A Clustering-based Evolutionary Algorithm for Many-objective Optimization Problems

Qiuzhen Lin, Songbai Liu, Ka-Chun Wong, Maoguo Gong, *Senior Member, IEEE*, Carlos A. Coello Coello, *Fellow, IEEE*, Jianyong Chen, *Member, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—This paper suggests a novel clustering-based evolutionary algorithm for many-objective optimization problems. Its main idea is to classify the population into a number of clusters, which is expected to solve the difficulty of balancing convergence and diversity in high dimensional objective space. The individuals showing high similarities on the vector angles are gathered into the same cluster, such that the population's distribution can be well portrayed by the clusters. To efficiently find these clusters, partitional clustering is first used to classify the union population into *m* main clusters based on the *m* axis vectors (*m* is the number of objectives), and then hierarchical clustering is further run on these *m* main clusters to get *N* final clusters (*N* is the population size and *N* > *m*). At last, in environmental selection, one individual from each of *N* clusters closest to the axis vectors is selected to maintain diversity, while one individual from each of the other clusters is preferred by a simple convergence indicator to ensure convergence. When tackling some well-known test problems with 5 to 15 objectives, extensive experiments validate the superiority of our algorithm over six competitive many-objective evolutionary algorithms, especially on problems with incomplete and irregular Pareto-optimal fronts.**

***Index Terms*—Many-objective optimization, Evolutionary algorithm, Partitional clustering, Hierarchical clustering.**

I. Introduction

MANY-objective optimization problems (MaOPs) contain more than three objectives to be optimized simultaneously, which are extended from multi-objective optimization problems (MOPs), as defined by

$$\begin{array}{ll} \text{minimize} & F(x) = (f_1(x), ..., f_m(x)) \\ \text{subject to} & x \in \Omega \end{array} \quad (1)$$

where $x = (x_1, \ldots, x_n)$ is a decision vector from the search space Ω (n is the number of decision variables), and $F(x)$ defines m objective functions. Due to the conflicts often arising in different objectives, there is no a single optimal solution, but a set of trade-off solutions termed Pareto-optimal set (PS), whose mapping onto objective space is termed Pareto-optimal front (PF) [1]. During the last decades, evolutionary algorithms (EAs) have become a popular and effective approach to tackle MOPs and MaOPs [2]-[3].

Three main kinds of multi-objective EAs (MOEAs), i.e., Pareto-based MOEAs [4]-[5], decomposition-based MOEAs [6], and indicator-based MOEAs [7], have shown their effectiveness and efficiency in tackling MOPs with two or three objectives. Pareto-based MOEAs use a Pareto-based ranking scheme to sort the population into different convergence layers, and then adopt a diversity maintenance strategy to enhance the population's diversity. Decomposition-based MOEAs transform an MOP into a number of sub-problems and then solve them simultaneously using a collaborative search process. Indicator-based MOEAs apply a single performance indicator, e.g., hypervolume (HV) [8], to effectively guide the evolution of the population. However, when dealing with MaOPs, their performance deteriorates significantly, mainly due to the curse of dimensionality [9]-[10]. Pareto-based MOEAs fail to provide sufficient selection pressure toward the PFs, as the effect of Pareto-based ranking becomes insignificant when handling a large proportion of non-dominated solutions [11]-[13]. For decomposition-based MOEAs, it is hard to specify a set of weight vectors in high dimensional objective space, and their performance strongly depends on the consistency to the shapes of weight vectors and PFs [14]-[16]. Indicator-based MOEAs often suffer from a high computational cost, which is more serious in tackling MaOPs [17]-[18]. To deal with the above problems, a number of many-objective evolutionary algorithms (MaOEAs) have been recently developed.

On Pareto-based MOEAs, there are two main approaches to enhance them for solving MaOPs. The first one is to modify or renew the definition of the Pareto-based dominance relation, as is the case of fuzzy-dominance [19], corner sort [20], a new ensemble fitness ranking (EFR-RR) [21], θ-dominance [22], reference point-based dominance [23], or a generalized form of Pareto-optimality [24]. The other one tries to alleviate the loss of selection pressure by enhancing diversity management. Examples of this approach are the use of shift-based density estimation in SPEA2-SDE [25], the use of associated reference points in NSGA-III [26], or the use of a reference direction-based density estimator in SPEA/R [27].

