

## Research Article

# An Evolutionary Algorithm with Clustering-Based Assisted Selection Strategy for Multimodal Multiobjective Optimization

Naili Luo <sup>1,2</sup>, Wu Lin <sup>1</sup>, Peizhi Huang <sup>1</sup> and Jianyong Chen<sup>1</sup>

<sup>1</sup>College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

<sup>2</sup>College of Optoelectronic Engineering, Shenzhen University, Shenzhen, China

Correspondence should be addressed to Peizhi Huang; hpz@szu.edu.cn

Received 22 July 2020; Revised 12 December 2020; Accepted 30 December 2020; Published 13 January 2021

Academic Editor: Guang Li

Copyright © 2021 Naili Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In multimodal multiobjective optimization problems (MMOPs), multiple Pareto optimal sets, even some good local Pareto optimal sets, should be reserved, which can provide more choices for decision-makers. To solve MMOPs, this paper proposes an evolutionary algorithm with clustering-based assisted selection strategy for multimodal multiobjective optimization, in which the addition operator and deletion operator are proposed to comprehensively consider the diversity in both decision and objective spaces. Specifically, in decision space, the union population is partitioned into multiple clusters by using a density-based clustering method, aiming to assist the addition operator to strengthen the population diversity. Then, a number of weight vectors are adopted to divide population into  $N$  subregions in objective space ( $N$  is population size). Moreover, in the deletion operator, the solutions in the most crowded subregion are first collected into previous clusters, and then the worst solution in the most crowded cluster is deleted until there are  $N$  solutions left. Our algorithm is compared with other multimodal multiobjective evolutionary algorithms on the well-known benchmark MMOPs. Numerical experiments report the effectiveness and advantages of our proposed algorithm.

## 1. Introduction

Many real-world optimization problems often involve multiple (often conflicting) objectives [1–3], which are usually modeled as follows:

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

where  $x$  is an  $n$ -dimensional decision vector in  $\Omega \in \mathbb{R}^n$  and  $f_1(x), \dots, f_m(x)$  are, respectively, the objective values of  $x$  by using a specific function mapping. Pareto optimal solutions (PS) are a set of nondominated solutions for an MOP, and Pareto optimal front (PF) represents their objective values [4–6]. To get the optimal solutions of an MOP, many multiobjective evolutionary optimization algorithms (MOEAs) are proposed to evolve a population with multiple solutions, aiming to approximate the true PF [7–9]. Due to the advantage of being able to optimize multiple solutions simultaneously, MOEAs have been a widely used approach

to solve various kinds of optimization problems, that is, MOEAs based on Pareto dominance relation [10–13], MOEAs based on performance indicator [14–16], and MOEAs based on decomposition [17–25].

Recently, some MOPs with different PSs or some accepted local PSs [26–28] have drawn much attention, which are called multimodal multiobjective optimization problems (MMOPs). In recent years, several studies considering the diversity in decision space are proposed to solve MMOPs. In Omni-optimizer [29], the crowding distance is regarded as the diversity indicator of solution in decision space and embedded into NSGA-II, which strengthens diversity of population. In DN-NSGA-II [30], it aims to maintain distinct solutions in decision space, while their objective values are equal. Moreover, in MOEA/D-AD [31], by incorporating the addition and deletion operators into MOEA/D, it can obtain multiple nondominated solutions being different in decision space, which strengthens the diversity of population in decision space. Furthermore, there are other heuristic

algorithms having been getting more attention for MMOPs, that is, particle swarm optimization (PSO) algorithms [32–34]. For example, a PSO algorithm is proposed in [32] as an effective niching algorithm, which does not need any niching parameter. In MO\_Ring\_PSO\_SCD [33], a ring topology and special crowding distance are incorporated into PSO algorithm, helping to ensure greater diversity. Similarly, in [34], the neighborhood in decision space is constructed by a self-organizing map network, aiming to approximate multiple PSs for solving MMOPs. Recently, in TriMOEA-TA&R [35], two archives are used to cooperatively balance the convergence in objective space and the diversity in both decision and objective spaces. Due to the complexities and unpredictability in decision-making, more solutions with good performance, that is, multiple distinct global PSs and even some local PSs with good quality, are supposed to provide for decision-making [35].

However, for tackling MMOPs with at least one local PS, once global PSs are found in evolutionary process of the above multimodal MOEAs (MMOEAs) [29–35], all local PSs will not survive in next-generation population as local PSs are often dominated by global PS and they are usually replaced by some solutions with good convergence. To solve MMOPs, we develop a novel evolutionary algorithm with clustering-based assisted selection strategy for multimodal multiobjective optimization (MMOEA-CAS). In decision space, the union population is first partitioned into multiple clusters by a density-based clustering method, which will assist the addition operator to strengthen diversity of population. Then, a number of weight vectors divide population into  $N$  subregions in objective space ( $N$  is the population size), which is helpful to extend diversity of population. Moreover, in the deletion operator, the solutions in the most crowded subregion are collected into previous clusters, and then the worst solution in the most crowded cluster is deleted until there are  $N$  solutions left. Therefore, our approach is able to prevent the loss of local optimal solutions and help to realize a good balance in diversity of population. Here, the main contributions of our approach are stated as follows:

- (1) Different from traditional environmental selection mechanisms, in MMOEA-CAS, a density-based clustering method is applied on the population to obtain multiple clusters according to their neighboring relationship in decision space. Based on those classified results, those solutions with better performance in each cluster have the priority to be selected in the addition operator. This way, the diversity in decision space can be ensured.
- (2) The deletion operator proposed in our method considers the solutions' density information from both decision and objective spaces. Firstly, a set of weight vectors are used to divide candidate solutions into different regions based on their locations in objective space; then solutions in most crowded subregion are classified into previous clusters and the crowded solution in this cluster is deleted. This way, the more promising solutions are reserved into next generation.

Next, we give a brief introduction for organization of this paper. Some definitions about MMOPs and the introductions about the used clustering method are given in Section 2. Then, the framework of our algorithm with the addition and deletion operators is described in Section 3. Experimental setup and results are presented and analyzed in Sections 4 and 5, respectively. At last, some conclusions and further directions are discussed in Section 6.

## 2. Preliminaries

**2.1. MMOPs.** In this subsection, we give a case of MMOPs, which is shown in Figure 1. If any solution in a set  $PS_1$  cannot be dominated by any other solution in feasible regions,  $PS_1$  is regarded as global PS. The global PF is its objective vectors. Note that we marked global PS/PF with red color as shown in Figure 1. If any solution in a set PS cannot be dominated by any other solution in PS, PS is said to be local PS. The local PF is its objective vectors. As shown in Figure 1, we marked local PS/PF with blue color. Note that the local population is defined as the set including the neighboring solutions in the decision space. Thus, some MOPs with different PSs or local PSs [26–28] are called multimodal multiobjective optimization problems (MMOPs). For MMOPs, multiple different PSs, including global PSs and some good local PSs, should be provided for the decision-maker, which help to find more suitable solutions in making decision. Therefore, diversity of solutions in both objective and decision spaces should be considered equally important in the selection mechanisms of MOEAs for solving MMOPs.

**2.2. Density-Based Clustering Method.** It is well known that DBSCAN [36] with the neighborhood parameters  $\varepsilon$  and  $MinPts$  ( $MinPts$  is positive integer) is a density-based clustering method, aiming to find the cluster defined to the maximal density-connected set of sample data deduced by the relationship of density-reachable. Considering each sample data  $x = (x_1, \dots, x_n)$  in a data set  $D$ , related definitions are given as follows:

$\varepsilon$ -Neighborhood: the  $\varepsilon$ -neighborhood  $B(x)$  of  $x$  is defined as follows:

$$B(x) = \{y \in U \mid \text{dist}(x, y) \leq \varepsilon\}, \quad (2)$$

where  $\text{dist}(x, y)$  is Euclidean distance between  $x$  and  $y$ .

