The evolution process in DE [12] is executed via the calculations of the difference between individuals. Similar to other EC algorithms, at the initialization stage, individuals are randomly generated in the search space as

$$\boldsymbol{x}_{i,j} = \text{rand}(0, 1) \times \left(U_j - L_j\right) + L_j \tag{1}$$

where $\boldsymbol{x}_{i,j}$ indicates the jth element of the ith individual, and U_j and L_j are the upper bound and lower bound of the jth dimension, respectively. Here, rand(0, 1) is generated uniformly from interval [0, 1]. After initialization, the evolutionary operations in the main loop of DE, which are mutation, crossover, and selection operators, are executed iteratively before the termination.

The mutation operator generates mutation vector $\boldsymbol{v}_i$ according to the difference between randomly selected individuals. Some well-known mutation strategies are shown as follows.

1) *DE/rand/1:*

$$\boldsymbol{v}_i = \boldsymbol{x}_{r1} + F \times (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{2}$$

2) *DE/rand/2:*

$$\boldsymbol{v}_i = \boldsymbol{x}_{r1} + F \times (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) + F \times (\boldsymbol{x}_{r4} - \boldsymbol{x}_{r5}) \tag{3}$$

3) *DE/best/1:*

$$\boldsymbol{v}_i = \boldsymbol{x}_{best} + F \times (\boldsymbol{x}_{r1} - \boldsymbol{x}_{r2}) \tag{4}$$

4) *DE/current-to-rand/1:*

$$\boldsymbol{u}_i = \boldsymbol{x}_i + \text{rand}(0, 1) \times (\boldsymbol{x}_{r1} - \boldsymbol{x}_i) + F \times (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{5}$$

where the parameter F represents the amplifier factor. $r1$, $r2$, $r3$, $r4$, and $r5$ are five different indexes, which are randomly selected. $\boldsymbol{x}_{best}$ indicates the fittest individual.

In the crossover operator, the trial vector $\boldsymbol{u}_i$ is generated via the recombination of the mutant vector $\boldsymbol{v}_i$ and the parental individual $\boldsymbol{x}_i$, as follows:

$$\boldsymbol{u}_{i,j} = \begin{cases} \boldsymbol{v}_{i,j}, & \text{if rand}(0, 1) \leq CR \text{ or } j == j_{\text{rand}} \\ \boldsymbol{x}_{i,j}, & \text{otherwise} \end{cases} \tag{6}$$

where CR indicates the crossover rate, and j_{rand} is a randomly selected integer in [1, D] to avoid the trial vector $\boldsymbol{u}_i$ being the same as the parental individual $\boldsymbol{x}_i$.

After the crossover, the selection operator is applied to retain the optimal individual from the offspring and the parent. If the fitness of the offspring $\boldsymbol{u}_i$ is better than that of the parental individual $\boldsymbol{x}_i$, the individual $\boldsymbol{x}_i$ is replaced by $\boldsymbol{u}_i$. When dealing with a maximization optimization problem, the selection operator is shown as follows:

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{u}_i, & \text{if } f(\boldsymbol{u}_i) > f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i, & \text{otherwise} \end{cases} \tag{7}$$

where $f(.)$ represents the fitness function.

C. Motivation

How to optimally divide the population into several niches is the key issue to improve the performance of the MMOP algorithms. As the related works reviewed above, some clustering approaches have been proposed to divide the population into several niches. Although the generated niches are suitable for effectively and efficiently solving the MMOP, there are still gaps between the generated niches and the optimal niches for the MMOP.

We note that finding the optimal niches is equivalent to searching for the optimal niches among multiple candidate options. Also, the aiming of solving an optimization problem is searching out an optimal solution among the search space. Considering this, we can build an optimization problem whose optimal solution is corresponding to the optimal niches. However, it is hard to construct an optimization problem that is directly relative to the niche-dividing problem. In MMOP, we can divide the population into niches by first selecting several individuals as niche centers and then assigning the other individuals to the nearest niche centers to generate niches. The problem of selecting optimal niche centers can

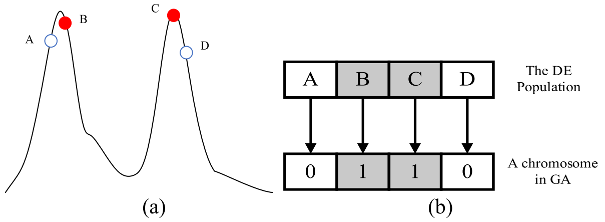


Fig. 1. Relationship between a chromosome in GA and the population of DE. (a) Landscape of a population contains four individuals in DE. Red solid points (B and C) are two feasible niche centers, whereas the two blue hollow points (A and D) are not niche centers. (b) Chromosome represents the distinguished niche centers.

be intuitively transformed into a binary-encoded optimization problem. Solving the binary-encoded optimization problem can provide a set of promising niche centers, and the niches can be generated according to these niche centers. Based on the above consideration, we can propose a novel and interesting idea that automatically divides the population into several niches via solving the optimization problem. The remained problems only include how to select the suitable optimization algorithm and the objective function to evaluate the niche centers, which are solved in the next section.

III. NCD-DE

This section details the proposed NCD-DE algorithm. In this section, the first part describes the NCD optimization. The NCD optimization problem is solved by a GA, namely, "internal GA," to distinguish the optimal niche centers and partition the population according to the niche centers. In internal GA, the FEM objective function is designed to evaluate the selected niche centers. The NGCM strategy, including the niching mutation strategy with dual-scale local search to improve the exploitation ability and global mutation strategy to enhance the exploration ability, is introduced later. Then, the description of the archive technique is given. Furthermore, the complete NCD-DE algorithm pseudocode and the advantages are provided. Finally, the time complexity of NCD-DE is analyzed.

A. NCD Optimization

1) Binary Encoding of Chromosomes: When using NCD-DE to solve the MMOPs, the NCD optimization is carried out in every generation to determine the optimal niche centers. The NCD problem is transformed into an optimization problem and is solved by an internal GA approach. To avoid confusion, a solution in GA is called a chromosome, whereas a solution in DE is called an individual.

The chromosome of the internal GA is encoded as a 0/1 binary vector with the length as *NP*, where *NP* is the number of individuals in DE. Each gene in the chromosome is corresponding to an individual of the DE population. If the value of the gene is 1, then the corresponding individual is distinguished as a niche center; otherwise, if the gene is 0, the corresponding individual is not selected as a niche center. For example, as illustrated in Fig. 1, the DE population with the size of 4 contains individuals A, B, C, and D, and a feasible set of niche centers are individuals B and C, so that the corresponding chromosome is [0, 1, 1, 0]. This way, each chromosome of the internal GA encodes a set of feasible niche centers and the optimal niche centers can be optimized by internal GA. Therefore, the number of niche centers is determined by number of 1s in the chromosome. Note that a niche center does not need to be the best individual in its niche, but the entire set of niche centers needs to make the corresponding chromosome in internal GA have a good quality to partition the population.

2) FEM Objective Function: The quality of a chromosome is determined by the objective function of internal GA. A set of promising niche centers should possess both good fitness and large diversity. First, the fitness of the individuals that are selected as the niche centers should be good enough since a niche center with better fitness means it is more likely to dominate a peak, enhancing the exploitation ability around this niche center. Second, the diversity of all the selected niche centers should be large. That is, because niche centers with more enormous diversity mean that they can cover a broader search space range, which helps discover more global peaks. Therefore, the objective function in internal GA should have two parts, which both focus on the fitness and the diversity of the selected niche centers. The fitness measurement of the selected individuals can be directly reflected by their fitness values on the MMOP, but the diversity measurement should be designed prudently.

Many MMOP algorithms [22], [23] use the Euclidean distance to measure the diversity among individuals. According to the chromosome in GA, the average Euclidean distance among selected niche centers can be calculated as follows:

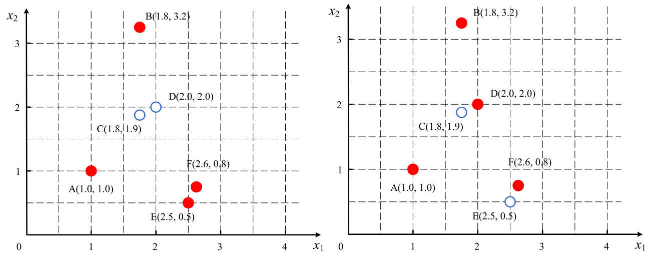
$$\text{avgDistance}(Ch_i) = \frac{1}{NN} \sum_{j=0;\, Ch_{i,j}==1}^{NP} \text{distance}(j) \tag{8}$$

where $Ch_{i,j}$ is the jth gene on the ith chromosome. Since we only calculate the average Euclidean distance among the niche centers, the distance (j) is added when the value of $Ch_{i,j}$ equals 1. *NN* is the number of niche centers distinguished by this chromosome, which can be calculated as number of 1s in the chromosome. Here, distance(j) is the average distance from the jth individual to other niche centers. The formula of distance(j) is given as follows:

$$\text{distance}(j) = \frac{1}{NN} \sum_{k=0;\, Ch_{i,k}==1}^{NP} \text{dis}\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right) \tag{9}$$

where dis($\boldsymbol{x}_j$, $\boldsymbol{x}_k$) is the value of the Euclidean distance between individuals $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$. Intuitively, suppose the average Euclidean distance of a group of niche centers is relatively larger, niche centers can be distributed in a broad region, where most of the peaks might be covered. Therefore, the average Euclidean distance seems to be a feasible measurement for the niche centers' diversity.