For decomposition-based MOEAs, two approaches have been proposed for generating uniformly distributed weight vectors in MOEA/DD [28] and I-DBEA [29]. Moreover, Pareto-based dominance was combined in MOEA/DD, while in I-DBEA, two independent distance measures were used to

Manuscript received XX. XX. 2017; revised XX. XX. 2017; accepted XX. XX. 2018. This work was supported by the National Natural Science Foundation of China under Grants 61876110 and 61772393, the Research Grants Council of the Hong Kong Special Administrative Region [CityU 21200816], [CityU 11203217], and [CityU 11200218], and CONACyT grant no.221551.

Q.Z. Lin, S.B. Liu, and J.Y. Chen are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail of Jianyong Chen: jychen@szu.edu.cn).

S.B. Liu, and K.-C. Wong are with the Department of Computer Science, City University of Hong Kong, Hong Kong.

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

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

J. Zhang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

balance convergence and diversity, and a simple pre-emptive distance comparison scheme was employed for association. To balance convergence and diversity in the solution update mechanism used for MaOPs, a chain-reaction solution update strategy was designed in MOEA/D-CRU [30], a distance based updating strategy was used in MOEA/D-DU [21], two distinctive components (i.e., decomposition-based sorting and angle-based selection) were reported in MOEA/D-SAS [31], and a weighted sum method was employed on a local manner in MOEA/D-LWS [32]. Although all the above algorithms showed competitive performance for MaOPs, they still faced the same problem to specify a set of weight vectors, which strongly impacts their performance due to the consistency between the shapes of weight vectors and PFs [15]. Therefore, a reference vector regeneration strategy was used in RVEA [33] and its improved version K-RVEA [34], and an adaptive method of adjusting the weight vectors was designed based on a self-organizing map in MOEA/D-SOM [35].

Regarding indicator-based MOEAs, some approaches have been presented to compute the HV in more efficient ways [18], [36]-[38]. However, they are still not so efficient to tackle MaOPs with a large number of objectives. Other performance indicators (e.g., R2 [39]-[40] and the additive approximation [41]) and the combination of two performance indicators (e.g., Two_Arch2 [42] and SRA [43]) have also been reported for solving MaOPs.

Moreover, there are several new MaOEAs proposed to use the vector angles between individuals for population update. In VaEA [44], the search directions extracted from the population were employed. Then, a *maximum-vector-angle-first* principle was used to maintain diversity, while a *worse-elimination* principle was adopted to balance convergence and diversity. In MaOEA-DDFC [45], the ideas of directional diversity (DD) and favorable convergence (FC) were presented. A mating selection method based on FC was used to enhance the selection pressure, while the environmental selection based on DD and FC was applied to balance convergence and diversity. In MaOEA-CSS [46], a coordinated selection strategy was reported, with a new convergence measure based on distance and a new diversity measure based on vector angle to enhance performance on MaOPs. In 1by1EA [47], an efficient convergence indicator was used to select individuals one by one, with the aim to strengthen the selection pressure. When one individual was chosen, its neighbors were de-emphasized to ensure diversity for tackling MaOPs, which was realized by an angle-based similarity niche method.