**Core objective:** a sample data  $x$  is said to be core objective if it satisfies equation (3) as follows:

$$B(x) \geq MinPts. \quad (3)$$

**Directly density-reachable:** if a sample data  $y$  is direct density-reachable from  $x$ , it must meet

$$y \in B(x), \quad (4)$$

$$B(y) \geq MinPts.$$

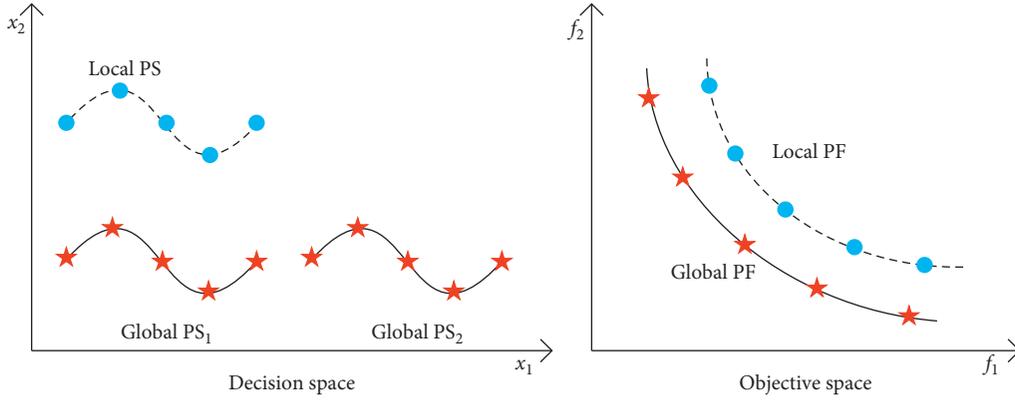


FIGURE 1: A case of MMOPs.

**Density-reachable:** the sample data  $y$  is said to be density-reachable from  $x$  if there are a set of sample data  $p_1, p_2, \dots, p_n$ , where  $p_1 = x, p_n = y$ , and  $p_{i+1}$  is direct density-reachable from  $p_i$ .

**Density-connected:** the sample data  $x$  and  $y$  are said to be density-connected, if there is a sample data  $p$  where  $x$  and  $y$  are density-reachable from it.

### 3. The Proposed Algorithm

**3.1. General Framework.** In this section, we introduce the general framework of MMOEA-CAS (Algorithm 1). At first,  $N$  solutions are randomly generated in  $\Omega \in \mathbb{R}^n$ , which form a population  $P = \{x^1, \dots, x^N\}$  in line 1. In line 2, multiple weight vectors  $w^1, \dots, w^N$  are initialized [37]. Then, the recombination operators are applied on parent population to generate offspring population  $P'$ . Next, a union population  $U$  is formed by merging  $P'$  and  $P$  in line 5. In line 6, according to Pareto dominated relationship [10], those solutions in  $U$  are ranked into different Pareto fronts  $F_1, \dots, F_t$ . As shown in lines 7-8, after normalizing  $U$ , DBSCAN is used to divide normalized population into multiple clusters  $C_1, \dots, C_K$  in decision space. Then, the addition operator is used to select the promising solutions into next-generation population  $P$  with more than  $N$  solutions in line 9. Next, in lines 10-11, after normalizing  $P$ , the association process with a set of weight vectors is used to divide  $P$  into  $N$  subregions in objective space. Finally, in line 12, the deletion operator is run on  $P$  to delete the most crowded solution each time until there are  $N$  solutions left in  $P$ . If the maximal number of evaluations is reached,  $P$  is regarded as the final solution set as shown in line 14. If the current evaluations are still smaller than the maximal evaluations, the above procedure in lines 4-12 will continue until the stopping criterion is met.

**3.2. Addition Operator.** In this section, the addition operator (Algorithm 2) is introduced to select some promising solutions to maintain diversity of population. As described in lines 1-2, the nondominated solutions are first selected and then add them into an empty set  $P$  in line 2. To further strengthen the diversity, in lines 3-6, the nondominated solutions in each cluster  $C_k (k = 1, \dots, K)$  are also added to  $P$ .

In addition, solutions with better convergence in objective space are further selected with higher priority until the size of solutions in  $P$  is not less than  $N$ . As shown in lines 8-11, solutions from  $F_2$  to  $F_t$  are used to fill  $P$  according to their respective priorities. When there are more than  $N$  solutions, the addition operator ends with returning the temporary population  $P$ .

**3.3. Association Procedure.** After the addition operator is run, the association procedure (Algorithm 3) will be applied on normalized  $P$  to divide population into  $N$  subregions in objective space by a set of weight vectors. At first,  $N$  empty subregions  $\Delta^1, \dots, \Delta^N$  are initialized in line 1. Next, for each solution  $x \in P$ , its distances to all weight vectors are computed, and then  $x$  is associated with its closest subregion as shown in lines 3-7. When all solutions are associated with their closest subregions, association procedure ends and  $\Delta^1, \dots, \Delta^N$  are returned in line 9.

**3.4. Deletion Operator.** After  $\Delta^1, \dots, \Delta^N$  are got by association procedure, deletion operator (Algorithm 4) will be used to prune  $P$  to get  $N$  final solutions. The HAD method [38] is used to reflect the crowding distances of each solution  $x$  in decision space. Thus, as shown in line 2-3, after normalizing  $P$ , the HAD values of solutions are computed by equation (5) as follows:

$$\text{HAD}(x) = \frac{|P| - 1}{\sum_{y \in P_i \text{ and } y \neq x} (1/\text{dist}(x, y))}, \quad (5)$$

where  $\text{dist}(\cdot)$  is the Euclidian distance between two normalized solutions. In line 4, find the most crowded subregion  $\Delta^i$  by equation (6) as follows:

$$i = \arg \max_{i \in \{1, \dots, N\}} \{|\Delta^i|\}. \quad (6)$$

In line 5, solutions in  $\Delta^i$  are classified into  $S_1, \dots, S_K$  by equation (7) as follows:

$$S_k = \{x | x \in C_k, k = 1, \dots, K\}. \quad (7)$$

Then, in line 6, find the most crowded cluster  $S_k$  by equation (8) as follows:

$$k = \arg \max_{k \in \{1, \dots, K\}} \{|S_k|\}. \quad (8)$$

In lines 7-8, the solution  $x$  with the smallest HAD value in  $S_k$  will be deleted from  $\Delta^i$  and  $P$ . Finally, deletion operator ends until there are  $N$  solutions left and  $P$  will be returned in line 10 as the final population.

## 4. Experimental Settings

**4.1. Test Instances and Parameter Settings.** In experiments, a series of MMOPs from CEC 2019 [28] are used to validate the performance of all compared algorithms, that is, some MMOPs with many distinct PSs, some MMOPs with irregular shape of PSs (i.e., MMF1\_e, MMF1\_z, and MMF5-7), and some MMOPs with existing global and local PSs (MMF10-13, MMF15, and MMF15\_a).

Our proposed algorithm MMOEA-CAS is, respectively, compared with four MMOEAs (i.e., Omni-optimizer [29], DN-NSGA-II [30], MO\_Ring\_PSO\_SCD [33], and TriMOEA-TA&R [35]) on all the MMOPs adopted. The parameter settings of compared algorithms are suggested in their references, which are given in Table 1.

