Multimodal Multi-objective Evolutionary Optimization with Dual Clustering in Decision and Objective Spaces

Qiuzhen Lin, *Member, IEEE*, Wu Lin, Zexuan Zhu, Maoguo Gong, *Senior Member, IEEE*, Jianqiang Li, and Carlos A. Coello Coello, *Fellow, IEEE*

***Abstract*—This paper suggests a multimodal multi-objective evolutionary algorithm with dual clustering in decision and objective spaces. One clustering is run in decision space to gather nearby solutions, which will classify solutions into multiple local clusters. Non-dominated solutions within each local cluster are first selected to maintain local Pareto sets, and the remaining ones with good convergence in objective space are also selected, which will form a temporary population with more than *N* solutions (*N* is the population size). After that, a second clustering is run in objective space for this temporary population to get *N* final clusters with good diversity in objective space. Finally, a pruning process is repeatedly run on the above clusters until each cluster has only one solution, which removes the most crowded solution in decision space from the most crowded cluster in objective space each time. This way, the clustering in decision space can distinguish all Pareto sets and avoid the loss of local Pareto sets, while that in objective space can maintain diversity in objective space. When solving all the benchmark problems from the competition of multimodal multi-objective optimization in IEEE Congress on Evolutionary Computation 2019, the experiments validate our advantages to maintain diversity in both objective and decision spaces.**

***Index Terms*—Multi-objective optimization, Multimodal optimization, Evolutionary algorithm, Clustering.**

I. INTRODUCTION

IN recent decades, multi-objective evolutionary algorithms (MOEAs) have become a popular and effective approach for solving multi-objective optimization problems (MOPs). MOEAs are population-based approaches that aim to find a set of non-dominated solutions that approximate the Pareto front (PF) of an MOP with an even distribution in objective space [1]-[3]. Then, decision makers can select one final solution from this approximation set according to their preferences.

Generally, most MOEAs include a recombination operator to generate offspring and a selection operator to update the inferior parents. Simulated binary crossover (SBX) [4], differential evolution (DE) [5], local or gradient based search strategies [6]-[8] are often used as the recombination operator, while three main kinds of selection criteria are employed in the selection operator of most MOEAs, including Pareto-based strategy [9]-[11], decomposition-based strategy [12]-[20], and indicator-based strategy [21]-[23]. However, these selection criteria only assess performance of solutions in objective space, but rarely consider their diversity in decision space. This design principle in MOEAs will face some challenges for solving multimodal MOPs (MMOPs), as some Pareto optimal solution sets (PSs) for the same PF may be missed [24].

In fact, MMOPs arise in many real-world applications, e.g., multi-objective knapsack optimization [25], flow shop scheduling [26], rocket engine optimization [27], and space mission design [28], among others. These MMOPs have different PSs for the same PF or even possess some local and global PSs simultaneously [29]-[31]. As some uncertain factors are difficult to capture in the modeling process of real-world applications, the optimal solutions obtained by optimizing the target MMOPs may become infeasible in practice due to environmental changes or to the presence of constraints. Thus, it is very helpful if the optimizer can provide multiple different PSs with the same quality in objective space or even some local PSs with acceptable quality as an alternative choice, which can provide the decision maker more options to eliminate uncertain risks that arise in practice [32]-[34].

As stated in [24], a diverse set of solutions should be considered in both decision and objective spaces when designing MOEAs for solving MMOPs. Based on this principle, some multimodal MOEAs (MMOEAs) [35]-[38] have been proposed, which can effectively maintain a number of global PSs for the same PF. However, they still show some difficulties in saving good local PSs with acceptable quality, as global PSs with good convergence always dominate and replace all local PSs. However, as mentioned above, some good local PSs with acceptable quality should be also provided to eliminate the risk in case global PSs become infeasible in practice. In the CEC 2019 competition on MMOPs [31], it was suggested to approximate their global PSs as well as to maintain some good local PSs.

This work was supported by the National Nature Science Foundation of China under Grants 61876110, 61871272 and 61836005, the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, the Shenzhen Fundamental Research Program under Grants JCYJ20190808173617147 and JCYJ20190808164211203, and CONACyT grant no. 2016-01-1920. (*Corresponding author: Zexuan Zhu*)

Q.Z. Lin, W. Lin, and J.Q. Li are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

Z. Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China; Shenzhen Pengcheng Laboratory, Shenzhen 518055, China; and also with the SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen University, Shenzhen 518060, China (e-mail: zhuzx@szu.edu.cn).

M. Gong is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China.

C.A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), México, D.F. 07300, MÉXICO.

In recent years, niching methods have been the most popular tool to maintain global and local PSs, adopting techniques such as fitness sharing [39]-[40], crowding [41] and clustering [42]-[44]. For example, density-based clustering was proposed in [42] as a niching method to find equivalent PSs for the same PF, and an affinity propagation clustering was presented in [43] as a parameter-free automatic niching method to capture the peaks in multimodal optimization. In [44], a hill-valley clustering approach was introduced to adaptively cluster the search space into a number of niches, and then each of them is optimized by a core search algorithm. Inspired by the above studies of using clustering to distinguish multiple PSs, this paper suggests a novel MMOEA with dual clustering in decision and objective spaces, termed MMOEA/DC. Different from the existing MMOEAs [35-38] that focus on searching multiple global PSs, our algorithm can properly balance the maintenance of global PSs and some good local PSs with acceptable quality during the evolutionary search. In our algorithm, a neighborhood-based clustering method (NCM) modified from DBSCAN [45] is run in decision space to classify the union population into multiple clusters based on their neighborhood relationship, trying to maintain diversity in decision space. Then, the non-dominated solutions in each of these clusters are first selected to maintain good local PSs, while some remaining solutions with good convergence are further selected by applying non-dominated sorting [9] in objective space to approximate global PSs, which composes a temporary population with global and local PSs (its size is larger than the population size N). Then, a hierarchical clustering method (HCM) [46]-[47] is run in objective space to classify this temporary population into N final clusters. At last, the most crowded solution in decision space from the most crowded cluster in objective space will be removed each time according to the harmonic averaged distance (HAD) [48]. The above deletion process will be repeated until each cluster has only one solution. This way, the final solutions in clusters can properly balance the maintenance of global PSs and good local PSs with acceptable quality. After evaluating the performance of MMOEA/DC on all the CEC 2019 MMOPs [31], our experimental results validate the advantages of our proposed approach to maintain global and local PSs. Moreover, the search behavior of clustering on MMOPs and the impact of crowding-based mating selection are also discussed in order to verify their effectiveness.

The rest of this paper is organized as follows. Section II introduces some related background of MMOPs and clarifies the motivations to design MMOEA/DC. Section III introduces the dual clustering methods adopted in both decision and objective spaces, while Section IV provides the details of our algorithm. Section V presents our experimental results and their corresponding discussion. Finally, this paper is concluded in Section VI.

II. Related Background and Motivation

A. Multimodal Multi-objective Optimization Problems

In general, an MOP can be modeled as follows:

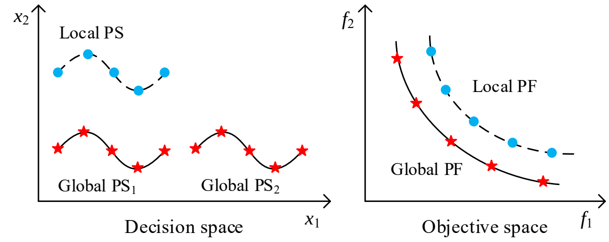


Fig. 1. Illustrations of local PS, global PS, local PF and global PF.

$$\min_{x\in\Omega} F(x) = (f_1(x), ..., f_m(x))^T, \quad (1)$$

where $x = (x_1, ..., x_n)$ is a variable vector in decision space Ω (n is the number of decision variables), and $F(x)$ defines m (often conflicting) objective functions $f_1(x), ..., f_m(x)$. Some MOPs from real-world scenarios may have different PSs for the same global PF or even possess some local PSs simultaneously [29]-[31]. Such problems are often called multimodal MOPs (MMOPs). In [31], local PS, local PF, global PS, and global PF of MMOPs are respectively defined as follows.

Local PS and Local PF: For any solution x in a set P_L, if no neighboring solution y satisfying $\|y-x\|_\infty \le \delta$ (δ is a small positive value) can dominate x, P_L is called local PS and its mapping in objective space is called local PF.

Global PS and Global PF: For any solution x in a set P_G, if other solution in feasible space cannot dominate x, P_G is called global PS and its mapping in objective space is called global PF.

For clarity, a simple example is given in Fig. 1, which shows an MMOP with one local PS and two global PSs (i.e., PS_1 and PS_2). Dashed lines with blue circle dots represent local PS or PF, while solid lines with red stars are global PS or PF. As suggested in [32]-[34], all global PSs and good local PSs with acceptable quality for MMOPs should be provided to support decision making in various application scenarios.

B. Related Work

In recent years, a number of MMOEAs have been proposed to solve MMOPs, which can be roughly classified into three main categories: Pareto-based MMOEAs, decomposition-based MMEAs, and others.

1) Pareto-based MMOEAs: There are two extensions from NSGA-II [9] for solving MMOPs, i.e., Omni-optimizer [35] and DN-NSGA-II [36]. The concept of crowding distance in decision space was presented in Omni-optimizer and a decision space based niching method was proposed in DN-NSGA-II. Both mechanisms are able to find multiple global PSs for the same PF. In [37], a niche sharing method was simultaneously employed for diversity maintenance in both objective and decision spaces, allowing to diversify solutions for solving MMOPs. In SPEA2+ [38], two archives were adopted to maintain the diversity of solutions for solving MMOPs, which are updated according to the density of solutions respectively in objective space and in decision space for the environmental selection.