However, the average Euclidean distance may not be able to distinguish the most suitable niche centers. That is, because a relatively larger value of the average Euclidean distance only indicates the niche centers are distributed in a broad region, while the niche centers may not be uniformly distributed in



(a) (b)

Fig. 2. Two examples of selecting four niche centers among six individuals, and the niche centers are colored red and solid. (a) Individuals A, B, E, and F are selected as niche centers. However, E and F might belong to one niche. (b) Individuals A, B, D, and F are selected as niche centers. The four niche centers are distributed diversely.

the region. For example, as illustrated in Fig. 2, there are six individuals in the DE population, and both GA chromosomes in Fig. 2(a) and (b) select four individuals as niche centers, which are (A, B, E, F) and (A, B, D, F), respectively. If the Euclidean measurement is applied, that is, (8), the average distance of niche centers in Fig. 2(a) is 1.396. In Fig. 2(b), the average distance is 1.307. In such a case, the niche centers in Fig. 2(a) are preferred because their average distance is relatively large. However, as we can see from these two figures and based on common sense, the distribution of niche centers in Fig. 2(b) is preferred because the niche centers are uniformly distributed and dominate a broad region. Therefore, the niche centers in Fig. 2(b) are more dispersed than those in Fig. 2(a), but the Euclidean distance diversity measurement may select niche centers with poor diversity.

To select the most suitable niche centers to maintain diversity, another diversity measurement, the entropy measurement, is employed in our proposed algorithm. The calculation of average entropy among niche centers is shown as

$$\text{avgEntropy}(Ch_i) = \frac{1}{NN} \sum_{j=0;\, Ch_{i,j}==1}^{NP} \text{entropy}(j) \tag{10}$$

where the entropy(j) is the entropy function of the jth individual. The value of entropy of the jth individual can be calculated as

$$\text{entropy}(j) = \frac{1}{NN} \sum_{k=0;\, k\neq j,\, Ch_{i,k}==1}^{NP} -p_{j,k} \times \ln\left(p_{j,k}\right) \tag{11}$$

$$p_{j,k} = \frac{\exp\left(-\text{dis}\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right)\right)}{\sum_{z=0;\, z\neq j,\, Ch_{j,z}==1}^{NP} \exp\left(-\text{dis}\left(\boldsymbol{x}_j, \boldsymbol{x}_z\right)\right)} \tag{12}$$

where $p_{j,k}$ can be viewed as the similarity between individuals $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$. exp(.) represents the exponential function, and ln(.) is the natural logarithm function. Note that the entropy function is calculated only when $Ch_{j,k}$ equals 1. A larger entropy indicates that the niche centers are with higher diversity.

When encountered with the case noted in Fig. 2, the entropy measurement of (10) can obtain a better result than the average Euclidean distance measurement. The average entropy of niche centers in Fig. 2(a) is 0.196. In Fig. 2(b), the average entropy is 0.202. Therefore, the selection of niche centers in Fig. 2(b) is preferred. This finding indicates that the entropy measurement shows better performance than the Euclidean distance measurement in the above situation. Moreover, to demonstrate the general performance of the entropy measurement, comparison experiments between Euclidean distance and entropy are conducted on commonly used CEC2013 MMOP benchmark functions. The experimental results can be found in Section IV-C.

Algorithm 1: Internal GA

```
Input: Size of GA population $m_{GA}$; length of the chromosome $NP$;
       crossover rate $P_c$; mutation rate $P_m$; iteration epochs $T_{GA}$;
Output: The best chromosome
Begin
1:  Randomly initialize and evaluate the population $P$ of size $m_{GA}$;
2:  For $t = 1$ to $T_{GA}$
3:      Initialize two empty sets $Q$ and $S$;
4:      Evaluate the population $P$;
5:      For $i = 1$ to $m_{GA}$
6:          Select two parents from $P$ by binary tournament selection;
7:          If rand(0, 1) $\leq P_c$
8:              Crossover the parents to generate two offspring, then
                randomly choose an offspring, mutate it and add it to $Q$;
9:          Else
10:             Choose the better parent, mutate it and add it into $Q$;
11:         End If
12:     End For
13:     Add the chromosome with the best FEM from $P \cup Q$ into $S$;
14:     For $j = 1$ to $m_{GA} - 1$
15:         Select a chromosome from $P \cup Q$ by binary tournament
            selection and add it into $S$;
16:     End For
17:     $P \leftarrow S$;
18: End For
19: Return the best chromosome of $P$;
End
```

Based on the above case studies, the FEM objective function, including the fitness part and the entropy part for internal GA, is proposed and shown as follows:

$$\text{FEM}(Ch_i) = \frac{1}{NN^2} \sum_{j=0;\, Ch_{i,j}==1}^{NP} \text{fitness}(j) \times \text{entropy}(j) \tag{13}$$

where fitness(j) indicates the fitness of individual $\boldsymbol{x}_j$ to accomplish the goal of making the fitness of niche centers better, and entropy(j) is the entropy given in (11) between individual $\boldsymbol{x}_j$ and other niche centers. The evolution target of each chromosome in the internal GA is to make its FEM value optimal, which indicates that the corresponding niche centers not only obtain better fitness but also yield promising diversity.

3) Framework of Internal GA: To efficiently solve the 0/1 optimization problem above, the internal GA is carried out as a variant of the classical GA [8]–[10]. The internal GA is executed to distinguish a set of niche centers with maximum FEM value at the beginning of each DE generation. Similar to the basic GA, the internal GA maintains a population of chromosomes to search for the optimal niche centers via crossover, mutation, and selection operators.

The process of internal GA is described in Algorithm 1. First, m_{GA} chromosomes that contain information about feasible niche centers are randomly initialized at the beginning, and each gene is a 0/1 binary number (line 1). After the initialization, the main loop (lines 2–18) of the evolution process

Algorithm 2: Population Partition

```
    Input: Size of population $NP$; The best chromosome $Ch_{Best}$ from
        internal GA
    Output: The niches after partition
    Begin
1:  For $i$ = 1 to $NP$
2:    If $Ch_{Best,i}$ == 1
3:      Add this individual into the set of niche centers;
4:    End If
5:  End For
6:  For $i$ = 1 to $NP$
7:    If $Ch_{Best,i}$ == 0
8:      Add this individual to the niche of the nearest niche center;
9:    End If
10: End For
11: Return the niches;
    End
```

in the internal GA is executed for T_{GA} generations. After the evaluation of population P according to FEM (line 4), the crossover and mutation operators are executed (lines 5–12). Specifically, two parental chromosomes are selected via binary tournament selection (line 6). Later, the crossover and the mutation operation are executed orderly (lines 7–11).

With the crossover probability P_c, the crossover operator is executed on the two parental chromosomes as follows:

$$Ch_{o1,j} = \begin{cases} Ch_{p1,j}, & \text{if } j \le j_{\text{rand}} \\ Ch_{p2,j}, & \text{otherwise} \end{cases} \tag{14}$$

$$Ch_{o2,j} = \begin{cases} Ch_{p2,j}, & \text{if } j \le j_{\text{rand}} \\ Ch_{p1,j}, & \text{otherwise} \end{cases} \tag{15}$$

where $Ch_{p1,j}$ and $Ch_{p2,j}$ are the jth genes of the parental chromosomes, while $Ch_{o1,j}$ and $Ch_{o2,j}$ are the jth genes of the offspring chromosomes. j_{rand} is the index of a randomly selected dimension, indicating the crossover position of the chromosomes. After that, one of the offspring is randomly selected and forwarded to the mutation operator. Otherwise, if the random number rand(0, 1) is larger than P_c, the better one of the parental chromosomes is selected and forwarded to the mutation operator without crossover.

In the mutation operator, each gene of the offspring chromosomes can be changed to another value according to the mutation probability P_m. Due to the 0/1 binary encoding of the chromosomes, each gene is either 0 or 1. Therefore, if the current gene is 0, its value is changed to 1 after mutation and vice versa, as follows:

$$Ch_{i,j} = \begin{cases} 1 - Ch_{i,j}, & \text{if rand}(0, 1) \le P_m \\ Ch_{i,j}, & \text{otherwise} \end{cases} \tag{16}$$

where P_m is the value of the mutation rate.

After crossover and mutation, in line 13, the best chromosome of the union set of the population P and the offspring Q is directly added in the next generation population S. Then, other $m_{GA} - 1$ chromosomes of the next generation are chosen via binary tournament selection from the union set of P and Q (lines 14–16). Finally, the internal GA returns the chromosome with the largest FEM value.