Note that simulated binary crossover (SBX) and polynomial-based mutation [10] are used as recombination operators for producing offspring in MMOEA-CAS. For DBSCAN, the neighborhood parameter  $MinPts$  is set to be 3, and  $\epsilon$  is set by an adaptive strategy as follows:

$$r = \frac{\sum_{i=1}^n (x_i^{\max} - x_i^{\min})}{n} \times 0.1, \quad (9)$$

where  $x_i^{\min}$  and  $x_i^{\max}$  are the minimum and maximum values in  $i$ -th dimension.

In addition, the population size is equal to  $100 \times n$  for different test problems with  $n$ -dimensional decision variables and the maximum evaluation is equal to  $5000 \times n$ . In order to avoid exceptional situation, we collect 21 times of experimental results for each algorithm on solving each test problem.

**4.2. Performance Indicator.** In our experiments, we select the inverted generational distance (IGD [39, 40]) as performance indicator. To evaluate the quality of obtained solutions in decision space or objective space, IGD is, respectively, applied on two different spaces. When computing IGD, sampling reference points on the true PS or PF is necessary.

If  $P^*$  is composed of sampling reference points on PS and  $P$  is the final set of decision variables in decision space, the distance between  $P^*$  and  $P$  is named IGDX. If  $P^*$  is composed of sampling reference points on PF and  $P$  is the final set of objective values in objective space, the distance between them is named IGDF. In general, we compute the IGD value according to the following equation:

$$IGD(P, P^*) = \frac{\sum_{i=1}^{|P^*|} \min_{j=1}^{|P|} d(p^i, x^j)}{|P^*|}, \quad (10)$$

where  $d(p^i, x^j)$  is Euclidean distance between  $p^i$  and  $x^j$  ( $p^i$  belongs to  $P^*$  and  $x^j$  belongs to  $P$ ). The lower IGDX or IGDF values indicate the better performance of  $P$ .

## 5. Experimental Studies

**5.1. Results and Discussion.** As shown in Table 2, the compared results of other MMOEAs (i.e., Omni-optimizer, DN-NSGA-II, MO\_Ring\_PSO\_SCD, and TriMOEA-TA&R) and MMOEA-CAS in terms of IGDX and IGDF metrics are summarized. In order to get more intuitional information from the statistical data, we use “+/-/~” to record the compared results between other four competitors and MMOEA-CAS, where Wilcoxon’s rank-sum test with  $\alpha=0.05$  is used in comparison. According to the overall statistic results in the last row in Table 2, we can observe that MMOEA-CAS showed better performance than four other compared algorithms in 31, 32, 25, and 32 out of 44 cases, and it was similar to them in 4, 3, 5, and 2 cases. Conversely, the results obtained by MMOEA-CAS were significantly worse than those obtained by other MMOEAs in 9, 9, 14, and 10 out of 44 cases. The above comparison results demonstrate that our proposed MMOEA-CAS can get more uniformly and widely distributed solutions in decision space without deteriorating distributions in objective space, which confirms its superiority and effectiveness in solving most of MMOPs adopted.

The detailed IGDX and IGDF values of MMOEA-CAS and four MMOEAs are reported in Tables 3 and 4, respectively. To be much clearer, we marked the best value in bold for each test problem. In Table 3, MMOEA-CAS achieved the best IGDX values in 10 cases, that is, Omnitest, MMF2, MMF4, MMF7, MMF10-13, MMF15, and MMF15\_a; MO\_Ring\_PSO\_SCD got best IGDX values in 8 cases, that is, SYM-PART rotated, MMF5-6, and MMF8; TriMOEA-TA&R only obtained best IGDX values in 4 cases, that is, SYM-PART simple and MMF9. The remaining compared algorithms could not obtain best IGDX value in any case. Similarly, we could also observe from Table 4 that MMOEA-CAS obtained the best IGDF values in 10 cases, that is, MMF4, MMF7, MMF10-15, MMF14\_a, and MMF15\_a; Omnitest got the smallest IGDF values in 9 cases, that is, SYM-PART simple, MMF1\_z, MMF2, MMF5-6, and MMF8-9; MO\_Ring\_PSO\_SCD obtained the smallest IGDF values on MMF1 and MMF3, while TriMOEA-TA&R only obtained the smallest IGDF value on MMF1\_e; DN-NSGA-II could not obtain best IGDX value in any case. Thus, we could conclude that MMOEA-CAS achieved best IGDX values and best IGDF values on MMOPs with existing global and local PSs, that is, MMF10-13, MMF15, and MMF15\_a. By collecting the best values on all MMOPs in terms of IGDX and IGDF, we can conclude that the overall performance of our proposed MMOEA-CAS is better than other compared algorithms on these test problems. In addition, the above comparison results based on the best

**Input:**  $N$  (the number of solutions in population)  
**Output:**  $P$  (final population)

- (1) initialize a population  $P$
- (2) initialize weight vectors  $W = \{w^1, \dots, w^N\}$
- (3) **while** the stopping criterion is not met
- (4)    $P' = \text{Crossover} + \text{Mutation}(P)$
- (5)    $U = P \cup P'$
- (6)    $(F_1, \dots, F_t) = \text{Non-dominated Sorting}(U)$
- (7)   normalize  $U$
- (8)    $(C_1, \dots, C_K) = \text{DBSCAN}(U, \varepsilon, \text{MinPts})$
- (9)    $P = \text{Addition Operator} // \text{Algorithm 2}$
- (10)   normalize  $P$
- (11)    $(\Delta^1, \dots, \Delta^N) = \text{Association Process}(P, W) // \text{Algorithm 3}$
- (12)    $P = \text{Deletion Operator} // \text{Algorithm 4}$
- (13) **end while**
- (14) **return**  $P$

ALGORITHM 1: General framework.

**Output:**  $P$  (updated population)

- (1) set  $P$  to be an empty set
- (2)  $P = P \cup F_1$
- (3) **for**  $k = 1$  **to**  $K$
- (4)    $(f_1, \dots, f_t) = \text{Non-dominated Sorting}(C_k)$
- (5)    $P = P \cup f_1$
- (6) **end for**
- (7)    $i = 1$
- (8) **while**  $|P| < N$
- (9)    $i = i + 1$
- (10)    $P = P \cup F_i$
- (11) **end while**
- (12) **return**  $P$

ALGORITHM 2: Addition operator.

**Input:**  $P$  (population),  $W$  (weight vectors)  
**Output:**  $\Delta^1, \dots, \Delta^N$  (clusters)

- (1) initialize  $N$  empty subregions  $\Delta^1, \dots, \Delta^N$
- (2) **for each**  $x \in P$
- (3)   **for each**  $w \in W$
- (4)     compute  $d(x, w) = x - (w^T x) / \|w\|$
- (5)   **end for**
- (6)    $i = \arg \min d(x, w)$
- (7)    $\Delta^i = \Delta^i \cup_{w \in W} \{x\}$
- (8) **end for**
- (9) **return**  $\Delta^1, \dots, \Delta^N$

ALGORITHM 3: Association procedure.

performance on each test problem demonstrate that the environmental selection mechanism of MMOEA-CAS can reserve more promising solutions, while those classical environmental selection criteria have no ability to solve these MMOPs well.