2) Decomposition-based MMOEAs: In [49], environmental selection was conducted based on a fitness value combining the PBI function value [12] in objective space and two distance values in decision space. This way, multiple dissimilar solutions can be associated to one subproblem, which helps to

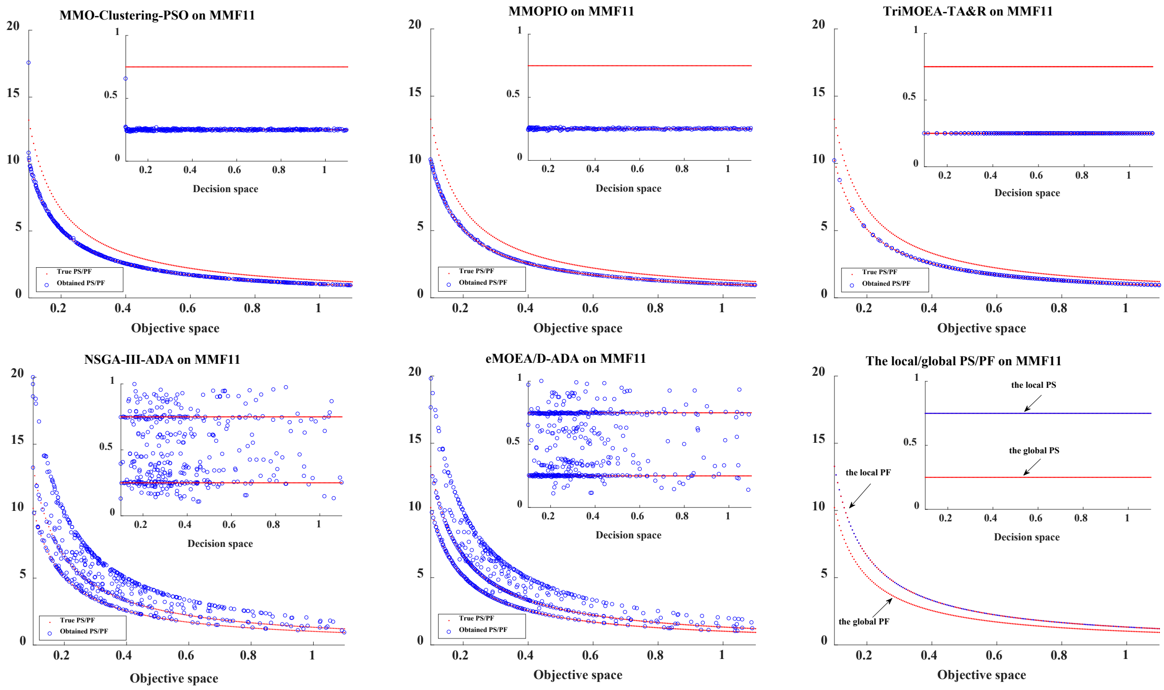


Fig. 2. The final solutions obtained by five different MMOEAs and the true local/global PS/PF on MMF11.

extend diversity for solving MMOPs. In MOEA/D-AD [50], each solution was associated to a subproblem with the closest weight vector based on the perpendicular distance and a relative neighborhood size *L* was set in its decision space. Then, each offspring only updates the original solutions associated to the same subproblem from its *L* nearest individuals in decision space. Thus, multiple PSs can be preserved for each subproblem. This idea was generalized as a framework in [34] based on three operations: assignment, deletion, and addition, which can enhance the performance of decomposition-based MOEAs when solving MMOPs.

3) Others: There are also some other MOEAs designed for MMOPs, which are difficult to classify into any of the two above categories. A multi-objective optimizer was designed in [51], where local PSs can assist to find global PSs by sliding down the multi-objective gradient hill and moving along local PSs. A general framework using two archives and recombination was designed in TriMOEA-TA&R [32]. The diversity archive uses a clustering strategy and a niche-based clearing strategy to maintain diversity respectively in objective space and decision space, while the convergence archive aims to maintain diverse converged solutions from independent decision subspaces. By recombining solutions in both archives, this algorithm is able to find multiple PSs. An MOEA using a convergence-penalized density method (CPDEA) was proposed in [33], which transforms the distances among solutions in decision space based on their local convergence quality. Using these transformed distances, the density values of solutions are estimated and used as the selection criterion to maintain multiple PSs.

Moreover, other heuristic algorithms have also been adopted for tackling MMOPs in recent years. For example, a new particle swarm optimization (PSO) algorithm was introduced in [52] with an index-based ring topology, which can produce some stable niches to find a number of PSs. A self-organizing map network was embedded into a PSO algorithm in [53] and into an improved pigeon-inspired optimization algorithm in [54], building a good neighborhood relationship in decision space, which helps to approximate multiple PSs for solving MMOPs. Please note that the non-dominated sorting method [9] based on the special crowding distance (SCD) is used in all the above PSO algorithms.

C. *Motivations and Contributions*

However, the above algorithms only focus on maintaining global PSs, but often neglect good local PSs with acceptable quality in objective space. In this subsection, the performance of some representative algorithms was investigated for solving MMOPs with coexistence of global and local PSs. Two algorithms based on Pareto dominance (MMO-Clustering-PSO[1] and MMOPIO [54]), two decomposition-based MOEAs (NSGA-III-ADA [34] and eMOEA/D-ADA [34]) and TriMOEA-TA&R [32] were adopted here to solve MMF11 from the CEC 2019 MMOPs [31]. The initial population size was set to 100 for NSGA-III-ADA and eMOEA/D-ADA and to 200 for the other algorithms. Please note that the numbers of solutions are changed at each generation of NSGA-III-ADA and eMOEA/D-ADA and the setting in this paper will finally produce more than 200 solutions. All the algorithms were independently run 31 times with a maximum number of function evaluations of 10000. Their final solutions based on the median performance obtained with respect to the IGDX indicator [36] in 31 runs are plotted in Fig. 2. As observed in Fig. 2, all the algorithms faced some difficulties in maintaining a good local PS with acceptable quality for MMF11. In MMO-Clustering-PSO and MMOPIO, the non-dominated sorting method [9] was the primary selection criterion in objective space, which allows the local PS to be removed as it

[1] http://www5.zzu.edu.cn/ecilab/info/1036/1211.htm

is dominated by the global PS during the evolutionary process. NSGA-III-ADA and eMOEA/D-ADA could maintain some local PSs if they were far away from the global PS in decision space, but they preserved some local PSs with poor quality, which cannot evenly cover the true global and local PSs of MMF11. For TriMOEA-TA&R, the local PS was totally missed for MMF11, as the non-dominated sorting method [9] was used to update its diversity archive in objective space.

Based on the above experiments, it is clear that these algorithms faced difficulties in maintaining a good local PS with acceptable quality for solving MMF11. In fact, most MMOEAs [32-33,35-38,52-54] often ignore local PSs, as the maintenance of local PSs may easily lead the algorithm to get trapped into the local optimal regions and local PSs are easily removed due to the Pareto dominance by global PSs. In order to fill this research gap for MMOPs, this paper suggests a new MMOEA with dual clustering in decision and objective spaces, called MMOEA/DC, which considers two key tasks on solving MMOPs: (i) locating multiple global PSs in decision space with the same quality in objective space; (ii) locating good local PSs with acceptable quality in objective space. To solve the tasks, one clustering is applied in decision space to gather nearby solutions, which will classify solutions into multiple local clusters, so as to maintain the local PSs. A second clustering is applied in objective space so that the temporary population selected from these local clusters can get N final clusters, which aims to maintain diversity in objective space. At last, a pruning process is performed for the above clusters until each cluster only has one solution, by iteratively removing the most crowded solution in decision space from the most crowded cluster in objective space. This way, MMOEA/DC can strike a good balance on solving two key tasks of MMOPs.

To summarize, our main contributions which distinguish us from the previous works [32-38, 49-54] are the following:

1) This paper suggests a novel MMOEA with dual clustering in decision and objective spaces. An NCM is applied in decision space to distinguish global or local PSs, while an HCM is applied in objective space to maintain their diversity. This way, our environmental selection using dual clustering can properly balance the maintenance of global PSs and good local PSs with acceptable quality during the evolutionary process.

2) A crowding-based mating selection is proposed in this paper, which uses a binary tournament selection strategy based on the HAD values in decision space. Thus, more uniformly distributed mating parents can be selected, which can produce offspring with strengthened diversity in decision space.

III. Dual Clustering in Decision and Objective Spaces

In this section, the details of the dual clustering methods (NCM and HCM) are introduced, which are respectively applied in decision and objective spaces. NCM aims to distinguish local or global PSs, by classifying the union of parents and offspring into multiple clusters according to their neighborhood relations in decision space. After collecting the non-dominated solutions in each of these clusters and other remaining solutions with good convergence, HCM is applied in objective space to separate these solutions into N final clusters

Algorithm 1 *Neighborhood-based Clustering*(U, r)

```
Input: U, r
Output: C_1,…, C_K
1:  for each solution x ∈ U
2:     identify its r-neighborhood B(x) by (2)
3:  end for
4:  K = 1
5:  while U is not empty
6:     randomly select one solution x from U
7:     initialize Q = C_K = { x }
8:     while Q is not empty
9:        randomly select one solution x from Q
10:       T = B(x) \ C_K
11:       Q = Q ∪ T and C_K = C_K ∪ T
12:       delete x from Q
13:    end while
14:    U = U \ C_K
15:    K = K + 1
16: end while
17: return C_1,…, C_K
```

(N is the population size), aiming to ensure diversity in this space.

A. NCM in Decision Space

For a given dataset, NCM will build a partition of this dataset into multiple subsets with each subset representing a cluster. When solving MMOPs, NCM is run to classify solutions into a number of clusters. To distinguish local or global PSs, NCM follows the idea from DBSCAN [45] to identify density connected clusters. For each solution $x=(x_1,...,x_n)$ in a solution set U and the neighborhood radius $r=(r_1,...,r_n)$ (n is the number of decision variables), some concepts related to NCM are introduced as follows:

***r*-neighborhood**: The r-neighborhood $B(x)$ of x is defined as follows:

$$B(x)=\{y \mid \forall i\in\{1,...,n\}, |x_i-y_i|\le r_i, y\in U\}, \tag{2}$$

where $|x_i-y_i|$ is the distance between the i-th variables of x and y.

Direct density-reachable: For each solution $y\in B(x)$, y is said to be direct density-reachable from x.

Density-reachable: For one solution $y\in U$, y is said to be density-reachable from x if and only if there exists a path with k solutions $p_1, p_2,..., p_k$ ($k\ge 2$), where the first solution p_1 is x, the neighboring solutions are **direct density-reachable**, and the last solution p_k is y.

Based on the above concepts, for each solution $x\in U$, one cluster is constructed in NCM to collect all solutions which are density-reachable from x. To clarify the way in which NCM works, its pseudo-code is provided in Algorithm 1 with the inputs: U (a solution set) and $r=(r_1,...,r_n)$ (the neighborhood radius). At first, the neighborhood $B(x)$ of each solution $x\in U$ is identified by (2) in lines 1-3. Please note that the scopes of these neighborhoods are determined by the neighborhood radius r, which will significantly affect the clustering results in NCM. In lines 6-13, one cluster will be constructed at each iterative running, by randomly selecting a solution x from U and then collecting all the density-reachable solutions from x into the K-th cluster C_K, where the cluster index K is initialized as 1 in line 4. In this process, the solution set T in line 10 temporarily reserves the newly found direct density-reachable solutions of x that are also not included in the cluster

Algorithm 2 *Hierarchical Clustering*(P, N)

Input: P, N
Output: $H_1,\ldots,H_N$
1: initialize each x^i in P as a cluster H_i and centroid c_i
2: $S = |P|$
3: **while** $S > N$
4: **for** $i = 1$ to S
5: record the corresponding $d_{\min}(H_i)$ by (5)
6: record the corresponding $index(H_i)$ by (6)
7: **end for**
8: $a = b = -1$, $min = \infty$
9: **for** $i = 1$ to S
10: **if** $d_{\min}(H_i) < min$
11: $min = d_{\min}(H_i)$, $a = i$, $b = index(H_i)$
12: **end if**
13: **end for**
14: $H_a = H_a \cup H_b$ and update its centroid c_a by (4)
15: delete H_b, and S--
16: renumber the S clusters
17: **end while**
18: **return** $H_1,\ldots,H_N$

C_K, and the solution set Q in lines 8-13 is a temporary stack of observations, which will be visited when filling the cluster C_K. After the cluster C_K is built, its solutions should be removed from U as described in line 14, and the value of K is increased by 1 in line 15 to find the next cluster. At last, when all solutions in U are classified into the clusters (i.e., U will be empty in line 5), K clusters $C_1,\ldots,C_K$ are returned in line 17.