4) Population Partition to Form Niches: Until the completion of the internal GA, the population in DE is partitioned into several niches according to the optimal niche centers distinguished by the internal GA. Algorithm 2 is the pseudocode of the population partition method. First, the niche centers are selected according to the best chromosome Ch_{Best} generated from the internal GA (lines 1–5). Then, each individual that is not a niche center is added to the niche whose niche center is nearest to this individual (lines 6–10).

B. NGCM Strategy

In many MMOP algorithms [23], [34], [36], the niching-based DE mutation strategy generates a differential vector according to the parents chosen in the same niche. The niching mutation strategy can better exploit the local region to enhance the fitness value of the individuals. If the niching mechanism is not adopted in DE, the global mutation strategy [i.e., DE/rand/1 in (2)] generates differential vectors according to the parents globally selected from all individuals. This way, the search space is enlarged, individuals can migrate from one peak to another remote peak. Therefore, the global mutation strategy in DE can better explore more potential peaks. To combine the advantages of both the niching mutation and the global mutation, the NGCM strategy composed of the niching mutation strategy with dual-scale local search and the global mutation strategy is designed.

First, the niching mutation strategy with dual-scale local search is described, which corresponds to lines 7–22 in Algorithm 3. If the niche size is not smaller than three, the niching mutation strategy is adopted in this niche to generate offspring as follows:

$$\boldsymbol{v}_i = \boldsymbol{x}_{nr1} + F \times (\boldsymbol{x}_{nr2} - \boldsymbol{x}_{nr3}) \tag{17}$$

where $nr1$, $nr2$, and $nr3$ are three different indexes that are randomly selected in the niche of the individual $\boldsymbol{x}_i$. After niching mutation, the crossover operator in (6) is executed to recombine a new offspring. Once the offspring is created, it is forwarded into the crowding-based selection operator.

The crowding-based selection strategy is suitable here for the following reasons. As the generated offspring $\boldsymbol{u}_i$ may be far away from its parental individual $\boldsymbol{x}_i$, directly applying the selection operator in (7) may not be reasonable because $\boldsymbol{u}_i$ and $\boldsymbol{x}_i$ are more likely in the different niches. In such a situation, the crowding-based selection strategy that selects a better one within the offspring $\boldsymbol{u}_i$ and its nearest individual $\boldsymbol{x}_{nn}$ may be more suitable. When dealing with a maximization optimization problem, the crowding-based selection operator is shown as

$$\boldsymbol{x}_{nn} = \begin{cases} \boldsymbol{u}_i, & \text{if } f(\boldsymbol{u}_i) > f(\boldsymbol{x}_{nn}) \\ \boldsymbol{x}_{nn}, & \text{otherwise} \end{cases} \tag{18}$$

where $\boldsymbol{x}_{nn}$ is the most similar parental individual of $\boldsymbol{u}_i$. Herein, the most similar refers to the nearest individual measured by the Euclidean distance.

However, the niching mutation strategy cannot be carried out when the niche size is smaller than three. In such a situation, the dual-scale local search is performed to generate offspring. The niching mutation strategy is carried out by following (19) and (20), meaning the wide-scale and narrow-scale, respectively. Specifically, with a half probability, the offspring is generated by a wide scale as follows:

$$\boldsymbol{u}_{i,j} = \begin{cases} \boldsymbol{x}_{i,j} + \text{normal}(0, 1) \times \text{dis}(\boldsymbol{x}_i, \boldsymbol{x}_{nn}), & \text{if rand}(0, 1) \le 0.5 \\ \boldsymbol{x}_{i,j}, & \text{otherwise} \end{cases} \tag{19}$$

Algorithm 3: NCD-DE

```
    Input: Size of population $NP$; amplification factor $F$; crossover rate $CR$
        of DE; period $\lambda$;
    Output: Individuals in the archive
    Begin
1:  Randomly initialize and evaluate the population;
2:  gen = 0;
3:  While $FEs \leq MaxFEs$
4:  Execute Algorithm 1 and Algorithm 2 for niching;
5:  For each niche $S_i$
6:      Copy the best individual in $S_i$ into the archive;
7:      If size($S_i$) $\geq$ 3
            //Niching mutation strategy
8:          For each individual in $S_i$
9:              Niching mutation strategy via (17);
10:             Crossover using (6);
11:             Crowding-based selection using (18);
12:         End For
13:     Else
            //Dual-scale local search strategy
14:         For each individual in $S_i$
15:             If rand(0, 1) $\leq$ 0.5
16:                 Create offspring by wide-scale local search in (19);
17:             Else
18:                 Create offspring by narrow-scale local search in (20);
19:             End If
20:             Selection using (7);
21:         End For
22:     End If
23: End For
    //Global mutation strategy
24: If gen % $\lambda$ == 0
25:     For each individual
26:         Generate a new offspring using DE/rand/1 via (2);
27:         Crossover using (6);
28:         Crowding-based selection using (18);
29:     End For
30: End If
31: gen = gen + 1;
32: End While
33: Return archive;
    End
```

where $\boldsymbol{x}_{nn}$ is the most similar individual of $\boldsymbol{x}_i$. The normal (0, 1) is a random number sampled from the standard normal distribution, and dis($\boldsymbol{x}_i$, $\boldsymbol{x}_{nn}$) is the Euclidean distance between individuals $\boldsymbol{x}_i$ and $\boldsymbol{x}_{nn}$.

With another half probability, the new offspring is generated by a narrow scale as follows:

$$u_{i,j} = x_{i,j} + 0.5 \times \text{normal}(0, 1) \times \left(x_{nn,j} - x_{i,j}\right) \tag{20}$$

where *nn* is the index of $\boldsymbol{x}_i$'s nearest neighbor.

In the dual-scale local search strategy, as the offspring $\boldsymbol{u}_i$ is sampled from a normal distribution centralized at its parental individual $\boldsymbol{x}_i$, the offspring $\boldsymbol{u}_i$ is very similar to its parental individual $\boldsymbol{x}_i$. Therefore, the crossover operator between the new offspring $\boldsymbol{u}_i$ and its parent $\boldsymbol{x}_i$ is not necessary.

Similarly, as the new offspring $\boldsymbol{u}_i$ is very similar to the parental individual $\boldsymbol{x}_i$, the crowding-based selection is also not necessary in this case. Therefore, the basic selection operator in (7) is carried out, where the comparison is performed between the offspring $\boldsymbol{u}_i$ and its parental individual $\boldsymbol{x}_i$.

Second, the description of the global mutation strategy is given here, which corresponds to lines 25–29 in Algorithm 3. The global mutation strategy is executed periodically (i.e., by using the DE/rand/1 mutation strategy every λ generations) in the NGCM to help better discover more potential peaks. Specifically, after the interval of λ generations, each individual is updated via the global DE/rand/1 mutation strategy [i.e., (2)], and then the crossover operator in (6) and the crowding-based selection in (18) are carried out to generate offspring. Since each individual can mutate with any other individuals among the entire population, the individuals can vary in a broader region among the search space. Therefore, the global mutation strategy enables individuals to discover more potential peaks.

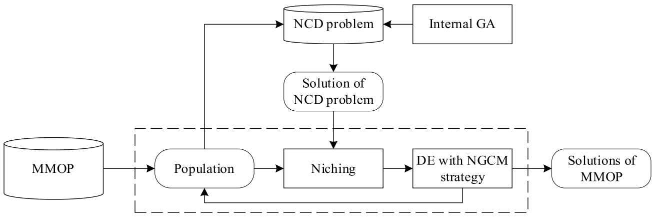


Fig. 3. Illustration of the relationship among NCD optimization, internal GA, and DE with NGCM strategy.

C. Archive Technique

Multiple algorithms designed for multiobjective optimization problems adopt an archive to preserve nondominated solutions [44]–[47]. Recently, this technique has also been used in a few MMOP algorithms to save historical information [48] or store stagnated individuals and reinitialize it to enhance diversity [49], [50].

In the proposed NCD-DE, it is observed that some niches may be extinct after the global mutation strategy because the individuals may jump to other niches or potential peaks. Even in the niching mutation strategy with dual-scale local search, the individuals may also jump to other nearby niches. Therefore, the archive technique is adopted in our algorithm to store optimal solutions. The idea of the archive technique is, after the population partition and before the NGCM strategy, the best individual in each niche is copied and stored into the archive to protect the found optimum. The effect of the archive technique is investigated in Section IV-C.

D. Complete NCD-DE

The complete NCD-DE algorithm is given in Algorithm 3. For a better understanding of the proposed algorithm, we illustrate the relationship among the NCD optimization, internal GA, and DE with NGCM strategy in Fig. 3. The advantages of the NCD-DE are summarized as follows.

1) Transforming the NCD problem into an optimization problem enables the NCD-DE to dynamically determine the number and locations of the niche centers depending on not only the optimization problem but also the distribution of the population, enhancing the performance on different problems and during different stages of the evolution.
2) Due to the FEM objective function in internal GA, the NCD optimization can select niche centers with both better quality and diversity. Compared with the average Euclidean measurement, the entropy measurement in the

FEM objective function can better measure the diversity of the niche centers, which helps to distinguish the promising niche centers.