Based on the experimental results in terms of IGDX, we can find that Omni-optimizer and DN-NSGAI could not

perform better than MMOEA-CAS on any case, showing the drawbacks of their selection mechanisms when solving those test problems. It can be concluded that the fast non-dominated sorting approach used in Omni-optimizer and DN-NSGA-II failed to maintain the diversity in decision space, since it always intended to choose the solutions with better convergence and diversity in objective space. Thus,

**Output:**  $P$  (updated population)

- (1) **while**  $|P| > N$
- (2)   normalize  $P$
- (3)   compute HAD values of solutions in  $P$  by equation (5)
- (4)   find the most crowded subregion  $\Delta^i$  by equation (6)
- (5)   divide all solutions in  $\Delta^i$  into  $S_1, \dots, S_K$  by equation (7)
- (6)   find the most crowded cluster  $S_k$  by equation (8)
- (7)   find  $x$  with the smallest HAD value in  $S_k$
- (8)   delete  $x$  from  $\Delta^i$  and  $P$
- (9) **end while**
- (10) **return**  $P$

ALGORITHM 4: Deletion operator.

TABLE 1: Parameters setting of all compared algorithms.

Omnioptimizer	$\eta_c = \eta_m = 20, p_c = 1, p_m = (1/n)$
DN-NSGA-II	CF = 0.5, $\eta_c = \eta_m = 20, p_c = 1, p_m = (1/n)$
MO-ring-PSO-SCD	$C_1 = C_2 = 2.05, W = 0.7298$
TriMOEA-TA&R	$p_{con} = 0.5, \delta_{niche} = 0.1, \epsilon_{peak} = 0.01, NR = 100, NCA = 20, NIA = 6$
MMOEA-CAS	$\eta_c = \eta_m = 20, p_c = 1, p_m = (1/n), MinPts = 3$

TABLE 2: The summarized results of other MMOEAs with MMOEA-CAS.

Algorithm	Omnioptimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R
+/-/~(IGDX)	0/21/1	0/21/1	9/11/2	5/17/0
+/-/~(IGDF)	9/10/3	9/11/2	5/14/3	5/15/2
+/-/~(All)	9/31/4	9/32/3	14/25/5	10/32/2

Wilcoxon's rank-sum test ( $\alpha = 0.05$ ) is used for experimental comparison where +, -, and = represent that the performance of the results obtained by competitors is better, worse, or similar.

TABLE 3: Results obtained by four other MMOEAs and MMOEA-CAS on IGDX.

Benchmark	Omnioptimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R	MMOEA-CAS
SYM-PART simple	5.558e+00 (2.61e+00)-	3.810e+00 (9.39e-01)-	1.663e-01 (2.19e-02)-	<b>2.181e-02 (3.01e-03)</b> +	7.633e-02 (4.92e-03)
SYM-PART rotated	4.405e+00 (1.22e+00)-	4.613e+00 (1.99e+00)-	<b>1.972e-01 (3.50e-02)</b> +	1.674e+00 (9.48e-01)-	4.240e-01 (6.09e-01)
Omnitest	1.501e+00 (2.29e-01)-	1.410e+00 (2.10e-01)-	4.057e-01 (8.74e-02)-	2.536e-01 (1.09e-01)-	<b>1.092e-01 (3.23e-03)</b>
MMF1	9.905e-02 (1.81e-02)-	9.963e-02 (1.70e-02)-	<b>4.951e-02 (2.06e-03)</b> +	7.290e-02 (8.54e-03)-	5.350e-02 (5.06e-03)
MMF1_z	7.949e-02 (1.57e-02)-	7.898e-02 (1.01e-02)-	<b>3.576e-02 (2.13e-03)</b> +	7.040e-02 (1.17e-02)-	5.593e-02 (6.27e-03)
MMF1_e	1.314e+00 (7.26e-01)~	1.055e+00 (4.70e-01)~	<b>5.709e-01 (2.57e-01)</b> +	1.940e+00 (7.07e-01)-	1.074e+00 (2.73e-01)
MMF2	1.266e-01 (1.01e-01)-	1.389e-01 (8.51e-02)-	3.802e-02 (1.72e-02)~	7.150e-02 (2.82e-02)-	<b>3.370e-02 (1.36e-02)</b>
MMF3	9.213e-02 (4.50e-02)-	9.220e-02 (3.53e-02)-	<b>3.164e-02 (1.19e-02)</b> +	9.315e-02 (3.98e-02)-	4.730e-02 (1.46e-02)
MMF4	8.358e-02 (2.70e-02)-	8.963e-02 (2.55e-02)-	2.705e-02 (1.71e-03)-	1.008e-01 (1.43e-01)-	<b>2.221e-02 (1.91e-03)</b>
MMF5	1.763e-01 (2.39e-02)-	1.703e-01 (2.06e-02)-	<b>8.615e-02 (5.66e-03)</b> ~	1.114e-01 (8.42e-03)-	8.754e-02 (8.77e-03)
MMF6	1.441e-01 (1.68e-02)-	1.471e-01 (1.66e-02)-	<b>7.260e-02 (3.87e-03)</b> +	9.006e-02 (9.24e-03) +	1.136e-01 (2.14e-02)
MMF7	4.629e-02 (9.25e-03)-	5.681e-02 (1.40e-02)-	2.610e-02 (1.54e-03)-	4.468e-02 (2.13e-02)-	<b>2.428e-02 (3.96e-03)</b>

TABLE 3: Continued.

Benchmark	Omnioptimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R	MMOEA-CAS
MMF8	3.286e-01 (1.65e-01)-	3.012e-01 (9.95e-02)-	<b>6.719e-02 (5.03e-03)+</b>	3.438e-01 (1.04e-01)-	1.448e-01 (3.29e-02)
MMF9	2.200e-02 (1.09e-02)-	2.529e-02 (1.32e-02)-	7.882e-03 (3.55e-04)+	<b>3.124e-03 (9.15e-05) +</b>	8.759e-03 (1.30e-03)
MMF10	1.646e-01 (3.31e-02)-	1.617e-01 (4.48e-02)-	1.004e-01 (3.69e-02)-	2.014e-01 (7.52e-05)-	<b>2.755e-02 (4.78e-02)</b>
MMF11	2.502e-01 (3.58e-04)-	2.505e-01 (4.85e-04)-	2.169e-01 (2.72e-02)-	2.524e-01 (5.79e-05)-	<b>9.196e-03 (1.25e-03)</b>
MMF12	2.396e-01 (1.68e-02)-	2.467e-01 (9.44e-04)-	1.875e-01 (4.53e-02)-	2.477e-01 (6.84e-04)-	<b>3.120e-03 (1.84e-04)</b>
MMF13	2.858e-01 (9.42e-03)-	2.895e-01 (1.54e-02)-	2.325e-01 (1.43e-02)-	2.714e-01 (6.55e-03)-	<b>1.055e-01 (1.46e-02)</b>
MMF14	8.873e-02 (6.29e-03)-	9.794e-02 (9.76e-03)-	5.329e-02 (1.86e-03)-	<b>3.651e-02 (4.92e-04) +</b>	4.651e-02 (1.11e-03)
MMF14_a	1.110e-01 (9.66e-03)-	1.196e-01 (5.83e-03)-	5.997e-02 (1.97e-03)+	<b>5.604e-02 (1.59e-03) +</b>	7.757e-02 (4.54e-03)
MMF15	2.471e-01 (2.10e-02)-	2.238e-01 (2.59e-02)-	1.562e-01 (1.63e-02)-	2.711e-01 (2.53e-04)-	<b>5.256e-02 (7.43e-03)</b>
MMF15_a	2.133e-01 (1.99e-02)-	2.086e-01 (1.46e-02)-	1.641e-01 (1.31e-02)-	2.194e-01 (2.83e-03)-	<b>9.684e-02 (6.43e-03)</b>
<i>best/all</i>	0/22	0/22	8/22	4/22	10/22
<i>+/-/~</i>	0/21/1	0/21/1	9/11/2	5/17/0	- -

Wilcoxon's rank-sum test ( $\alpha = 0.05$ ) is used for experimental comparison where +, -, and = represent that the performance of the results obtained by competitors is better, worse, or similar.