B. HCM in Objective Space

HCM aims to maintain diversity of solutions in objective space. Initially, each solution is treated as a cluster and then two clusters are iteratively combined according to a certain linkage criterion. Instead of using a single linkage which tends to produce long thin clusters, in which nearby elements of the same cluster have small distances, but elements at opposite ends of a cluster may be much farther from each other [55], Ward's linkage is used in this paper as suggested in [46]. This way, the sum of squared errors within the same cluster is minimized, while the errors between two distinct clusters are maximized. The sum of squared errors for two clusters H_i and H_j is obtained by

$$d(H_i,H_j) = \sqrt{\frac{2\cdot n_i \cdot n_j}{(n_i+n_j)}}\,\| c_i - c_j \|, \tag{3}$$

where n_i and n_j are the numbers of solutions respectively in clusters H_i and H_j, and $\| c_i - c_j \|$ indicates the Euclidean distance between c_i (the centroid of H_i) and c_j (the centroid of H_j) in objective space. Here, the centroid $c_a = (c_a^1,\ldots,c_a^m)$ of a cluster H_a can be computed as follows:

$$c_a^i = \frac{\sum_{p\in H_a} f_i(p)}{|H_a|},\ i = 1,\ldots,m, \tag{4}$$

where m is the number of objectives and $f_i(p)$ is the i-th objective value of each solution p in cluster H_a.

To clarify the way in which HCM works, its pseudo-code is given in Algorithm 2 with the inputs: P (a set with more than N solutions) and N (the number of clusters). In line 1, each solution $x^i \in P$ ($i = 1,2,\ldots,|P|$) is initialized as a cluster H_i with centroid $c_i = (f_1(x^i),\ldots,f_m(x^i))$, including the objective values of x^i as defined in (1). In line 2, S equal to $|P|$ indicates the number of clusters at first. Then, the procedures in lines 3-17 will be iteratively repeated to produce N final clusters. Specifically, in lines 4-7, the nearest cluster to each cluster H_i is found, which will record the minimal distance to H_i by $d_{\min}(H_i)$ and the index of the nearest cluster to H_i by $index(H_i)$, as follows:

$$d_{\min}(H_i) = \min\ d(H_i,H_j), \tag{5}$$

$$index(H_i) = \arg\min_j\ d(H_i,H_j), \tag{6}$$

where H_j can be any of the clusters excluding H_i. Then, in lines 8-13, the two most similar clusters (i.e., H_a and H_b) are found and recorded, where a and b respectively indicate their indexes in the cluster set. In this process, the parameter *min* is used to record the minimal distance of two clusters in S clusters. In line 14, cluster H_b is combined into cluster H_a and its new centroid c_a is updated by (4). Then, cluster H_b is deleted and S is decreased by 1 in line 15. The remaining clusters will be renumbered as $H_1, \ldots, H_S$ in line 16.

After running the above procedures, only N final clusters will remain, which will be returned as the final clustering result obtained by HCM in line 18.

Algorithm 3 Main Framework of MMOEA/DC

Input: an MMOP with m objectives and n variables, N, $G_{\max}$, λ, β
Output: P
1: initialize P to have N solutions
2: $G = 1$
3: **while** $G \le G_{\max}$
4: P' = *Crowding-based Mating Selection*(P, N) // **Algorithm 4**
5: P' = *Crossover*(P')
6: P' = *Mutation*(P')
7: $U = P \cup P'$
8: P = *Environmental Selection*(U, N, λ, β) // **Algorithm 5**
9: G++
10: **end while**
11: **return** P

IV. The Proposed Algorithm

In this section, the details of our algorithm are presented. First, our main framework is given to have an overview of MMOEA/DC. Then, to clarify the way in which it works, the crowding-based mating selection in decision space is described, and the environmental selection using the above NCM and HCM is introduced, aiming to approximate global PSs and some good local PSs with acceptable quality. Finally, the computational complexity of MMOEA/DC is analyzed.

A. Our Main Framework

Here, the pseudo-code of our main framework is provided in Algorithm 3 with the inputs: an MMOP with m objectives and n variables, N (the population size), G_{max} (the pre-set maximum number of generations), λ (a parameter to control the neighborhood radius), and β (the minimum number of solutions in each local cluster). At first, a population P is initialized in line 1 by randomly generating N solutions and the generation counter G is set to 1 in line 2. Afterwards, the crowding-based mating selection (as introduced in Algorithm 4 with

Algorithm 4 *Crowding-based Mating Selection*(P, N)

Input: P, N
Output: P'
1: initialize P' as an empty set
2: **while** $|P'| < N$
3: randomly select two solutions x and y from P
4: normalize x, y in decision space to get x', y'
5: **if** $HAD(x') < HAD(y')$
6: $P' = P' \cup \{y\}$
7: **else if** $HAD(x') > HAD(y')$
8: $P' = P' \cup \{x\}$
9: **else**
10: randomly select one from x and y, and add it into P'
11: **end if**
12: **end while**
13: **return** P'

the inputs P and N) is run in line 4 to select a mating parent population P' with N solutions for offspring generation. Then, a crossover operator (SBX [4]) in line 5 and a mutation operator (polynomial-based mutation [4]) in line 6 are run sequentially to mutate the offspring population P', which is then combined with P to form a union population U in line 7. At last, the environmental selection (as introduced in Algorithm 5 with the inputs U, N, λ, β) is run in line 8 to select the next population, and the generation counter G is increased by 1 in line 9. While G is smaller than G_{max}, the above iterative process in lines 4-9 will be run. Otherwise, the final population P is reported as our approximate solution set in line 11.

B. Crowding-based Mating Selection

In our design, the crowding-based mating selection adopts a binary tournament selection strategy based on the crowding status in decision space, aiming to select solutions with good diversity in decision space as mating parents. Its pseudo-code is provided in Algorithm 4 with the inputs: P (the population) and N (the population size). In line 1, the mating parent population P' is initialized as an empty set. When its size is smaller than N, the procedures in lines 3-11 will be run to select one solution into P' at each iteration. Two solutions x and y are randomly selected from P in line 3, which are normalized in line 4 respectively as x' and y' in decision space, as follows:

$$x_i' = \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}, i = 1, ..., n, \tag{7}$$

where x_i is the i-th variable of x and normalized as x_i', n is the number of decision variables, while $x_i^{\min}$ and $x_i^{\max}$ are respectively the minimum and maximum values of the i-th variable in P.

In lines 5-8, one normalized solution with a larger crowding distance in decision space is chosen by a binary tournament selection and then its original solution (i.e., x or y) will be added into P'. If they have the same crowding distance, one solution will be randomly selected in line 10.

Here, the HAD method [48] is used to reflect the crowding status of solutions in decision space. For a population P with N solutions, the HAD value of its normalized solution x' can be computed as follows:

$$HAD(x') = \frac{N-1}{\sum_{y' \in P \text{ and } y' \neq x'} 1/\|x' - y'\|}, \tag{8}$$

where $\|x' - y'\|$ is the Euclidean distance between the normalized solutions x' and y' from (7) in decision space. This HAD method considers the Euclidean distances to other normalized solutions, which can fairly reflect the crowding status of solutions in decision space. Generally, a smaller HAD value in (8) indicates a more crowded status of a solution in the population.

When the size of P' is equal to N in line 2, the mating parent population P' is returned in line 13.

Algorithm 5 *Environmental Selection*(U, N, λ, β)

Input: U, N, λ, β
Output: P
1: initialize P as an empty set
2: $(F_1, \ldots, F_t)$ = *Non-dominated Sorting* (U)
3: set r (neighborhood radius) by (9)
4: $(C_1, \ldots, C_K)$ = *Neighborhood-based Clustering*(U, r) //**Algorithm 1**
5: **for** i = 1 to K
6: **if** $|C_i| > \beta$
7: include non-dominated solutions of C_i into P
8: **end if**
9: **end for**
10: $i = 1$
11: **while** $|P| \le N$
12: $P = P \cup F_i$
13: $i = i + 1$
14: **end while**
15: normalize P in objective space by (10)
16: $(H_1, \ldots, H_N)$ = *Hierarchical Clustering*(P, N) //**Algorithm 2**
17: **while** $|P|$!= N
18: identify the most crowded cluster H_c by (11)
19: compute the *HAD* values of solutions in H_c by (8)
20: find one solution x in H_c with the smallest HAD value
21: $P = P \backslash \{x\}$, $H_c = H_c \backslash \{x\}$
22: **end while**
23: **return** P

C. Environmental Selection

The pseudo-code of our environmental selection is given in Algorithm 5 with the inputs: U (the combined population), N (the population size), λ (a parameter to control the neighborhood radius), and β (the minimum number of solutions in each local cluster). At first, P is initialized as an empty set in line 1 and non-dominated sorting [9] is run to find the non-dominated fronts (i.e., $F_1, \ldots, F_t$) from U in line 2. Then, in line 3, the neighborhood radius with n dimensions, i.e., $r = (r_1, ..., r_n)$, is computed as follows:

$$r_i = (x_i^{\max} - x_i^{\min}) \times \lambda, i = 1, ..., n, \tag{9}$$

where $x_i^{\min}$ and $x_i^{\max}$ are respectively the minimum and maximum values of the i-th variable in U, n is the number of decision variables, and λ is a parameter to control the neighborhood radius at each dimension. In line 4, NCM (Algorithm 1) is run to separate U into multiple local clusters in decision space, which will return K clusters $C_1, \ldots, C_K$.

After that, some promising solutions should be selected to fill the temporary population P until its size is larger than N. In lines 5-9, for each local cluster whose size is larger than β, non-dominated sorting [9] is run on this cluster to find its own non-dominated solutions, which are added into P to maintain local PSs. Afterwards, in lines 10-14, other solutions with good convergence in U are further selected when the size of P is not

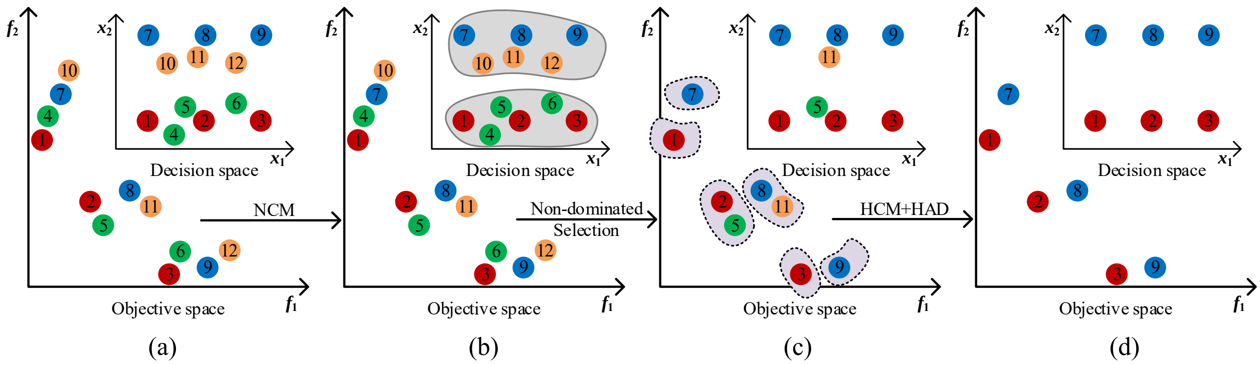


Fig. 3. An example of the way in which our environmental selection works using NCM and HCM for MMOPs with coexistence of local and global PSs.

larger than N. Solutions from F_1 to F_t in an ascending order but not in P are sequentially selected to fill P until its size is larger than N. After completing the above process, there are usually more than N solutions in P.