3) The NGCM strategy can enhance exploration and exploitation. The niche mutation strategy helps to exploit the nearby region, and the convergence speed is accelerated with the dual-scale local search strategy. The global mutation strategy can discover more potential optima to enhance exploration.
4) In the evolutionary process of NCD-DE, some individuals can jump into another peak, which may lead to the extinction of some niches. The archive technique can avoid the extinction of some niches by copying and storing the best individual of each niche in the archive.

E. Complexity Analysis

To analyze the complexity of NCD-DE, we denote the size of the DE population as *NP*, the population size of internal GA as m_{GA}, the iteration times of internal GA as T_{GA}, and the dimension of the problem as *D*. The time complexity of the NCD-DE is mainly affected by the NCD optimization and the NCGM strategy. For the NCD optimization, in the worst case, all the individuals are distinguished as niche centers, and thus, the upper bound of the time complexity to calculate the FEM function for each chromosome is $O(NP^2 \times D)$. Therefore, the time complexity for evaluating all chromosomes in internal GA is $O(m_{GA} \times NP^2 \times D)$. The mutation and crossover operations in the internal GA consume $O(m_{GA} \times NP)$, and the selection operation consumes $O(NP)$. Since the time complexity of the NCD optimization is mainly dominated by that of evaluating the chromosomes, the complexity of the NCD optimization is $O(T_{GA} \times m_{GA} \times NP^2 \times D)$. The time complexity of the population partition in Algorithm 2 is $O(NP^2 \times D)$. The time complexity of the NGCM strategy is composed of the time complexity of the niching mutation strategy with dual-scale local search and the time complexity of the global mutation strategy. For the niching mutation strategy with dual-scale local search, the time complexity is $O(NP^2 \times D)$. For the global mutation strategy, the time complexity is also $O(NP^2 \times D)$. Therefore, the upper bound of the time complexity of NCD-DE is $O(T_{GA} \times m_{GA} \times NP^2 \times D)$.

For clarity, the time complexities of the recent well-performing algorithms are listed in Table S.I in the supplementary material. Due to the space limitation, Table S.I is shown in the supplementary material. In Table S.I in the supplementary material, the time complexity of most algorithms is $O(NP^2 \times D)$, the time complexity of MOMMOP algorithm is $O(2NP^2 \times D^2 + 4NP^2 \times D)$, and that of ANDE is $O(NP^2 \times D + NP^3 \times NT)$, where *NT* is the iteration times of affinity propagation clustering in ANDE.

IV. Experimental Study

A. Experiment Settings

The experiments are conducted in the well-known CEC2013 MMOP benchmark functions to validate the effectiveness and efficiency of the NCD-DE. CEC2013 MMOP benchmark contains 20 functions with multiple global optima [51]. The basic properties of the 20 CEC2013 benchmark functions are provided in Table S.II in the supplementary material.

Two commonly adopted performance metrics: 1) peak ratio (PR) and 2) success rate (SR), are adopted to evaluate the performance of NCD-DE and the compared algorithms. Under the settings of the accuracy level ε and the maximum FEs (*MaxFEs*), PR is calculated as the mean proportion of the found global optima over all the global optima in several runs of the algorithm, and SR is calculated as the percentage of the successful runs where all the global optima are found over multiple runs. The formulas used to compute PR and SR can be found in [51].

For the settings of accuracy level, three accuracy levels $\varepsilon = 1.0\text{E} - 03$, $\varepsilon = 1.0\text{E} - 04$, and $\varepsilon = 1.0\text{E} - 05$ are adopted in the experiments to evaluate the performance of NCD-DE in different accuracy levels. The detailed settings of maximum FEs *MaxFEs* and population size *NP* on different problems are shown in Table S.III in the supplementary material [33], [34]. For the parameter settings in the NCD-DE, the population size m_{GA} of the internal GA is 30, and the crossover rate P_c and mutation rate P_m of the internal GA are, respectively, specified as 0.9 and 0.1, which are two common parameter settings for GA. The iteration number T_{GA} of the internal GA is set as 5. In the mutation and crossover operators of the NGCM strategy, the amplification factor *F* is 0.5 and the crossover rate *CR* is 0.9. The period λ of the global mutation strategy is set as 5. For the experimental results, each algorithm is independently executed 51 times to obtain the mean results.

B. Comparison With State-of-the-Art MMOP Algorithms

The experimental results for PR and SR at three different accuracy levels (i.e., $\varepsilon = 1.0\text{E} - 03$, $\varepsilon = 1.0\text{E} - 04$, and $\varepsilon = 1.0\text{E} - 05$) obtained by the NCD-DE on F_1–F_{20} are provided in Table I. For clarity, the results where PR equals 1.0 are emphasized by *boldface*.

To further demonstrate the effectiveness and efficiency of NCD-DE, we compare the proposed NCD-DE with a group of state-of-the-art MMOP algorithms. These algorithms include nine DE-based algorithms, which are CDE [22], SDE [26], NCDE, NSDE [29], Self-CCDE, Self-CSDE [30], LoICDE, LoISDE [35], and ANDE [34]. Two EDA-based MMOP algorithms, including LMSEDA and LMCEDA [33], are also compared. Furthermore, a multiobjective-based MMOP algorithm MOMMOP [42] is included for comparison. Besides, NMMSO [52] that wins first place in the CEC2015 MMOP competition is also compared with NCD-DE. Note that the benchmark functions of the CEC2015 MMOP competition are the same as the CEC2013 MMOP benchmark functions [34]. The parameter settings for all the comparison algorithms are adopted from their corresponding papers. The *NP* and *MaxFEs* of all these algorithms are provided in Table S.III in the supplementary material. The results of the compared algorithms are derived from [34] and [50].

The experimental results of the NCD-DE and other algorithms at $\varepsilon = 1.0\text{E} - 04$ are provided in Table II, and the rest

TABLE I
EXPERIMENTAL RESULTS OF NCD-DE ON CEC2013 BENCHMARK FUNCTIONS $F_1 - F_{20}$ AT $\varepsilon = 1.0\text{E} - 03$, $\varepsilon = 1.0\text{E} - 04$, AND $\varepsilon = 1.0\text{E} - 05$

Func	ε= 1.0E – 03		ε= 1.0E – 04		ε= 1.0E – 05	
	PR	SR	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_2	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_3	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_4	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_5	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_6	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_7	0.913	0.078	0.905	0.078	0.895	0.059
F_8	0.965	0.118	0.961	0.098	0.956	0.059
F_9	0.572	0.000	0.553	0.000	0.535	0.000
F_{10}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{11}	**1.000**	1.000	**1.000**	1.000	**1.000**	1.000
F_{12}	0.993	0.941	0.993	0.941	0.991	0.922
F_{13}	0.902	0.471	0.892	0.412	0.882	0.412
F_{14}	0.683	0.000	0.683	0.000	0.670	0.000
F_{15}	0.654	0.000	0.640	0.000	0.632	0.000
F_{16}	0.667	0.000	0.667	0.000	0.667	0.000
F_{17}	0.522	0.000	0.522	0.000	0.522	0.000
F_{18}	0.667	0.000	0.667	0.000	0.667	0.000
F_{19}	0.505	0.000	0.505	0.000	0.502	0.000
F_{20}	0.255	0.000	0.252	0.000	0.248	0.000

of the results at $\varepsilon = 1.0\text{E} - 03$ and $\varepsilon = 1.0\text{E} - 05$ are provided in Tables S.IV and S.V in the supplementary material, respectively. To evaluate the performance of each algorithm in the statistic view, Wilcoxon's rank-sum test [53] at $\alpha = 0.05$ is adopted on the values of PR between NCD-DE and other algorithms. The symbols "+ / ≈ / −" indicate the PR obtained by NCD-DE is "significantly superior/ equal/significantly inferior" to that obtained by the other algorithm, respectively. The analysis of the comparison results is given as follows.

1) For the first ten low-dimensional benchmark functions F_1–F_{10}, the experimental result of NCD-DE is significantly better than those obtained by most algorithms, while NMMSO dominates NCD-DE on F_7 and MOMMOP dominates NCD-DE on F_7, F_8, and F_9. However, as the accuracy level increases, the results of MOMMOP rapidly decrease, while the results obtained by NCD-DE are more stable.
2) For the next five functions F_{11}–F_{15}, which are low-dimensional functions with many optima, NCD-DE also achieves promising performance. Although NCD-DE does not achieve the best PR on every function, the experimental result of NCD-DE is significantly better than those of most algorithms. Compared with the results on different accuracy levels, many algorithms' performance deteriorates as the accuracy level increases, such as CDE, SDE, NCDE, Self-CCDE, Self-CSDE, LoICDE, and MOMMOP. However, the performance of NCD-DE only slightly decreases with increasing accuracy levels.
3) On the last five high-dimensional functions F_{16}–F_{20}, NCD-DE also outperforms most of the compared algorithms. Although the experimental result of NCD-DE is dominated by that of LMSEDA on F_{17}, NCD-DE achieves significantly superior performance than most algorithms. On these five high-dimensional functions, the results of most algorithms rapidly decrease as the accuracy level increases. For example, compared with the results on accuracy level $\varepsilon = 1.0\text{E} - 03$ and $\varepsilon = 1.0\text{E} - 05$, it is evident that the performance of many algorithms (such as Self-CCDE, Self-CSDE, and LMCEDA) deteriorates rapidly when the accuracy level increases. In contrast, NCD-DE's performance is stable.