TABLE 4: Results obtained by four other MMOEAs and MMOEA-CAS on IGDF.

Benchmark	Omnioptimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R	MMOEA-CAS
SYM-PART simple	<b>1.216e-02 (1.72e-03)+</b>	1.230e-02 (1.78e-03)+	4.042e-02 (5.50e-03)-	3.414e-02 (4.99e-03)-	2.628e-02 (3.07e-03)
SYM-PART rotated	<b>1.252e-02 (1.70e-03)+</b>	1.539e-02 (2.12e-03)+	4.604e-02 (4.32e-03)-	2.843e-02 (3.88e-03)+	3.201e-02 (5.92e-03)
Omnitest	<b>6.776e-03 (2.47e-04)+</b>	7.911e-03 (5.10e-04)+	4.232e-02 (5.35e-03)-	1.841e-02 (4.18e-03)+	2.953e-02 (2.03e-03)
MMF1	3.788e-03 (5.27e-04)+	4.451e-03 (4.07e-04)+	<b>3.707e-03 (1.58e-04)+</b>	4.926e-03 (9.72e-04)~	4.762e-03 (5.42e-04)
MMF1_z	<b>3.231e-03 (4.30e-04)+</b>	3.649e-03 (4.91e-04)+	3.603e-03 (1.57e-04)+	4.418e-03 (9.38e-04)+	5.427e-03 (1.06e-03)
MMF1_e	1.701e-02 (9.98e-03)+	2.171e-02 (1.52e-02)+	1.229e-02 (1.50e-03)+	<b>6.981e-03 (1.42e-03)+</b>	1.191e-01 (5.32e-02)
MMF2	<b>1.801e-02 (9.68e-03)~</b>	3.691e-02 (3.43e-02)~	1.994e-02 (4.34e-03)~	2.521e-02 (8.18e-03)-	1.839e-02 (3.99e-03)
MMF3	2.203e-02 (2.40e-02)~	2.320e-02 (1.42e-02)~	<b>1.614e-02 (3.22e-03)~</b>	5.876e-02 (7.78e-02)-	1.990e-02 (9.70e-03)
MMF4	2.818e-03 (2.63e-04)-	3.209e-03 (1.81e-04)-	3.652e-03 (3.93e-04)-	3.445e-02 (6.70e-02)-	<b>2.694e-03 (4.49e-04)</b>
MMF5	<b>3.471e-03 (6.47e-04)+</b>	3.793e-03 (3.10e-04)+	3.748e-03 (1.94e-04)+	4.220e-03 (6.03e-04)~	4.284e-03 (5.54e-04)
MMF6	<b>3.154e-03 (3.20e-04)+</b>	3.747e-03 (2.97e-04)+	3.538e-03 (1.34e-04)+	4.602e-03 (2.06e-03)+	7.700e-03 (2.43e-03)
MMF7	3.162e-03 (2.96e-04)-	3.957e-03 (4.17e-04)-	3.830e-03 (3.43e-04)-	4.273e-03 (1.25e-03)-	<b>2.937e-03 (6.51e-04)</b>
MMF8	<b>3.145e-03 (1.88e-04)+</b>	3.983e-03 (5.03e-04)+	4.838e-03 (2.83e-04)~	5.667e-03 (7.00e-03)-	5.583e-03 (1.69e-03)
MMF9	<b>1.271e-02 (1.28e-03)~</b>	1.441e-02 (1.84e-03)-	1.562e-02 (1.42e-03)-	6.975e-02 (4.17e-03)-	1.332e-02 (1.42e-03)
MMF10	1.822e-01 (4.17e-02)-	1.938e-01 (4.11e-02)-	1.297e-01 (1.51e-02)-	2.285e-01 (5.05e-03)-	<b>3.839e-02 (3.63e-02)</b>

TABLE 4: Continued.

Benchmark	Omnioptimizer	DN-NSGA-II	MO-Ring-PSO-SCD	TriMOEA-TA&R	MMOEA-CAS
MMF11	9.565e-02 (9.39e-04)-	9.823e-02 (2.07e-03)-	8.508e-02 (6.16e-03)-	1.633e-01 (7.27e-03)-	<b>2.827e-02</b> <b>(3.10e-03)</b>
MMF12	8.611e-02 (8.13e-03)-	8.404e-02 (2.68e-03)-	6.431e-02 (1.27e-02)-	8.601e-02 (1.47e-03)-	<b>5.118e-03</b> <b>(3.54e-04)</b>
MMF13	1.472e-01 (2.08e-03)-	1.507e-01 (4.40e-03)-	9.329e-02 (1.73e-02)-	2.431e-01 (6.87e-03)-	<b>4.139e-02</b> <b>(9.33e-03)</b>
MMF14	9.765e-02 (3.95e-03)-	1.080e-01 (8.20e-03)-	8.042e-02 (2.94e-03)-	8.616e-02 (1.12e-03)-	<b>6.788e-02</b> <b>(1.14e-03)</b>
MMF14_a	1.049e-01 (4.57e-03)-	1.178e-01 (6.69e-03)-	7.818e-02 (2.50e-03)-	7.878e-02 (1.15e-03)-	<b>7.506e-02</b> <b>(1.69e-03)</b>
MMF15	2.010e-01 (7.37e-03)-	2.162e-01 (1.14e-02)-	1.741e-01 (3.27e-03)-	2.067e-01 (7.10e-04)-	<b>1.134e-01</b> <b>(1.21e-02)</b>
MMF15_a	2.074e-01 (8.92e-03)-	2.259e-01 (1.05e-02)-	1.744e-01 (3.72e-03)-	1.936e-01 (3.67e-03)-	<b>1.483e-01</b> <b>(3.59e-03)</b>
<i>best/all</i>	9/22	0/22	2/22	1/22	10/22
<i>+/-/~</i>	9/10/3	9/11/2	5/14/3	5/15/2	--

Wilcoxon's rank-sum test ( $\alpha = 0.05$ ) is used for experimental comparison where +, -, and = represent that the performance of the results obtained by competitors is better, worse, or similar.