To mitigate the impact of different scaled objectives in MMOPs, the i-th objective value $f_i(x)$ of each solution x in P is normalized in line 15 using

$$f_i'(x) = \frac{f_i(x) - z_i^{\min}}{z_i^{\max} - z_i^{\min}}, i = 1, \ldots, m, \quad (10)$$

where $f_i'(x)$ is the i-th normalized objective value, m is the number of objectives, while $z_i^{\min}$ and $z_i^{\max}$ are respectively the minimum and maximum values of the i-th objective in P. Then, in line 16, HCM (Algorithm 2) is further run on P to get N clusters $H_1, \ldots, H_N$ based on the normalized objective values. In lines 17-22, the procedures are run to iteratively remove the most crowded solution in decision space from the most crowded cluster in objective space, until the size of P is equal to N. In detail, the most crowded cluster H_c is selected from $H_1, \ldots, H_N$ in line 18, with its index c found as follows:

$$c = \underset{i \in \{1,\ldots,N\}}{\arg\max} \{|H_i|\}, \quad (11)$$

where $|H_i|$ returns the number of solutions in H_i. If more than one cluster has the same maximal size of solutions, only one of them is randomly selected. Then, the HAD values of solutions in H_c are computed by (8) in line 19, which reflect their crowded status in decision space. Also, one solution x with the smallest HAD value is selected from H_c in line 20, which will be removed from P and H_c in line 21. When the size of P is reduced to N in line 17, P is returned in line 23 as the next population.

In order to show the way in which our environmental selection works, a simple example is provided in Figs. 3(a)-3(d) for solving MMOPs with local and global PSs. The union population has twelve solutions marked from 1 to 12 with various colors in Fig. 3(a), where half of the solutions should be selected into the next generation. As shown in Fig. 3(b), NCM (Algorithm 1) is first used in our environmental selection to divide the union population into two local clusters in decision space. In each local cluster, the non-dominated solutions are selected as candidates to maintain local or global PSs. Next, in Fig. 3(c), HCM (Algorithm 2) is applied in objective space to classify these candidates into N clusters (N=6 in this case). Based on the HAD value in (8), one solution in the most crowded cluster with the smallest HAD value will be removed iteratively until there is only one solution in each cluster. This way, our environmental selection can properly approximate local and global PSs in this case, as illustrated in Fig. 3(d). Other traditional environmental selection methods based on Pareto dominance [9]-[11], decomposition [12]-[20], and performance indicators [21]-[23] cannot maintain local PSs, since such local PSs are often dominated by global PSs and are consequently removed.

D. *Computational Complexity Analysis of MMOEA/DC*

The computational complexity of MMOEA/DC in one generation is mainly determined by the crowding-based mating selection in Algorithm 4 and the environmental selection in Algorithm 5. As shown in Algorithm 4, the mating selection requires a time complexity of $O(nN^2)$ (n is the number of decision variables and N is the population size) to select N parent solutions for offspring generation. As shown in Algorithm 5, the computational complexity of this environmental selection is mainly determined by non-dominated sorting in line 2, the dual clustering methods (i.e., NCM in line 4 and HCM in line 16), and the iterative procedures of pruning P in lines 17-22. Non-dominated sorting requires a time complexity of $O(N\log_{m-2} N)$ (m is the number of objectives) [9] in line 2. NCM needs a time complexity of $O(nN^2)$ to run Algorithm 1 in line 4, and HCM needs a time complexity of $O(mN^2)$ to run Algorithm 2 in line 16. In lines 17-22, the iterative procedures of pruning P need a time complexity of $O(nN^3)$ in the worst case. Thus, the overall worst time complexity of MMOEA/DC is $O(nN^3)$ in one generation.

V. Experimental Studies

A. *Benchmark Problems and Performance Indicators*

In this paper, a test suite of MMOPs from the CEC 2019 competition [31] was employed. This benchmark is characterized for including various shapes of PSs and PFs, coexistence of local and global PSs, and scalable numbers of PSs, decision variables and objectives.

Inverted generational distances in decision space (IGDX [36]) and in objective space (IGDF [56]-[57]) were adopted as performance indicators in this paper. These indicators respectively reflect the performance of algorithms in decision and objective spaces. The smaller the values of IGDX and IGDF

TABLE I
COMPARISON OF RESULTS OF MMOEA/DC AND FIVE COMPETITIVE MMOEAS ON THE CEC 2019 MMOPS USING IGDX

Problem	Omni-Optimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R	MMOPIO	MMOEA/DC
SYM-PART simple	4.992e+00(1.65e+00)-	4.524e+00(1.56e+00)-	1.708e-01(2.32e-02)-	**2.346e-02(8.51e-03)+**	1.387e-01(2.31e-01)-	5.017E-02(3.34E-03)
SYM-PART rotated	4.630e+00(1.64e+00)-	4.118e+00(1.91e+00)-	2.610e-01(2.37e-01)-	1.725e+00(1.35e+00)-	9.579e-02(1.79e-02)-	**8.169E-02(6.09E-03)**
Omni-test	1.548e+00(2.10e-01)-	1.456e+00(1.53e-01)-	3.919e-01(7.86e-02)-	2.665e-01(1.73e-01)-	3.185e-01(1.17e-01)-	**1.013E-01(3.77E-03)**
MMF1	9.360e-02(1.46e-02)-	9.758e-02(1.58e-02)-	4.905e-02(2.27e-03)-	7.116e-02(8.30e-03)-	**4.189e-02(1.95e-03)+**	4.561E-02(1.99E-03)
MMF1_z	7.616e-02(2.42e-02)-	8.265e-02(1.28e-02)-	3.560e-02(1.44e-03)-	6.967e-02(1.14e-02)-	3.102e-02(2.07e-03)~	**3.102E-02(1.39E-03)**
MMF1_e	1.265e+00(5.45e-01)-	1.213e+00(5.47e-01)-	5.267e-01(1.30e-01)~	1.782e+00(7.15e-01)-	**4.055e-01(1.17e-01)+**	5.397E-01(2.50E-01)
MMF2	1.056e-01(6.81e-02)-	1.098e-01(7.05e-02)-	4.181e-02(1.19e-02)-	6.484e-02(2.64e-02)-	**1.618e-02(6.45e-03)+**	1.978E-02(6.50E-03)
MMF3	9.915e-02(4.43e-02)-	8.647e-02(3.22e-02)-	3.229e-02(1.37e-02)~	9.342e-02(3.76e-02)-	**1.206e-02(2.58e-03)+**	3.267E-02(6.44E-03)
MMF4	8.658e-02(3.10e-02)-	8.590e-02(1.83e-02)-	2.742e-02(1.83e-03)~	1.038e-01(1.46e-01)-	2.787e-02(3.68e-03)~	**2.646E-02(2.73E-03)**
MMF5	1.683e-01(1.33e-02)-	1.688e-01(1.86e-02)-	8.527e-02(3.95e-03)-	1.142e-01(1.15e-02)-	8.343e-02(9.25e-03)-	**7.834E-02(2.70E-03)**
MMF6	1.464e-01(1.50e-02)-	1.417e-01(1.79e-02)-	7.266e-02(3.98e-03)-	9.165e-02(9.98e-03)-	7.036e-02(4.32e-03)-	**6.727E-02(3.22E-03)**
MMF7	5.171e-02(1.35e-02)-	5.179e-02(1.07e-02)-	2.645e-02(1.43e-03)+	4.775e-02(2.24e-02)-	**2.210e-02(1.87e-03)+**	2.867E-02(2.84E-03)
MMF8	2.682e-01(1.07e-01)-	2.720e-01(1.08e-01)-	6.836e-02(5.86e-03)~	3.501e-01(9.97e-02)-	**6.214e-02(9.72e-03)+**	7.200E-02(1.48E-02)
MMF9	2.147e-02(1.10e-02)-	2.311e-02(1.29e-02)-	7.987e-03(5.00e-04)-	**3.127e-03(8.43e-05)+**	6.732e-03(1.24e-03)~	6.629E-03(2.52E-04)
MMF10	1.753e-01(3.49e-02)-	1.448e-01(3.73e-02)-	1.037e-01(4.18e-02)-	2.014e-01(7.04e-05)-	1.595e-01(3.50e-02)-	**3.620E-02(6.18E-02)**
MMF11	2.502e-01(3.82e-04)-	2.506e-01(4.18e-04)-	2.112e-01(2.61e-02)-	2.524e-01(8.18e-05)-	2.481e-01(1.48e-03)-	**7.570E-03(2.98E-04)**
MMF12	2.431e-01(1.20e-02)-	2.465e-01(6.20e-04)-	1.862e-01(4.18e-02)-	2.478e-01(6.01e-04)-	2.327e-01(3.03e-02)-	**3.137E-03(1.83E-04)**
MMF13	2.840e-01(8.73e-03)-	2.871e-01(1.32e-02)-	2.399e-01(1.51e-02)-	2.726e-01(6.24e-03)-	2.575e-01(8.06e-03)-	**9.026E-02(2.81E-02)**
MMF14	9.207e-02(7.75e-03)-	9.651e-02(8.39e-03)-	5.390e-02(1.66e-03)-	**3.655e-02(4.91e-04)+**	5.975e-02(2.67e-03)-	5.130E-02(1.32E-03)
MMF14_a	1.107e-01(8.03e-03)-	1.189e-01(8.02e-03)-	6.075e-02(1.50e-03)+	**5.655e-02(1.75e-03)+**	6.649e-02(3.00e-03)+	7.716E-02(3.69E-03)
MMF15	2.355e-01(2.92e-02)-	2.247e-01(2.62e-02)-	1.533e-01(1.35e-02)-	2.711e-01(2.94e-04)-	1.699e-01(2.65e-02)-	**5.356E-02(1.46E-03)**
MMF15_a	2.137e-01(1.38e-02)-	2.107e-01(1.59e-02)-	1.619e-01(1.28e-02)-	2.195e-01(2.65e-03)-	1.696e-01(1.42e-02)-	**1.002E-01(2.23E-02)**
best/all	0/22	0/22	0/22	4/22	6/22	12/22
+/–/~	0/22/0	0/22/0	2/16/4	4/18/0	7/12/3	– –

"+", "–", and "~" indicate that the results of the compared algorithms are significantly better than, worse than, and similar to that of MMOEA/DC respectively by Wilcoxon's rank sum test with $\alpha = 0.05$. In addition, the best metric values are highlighted in **bold face**.

the better the approximations to the true PS and PF, respectively. Due to page limitations, please refer to [36], [56] and [57] for details of IGDX and IGDF.