To summarize, most MaOEAs were enhanced from the three main kinds of MOEAs and several recent MaOEAs were proposed with the use of vector angles to maintain diversity, as the vector angles between individuals are more effective than the Euclidean distance to measure the individuals' distribution in high dimensional objective space [44]-[47]. Although these MaOEAs showed very competitive performance, they still met some difficulties when solving the more difficult MaOPs with incomplete and irregular PFs [13], such as the MaF test problems recently proposed in [48]. Moreover, there has been some previous interesting works on embedding clustering into MOEAs [49]-[55]. For example, clustering was embedded in SPEA [49] to tailor the non-dominated archive for tackling MOPs, while clustering was used in [50] with a modeling procedure to promote the population's diversity. In [51], a *k*-means clustering method was employed to find the population structure by partitioning the solutions into several clusters, in which the solutions under the same cluster were allowed to reproduce. Two clustering methods were applied in [52] to give interdependence variable analysis and control variable analysis when tackling MOPs. In [53] and [54], the crowding-distance based truncation procedure in NSGA-II was replaced by a hierarchical clustering method to cope with MOPs and MaOPs, respectively. Another related study was given in [55] to scrutinize the impact of clustering with different similarity metrics in different spaces (i.e., variable space, objective space, and a combination of both) when clustering was embedded into a multi-objective particle swarm optimization algorithm. Unlike these previous approaches, this paper suggests a novel MaOEA based on clustering, called MaOEA/C, which is more flexible to balance convergence and diversity in high dimensional objective space. The main novelties of MaOEA/C distinguished from the previous works [49]-[55] are the used acute angle as the similarity metric in clustering and the proposed two-step clustering strategy to speed up its execution. In environmental selection, all the offspring and parents are combined and then they are finally classified into N clusters (N is the population size) using the vector angles of individuals to reflect their similarities, which helps to portray the population's distribution. In more detail, the partitional clustering method (PCM) is first run to separate the union population into m main clusters (m is the number of objectives), using m axis vectors as m centroids. Then, the hierarchical clustering method (HCM) is used to classify each of these m clusters, aiming to efficiently get N final clusters. At last, the clusters closest to m axis vectors are found and one individual from each of them is selected to maintain diversity. For each of remaining clusters, one individual is chosen by a simple convergence indicator to ensure convergence. Moreover, when running recombination, the parent population is also classified to m clusters using PCM and then the mating parents are selected from the same cluster with a high probability to encourage exploitation. The performance of MaOEA/C was studied when tackling the well-known WFG [56] and MaF [48] test problems with 5 to 15 objectives. When compared to six competitive MaOEAs (NSGA-III [26], MOEA/D-DU [21], EFR-RR [21], θ-DEA [22], SRA [43], and VaEA [44]), MaOEA/C showed some advantages on most cases, especially on problems with incomplete and irregular PFs.

To conclude, our main contributions are clarified as follows.

1) This paper suggests a novel clustering-based evolutionary algorithm for tackling MaOPs. With the clustering methods, the union population is finally classified into N clusters and then one individual is only selected from each cluster to maintain convergence or diversity in environmental selection.

Algorithm 1 *Partitional_Clustering* ($\boldsymbol{S}$, m)

1: initialize $C_1^{PCM}, C_2^{PCM} ... C_m^{PCM}$ as the empty set
2: set m axis vectors as centroids c_i^{PCM} for C_i^{PCM}
3: set the cluster sizes (cs) as $|\boldsymbol{S}|/m$
4: **for** each c_i^{PCM} selected randomly
5: calculate the angles of $s_j \in \boldsymbol{S}$ to c_i^{PCM} by (2)
6: select cs individuals closest to c_i^{PCM} from $\boldsymbol{S}$ into C_i^{PCM}
7: remove the selected cs individuals from $\boldsymbol{S}$
8: **end for**
9: **return** $C_1^{PCM}, C_2^{PCM} ... C_m^{PCM}$

Therefore, the trade-off of convergence and diversity in solving MaOPs is well balanced by using the clustering methods adopted in this paper.

2) Two clustering approaches, i.e., PCM and HCM, are used in this paper. The adopted two-step clustering strategy aims to efficiently classify the union of parent and offspring populations into N clusters, requiring a computational cost similar to that of most state-of-the-art MaOEAs [44].

The rest of this paper is organized as follows. Section II introduces the related background on clustering methods. The details of MaOEA/C are given in Section III, and the experimental results are provided in Section IV. Finally, our conclusions and future work are presented in Section V.

II. Related Background on Clustering Methods

Due to the high computational cost of HCM [57] when it is run on the union population, a PCM modified from K-means [58] is first used to divide the union population into m main clusters (m is the number of objectives). Then, HCM is run to classify each of m main clusters into k small clusters, where $k = N/m$ and N is manually set as a multiple of m in this paper. Thus, it finally gets N clusters. At last, one individual showing good convergence or diversity is selected from each of these N clusters to form the new population. Here, the used PCM and HCM in this paper are respectively introduced below.