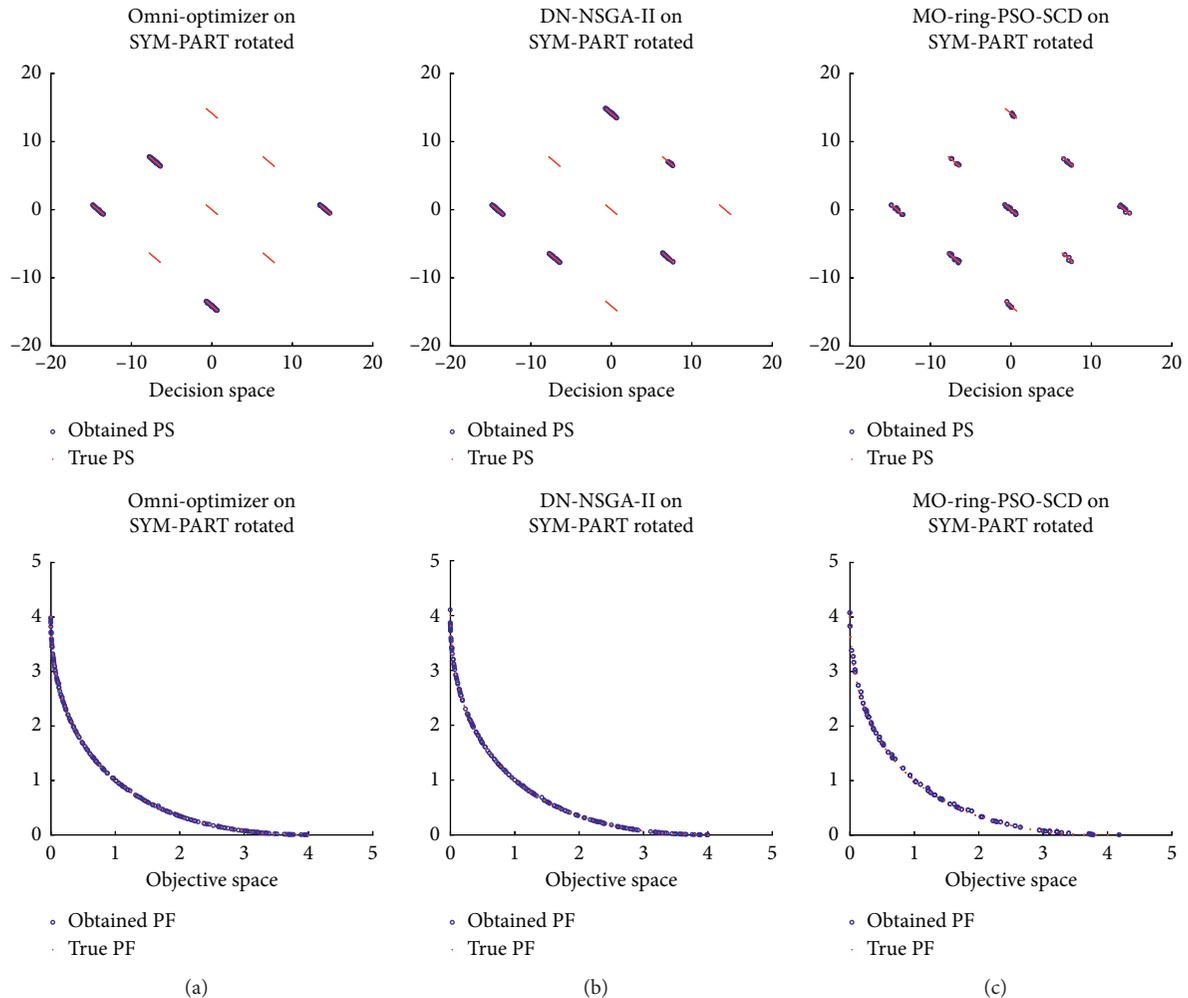


FIGURE 2: Continued.

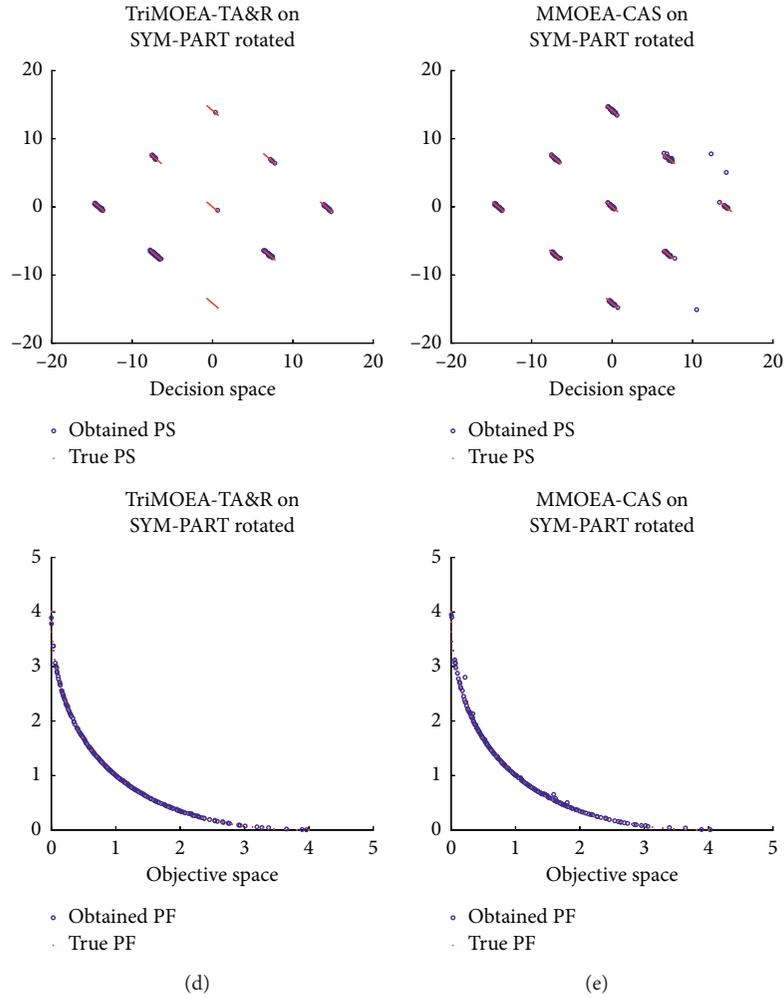


FIGURE 2: The final populations obtained in decision and objective spaces on SYM-PART rotated.

regarding the performance of final population obtained in decision space, the above two algorithms always obtain poor performance on all test problems because they neglect the importance of diversity in decision space. To address this issue, MMOEA-CAS used the density-based clustering method to divide clusters where the similar solutions are likely to be collected into the same cluster. At the same time, in the addition operator, the solutions with good performance in the same cluster even if it is worse in the whole population can be still seen as promising candidate solutions. This way, MMOEA-CAS can guarantee the diversity of population in decision space. This also explains why Omni-optimizer and DN-NSGA-II showed the advantages on MMOPs with multiple PSs (i.e., SYM-PART simple, SYM-PART rotated, Omni-test, MMF1, MMF5-6, and MMF8) when comparing the quality of population in objective space, while they showed poor performance on some MMOPs with local and global PSs (i.e., MMF10-13, MM15, and MMF15\_a) in terms of IGDX. Due to the comprehensive consideration of the diversity in both decision and objective spaces, MMOEA-CAS achieves better performance on most MMOPs at the expense of diversity in objective space when solving some problems with multiple PSs.

In order to improve the diversity in decision space, MO\_Ring\_PSO\_SCD used the special crowding distance, while it still followed the principle of convergence first in objective space. The local Pareto optimal solutions are always lost during the evolutionary process of MO\_Ring\_PSO\_SCD. Similarly, in TriMOEA-TA&R, nondominated solutions still replace dominated solutions in objective space, so it also failed to strike the balance between the diversity in decision and objective spaces. To avoid the local Pareto optimal solutions being replaced by the solutions with better convergence during the evolutionary process, the proposed addition and deletion operators in MMOEA-CAS collaboratively select more promising solution into next generation according to the information of clustering results classified by the density-based clustering method in decision space and the subregions divided by weight vectors in objective space. Therefore, only the solutions that are always performing worse in both two spaces are abandoned, and those solutions having good performance in either space could get to survive. Although MO\_Ring\_PSO\_SCD and TriMOEA-TA&R performed better than MMOEA-CAS on some problems with multiple PSs, they still obtained poor performance when compared with MMOEA-CAS on MMOPs having global and local PSs. As shown in