B. Parameters Settings for the Compared Algorithms

In this paper, five competitive algorithms (Omni-optimizer [35], DN-NSGA-II [36], MO_Ring_PSO_SCD [52], TriMOEA-TA&R [32] and MMOPIO [54]) are included for performance comparison. The source codes of these compared algorithms were provided by their authors and our source code is available at https://github.com/wulinszu/MMOEA-DC.git. All these algorithms were implemented in MATLAB and run on a personal computer with an Intel (R) Core (TM) i7-7700 CPU, 3.60GHz (processor), and 24 GB (RAM). The parameters settings of the compared algorithms are listed in Table S-I of the supplementary file, as suggested in their references. In DN-NSGA-II, the crowding factor (*CF*) is set to half of the population size (*N*). In MO_Ring_PSO_SCD, the coefficients C_1 and C_2 in the velocity update equation are all set to 2.05 and the inertia weight *W* is set to 0.7298. In TriMOEA-TA&R, the probability to select parents from the convergence archive (p_{con}), the niche radius in decision space (δ_{niche}), the accuracy level (ε_{peak}), the number of reference points (*NR*), the number of sampling solutions in control variable analysis (*NCA*), and the maximum number of trails required to judge the interaction (*NIA*) are set to 0.5, 0.1, 0.01, 100, 20 and 6, respectively. In MMOPIO, the maximum size of the personal best archive (n_{PBA}) is set to 5 and the archive size (*A*) is set to the population size *N*. For MMOEA/DC, the parameter λ used to control the neighborhood radius in NCM is set to 0.1, while the parameter β controlling the minimum number of solutions in each local cluster is set to 5, the impact of which will be studied in Section IV.F. Please note that Omni-optimizer, DN-NSGA-II and MMOEA/DC use two evolutionary operators (i.e., SBX with the crossover probability $p_c = 1.0$ and the distribution index $\eta_c = 20$, and polynomial-based mutation with the mutation probability $p_c = 1/n$ and the distribution index $\eta_m = 20$, where *n* is the number of decision variables), while MO_Ring_PSO_SCD, TriMOEA-TA&R, and MMOPIO respectively adopt a particle position update method, a novel recombination strategy, and an elite learning strategy to generate new solutions.

As suggested in [31], the population size *N* and the maximum number of function evaluation *MaxFES* for different test problems were set by *N* = 100×*n* and *MaxFES* = 5000×*n*, where *n* is the number of decision variables. Thus, the maximum number of generations G_{max} was set to 50. All the compared MMOEAs were run independently 31 times on each test problem. The mean values and standard deviations (included in brackets after the mean values) for IGDX and IGDF from 31 runs are collected for comparison. In order to obtain a statistically sound conclusion, a Wilcoxon rank sum test is run with a significance level $\alpha = 0.05$, which shows the statistically

TABLE II
COMPARISON OF RESULTS OF MMOEA/DC AND FIVE COMPETITIVE MMOEAS ON THE CEC 2019 MMOPS USING IGDF

Problem	Omni-Optimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R	MMOPIO	MMOEA/DC
SYM-PART simple	1.223e-02(1.61e-03)-	1.295e-02(1.67e-03)-	4.070e-02(5.55e-03)-	3.328e-02(5.39e-03)-	1.423e-02(1.83e-03)-	**1.106E-02(1.19E-03)**
SYM-PART rotated	**1.300e-02(1.55e-03)+**	1.534e-02(2.19e-03)+	4.722e-02(5.12e-03)-	2.810e-02(4.49e-03)-	1.646e-02(2.61e-03)~	1.700E-02(2.66E-03)
Omni-test	**6.939e-03(5.29e-04)+**	7.774e-03(4.92e-04)+	4.051e-02(4.11e-03)-	1.803e-02(3.85e-03)+	1.098e-02(1.65e-03)+	2.330E-02(1.24E-03)
MMF1	3.663e-03(4.40e-04)~	4.476e-03(6.12e-04)-	3.738e-03(1.77e-04)-	5.046e-03(1.33e-03)-	**2.748e-03(1.00e-04)+**	3.499E-03(1.44E-04)
MMF1_z	3.141e-03(4.12e-04)+	4.012e-03(1.49e-03)-	3.594e-03(2.00e-04)-	4.380e-03(9.02e-04)-	**2.597e-03(1.30e-04)+**	3.325E-03(3.45E-04)
MMF1_e	2.144e-02(1.59e-02)-	2.230e-02(1.21e-02)-	1.190e-02(1.41e-03)-	7.900e-03(3.17e-03)-	6.047e-03(1.02e-03)~	**5.627E-03(6.89E-04)**
MMF2	2.043e-02(1.96e-02)~	2.562e-02(1.56e-02)-	2.185e-02(6.92e-03)-	2.361e-02(8.31e-03)-	**8.308e-03(1.82e-03)+**	1.039E-02(3.23E-03)
MMF3	1.781e-02(1.35e-02)-	1.958e-02(1.07e-02)-	1.666e-02(4.97e-03)-	4.829e-02(6.74e-02)-	**6.828e-03(1.28e-03)+**	9.109E-03(1.40E-03)
MMF4	2.881e-03(1.68e-04)~	3.223e-03(2.59e-04)~	3.551e-03(2.12e-04)-	3.588e-02(6.85e-02)~	**2.690e-03(1.34e-04)~**	2.982E-03(6.39E-04)
MMF5	3.338e-03(4.66e-04)+	3.821e-03(5.75e-04)-	3.720e-03(1.79e-04)-	4.837e-03(2.05e-03)-	**2.719e-03(1.13e-04)+**	3.542E-03(1.42E-04)
MMF6	3.245e-03(2.85e-04)+	3.727e-03(3.31e-04)-	3.507e-03(1.64e-04)~	4.379e-03(1.77e-03)-	**2.608e-03(8.89e-05)+**	3.529E-03(1.41E-04)
MMF7	3.093e-03(2.50e-04)+	4.019e-03(4.27e-04)-	3.690e-03(2.39e-04)~	4.385e-03(1.38e-03)~	**2.673e-03(6.87e-05)+**	3.560E-03(5.25E-04)
MMF8	3.269e-03(3.34e-04)-	3.960e-03(5.16e-04)-	4.831e-03(2.56e-04)-	5.797e-03(7.34e-03)-	3.043e-03(1.94e-04)-	**2.779E-03(3.63E-04)**
MMF9	1.277e-02(9.10e-04)-	1.422e-02(1.76e-03)-	1.575e-02(1.76e-03)-	6.992e-02(4.05e-03)-	1.248e-02(1.24e-03)-	**1.032E-02(5.82E-04)**
MMF10	1.929e-01(3.60e-02)-	1.867e-01(4.05e-02)-	1.345e-01(1.64e-02)-	2.283e-01(4.90e-03)-	1.549e-01(2.77e-02)-	**3.891E-02(4.90E-02)**
MMF11	9.604e-02(9.94e-04)-	9.874e-02(2.18e-03)-	8.653e-02(7.74e-03)-	1.628e-01(8.42e-03)-	9.195e-02(5.04e-03)-	**2.089E-02(6.02E-04)**
MMF12	8.448e-02(5.76e-03)-	8.330e-02(3.01e-04)-	6.865e-02(1.39e-02)-	8.589e-02(1.30e-03)-	8.052e-02(1.01e-02)-	**3.881E-03(1.13E-04)**
MMF13	1.472e-01(2.39e-03)-	1.511e-01(2.93e-03)-	1.000e-01(2.25e-02)-	2.435e-01(7.31e-03)-	1.166e-01(2.56e-02)-	**2.242E-02(3.42E-03)**
MMF14	9.996e-02(4.73e-03)-	1.109e-01(7.50e-03)-	8.056e-02(2.77e-03)-	8.631e-02(1.12e-03)-	8.067e-02(2.48e-03)-	**6.673E-02(1.99E-03)**
MMF14_a	1.047e-01(6.05e-03)-	1.191e-01(6.62e-03)-	7.823e-02(1.73e-03)-	7.894e-02(1.31e-03)-	7.731e-02(2.25e-03)-	**6.731E-02(2.23E-03)**
MMF15	2.020e-01(8.82e-03)-	2.148e-01(8.31e-03)-	1.736e-01(2.77e-03)-	2.066e-01(7.90e-04)-	1.735e-01(3.58e-03)-	**1.004E-01(1.76E-03)**
MMF15_a	2.058e-01(8.10e-03)-	2.240e-01(9.12e-03)-	1.734e-01(3.27e-03)-	1.945e-01(3.99e-03)-	1.751e-01(3.20e-03)-	**1.392E-01(1.35E-02)**
best/all	2/22	0/22	0/22	0/22	8/22	12/22
+/–/~	6/13/3	2/19/1	0/20/2	1/19/2	8/11/3	– –

"+", "–", and "~" indicate that the results of the compared algorithms are significantly better than, worse than, and similar to that of MMOEA/DC respectively by Wilcoxon's rank sum test with $\alpha = 0.05$. In addition, the best metric values are highlighted in **bold face**.

significant differences on the results of MMOEA/DC and other algorithms. In the following tables, the symbols "+", "–", and "~" indicate that the results of other algorithms are significantly better than, worse than, and similar to those of MMOEA/DC, respectively.

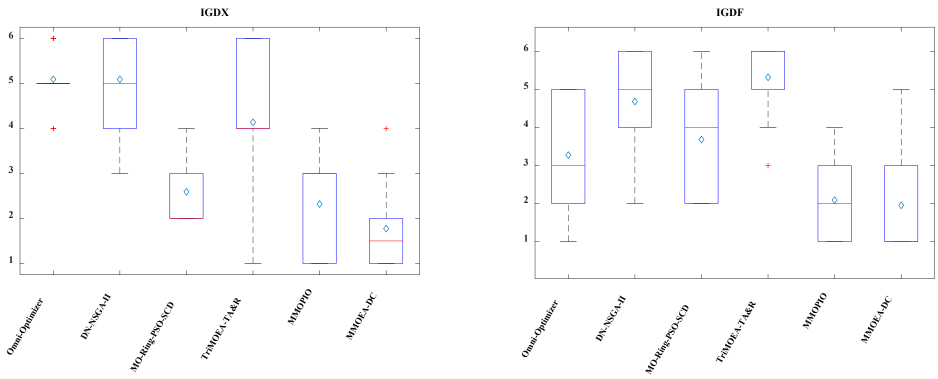
C. Comparison with Five Competitive MMOEAs

1) Comparison of Results on MMOPs

The detailed IGDX and IGDF results are listed in Table I and Table II, respectively. In the second to last row of each table, the numbers of problems in which the compared algorithms performed best are summarized. From this table, it can be seen that MMOEA/DC performed best on 12 out of 22 cases regarding both IGDX and IGDF. Thus, MMOEA/DC showed the best overall performance. In the last row of each table, a summary of the significance test was also provided for the compared algorithms, using the symbol "+/–/~" to summarize the numbers of problems in which other algorithms respectively performed better than, worse than and similarly to MMOEA/DC. From these summarized results, MMOEA/DC also performed significantly better than the other algorithms when solving most of the test problems adopted, in terms of both IGDX and IGDF. Thus, it is reasonable to conclude that MMOEA/DC showed a superior performance over the five competitors for solving most of the MMOPs adopted. Moreover, in order to visually show the distributions of these results, the box plots of these IGDX and IGDF results for all the compared algorithms are respectively provided in Figs. S-1 and S-2 of the supplementary file, due to page limitations.

For MMOPs with coexistence of local and global PSs, i.e., MMF10, MMF11, MMF12, MMF13, MM15 and MMF15_a, MMOEA/DC offered some advantages as it obtained the best results in all cases regarding IGDX and IGDF. As NCM is run in decision space, MMOEA/DC can distinguish more local and global PSs, which can strengthen diversity in both decision and objective spaces. Omni-optimizer and DN-NSGA-II used the non-dominated sorting method, which preferred to select the solutions with better convergence in objective space. Thus, local PSs of these problems hardly survived during the evolutionary process. MO_Ring_PSO_SCD and MMOPIO considered the diversity of solutions in decision space for the problems with multiple PSs for the same PF, while paying little attention to local PSs. Although two archives and recombination strategies were adopted, TriMOEA-TA&R still gave priority to the solutions with better convergence in objective space. Thus, the above compared algorithms can well approximate global PSs, but ignore local PSs in their evolutionary search.