According to the results and analyses above, the effectiveness and efficiency of NCD-DE are demonstrated. NCD-DE outperforms most of the compared algorithms not only on the low-dimensional functions but also on the high-dimensional functions. Furthermore, the stable results indicate that the results obtained by NCD-DE do not deteriorate rapidly as the increasing of accuracy level ε, whereas the experimental results of other algorithms decrease quickly as ε increases. Therefore, it is concluded that the performance of NCD-DE is stable. Besides, the NCD-DE algorithm can find almost global optima and ensure the accuracy of solutions, which indicates that NCD-DE can achieve both great exploration and great exploitation abilities.

C. Effects of Components in NCD-DE

There are three independent components in NCD-DE: 1) the NCD optimization; 2) the NGCM strategy; and 3) the archive technique. Their properties and effects are analyzed in this section.

1) Effect of NCD Optimization: The effect of NCD optimization is mainly influenced by the FEM objective function. Although the motivation of the proposed FEM objective function in maintaining the diversity of niche centers has been discussed in Fig. 2 and its effectiveness in helping NCD-DE solve MMOPs has been demonstrated in Section IV-B, the effect of the FEM in the NCD optimization is further studied here. To demonstrate the FEM is superior to the fitness-distance measurement (FDM), a comparison is conducted between the NCD-DE and a variant of NCD-DE with FEM objective function. The variant of NCD-DE is called NCD-DE-FDM (NCD-DE with FDM objective function). In NCD-DE-FDM, the average Euclidean distance between the niche centers is applied to replace the entropy in the FEM. The FDMs objective function FDM(.) can be calculated as follows:

$$\text{FDM}(Ch_i) = \frac{1}{NN^2} \sum_{j=0;\, Ch_{i,j}==1}^{NP} \text{fitness}(j) \times \text{distance}(j) \quad (21)$$

where the formula of distance(j) is shown in (9).

Another NCD-DE variant called NCD-DE-bq (NCD-DE with the best quarter of the population as niche centers) is designed to demonstrate the importance of the diversity of the selected niche centers. In NCD-DE-bq, the diversity of the niche centers is ignored. Instead, a quarter (i.e., 25%) of the total individuals with optimal fitness value are selected as niche centers. The comparison between NCD-DE and NCD-DE-bq reveals the necessity of the diversity of niche centers.

TABLE II
EXPERIMENTAL RESULTS OF NCD-DE AND 13 OTHER MMOP ALGORITHMS ON CEC2013 BENCHMARK FUNCTIONS AT $\varepsilon = 1.0$E-04

Func	NCD-DE		CDE		SDE		NCDE		NSDE		Self-CCDE		Self-CSDE	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{1}	**1.000**	1.000	**1.000(≈)**	1.000	0.657(+)	0.373	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{2}	**1.000**	1.000	**1.000(≈)**	1.000	0.737(+)	0.529	**1.000(≈)**	1.000	0.776(+)	0.667	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{3}	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{4}	**1.000**	1.000	**1.000(≈)**	1.000	0.284(+)	0.000	**1.000(≈)**	1.000	0.240(+)	0.000	**1.000(≈)**	1.000	0.686(+)	0.294
F_{5}	**1.000**	1.000	**1.000(≈)**	1.000	0.922(+)	0.843	**1.000(≈)**	1.000	0.745(+)	0.490	**1.000(≈)**	1.000	0.961(+)	0.922
F_{6}	**1.000**	1.000	**1.000(≈)**	1.000	0.056(+)	0.000	0.305(+)	0.000	0.056(+)	0.000	0.942(+)	0.490	0.699(+)	0.020
F_{7}	0.905	0.078	0.861(+)	0.000	0.054(+)	0.000	0.873(+)	0.000	0.053(+)	0.000	0.884(+)	0.020	0.695(+)	0.000
F_{8}	0.961	0.098	0.000(+)	0.000	0.015(+)	0.000	0.001(+)	0.000	0.013(+)	0.000	0.994(−)	0.882	0.695(+)	0.000
F_{9}	0.553	0.000	0.474(+)	0.000	0.011(+)	0.000	0.461(+)	0.000	0.006(+)	0.000	0.459(+)	0.000	0.265(+)	0.000
F_{10}	**1.000**	1.000	**1.000(≈)**	1.000	0.147(+)	0.000	0.989(+)	0.863	0.098(+)	0.000	**1.000(≈)**	1.000	0.992(≈)	0.922
F_{11}	**1.000**	1.000	0.330(+)	0.000	0.314(+)	0.000	0.729(+)	0.059	0.248(+)	0.000	0.778(+)	0.137	0.399(+)	0.000
F_{12}	0.993	0.941	0.002(+)	0.000	0.208(+)	0.000	0.252(+)	0.000	0.135(+)	0.000	0.422(+)	0.000	0.321(+)	0.000
F_{13}	0.892	0.412	0.141(+)	0.000	0.297(+)	0.000	0.667(+)	0.000	0.225(+)	0.000	0.660(+)	0.000	0.317(+)	0.000
F_{14}	0.683	0.000	0.026(+)	0.000	0.216(+)	0.000	0.667(+)	0.000	0.190(+)	0.000	0.657(+)	0.000	0.304(+)	0.000
F_{15}	0.640	0.000	0.005(+)	0.000	0.108(+)	0.000	0.319(+)	0.000	0.125(+)	0.000	0.343(+)	0.000	0.186(+)	0.000
F_{16}	**0.667**	0.000	0.000(+)	0.000	0.108(+)	0.000	**0.667(≈)**	0.000	0.170(+)	0.000	0.657(+)	0.000	0.072(+)	0.000
F_{17}	0.522	0.000	0.000(+)	0.000	0.076(+)	0.000	0.250(+)	0.000	0.108(+)	0.000	0.248(+)	0.000	0.056(+)	0.000
F_{18}	**0.667**	0.000	0.167(+)	0.000	0.026(+)	0.000	0.500(+)	0.000	0.163(+)	0.000	0.337(+)	0.000	0.003(+)	0.000
F_{19}	**0.505**	0.000	0.000(+)	0.000	0.105(+)	0.000	0.348(+)	0.000	0.098(+)	0.000	0.113(+)	0.000	0.000(+)	0.000
F_{20}	**0.252**	0.000	0.000(+)	0.000	0.000(+)	0.000	0.250(≈)	0.000	0.123(+)	0.000	0.027(+)	0.000	0.000(+)	0.000
+ / ≈ / −			13 / 7 / 0		19 / 1 / 0		13 / 7 / 0		18 / 2 / 0		13 / 6 / 1		16 / 4 / 0	
Func	LoICDE		LoISDE		NMMSO		LMCEDA		LMSEDA		MOMMOP		ANDE	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{1}	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{2}	**1.000(≈)**	1.000	0.235(+)	0.039	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{3}	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{4}	0.975(+)	0.902	0.250(+)	0.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{5}	**1.000(≈)**	1.000	0.667(+)	0.333	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{6}	**1.000(≈)**	1.000	0.056(+)	0.000	0.992(+)	0.880	0.990(≈)	0.843	0.972(+)	0.588	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{7}	0.705(+)	0.020	0.029(+)	0.000	**1.000(−)**	1.000	0.734(+)	0.000	0.673(+)	0.000	**1.000(−)**	1.000	0.933(−)	0.176
F_{8}	0.000(+)	0.000	0.012(+)	0.000	0.899(+)	0.020	0.367(+)	0.000	0.613(+)	0.000	**1.000(−)**	1.000	0.944(+)	0.078
F_{9}	0.187(+)	0.000	0.005(+)	0.000	0.978(−)	0.120	0.284(+)	0.000	0.248(+)	0.000	**1.000(−)**	1.000	0.512(+)	0.000
F_{10}	**1.000(≈)**	1.000	0.083(+)	0.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000	0.998(≈)	0.980	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{11}	0.660(+)	0.000	0.167(+)	0.000	0.990(+)	0.940	0.667(+)	0.000	0.892(+)	0.392	0.716(+)	0.020	**1.000(≈)**	1.000
F_{12}	0.495(+)	0.000	0.125(+)	0.000	0.993(≈)	0.940	0.750(+)	0.000	0.990(+)	0.922	0.939(+)	0.549	**1.000(−)**	1.000
F_{13}	0.510(+)	0.000	0.167(+)	0.000	**0.983(−)**	0.900	0.667(+)	0.000	0.667(+)	0.000	0.667(+)	0.000	0.686(+)	0.000
F_{14}	0.657(+)	0.000	0.167(+)	0.000	**0.720(−)**	0.000	0.667(+)	0.000	0.667(+)	0.000	0.667(+)	0.000	0.667(+)	0.000
F_{15}	0.299(+)	0.000	0.125(+)	0.000	0.632(+)	0.000	0.696(−)	0.000	**0.738(−)**	0.000	0.618(+)	0.000	0.632(+)	0.000
F_{16}	0.559(+)	0.000	0.167(+)	0.000	0.660(+)	0.000	**0.667(≈)**	0.000	**0.667(≈)**	0.000	0.650(+)	0.000	**0.667(≈)**	0.000
F_{17}	0.223(+)	0.000	0.076(+)	0.000	0.468(+)	0.000	0.456(+)	0.000	**0.620(−)**	0.000	0.505(+)	0.000	0.397(+)	0.000
F_{18}	0.219(+)	0.000	0.157(+)	0.000	0.650(+)	0.000	0.657(+)	0.000	0.660(+)	0.000	0.497(+)	0.000	0.654(+)	0.000
F_{19}	0.037(+)	0.000	0.027(+)	0.000	0.450(+)	0.000	0.451(+)	0.000	0.458(+)	0.000	0.223(+)	0.000	0.363(+)	0.000
F_{20}	0.123(+)	0.000	0.088(+)	0.000	0.172(+)	0.000	0.059(+)	0.000	0.248(≈)	0.000	0.125(+)	0.000	0.248(≈)	0.000
	14 / 6 / 0		18 / 2 / 0		9 / 7 / 4		11 / 8 / 1		10 / 8 / 2		10 / 7 / 3		8 / 10 / 2	