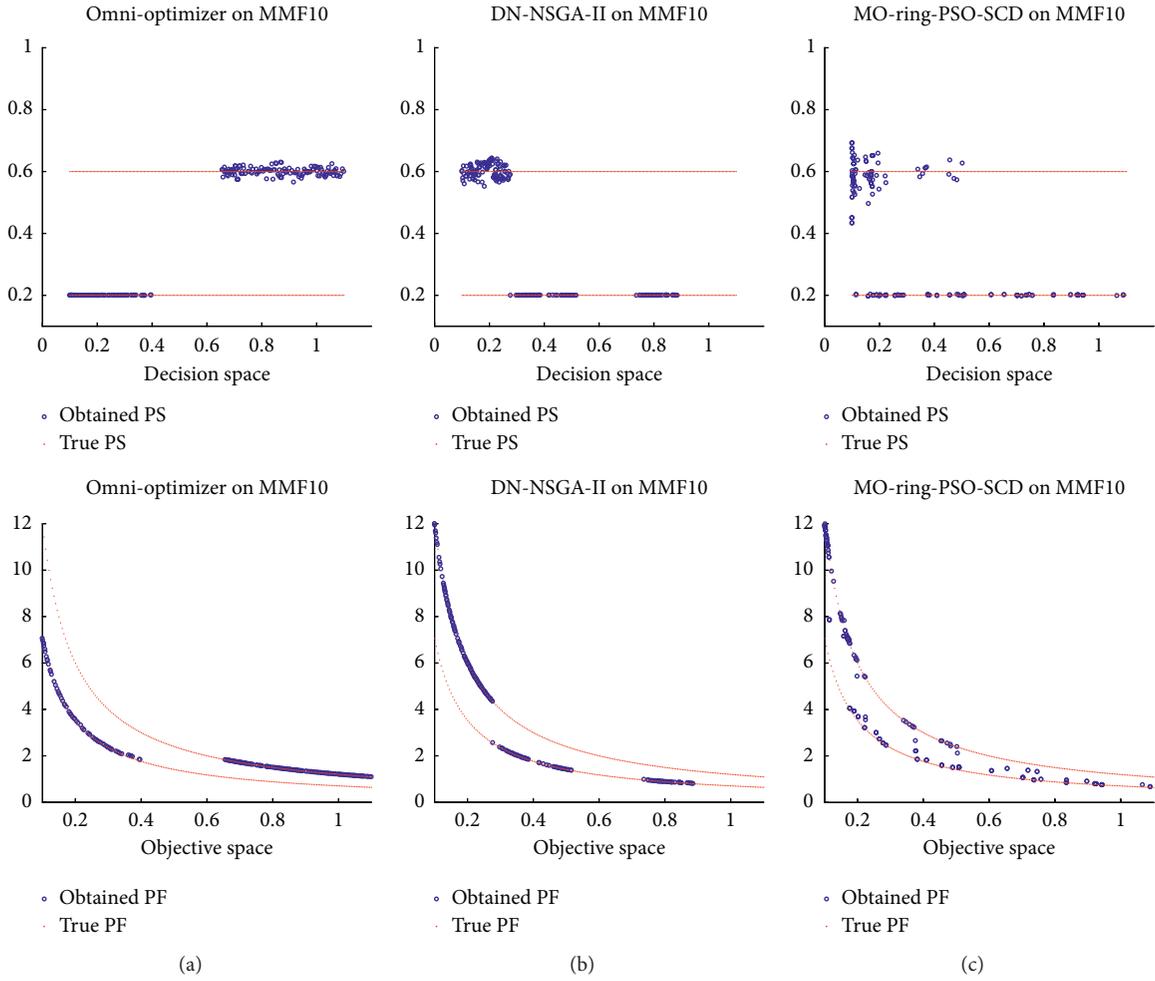


FIGURE 3: Continued.

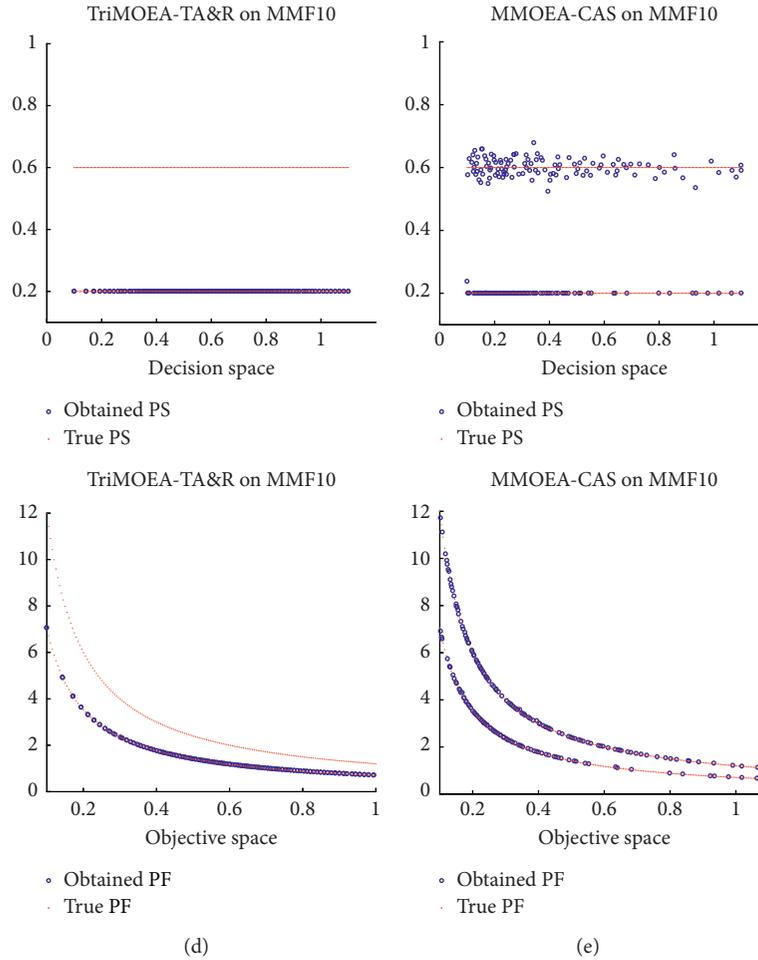


FIGURE 3: The final populations obtained in decision and objective spaces on MMF10.

Tables 3 and 4, when compared with MMOEA-CAS, it is obvious that MO\_Ring\_PSO\_SCD and TriMOEA-TA&R achieved worse results on most MMOPs, especially on these MMOPs having global and local PSs. Thus, these compared results validate the effectiveness of MMOEA-CAS and it is able to gain solutions with better overall performance in decision space and objective space when compared to other MMOEAs on most MMOPs.

**5.2. Further Discussion.** To clearly demonstrate the performance of different MMOEAs, the final populations obtained in decision and objective spaces by different algorithms are plotted in Figures 2–3, which qualitatively show the effectiveness and advantages of our proposed algorithm on solving these MMOPs. For example, a MMOP with multiple PSs, that is, SYM-PART rotated, is shown in Figure 2, and a MMOP with local PS, that is, MMF10, is shown in Figure 3, respectively. From Figure 2, it was found that the diversity of final population obtained by MMOEA-CAS is more uniform in decision space. Although Omnioptimizer and DN-NSGA-II found better solutions in objective space, they failed to maintain other PSs in decision space. Similarly, the population

obtained by TriMOEA-TA&R also missed some PSs in decision space. MO\_Ring\_PSO\_SCD found all PSs, but the final solutions cannot uniformly cover all the true PSs. Thus, we can conclude that MMOEA-CAS gets better results in terms of the comprehensive performance in both decision and objective spaces on SYM-PART rotated.

Furthermore, for MMF10, the final populations obtained are also shown in Figure 3. It is obvious that the performance of final population obtained by MMOEA-CAS is significantly better than other MMOEAs. Other compared algorithms mainly pick solutions with faster convergence rates in objective space, which will cause some local optimal solutions always to be eliminated by the global optimal solutions. Thus, only MMOEA-CAS maintained some promising local optimal solutions. To summarize, MMOEA-CAS shows some superiorities on two cases, especially on MM10.