For other MMOPs with multiple PSs to the same global PF, MMOEA/DC also showed advantages in the cases having a larger number of PSs. MMOEA/DC achieved the best results on SYM-PART rotated, Omni-test, MMF1_z, and MMF4-6



(a) (b)

Fig. 4. The boxplots of the performance ranks of Friedman's test on IGDX and IGDF.

using IGDX, and on SYM-PART simple, MMF1_e, MMF8, MMF9, MMF14, and MMF14_a using IGDF. However, using IGDX, MMOEA/DC was outperformed by TriMOEA-TA&R on 4 cases (e.g., SYM-PART simple and MMF9) and by MMOPIO on 7 cases (e.g., MMF1_e and MMF1-3). MMOPIO achieved the best results on MMF1, MMF1_z, and MMF2-7 using IGDF, while Omni-optimizer was best on SYM-PART simple and Omni-test. MMOEA/DC performed relatively worse on the above mentioned cases. This is reasonable as some noise in the evolutionary search may cause two poor clustering cases, i.e., solutions that should approximate multiple PSs for the same PF may be classified into one cluster and solutions with poor performance may be all classified into one cluster. In the first case, it will be difficult for our environmental selection to select solutions that can evenly cover the true PSs, whereas in the second case, solutions with poor performance will be always maintained by our environmental selection. Moreover, the impact of the neighborhood radius *r* also disturbs NCM to distinguish multiple PSs for the same PF, which may partially lose some local or global PSs.

Based on the above analysis and discussions, it is obvious that MMOEA/DC showed some advantages on solving most of the MMOPs adopted, especially on MMOPs with coexistence of local and global PSs. An NCM was applied in decision space to maintain local or global PSs and an HCM was applied in objective space to maintain diversity of local and global PFs, which could properly approximate local and global PSs during the evolutionary search. MMOPIO obtained the best results on MMF1, MMF2, MMF3 and MMF7 using IGDX and IGDF, but its local PSs were still lost during the evolutionary search. TriMOEA-TA&R achieved the best results on SYM-PART simple, MMF14 and MMF14_a using IGDX, as its convergence and diversity archives can cooperatively solve MMOPs, but could not show advantages for other kinds of MMOPs. MO_Ring_PSO_SCD showed a median performance on most MMOPs, because its special crowding distance in decision and objective spaces cannot distinguish between local and global PSs. Even though Omni-optimizer and DN-NSGA-II considered the crowding distance in decision space, they still employed non-dominated sorting to select solutions in objective space, which results in poor performance on most cases.

Moreover, to quantify the overall performance of each algorithm, their performance ranks obtained from Friedman's test [58] on all the test problems are illustrated by boxplots in Fig. 4(a) on IGDX and Fig. 4(b) on IGDF. Note that the red lines represent their median performance ranks in boxplots, while the blue rectangles indicate their average performance ranks. Obviously, the median and average performance ranks of MMOEA/DC are much smaller than those of other competitors on both IGDX and IGDF, which confirms the advantages of our proposed approach when solving all the test problems. Moreover, the *p*-values obtained from Bonferroni-Dunn's and Holm's post hoc procedure in the software tool KEEL [59] are provided in Table S-II and Table S-III of the supplementary file, which show the significant differences of the IGDX and IGDF results among these compared algorithms. Please note that a *p*-value closer to 0 means a more significant difference on the results. Most *p*-values in Tables S-II and S-III are very close to 0, indicating that MMOEA/DC showed a superior performance with statistical significance over its competitors.

To show the actual runtime of all the compared algorithms, their average running times (in seconds: s) from 31 runs are plotted in Fig. S-3 of the supplementary file for all the test problems. Obviously, TriMOEA-TA&R showed the fastest speed, as only N_A (the number of peak solutions) plus N_D (the size of diversity archive) solutions were maintained during the evolutionary search, which offers a significant advantage in terms of computational efficiency. Omni-optimizer and DN-NSGA-II used the same framework from NSGA-II, showing a similar running speed. MO_Ring_PSO_SCD and MMOPIO showed the worst running speed on most cases, as they used a complex density metric to maintain diversity in decision space. Although MMOEA/DC used two clustering methods (NCM and HCM), it still showed a median running speed in most cases. However, for the problems with three decision variables or three objectives like MMF13, MMF14, MMF14_a, MMF15 and MMF15_a in Fig. S-3, MMOEA/DC had the slowest execution speed as more time was required to run the clustering methods for MMOPs with high-dimensional decision or objective spaces.

2) A Further Discussion and Analysis on MMOEA/DC

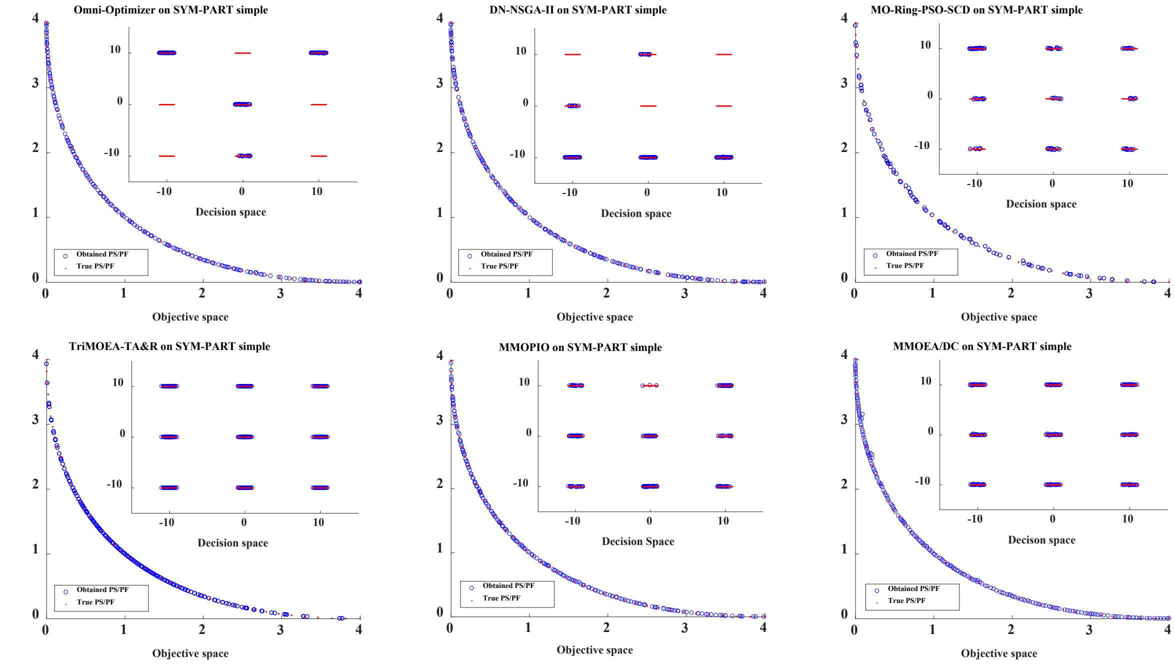


Fig. 5. The final solutions obtained by different MMOEAs on SYM-PART simple.

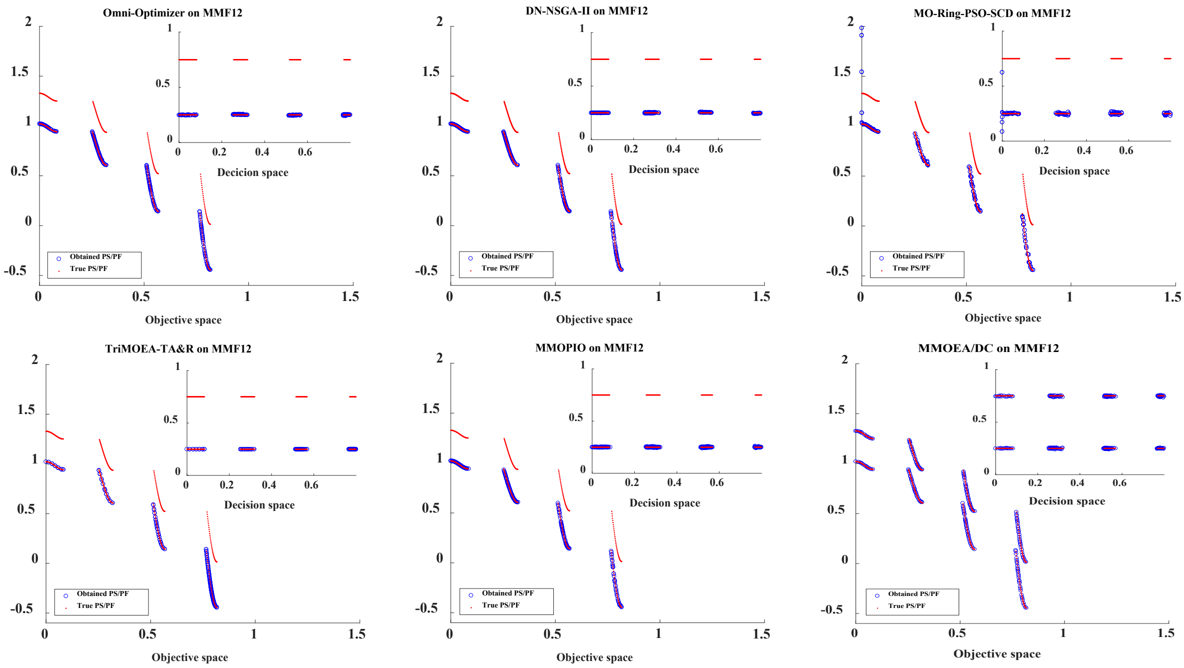
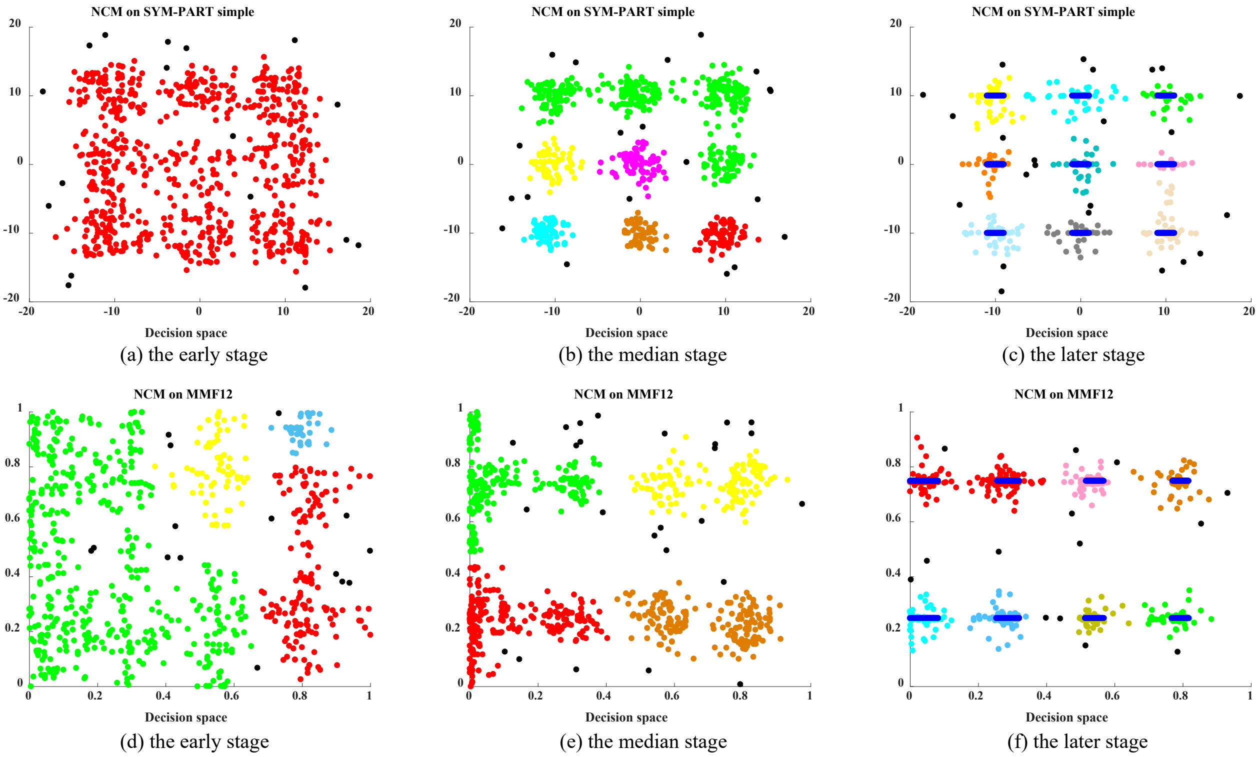


Fig. 6. The final solutions obtained by different MMOEAs on MMF12.