Table III provides the experimental results for the NCD-DE-FEM (the same as the NCD-DE), the NCD-DE-FDM, and the NCD-DE-bq. The results of NCD-DE-FEM are superior or equal to those of NCD-DE-bq and NCD-DE-FDM on all benchmark functions F_1–F_{20}. First, the experimental results of NCD-DE-FEM dominate those of NCD-DE-bq on 13 functions, which indicates that both the diversity and fitness of niche centers are important; second, the performance of NCD-DE is significantly superior to that of NCD-DE-FDM on 8 functions, while NCD-DE-FDM cannot dominate NCD-DE on any functions, which demonstrates the superiority of the FEM objective function in internal GA.

2) Effect of NGCM Strategy: The NGCM strategy can be decomposed into two different parts: 1) the niching mutation strategy with dual-scale local search and 2) the global mutation strategy. To verify each part's effects, two NCD-DE variants are designed: 1) NCD-DE-w/o-dls (NCD-DE without dual-scale local search) that eliminates the dual-scale local search and 2) NCD-DE-w/o-gms (NCD-DE without global mutation strategy) that eliminates the global mutation strategy. Note that since the niching mutation strategy is fundamental in NCD-DE, both the NCD-DE-w/o-dls and NCD-DE-w/o-gms contain the niching mutation strategy. The experimental results of NCD-DE and its variants are given in Table S.VI in the supplementary material.

First, the dual-scale local search strategy's effect is demonstrated by comparing the NCD-DE and the NCD-DE-w/o-dls. The experimental results of NCD-DE dominate or are at least equal to those of NCD-DE-w/o-dls on most functions. The performance of NCD-DE is significantly superior to that of NCD-DE-w/o-dls on ten functions, which are F_6, F_8, F_{11}–F_{15}, and F_{17}-F_{19}. The performance of NCD-DE is inferior to NCD-DE-w/o-dls on F_7 and F_9. Therefore, we can conclude that dual-scale local search can help to increase the accuracy of individuals. Hence, the dual-scale local search is important in NCD-DE.

In addition, the effect of the global mutation strategy is presented through the comparison between NCD-DE and NCD-DE-w/o-gms. On F_6–F_9, F_{12}, F_{14}, F_{15}, and F_{17}, the experimental results of NCD-DE are significantly superior to those of NCD-DE-w/o-gms, while NCD-DE-w/o-gms outperforms NCD-DE only on F_{20}. It is observed that the performance of NCD-DE is generally better than that of NCD-DE-w/o-gms. The comparison result indicates that the global mutation strategy enables the individuals to jump from one peak to another peak,

TABLE III
EXPERIMENTAL RESULTS ON CEC2013 BENCHMARK FUNCTIONS AMONG NCD-DEs WITH DIFFERENT OBJECTIVE FUNCTIONS AT $\varepsilon = 1.0\text{E-}04$

Func	NCD-DE-FEM		NCD-DE-FDM		NCD-DE-bq	
	PR	SR	PR	SR	PR	SR
F_1	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_2	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_3	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_4	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_5	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_6	**1.000**	1.000	**1.000(≈)**	1.000	0.609(+)	0.059
F_7	**0.905**	0.078	0.802(+)	0.000	0.864(+)	0.000
F_8	**0.961**	0.098	0.948(≈)	0.059	0.184(+)	0.000
F_9	**0.553**	0.000	0.478(+)	0.000	0.448(+)	0.000
F_{10}	**1.000**	1.000	**1.000(≈)**	1.000	**1.000(≈)**	1.000
F_{11}	**1.000**	1.000	**1.000(≈)**	1.000	0.676(+)	0.000
F_{12}	**0.993**	0.941	0.963(+)	0.706	0.669(+)	0.000
F_{13}	**0.892**	0.412	0.823(+)	0.137	0.578(+)	0.000
F_{14}	**0.683**	0.000	0.667(+)	0.000	0.588(+)	0.000
F_{15}	**0.640**	0.000	0.625(+)	0.000	0.235(+)	0.000
F_{16}	**0.667**	0.000	**0.667(≈)**	0.000	**0.667(≈)**	0.000
F_{17}	**0.522**	0.000	0.456(+)	0.000	0.250(+)	0.000
F_{18}	**0.667**	0.000	**0.667(≈)**	0.000	0.510(+)	0.000
F_{19}	**0.505**	0.000	0.473(+)	0.000	0.265(+)	0.000
F_{20}	**0.252**	0.000	0.250(≈)	0.000	0.066(+)	0.000
+ / ≈ / −			8 / 12 / 0		13 / 7 / 0	

which enables NCD-DE to search for more potential global optima.

3) Effect of Archive Technique: The archive technique plays a vital role in maintaining the identified solutions. In the archive technique, the best individuals of each niche are copied and stored to avoid extinction. The archive technique's effect is demonstrated by comparing the NCD-DE and the variant of NCD-DE without an archive technique, namely, NCD-DE-w/o-a (NCD-DE without archive). The experimental results of NCD-DE and NCD-DE-w/o-a are shown in Table S.VII in the supplementary material. It is easy to be observed from the experimental results that the NCD-DE achieves more promising performance than NCD-DE-w/o-a. Specifically, the experimental results of NCD-DE are significantly better than those of NCD-DE-w/o-a in functions F_7 and F_9, which contain many global optima (F_7 contains 36 global optima, and F_9 contains 216 global optima). In these functions, it is easy for individuals to jump from one peak to another, which leads to the extinction of several individuals. Therefore, the comparison result indicates that the archive technique helps to preserve individuals from extinction.

D. Sensitivity Analysis

1) Sensitivity Analysis of T_{GA}: The control parameter T_{GA} is an integer that determines the internal GA's iteration times in the NCD optimization. The setting of T_{GA} should be carefully considered. On the one hand, if the value of T_{GA} is relatively small, the internal GA cannot obtain promising niche centers with optimal FEM values in only a few iterations. On the other hand, if T_{GA} is set as a relatively large value, the optimization for optimal niche centers can cost a relatively large computation complexity.

To investigate the influence of the parameter T_{GA}, four NCD-DEs with different values of T_{GA} are designed, which are NCD-DEs with $T_{GA} = 5$ (original), 10, 15, and 20. The experimental results at $\varepsilon = 1.0\text{E} - 04$ of all the NCD-DE variants with different values of T_{GA} are provided in Table S.VIII in the supplementary material.

From Table S.VIII in the supplementary material, the experimental results of the NCD-DE variants are slightly improved as T_{GA} increases in F_7, F_9, F_{12}, F_{15}, and F_{20}. However, in F_8, F_{13}, and F_{17}, the PRs perform undulate results as T_{GA} increases. The performance of NCD-DE-T20 (i.e., NCD-DE with $T_{GA} = 20$) for PR is generally better than that of NCD-DE, which can be concluded from the result of Wilcoxon's rank-sum test. However, NCD-DE outperforms NCD-DE-T10 and NCD-DE-T15.

In summary, the results of NCD-DE slightly increase as the T_{GA} parameter increases. This finding is attributed to the fact that the internal GA chromosomes are refined during the evolutionary process iterations. If the iteration time increases, the final generation of chromosomes may obtain better FEM values. However, the performance increase is slight. Considering the balance between the time complexity and the algorithm's performance, the setting of the parameter T_{GA} is recommended as 5.

2) Sensitivity Analysis of λ: In the NGCM strategy, the control parameter λ is an integer that determines the gap for the execution of the global mutation strategy. The value of parameter λ should be carefully selected. On the one hand, the global mutation strategy is executed frequently if λ is too small, so as to cost a large amount of FEs. On the other hand, if λ is relatively large, the global mutation strategy is seldomly executed, which can deteriorate the exploration ability of NCD-DE.