## 6. Conclusion

We have developed a novel evolutionary algorithm with clustering-based assisted selection strategy for multimodal multiobjective optimization, in which addition operator and

deletion operator are proposed to comprehensively consider the diversity in both decision and objective spaces. Specifically, a density-based clustering method is used to divide population into multiple clusters in decision space, aiming to assist addition operator to strengthen diversity of population. Then, weight vectors are used to divide population into  $N$  subregions in objective space ( $N$  is population size). Moreover, in deletion operator, the solutions in the most crowded subregion are collected into previous clusters, and then the worst solution in the most crowded cluster is deleted until there are  $N$  solutions left. The numerical values on both IGD<sub>X</sub> and IGD<sub>F</sub> have reported the effectiveness of MMOEA-CAS. Most existing MMOEAs cannot search for local Pareto optimal solutions because they are often dominated and replaced by global optimal solutions with good convergence, while our proposed algorithm is capable of preserving more uniformly and widely distributed solutions in decision space without deteriorating their distributions in objective space, which confirms its superiority on most of MMOPs.

Finally, some more effective frameworks, enhanced search operators, and other techniques will be further studied and developed to fill in the gaps in the field of multimodal multiobjective optimization. In addition, we will further extend our proposed algorithm to solve other more complicated multimodal multiobjective optimization problems.

## Data Availability

All experimental results can be obtained upon request through email to the corresponding author.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61876110, 61836005, and 61672358, the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, and Shenzhen Technology Plan under Grant JCYJ20190808164211203. Also, this work was supported by the National Engineering Laboratory for Big Data System Computing Technology and the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University.

## References

- [1] Y. Zhang, D. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE-ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2015.
- [2] Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Computing Surveys*, vol. 47, no. 4, pp. 1–33, 2015.
- [3] Q. Zhang, J. Ding, W. Shen, J. Ma, and G. Li, "Multiobjective particle swarm optimization for microgrids pareto optimization dispatch," *Complexity*, vol. 2020, Article ID 5695917, 13 pages, 2020.
- [4] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK, 2001.
- [5] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: a review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41256–41279, 2018.
- [6] C. Coello, G. Lamont, and D. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, Berlin, Germany, 2007.
- [7] Z. Zhang, Q. Han, Y. Li, Y. Wang, and Y. Shi, "An evolutionary multiagent framework for multiobjective optimization," *Complexity*, vol. 2020, Article ID 9147649, 18 pages, 2020.
- [8] X. Wu, S. Zhang, Z. Gong, J. Ji, Q. Lin, and J. Chen, "Decomposition-based multiobjective evolutionary optimization with adaptive multiple Gaussian process models," *Complexity*, vol. 2020, Article ID 9643273, 22 pages, 2020.
- [9] W. Lin, Q. Lin, Z. Zhu, J. Li, J. Chen, and Z. Ming, "Evolutionary search with multiple utopian reference points in decomposition-based multiobjective optimization," *Complexity*, vol. 2019, Article ID 7436712, 22 pages, 2019.
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multi-objective optimization," in *Proceeding of Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems*, pp. 95–100, Athens, Greece, 2002.
- [12] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 4, pp. 577–601, 2014.
- [13] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular pareto fronts," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2758–2770, 2019.
- [14] E. Zitzler and S. Künzli, "Indicator-based selection in multi-objective search," in *Proceeding of the International Conference on Parallel Problem Solving from Nature-PPSN VIII*, pp. 832–842, Springer, Berlin, Germany, 2004.
- [15] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [16] K. Li, J. Zheng, M. Li, C. Zhou, and H. Lv, "A novel algorithm for non-dominated hypervolume-based multi-objective optimization," in *Proceeding of the IEEE International Conference Systems, Man, and Cybernetics*, pp. 5220–5226, San Antonio, TX, USA, October 2009.
- [17] Q. Zhang and H. Li, "MOEA/D: a multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [18] H. Zhang, X. Zhang, X.-Z. Gao, and S. Song, "Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble," *Neurocomputing*, vol. 173, pp. 1868–1884, 2016.
- [19] H. Li and Q. Zhang, "Multi-objective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE*

- Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [20] S. X. Zhang, L. M. Zheng, L. Liu, S. Y. Zheng, and Y. M. Pan, “Decomposition-based multi-objective evolutionary algorithm with mating neighborhood sizes and reproduction operators adaptation,” *Soft Computing*, vol. 21, no. 21, pp. 6381–6392, 2017.
- [21] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, “Stable matching-based selection in evolutionary multi-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.
- [22] C. Dai and X. Lei, “An improvement decomposition-based multi-objective evolutionary algorithm with uniform design,” *Knowledge-Based Systems*, vol. 125, no. 1, pp. 108–115, 2017.
- [23] H. Xu, W. Zeng, D. Zhang, and X. Zeng, “MOEA/HD: a multiobjective evolutionary algorithm based on hierarchical decomposition,” *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 517–526, 2019.
- [24] W. Li, L. Wang, Q. Jiang, X. Hei, and B. Wang, “Multi-objective cloud particle optimization algorithm based on decomposition,” *Algorithms*, vol. 8, no. 2, pp. 157–176, 2015.
- [25] Y. Zhang, D.-W. Gong, J.-Y. Sun, and B.-Y. Qu, “A decomposition-based archiving approach for multi-objective evolutionary optimization,” *Information Sciences*, vol. 430–431, pp. 397–413, 2018.
- [26] C. Yue, B. Qu, K. Yu, J. Liang, and X. Li, “A novel scalable test problem suite for multimodal multiobjective optimization,” *Swarm and Evolutionary Computation*, vol. 48, pp. 62–71, 2019.
- [27] H. Ishibuchi, Y. Peng, and K. Shang, “A scalable multimodal multi-objective test problem,” in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 310–317, Wellington, New Zealand, June 2019.
- [28] J. J. Liang, B. Y. Qu, D. W. Gong, and C. T. Yue, “Problem definitions and evaluation criteria for the cec 2019 special session on multimodal multiobjective optimization,” Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, 2019.
- [29] K. Deb and S. Tiwari, “Omni-optimizer: a procedure for single and multi-objective optimization,” in *Lecture Notes in Computer Science, Evolutionary Multi-Criterion Optimization*, pp. 47–61, Springer, Berlin, Germany, 2005.
- [30] J. J. Liang, C. T. Yue, and B. Y. Qu, “Multimodal multi-objective optimization: a preliminary study,” in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2454–2461, Vancouver, Canada, July 2016.
- [31] R. Tanabe and H. Ishibuchi, “A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization,” in *Parallel Problem Solving from Nature—PPSN XV*, pp. 249–261, Springer, Berlin, Germany, 2018.
- [32] X. Li, “Niching without niching parameters: particle swarm optimization using a ring topology,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.
- [33] C. Yue, B. Qu, and J. Liang, “A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 805–817, 2018.
- [34] Y. Hu, J. Wang, J. Liang et al., “A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm,” *Science China Information Sciences*, vol. 62, no. 7, pp. 1–17, 2019.
- [35] Y. Liu, G. G. Yen, and D. Gong, “A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 660–674, 2019.
- [36] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, Menlo Park, CA, USA, August 1996.
- [37] I. Das and J. E. Dennis, “Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [38] V. L. Huang, P. N. Suganthan, K. Qin, and S. Baskar, “Multi-objective differential evolution with external archive and harmonic distance-based diversity measure,” *Technical Report, Nanyang Technological University Singapore*, <https://www.researchgate.net/publication/228967624>, 2005.
- [39] P. A. N. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
- [40] M. Li and J. Zheng, “Spread assessment for evolutionary multi-objective optimization,” in *Proceeding of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 216–230, Nantes, France, April 2009.