To visually show the performance of the compared algorithms, the final solution sets with the median IGDF values from 31 runs are plotted in Fig. 5 for SYM-PART simple and in Fig. 6 for MMF12.

For SYM-PART simple in Fig. 5, Omni-optimizer and DN-NSGA-II could not find the whole PSs, which lost some PSs during the evolutionary search. MO_Ring_PSO_SCD and MMOPIO were able to maintain all the PSs for the same PF in objective space, mainly due to their special crowding distance mechanism. However, there is plenty of room for improving their diversity in decision space, as their approximate solutions did not uniformly cover the true PSs. TriMOEA-TA&R and MMOEA/DC could obtain the better approximate PSs in decision space, while MMOEA/DC even outperformed TriMOEA-TA&R in objective space.

Moreover, for the problems with coexistence of local and global PSs, MMOEA/DC showed the obvious advantages as its competitors did not consider maintaining local PSs in their design. When solving MMF12 with coexistence of disconnected local and global PSs in Fig. 6, MMOEA/DC was the only one that could find a good approximation for disconnected local and global PSs, while other compared algorithms



(a) the early stage (b) the median stage (c) the later stage

(d) the early stage (e) the median stage (f) the later stage

Fig. 7. The process of NCM on SYM-PART simple and MMF12 during different evolutionary stages.

always lost local PSs, as they were only designed to seek for multiple global PSs for the same PF.

Moreover, in order to further study the search behavior of MMOEA/DC, the process of NCM, which plays a crucial role to maintain diversity in decision space, is visually depicted in Fig. 7, where solutions are marked by black color when the number of solutions in their cluster is less than β, while other clusters are identified with different colors. Please note that solutions of the union population U in line 7 of Algorithm 3 are all plotted in Fig. 7, and the true PSs are all marked by blue color in Figs. 7(c) and 7(f). SYM-PART simple and MMF12 are selected as examples to show the evolutionary behavior of NCM in MMOEA/DC for tackling MMOPs. In these experiments, the population size is 400 and the maximum number of function evaluation is 2000. With the growth of the population, the solutions will be very crowded. Thus, the parameter λ in NCM is adjusted to 0.05 to get a suitable neighborhood radius. As shown in Figs. 7(a) and 7(d), at the early stage of the evolutionary process on SYM-PART simple and MMF12, it is difficult for NCM to divide the local clusters based on their neighborhood relationships, as solutions are randomly distributed in decision space. With the running of generations, solutions in decision space tend to approach the true PSs. For example, as shown in Figs. 7(b) and 7(e), in the median stage of evolution, more local clusters can be obtained by NCM in environmental selection, which maintains diversity in decision space. Specifically, on SYM-PART simple with nine PSs mapping to the same PF, non-dominated solutions in each local cluster will be selected in priority, which helps to approach all global PSs. As depicted in Fig. 7(c), MMOEA/DC can approximate all global PSs of SYM-PART simple at a later stage of the evolutionary process. Moreover, for MMF12 with coexistence of local and global PSs, NCM also plays a critical role to approximate local and global PSs, as they are classified into different clusters in Fig. 7(f), which will be selected simultaneously for the next generation in the environmental selection.

D. Comparison with the winner algorithm at the CEC 2019 competition on MMOPs

During the review process of this paper, the winner algorithm (MMO-Clustering_PSO) at the CEC 2019 competition on MMOPs was announced[1] and its source code was provided to us by its authors. Here, MMOEA/DC was compared with MMO-Clustering_PSO using the same population size and maximum number of function evaluation adopted in this paper. Due to page limitations, their detailed IGDX and IGDF results from 31 runs on all the CEC 2019 MMOPs are listed in Table S.IV of the supplementary file. From the summary of the significance test in the last row, MMOEA/DC had a competitive performance with respect to MMO-Clustering_PSO. MMOEA/DC was slightly better on IGDF but slightly worse on IGDX. However, it was found that MMO-Clustering_PSO also lost local PSs, as it was worse than MMOEA/DC on MMOPs with coexistence of local and global PSs (i.e., MMF10, MMF11, MMF12, MMF13, MM15 and MMF15_a). Moreover, MMO-Clustering_PSO performed relatively worse on MMOPs with a large number of global PSs (i.e., SYM-PART simple, SYM-PART rotated and Omni-test). To visually show their performance comparison on these kinds of MMOPs, their final solution sets with respect to the median

[1] http://www5.zzu.edu.cn/ecilab/info/1036/1211.htm

Table III
THE SUMMARIZED RESULTS OF MMOEA/DC AND ITS VARIANTS

Different Parameters setting	+/-/~(IGDX)	+/-/~(IGDF)
Random Mating Selection VS Crowding-based Mating Selection	3/14/5	3/9/10
BLX-alpha VS SBX	7/12/3	1/20/1
$\beta=1$ VS $\beta=5$	5/2/15	1/1/20
$\beta=10$ VS $\beta=5$	1/3/18	1/0/21
$\beta=20$ VS $\beta=5$	0/4/18	2/1/19
$\lambda=0.02$ VS $\lambda=0.1$	2/15/5	3/17/2
$\lambda=0.05$ VS $\lambda=0.1$	6/13/3	6/10/6
$\lambda=0.08$ VS $\lambda=0.1$	2/5/15	5/6/11
$\lambda=0.15$ VS $\lambda=0.1$	5/5/12	3/5/14
$\lambda=0.2$ VS $\lambda=0.1$	4/10/8	2/8/12

IGDF values from 31 runs are plotted for five low-dimensional cases in the supplementary file, i.e., MMF10, MMF11 and MMF12 in Fig. S-4, and SYM-PART simple and SYM-PART rotated in Fig. S-5. It can be easily observed that MMO-Clustering_PSO almost missed all the local PSs for solving MMF10, MMF11 and MMF12, and could not evenly cover the global PSs for solving SYM-PART simple and SYM-PART rotated. These experiments further confirm the advantages of our proposed approach in maintaining global PFs and some good local PFs.

E. The Impact of the Crossover Operator and Mating Selection

First, to study the impact of crossover operator, BLX-alpha [60] was embedded into MMOEA/DC to replace SBX. Their summarized results are collected in Table III, while the mean values and the corresponding standard deviations of IGDX and IGDF are provided in Table S-V of the supplementary file. From the experimental results, the use of SBX in MMOEA/DC was more effective as it performed better or similarly in 15 and 21 out of 22 cases regarding IGDX and IGDF, respectively.

Second, to study the impact of mating selection, our algorithm was compared to its variant with random mating selection. Their summarized results are collected in Table III, while the mean values and standard deviations of IGDX and IGDF are provided in Table S-VI and Table S-VII of the supplementary file. The effectiveness of crowding-based mating selection in MMOEA/DC was validated, as it performed better or similarly on 19 out of 22 cases regarding both of IGDX and IGDF.

F. Parameter Sensitivity Analysis of MMOEA/DC

In order to study the impact of parameter β (indicating the minimum number of solutions in each local cluster) and λ (controlling the neighborhood radius) in environmental selection, more comparative experiments were performed. Due to page limitations, our summarized results using different parameters are collected in Table III, while the mean values and standard deviations of IGDX and IGDF are provided in Tables S-VI to S-IX of the supplementary file.

First, keeping the same parameters settings as introduced in Section V.B, MMOEA/DC with different β values from $\{1,5,10,20\}$ were experimentally compared. As observed from Tables III, MMOEA/DC using β=5 had statistically similar IGDX results with respect to those using β=1, β=10 and β=20, respectively on 15, 18 and 18 out of 22 cases, and showed statistically similar IGDF results with respect to those using β=1, β=10 and β=20, respectively on 20, 21 and 19 out of 22 cases. It seems that MMOEA/DC is not so sensitive to the setting of β when considering all the adopted MMOPs. However, a too large β value may cause a loss of diversity in decision space, as only few clusters can be obtained, while a too small β value may produce lots of isolated clusters, which may highly exceed the actual number of discrete PSs. To guarantee diversity in decision space and get a reasonable number of clusters, β=5 is suggested for MMOEA/DC in this paper when solving all the adopted MMOPs.

Second, also keeping the same parameters settings as before, MMOEA/DC with different λ values from {0.02, 0.05, 0.08, 0.1, 0.15, 0.2} were compared. From Tables III, MMOEA/DC using $\lambda=0.1$ obtained significantly better or statistically similar IGDX results than those using $\lambda=0.02$, $\lambda=0.05$, $\lambda=0.08$, $\lambda=0.15$ and $\lambda=0.2$, respectively on 20, 16, 20, 17 and 18 out of 22 cases, and achieved significantly better or statistically similar IGDF results than those using $\lambda=0.02$, $\lambda=0.05$, $\lambda=0.08$, $\lambda=0.15$ and $\lambda=0.2$, respectively on 19, 16, 17, 19 and 20 out of 22 cases. From these summarized results, it is found that a too large λ value (e.g., $\lambda=0.2$) or a too small λ value (e.g., $\lambda=0.02$ or $\lambda=0.05$) will result in a failure to correctly classify local or global PSs by NCM. This is mainly because a too large λ value may lead to a loss of local PSs, while a too small λ value may induce difficulties to converge.Thus, λ values around 0.1 are suggested in MMOEA/DC when solving all the MMOPs adopted in this paper.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a novel MMOEA with dual clustering in decision and objective spaces (called MMOEA/DC) has been proposed. It runs dual clustering (NCM in decision space and HCM in objective space) to maintain diversity in both spaces, and it is able to approximate both global PSs and good local PSs with acceptable quality for solving MMOPs. When compared to five competitive algorithms (Omni-optimizer, DN-NSGA-II, MO_Ring_PSO_SCD, TriMOEA-TA&R and MMOPIO), MMOEA/DC showed some advantages, especially for solving MMOPs with coexistence of local and global PSs, which is mainly due to the use of dual clustering to effectively distinguish global PSs and good local PSs. Moreover, some experiments have been run to validate the effect of clustering and crowding-distance mating selection in MMOEA/DC. The impact of two parameters β (indicating the minimum number of solutions in local clusters) and λ (controlling the neighborhood radius) in environmental selection are also studied.