Therefore, the influence of the parameter λ should be investigated. Four NCD-DEs with different λ are designed, which are NCD-DEs with $\lambda = 1, 5$ (original), 10, and 20. These NCD-DEs are called NCD-DE (where $\lambda = 5$), NCD-DE-λ1, NCD-DE-λ10, and NCD-DE-λ20, respectively. The experiment results at $\varepsilon = 1.0\text{E} - 04$ of all the NCD-DEs with different values of λ are given in Table S.IX in the supplementary material.

On functions F_1–F_6, all these NCD-DEs with different values of λ can find all the global optima. However, on F_7–F_9, the performance of NCD-DE-λ1 dominates that of other NCD-DEs. On F_{12}–F_{19}, the performance of NCD-DE dominates that of other NCD-DE variants. On F_{20}, the best result is obtained by NCD-DE-λ20. From the experimental results, generally, the NCD-DE achieves the best performance. For example, the experimental results of NCD-DE with $\lambda = 5$ are significantly better than those of NCD-DE-λ1, NCD-DE-λ10, and NCD-DE-λ20 on seven functions, seven functions, and eight functions, respectively. We can find that the NCD-DE achieves the best performance when $\lambda = 5$. Therefore, the value of λ is recommended to be set as 5.

E. Real-World Application

To verify the effectiveness and efficiency of NCD-DE, the experiments on a real-world MMOP, that is, multiple competitive facilities location and design (MCFLD) problem [54], are conducted in this section. In the MCFLD problem, a chain wants to construct several new facilities in a given area. However, with the condition that many facilities offering the same service already exist or might be soon constructed in the given area, the new facilities of the chain have to compete with the existing facilities in the market. The goal of the MCFLD is to determine the optimal location and quality of the new facilities to maximize the total profit. Since the MCFLD problem is a nonlinear problem with multiple optima, the chain may want to find out several available solutions in the real-world application.

In specific, according to the model of MCFLD problem given in [54], the total profit that constructs p new facilities is formulated as

$$F\big(nf_1, nf_2, \ldots, nf_p\big) = c \cdot M\big(nf_1, nf_2, \ldots, nf_p\big) - \sum_{j=1}^{p} G\big(nf_j\big)$$

$$\begin{aligned} \text{s.t.} \quad & 0.5 \le \alpha_j \le 5 \\ & 0 \le x_j \le 10 \\ & 0 \le y_j \le 10,\ j = 1, 2, \ldots, p \end{aligned} \tag{22}$$

$$M\big(nf_1, nf_2, \ldots, nf_p\big) = \sum_{i=1}^{n} w_i \frac{\sum_{j=1}^{p} \gamma_i \alpha_j + \sum_{l=1}^{k} \alpha_{il}}{\sum_{j=1}^{p} \gamma_i \alpha_j + \sum_{l=1}^{m} \alpha_{il}} \tag{23}$$

$$G\big(nf_j\big) = \sum_{i=1}^{n} \frac{w_i}{2 + d_{iz_j}^2} + e^{\frac{\alpha_j}{\xi_0} + \xi_1} - e^{\xi_1} \tag{24}$$

where n, m, and k are the number of demand points, the number of existing facilities, and the number of facilities that belong to the chain, respectively. nf_j stands for the properties of the jth new facility, which includes three variables: 1) quality α_j; 2) x-coordinate x_j; and 3) y-coordinate y_j. Thus, the MCFLD problem includes $3p$ decision variables. The detailed description and settings of the parameters can be found in [54].

Since the location and the fitness of the real optima of the MCFLD problem are unknown, we first estimate the optimal fitness and then determine which solution is located at the peak. To estimate the real optimal fitness, the entire search space is divided into 10^6 grids, and a position is sampled in each grid. The fitness value of the best position in the 10^6 sampled positions is the estimated optimal fitness. A solution is determined to be peak if its fitness is larger than the estimated optimal fitness and there is no solution determined as a peak within 0.1 distance.

The results of mean peak number (PN) and the corresponding rank obtained by NCD-DE are compared to those obtained by CDE, NCDE, and ANDE. Each algorithm is independently executed 20 times to obtain the mean results. The experimental results on six MCFLD problems with settings $p = 2$ and $(n, m, k) = (21, 5, 2)$, (50, 2, 1), (50, 10, 4), (71, 5, 2), (100, 2, 1), and (100, 10, 4) are given in Table IV. From the results, it is obvious that the NCD-DE can find more global optima than the compared algorithms. In each problem, the NCD-DE can find more than 60 optima, while the number of optima found by CDE and NCDE is less than 5. Although the ANDE can find out nearly 40 optima when $(n, m, k) = (21, 5, 2)$, its performance in the other cases is not promising enough. Therefore, the effectiveness and efficiency of NCD-DE on real-world MMOP are verified.

TABLE IV
EXPERIMENTAL RESULTS ON SIX MCFLD PROBLEMS OF NCD-DE, CDE, NCDE, AND ANDE

Problem (n, m, k)	NCD-DE	CDE	NCDE	ANDE
	PN (Rank)	PN (Rank)	PN (Rank)	PN (Rank)
(21, 5, 2)	**80.05 (1)**	2.10 (4)	2.25 (3)	39.05 (2)
(50, 2, 1)	**102.60 (1)**	3.05 (3)	2.50 (4)	11.70 (2)
(50, 10, 4)	**60.50 (1)**	2.00 (4)	2.10 (3)	19.85 (2)
(71, 5, 2)	**86.50 (1)**	2.45 (4)	3.05 (3)	16.75 (2)
(100, 2, 1)	**93.85 (1)**	2.80 (4)	3.95 (3)	21.80 (2)
(100, 10, 4)	**111.75 (1)**	2.45 (4)	4.20 (3)	23.50 (2)
mean rank	**1.00**	3.83	3.17	2.00

V. CONCLUSION

In this article, we proposed an inspiring and practical mechanism that transforms the NCD problem into an optimization problem, and applies an extra EC algorithm to solve it. Through the transformation, we do not need to design ad hoc niching approaches for different MMOPs but only design the objective function of the optimization problem to fulfill our demands of the niche centers.

Based on the above mechanism, a novel NCD-DE algorithm composed of the NCD optimization, NGCM strategy, and archive technique has been proposed for solving MMOPs. In the NCD optimization, the NCD problem is transformed into an optimization problem and solved by the internal GA. The NCD optimization calculates the FEM of the selected niche centers to ensure that the niche centers with the optimal fitness value and great diversity are selected. The niche centers optimally selected by the NCD optimization are distributed widely and are with good fitness values on the MMOPs, which helps to locate more peaks and to increase the solution accuracy. To further increase the capability of exploration and exploitation, the NGCM strategy was proposed. On the one hand, the niching mutation strategy with dual-scale local search helps to search the circumambient region of each niche. On the other hand, the global mutation strategy enables each individual to jump from one niche to another niche or some undiscovered region. In addition, the archive technique was proposed to copy each niche's best individual and store it into an archive, which preserves the optimal individuals from extinction.

Based on the three components described, the proposed NCD-DE achieved better performance than multiple state-of-the-art and recent well-performing algorithms in dealing with MMOPs, as verified on the CEC2013 benchmark functions.

In future work, we will comprise the NCD optimization with other EC algorithms, such as PSO and EDA, to propose more alternative algorithms to deal with MMOPs. Moreover, we will make efforts to extend the NCD-DE to solve real-world MMOP applications. Besides, extensive research into updating the proposed NCD-DE as a distributed [13], [55] or parallel [56] algorithm, or using matrix-based framework [57], scale-adaptive fitness evaluation [58], or region-based encoding scheme [59] to reduce the running time are also worthy studied.

References

[1] W. Sheng, S. Swift, L. Zhang, and X. Liu, "A weighted sum validity function for clustering with a hybrid niching genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1156–1167, Dec. 2005.

[2] L. N. De Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2002, pp. 699–704.

[3] J. Yao, N. Kharma, and P. Grogono, "A multi-population genetic algorithm for robust and fast ellipse detection," *Pattern Anal. Appl.*, vol. 8, no. 1, pp. 149–162, 2005.

[4] E. Cuevas, M. González, and D. Zaldivar, "Multi-ellipses detection on images inspired by collective animal behavior," *Neural Comput. Appl.*, vol. 24, no. 5, pp. 1019–1033, 2014.

[5] D.-Z. Tan, W. N. Chen, J. Zhang, and W.-J. Yu, "Fast pedestrian detection using multimodal estimation of distribution algorithms," in *Proc. Genet. Evol. Comput. Conf.*, 2017, pp. 1248–1255.

[6] K. C. Wong, K. S. Leung, and M. H. Wong, "Protein structure prediction on a lattice model via multimodal optimization techniques," in *Proc. Genet. Evol. Comput. Conf.*, 2010, pp. 155–162.

[7] D.-K. Woo, J.-H. Choi, M. Ali, and H.-K. Jung, "A novel multimodal optimization algorithm applied to electromagnetic optimization," *IEEE Trans. Magn.*, vol. 47, no. 6, pp. 1667–1673, Jun. 2011.