A. Partitional Clustering Method

Given a dataset with N individuals, PCM builds m partitions for the dataset and each partition represents a cluster ($N \geq m$) [59]. In this paper, a classical PCM (K-means [58]) is modified to cluster the individuals of MaOPs and the vector angles of individuals are used to reflect their similarities in the objective space. Fixing the m axis vectors as centroids, our PCM aims to equally divide the input population into m clusters, which can be used in both the mating and the environmental selection. To clarify the process of PCM, its pseudo-code is given in **Algorithm 1** with the inputs: $\boldsymbol{S}$ as a population and m. In line 1, m clusters obtained by PCM as denoted by $C_1^{PCM}, C_2^{PCM}, ..., C_m^{PCM}$ are initialized as the empty set. Then, the m axis vectors, i.e., (1,0,…,0), (0,1,…,0),…, (0,0,…,1), are set as centroids c_i^{PCM} $(i = 1, 2, ..., m)$ in line 2, respectively for C_i^{PCM}. As each cluster is expected to include the same number of individuals, their cluster sizes (cs) are set to $|\boldsymbol{S}|/m$ in line 3. For each centroid c_i^{PCM}, the vector angles between it and all the individuals $s_j \in \boldsymbol{S}$ $(j = 1, 2, ..., |\boldsymbol{S}|)$ are computed in line 5 by (2).

$$angle(c_i^{PCM}, s_j) \triangleq \arccos\left|\frac{f_i^{'}(s_j)}{\sqrt{\sum_{k=1}^{m} f_k^{'}(s_j)^2}}\right| \tag{2}$$

Algorithm 2 *Hierarchical_Clustering* ($\boldsymbol{S}$, k)

1: set $s_i \in \boldsymbol{S}$ as a cluster C_i^{HCM} and centroid c_i^{HCM}
2: **for** i := 1 to $|\boldsymbol{S}|$
3: $index\left(C_i^{HCM}\right) = -1, \theta\left(C_i^{HCM}\right) = \infty, flag\left(C_i^{HCM}\right) =$ **false**
4: find the nearest cluster to C_i^{HCM} by Eq. (4)
5: record the corresponding $index\left(C_i^{HCM}\right)$ and $\theta\left(C_i^{HCM}\right)$
6: **end for**
7: $T_1 = T_2 = -1$, $minAngle = \infty$, $size = |\boldsymbol{S}|$
8: **for** i := 1 to $|\boldsymbol{S}|$
9: **if** $minAngle > \theta\left(C_i^{HCM}\right) \cap flag\left(C_i^{HCM}\right) ==$**false**
10: $minAngle = \theta\left(C_i^{HCM}\right)$, $T_1 = i$, $T_2 = index\left(C_i^{HCM}\right)$
11: **end if**
12: **end for**
13: **while** $size > k$
14: $C_{new}^{HCM} = C_{T_1}^{HCM} \cup C_{T_2}^{HCM}$ and update c_{new}^{HCM} by (5)
15: $flag\left(C_{T_1}^{HCM}\right) =$ **true**, $C_{T_2}^{HCM} = C_{new}^{HCM}$ and $size$--
16: **for** i := 1 to $|\boldsymbol{S}|$
17: **if** ($index\left(C_i^{HCM}\right) == T_1$ or T_2) $\cap$ $flag\left(C_i^{HCM}\right) ==$**false**
18: update $\theta\left(C_i^{HCM}\right)$ and $index\left(C_i^{HCM}\right)$ like lines 4-5
19: **end if**
20: **end for**
21: update T_1, T_2 like lines 7-12
22: **end while**
23: delete C_i^{HCM} whose $flag\left(C_i^{HCM}\right)$ is **true**
24: **return** $C_1^{HCM}, C_2^{HCM}, ..., C_k^{HCM}$

where $f_i^{'}(s_j)$ indicates the normalized value of the ith objective for s_j using (3).

$$f_i^{'}(s_j) = \frac{f_i(s_j) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \tag{3}$$

where $i = 1, 2, ..., m$, $f_i^{\max}$ and $f_i^{\min}$ are respectively the maximum and minimum values of the ith objective obtained by all the individuals in $\boldsymbol{S}$, as it is challenging to find the exact nadir point [43, 44]. Especially, when the number of objectives is less than four, $f_i^{\max}$ and $f_i^{\min}$ should be found only from non-dominated individuals. Then, for each centroid c_i^{PCM} selected randomly in line 4, a number of cs individuals with the closest angles to c_i^{PCM} by (2) will be selected into the cluster C_i^{PCM} in line 6, which are then removed from $\boldsymbol{S}$ in line 7. Other clusters can be similarly obtained by running the procedures in lines 5-7. Please note that this PCM only iterates once to get m main clusters, so as to reduce the computational cost. Finally, in line 9, m clusters $C_1^{PCM}, C_2^{PCM}, ..., C_m^{PCM}$ are returned.