In our future work, we will further study the performance of our algorithm on more complicated problems since the test MMOPs adopted in this paper only have low dimensionality in both decision and objective spaces. We will further enhance the performance of dual clustering by adaptively controlling the neighborhood radius, trying to reduce its time complexity especially for solving MMOPs with a large number of decision variables and/or objectives. Moreover, we will also extend our algorithm for solving some real-world engineering problems.

REFERENCES

[1] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms. New York, NY, USA: John Wiley & Sons, Inc., 2001

[2] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, 1995.

[3] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 440-462, 2017.

[4] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," in *Complex Syst.*, Apr. 1995, vol. 9, pp. 115–148.

[5] H. Li, and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.

[6] O. Schütze, A. Martín, A. Lara, S. Alvarado, E. Salinas, and C. A. Coello, "The directed search method for multi-objective memetic algorithms," *Computational Optimization and Applications*, vol. 63, no. 2, pp. 305-332, 2016.

[7] O. Schuetze, G. Sanchez, and C. A. Coello, "A new memetic strategy for the numerical treatment of multi-objective optimization problems," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008, pp. 705-712.

[8] F. Daolio, A. Liefooghe. S. Verel, H. Aguirre, and K. Tanaka, "Global vs Local Search on Multi-objective NK-Landscapes: Contrasting the Impact of Problem Features," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 369-376.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[10] K. Deb, and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints," *IEEE Trans. Evol. Comput.* vol. 18, no. 4, pp. 577–601, 2014.

[11] Y. Tian, R. Cheng, X. Zhang, Y. Sun, and Y. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 331–345, 2019.

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

[13] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff decomposition approaches for multi-objective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 226–244, 2018.

[14] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 376-390, 2019.

[15] Q. Lin, G. Jin, Y. Ma et al., "A diversity-enhanced resource allocation strategy for decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2388–2401, 2018.

[16] R. Wang, Z. B. Zhou, and H. Ishibuchi, "Localized weighted sum method for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 3-18, 2018.

[17] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 296–313, 2018.

[18] H. Xu, W. Zeng, and D. Zhang, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, 2019.

[19] M. Wu, K. Li, S. Kwong, and Q. Zhang, "Evolutionary many-objective optimization based on adversarial decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 753-764, 2020.

[20] X. He, Y. Zhou, Z. Chen, and Q. Zhang, "Evolutionary many-objective optimization based on dynamical decomposition," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 361–375, 2019.

[21] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.

[22] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "A indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 609–622, 2017.

[23] T. Pamulapati, R. Mallipeddi, and P. Suganthan, "I_{SDE}^{+} - An indicator for multi and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 346–352, 2019.

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

[25] 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, 2002.

[26] 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, 2019.

[27] 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 *IEEE Congress on Evolutionary Computation*, New Orleans, LA, 2011, pp. 2558-2562.

[28] O. Schütze, M. Vasile, and C. A. C. Coello, "Computing the set of epsilon-efficient solutions in multiobjective space mission design," *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. 3, pp. 53–70, 2011.

[29] C. T. Yue, B. Y. Qu, K. J. Yu, J. J. Liang, and X. D. Li. "A novel scalable test problem suite for multimodal multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 48, pp. 62–71, 2019.

[30] H. Ishibuchi, Y. Peng and K. Shang, "A Scalable Multimodal Multiobjective Test Problem," in *IEEE Congress on Evolutionary Computation*, Wellington, New Zealand, 2019, pp. 310-317.

[31] J. J. Liang, B. Y. Qu, D. W. Gong, C. T. Yue. Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Mar. 2019. DOI: 10.13140/RG.2.2.33423.64164.

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

[33] 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, 2020.

[34] R. Tanabe, and H. Ishibuchi, "A Framework to Handle Multi-modal Multi-objective Optimization in Decomposition-based Evolutionary Algorithms," *IEEE Trans. Evol. Comput.*, DOI: 10.1109/TEVC.2019.2949841.

[35] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2005, pp. 47–61.

[36] J. J. Liang, C. T. Yue, and B. Y. Qu, "Multimodal multi-objective optimization: A preliminary study," in *IEEE Congress on Evolutionary Comput*ation, Vancouver, BC, 2016, pp. 2454–2461.

[37] Y. Liu, H. Ishibuchi, Y. Nojima, N. Masuyama, and K. Shang, "A double-niched evolutionary algorithm and its behavior on Polygon-based problems," in PPSN, 2018, pp. 262–273.

[38] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2," in PPSN, 2004, pp. 742–751.

[39] R. K. Ursem, "Multinational evolutionary algorithms," in *IEEE Congress on Evolutionary Computation*, Washington, DC, USA, 1999, pp. 1633–1640.

[40] D. E. Goldberg, J. Richardson et al., "Genetic algorithms with sharing for multimodal function optimization," in *Proc. of the Second International Conference on Genetic Algorithms*, Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[41] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *IEEE Congress on Evolutionary Computation*, Portland, OR, USA, 2004, pp. 1382–1389.

[42] O. Kramer, and H. Danielsiek, "DBSCAN-based multiobjective niching to approximate equivalent Pareto-subsets," in *Proc. of 2010 Genetic and Evolutionary Computation Conference*, 2010, pp. 503-510.

[43] Z. Wang, Z. Zhan, Y. Lin, W. Yu, H. Wang, S. Kwong, and J. Zhang, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 114–128, 2020.

[44] S. C. Maree, T. Alderliesten, D. Thierens, and P. A. N. Bosman, "Real-valued evolutionary multi-modal optimization driven by hill-valley clustering," in *Proceedings of the Genetic and Evolutionary Computation Conference*, Kyoto, Japan, 2018, pp. 857-864.

[45] 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. Int. Conf. Knowl. Disc. Data Min.*, 1996, pp. 226–231.

[46] Y. Hua, Y. Jin, K. Hao, “A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts,” *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2758–2770, 2019.

[47] Q. Lin, S. Liu, K. C. Wong, M. Gong, C. Ceollo, J. Yomg, and J. Zhang, “A clustering-based evolutionary algorithm for many-objective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 391–405, 2019.

[48] V. L. Huang, P. N. Suganthan, K. Qin, and S. Baskar, “Differential evolution with external archive and harmonic distance-based diversity measure,” 2008, https://www.researchgate.net/publication/228967624.

[49] C. Hu and H. Ishibuchi, “Incorporation of a decision space diversity maintenance mechanism into MOEA/D for multi-modal multi-objective optimization,” in GECCO (Companion), 2018, pp. 1898–1901.

[50] R. Tanabe and H. Ishibuchi, “A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization,” in PPSN, 2018, pp. 249–261.

[51] C. Grimme, P. Kerschke, and H. Trautmann, “Multimodality in multi-objective optimization-more boon than bane?” *International Conference on Evolutionary Multi-Criterion Optimization*, 2019, pp. 126-138.

[52] C. Yue, B. Qu, and J. Liang, “A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 805-817, 2018.

[53] J. Liang and Q. Guo, et al, “A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems,” *Int. Conf. on Swarm Intelligence*, 2018, pp. 550-560.

[54] Y. Hu, J. Wang, J. Liang, K. Yu, H. Song, Q. Guo, C. Yue, and Y. Wang, “A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm,” *SCIENCE CHINA Information Sciences*, vol. 62, no. 7, pp. 1–17, 2019.

[55] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, Cluster analysis (5th edition). Chichester, UK: John Wiley&Sons, Ltd, 2011.

[56] P. A. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.

[57] Q. Zhang, A. Zhou, and Y. Jin, “RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, 2008.

[58] M., Friedman, “A comparison of alternative tests of significance for the problem of m rankings,” *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86-92, 1940.

[59] J. Alcalá-Fdez, L. Sánchez, S. García, et al, “KEEL: a software tool to assess evolutionary algorithms for data mining problems,” *Soft Comput.*, vol. 13, pp. 307–318, 2009.

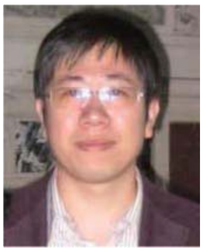
[60] M. Takahashi and H. Kita, “A crossover operator using independent component analysis for real-coded genetic algorithms,” in *IEEE Congress on Evolutionary Computation*, Seoul, South Korea, 2001, pp. 643-649.

Qiuzhen Lin received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

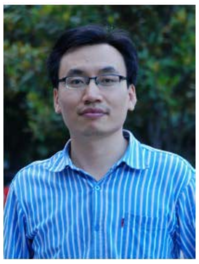
He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over twenty research papers since 2008. His current research interests include artificial immune system, multi-objective optimization, and dynamic system.

Wu Lin received the B.S. degree from Hubei University of Technology, Wuhan, China, in 2017, and received the M.S. degree from Shenzhen University, Shenzhen, China, in 2020.

He is currently a Research Assistant in College of Computer Science and Software Engineering, Shenzhen University. His current research interests are in evolutionary computation, multimodal multi-objective optimization, machine learning and transfer optimization.



Zexuan Zhu (M’12-SM’20) received the B.S. degree in computer science and technology from Fudan University, China, in 2003 and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008. He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. His research interests include computational intelligence, machine learning, and bioinformatics. Dr. Zhu is an Associate Editor of IEEE Transactions on Evolutionary Computation and IEEE Transactions on Emerging Topics in Computational Intelligence. He is also the Chair of the IEEE CIS Emergent Technologies Task Force on Memetic Computing.



Maoguo Gong (M’07–SM’14) received the B.S. and Ph.D. degrees in electronic science and technology from Xidian University, Xi’an, China, in 2003 and 2009, respectively. Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted as an Associate Professor and a Full Professor, respectively, both with exceptive admission. His current research interests include computational intelligence with applications to optimization, learning, data mining, and image understanding.

Dr. Gong was a recipient of the Prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is the Vice Chair of the IEEE Computational Intelligence Society Task Force on Memetic Computing, an Executive Committee Member of the Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation. He is also an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Jianqiang Li received his B.S. and Ph.D. Degree in automation major from South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is a professor at the College of Computer and Software Engineering of Shenzhen University. He led a project of the National Natural Science Foundation, and a project of the Natural Science Foundation of Guangdong Province, China. His major research interests include embedded systems and Internet of Things.

Carlos A. Coello Coello (M’98-SM’04-F’11) received Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996.

He is a Professor (CINVESTAV-3F Researcher) with the Department of Computer Science of CINVESTAV-IPN, Mexico City, México. He has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition, Springer, 2007). His publications currently report over 48,900 citations in Google Scholar (his h-index is 80). His research interests include evolutionary multiobjective optimization and constraint-handling techniques for evolutionary algorithms.

Dr. Coello Coello was a recipient of the 2007 National Research Award from the Mexican Academy of Sciences in the area of *Exact Sciences*, the 2013 IEEE Kiyo Tomiyasu Award and the 2012 National Medal of Science and Arts in the area of Physical, Mathematical and Natural Sciences. He is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and serves on the editorial board of 12 other international journals. He is a member of the Association for Computing Machinery, Sigma Xi, and the Mexican Academy of Science.