[8] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.

[9] J.-Y. Li, Z.-H. Zhan, C. Wang, H. Jin, and J. Zhang, "Boosting data-driven evolutionary algorithm with localized data generation," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 923–937, Oct. 2020.

[10] J. Y. Li, Z. H. Zhan, H. Wang, and J. Zhang, "Data-driven evolutionary algorithm with perturbation-based ensemble surrogates," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3925–3937, Aug. 2021.

[11] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[12] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[13] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4633–4647, Nov. 2020.

[14] W.-J. Yu *et al.*, "Differential evolution with two-level parameter adaptation," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1080–1099, Jul. 2014.

[15] X. Xia *et al.*, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[17] X. Zhang, K. J. Du, Z. H. Zhan, S. Kwong, T. L. Gu, and J. Zhang, "Cooperative coevolutionary bare-bones particle swarm optimization with function independent decomposition for large-scale supply chain network design with uncertainties," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4454–4468, Oct. 2020.

[18] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," *Artif. Intell. Rev.*, to be published, doi: 10.1007/s10462-021-10042-y.

[19] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proc. IEEE Congr. Evol. Comput.*, 1996, pp. 798–803.

[20] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proc. Int. Conf. Genet. Algorithms*, 1995, pp. 24–31.

[21] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.

[22] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, 2004, pp. 1382–1389.

[23] H. Zhao *et al.*, "Local binary pattern-based adaptive differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3343–3357, Jul. 2020.

[24] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1726–1737, Oct. 2014.

[25] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. Int. Conf. Genet. Algorithms*, 1987, pp. 41–49.

[26] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proc. Conf. Genet. Evol. Comput.*, 2005, pp. 873–880.

[27] J. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 207–234, 2002.

[28] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu, "Multimodal optimization by means of a topological species conservation algorithm," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 842–864, Dec. 2010.

[29] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[30] W. Gao, G. G. Yen, and S. Liu, "A cluster-based differential evolution with self-adaptive strategy for multimodal optimization," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1314–1327, Aug. 2014.

[31] R. K. Ursem, "Multinational evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, 1999, pp. 1633–1640.

[32] J. Yao, N. Kharma, and Y. Q. Zhu, "On clustering in evolutionary computation," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 1752–1759.

[33] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, "Multimodal estimation of distribution algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 636–650, Mar. 2017.

[34] Z.-J. Wang *et al.*, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, Feb. 2020.

[35] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.

[36] X. Lin, W. Luo, and P. Xu, "Differential evolution for multimodal optimization with species by nearest-better clustering," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 970–983, Feb. 2021.

[37] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 708–719, Aug. 2020.

[38] W. Luo, Y. Qiao, X. Lin, P. Xu, and M. Preuss, "Hybridizing niching, particle swarm optimization, and evolution strategy for multimodal optimization," *IEEE Trans. Cybern.*, early access, Dec. 15, 2020, doi: 10.1109/TCYB.2020.3032995.

[39] T. Huang, Y. J. Gong, W.-N. Chen, H. Wang, and J. Zhang, "A probabilistic niching evolutionary computation framework based on binary space partitioning," *IEEE Trans. Cybern.*, early access, Mar. 11, 2020, doi: 10.1109/TCYB.2020.2972907.

[40] J. Yao, N. Kharma, and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 80–102, Feb. 2010.

[41] K. Deb and A. Saha, "Multimodal optimization using a bi-objective evolutionary algorithm," *Evol. Comput.*, vol. 20, no. 1, pp. 27–62, Mar. 2012.

[42] Y. Wang, H. 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.

[43] R. Cheng, M. Li, K. Li, and X. Yao, "Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 692–706, Oct. 2018.

[44] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[45] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.

[46] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.

[47] J. Cheng, G. G. Yen, and G. Zhang, "A many-objective evolutionary algorithm with enhanced mating and environmental selections," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 592–605, Aug. 2015.

[48] L. Li and K. Tang, "History-based topological speciation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 136–150, Feb. 2015.

[49] Y.-H. Zhang, M.-T. Li, Y.-J. Gong, and J. Zhang, "Differential evolution with random walk mutation and an external archive for multimodal optimization," in *Proc. IEEE Symp. Series Comput. Intell.*, 2015, pp. 1868–1875.

[50] Z.-J. Wang *et.al.*, "Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 894–908, Dec. 2018.

[51] X. Li, A. Engelbrecht, and M. G. Epitropakis, "Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization," Evol. Comput. Mach. Learn. Group, RMIT Univ., Melbourne, VIC, Australia, Rep., 2013. [Online]. Available: http://titan.csit.rmit.edu.au/∼ e46507/cec13-niching/competition/cec2013-niching-benchmark-tech-report.pdf

[52] J. E. Fieldsend, "Running up those hills: Multi-modal search with the niching migratory multi-swarm optimizer," in *Proc. IEEE Congr. Evol. Comput.*, 2014, pp. 2593–2600.

[53] J. Carrasco, S. Garcia, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665.

[54] J. L. Redondo, J. Fernández, I. García, and P. M. Ortigosa, "Solving the multiple competitive facilities location and design problem on the plane," *Evol. Comput.*, vol. 17, no. 1, pp. 21–53, Mar. 2009.

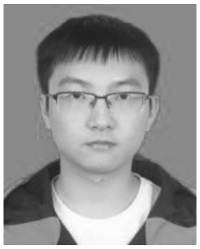
[55] Z.-H. Zhan *et al.*, "Cloudde: A heterogeneous differential evolution algorithm and its distributed cloud version," *IEEE Trans. Parallel. Distrib. Syst.*, vol. 28, no. 3, pp. 704–716, Mar. 2017.

[56] J.-Y. Li, Z.-H. Zhan, R.-D. Liu, C. Wang, S. Kwong, and J. Zhang, "Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4848–4859, Oct. 2021.

[57] Z.-H. Zhan *et al.*, "Matrix-based evolutionary computation," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, Jan. 21, 2021, doi: 10.1109/TETCI.2020.3047410.

[58] S. H. Wu, Z. H. Zhan, and J. Zhang, "SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 3, pp. 478–491, Jun. 2021.

[59] J.-R. Jian, Z.-G. Chen, Z.-H. Zhan, and J. Zhang, "Region encoding helps evolutionary computation evolve faster: A new solution encoding scheme in particle swarm for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 779–793, Aug. 2021.



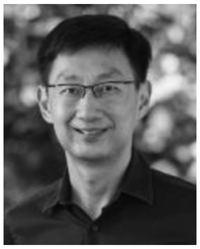
Yi Jiang (Student Member, IEEE) received the B.S. degree in computer science and technology from the South China University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests mainly include computational intelligence, machine learning, and their applications in real-world problems.

Zhi-Hui Zhan (Senior Member, IEEE) received the bachelor's degree and the Ph.D. degree in computer science from Sun Yat-Sen University, Guangzhou, China, in 2007 and 2013, respectively.

He is currently the Changjiang Scholar Young Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include evolutionary computation, swarm intelligence, and their applications in real-world problems and in environments of cloud computing and big data.

Dr. Zhan was a recipient of the IEEE Computational Intelligence Society Outstanding Early Career Award in 2021, the Outstanding Youth Science Foundation from National Natural Science Foundations of China in 2018, and the Wu Wen-Jun Artificial Intelligence Excellent Youth from the Chinese Association for Artificial Intelligence in 2017. His doctoral dissertation was awarded the IEEE CIS Outstanding Ph.D. Dissertation and the China Computer Federation Outstanding Ph.D. Dissertation. He is listed as one of the Highly Cited Chinese Researchers in Computer Science. He is currently an Associate Editor of the IEEE Transactions on Evolutionary Computation, *Neurocomputing*, and *Memetic Computing*.



Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) and the Ph.D. degree from the University of Glasgow, U.K., in 1994 and 1997, respectively.

He is currently a Chair Professor (Computational Intelligence) with the Department of Computing, the Hong Kong Polytechnic University, Hong Kong. He has published over 300 refereed articles and seven books.

Prof. Tan is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, USA. He served as the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013 and the *IEEE Transactions on Evolutionary Computation* from 2015 to 2020. He currently serves as an Editorial Board Member for more than ten journals. He is an IEEE Distinguished Lecturer Program Speaker and the Chief Co-Editor of Springer Book Series on *Machine Learning: Foundations, Methodologies, and Applications*

Jun Zhang (Fellow, IEEE) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Korea Brain Pool Fellow Professor with Hanyang University, Ansan, South Korea, and a Visiting Professor with the Chaoyang University of Technology, Taichung, Taiwan. He has published over more than 150 IEEE Transactions papers in his research areas. His current research interests include computational intelligence, cloud computing, operations research, and power electronic circuits.

Dr. Zhang was a recipient of the Changjiang Chair Professor from the Ministry of Education, China, in 2013, The National Science Fund for Distinguished Young Scholars of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of theIEEE Transactions on Cybernetics and the IEEE Transactions on Evolutionary Computation.