B. Hierarchical Clustering Method

The HCM starts with each point as a cluster and then combines similar clusters into one cluster [60]. In our HCM, the vector angles among individuals are also used to measure the similarities in the objective space, while average-link [61] is employed to define the similarity of two clusters by using the average of vector angles among all their individuals. As mentioned above, this HCM is run after the above PCM, which classifies each of $C_1^{PCM}, C_2^{PCM}, ..., C_m^{PCM}$ into $k=N/m$ small clusters. To clarify the process of HCM, its pseudo-code is given in **Algorithm 2** with the inputs: $\boldsymbol{S}$ as a population and k. In line 1, each individual $s_i \in \boldsymbol{S}$ is set as a cluster C_i^{HCM} with centroid c_i^{HCM} $(i = 1, 2, ..., |\boldsymbol{S}|)$. Then, in lines 2-6, the minimal vector angle to each centroid c_i^{HCM} is obtained. The vector angle between two centroids c_i^{HCM} and c_j^{HCM} can be computed by (4), to define the similarity for two clusters C_i^{HCM} and C_j^{HCM}.

$$angle(c_i^{HCM}, c_j^{HCM}) \triangleq \arccos\left| \frac{\sum_{l=1}^{m} f_l^{'}\left(c_i^{HCM}\right)\cdot f_l^{'}\left(c_j^{HCM}\right)}{\sqrt{\sum_{l=1}^{m} f_l^{'}\left(c_i^{HCM}\right)^2}\cdot\sqrt{\sum_{l=1}^{m} f_l^{'}\left(c_j^{HCM}\right)^2}} \right| \quad (4)$$

where $f_l^{'}(c_i^{HCM})$ indicates the normalized value of the lth objective for c_i^{HCM} by (3). In line 3, $index(C_i^{HCM})$ (recording the index of the nearest cluster to C_i^{HCM}) is initialized to -1, $\theta(C_i^{HCM})$ (memorizing the minimal angle to C_i^{HCM}) is initialized to ∞, and $flag(C_i^{HCM})$ (indicating whether C_i^{HCM} has been combined or not) is initialized to **false**. For lines 4-5, the nearest cluster to C_i^{HCM} is found by respectively recording its index and the minimal vector angle as $index(C_i^{HCM})$ and $\theta(C_i^{HCM})$. Then, in lines 7-12, two most similar clusters (i.e., $C_{T_1}^{HCM}$ and $C_{T_2}^{HCM}$) are found and recorded, where T_1 and T_2 indicate their indexes in the clusters set, while *size* initialized as $|\boldsymbol{S}|$ is the number of clusters at first. When *size* is larger than k in line 13, the procedures in lines 14-21 will be run iteratively. In line 14, two clusters ($C_{T_1}^{HCM}$ and $C_{T_2}^{HCM}$) are combined as a new cluster C_{new}^{HCM}, and then the objective values for its new centroid c_{new}^{HCM} are updated by

$$f_i\left(c_{new}^{HCM}\right) = \frac{\sum f_i(p)}{\left|C_{new}^{HCM}\right|} \quad (5)$$

where $i = 1, 2, \ldots, m$, and $p \in C_{new}^{HCM}$. Then, set $flag(C_{T_1}^{HCM})=$ **true** to indicate that $C_{T_1}^{HCM}$ is combined, replace $C_{T_2}^{HCM}$ with C_{new}^{HCM}, and decrease *size* by 1 in line 15. In lines 16-20, the clusters C_i^{HCM} whose nearest cluster is now $C_{T_1}^{HCM}$ or $C_{T_2}^{HCM}$ will update its minimal angle $\theta(C_i^{HCM})$ and the index of its nearest neighbor $index(C_i^{HCM})$, by repeating the procedures in lines 4-5. After that, lines 7-12 are run again to find the two most similar clusters and record their indexes (T_1 and T_2), as shown in line 21. This iterative running of lines 13-22 will finally keep a set of k clusters. At last, remove the clusters C_i^{HCM} which were combined and marked with $flag(C_i^{HCM})$ as **true** in line 23 and return the results of clustering obtained by HCM (i.e., $C_1^{HCM}, C_2^{HCM}, \ldots, C_k^{HCM}$) in line 24.

III. The Proposed Algorithm

In this section, the proposed algorithm MaOEA/C is introduced. First, using the above PCM and HCM in the environmental selection is described. In this operator, the main idea is to divide the union of parents and offspring populations into N clusters. Then, one individual aiming to maintain diversity or convergence is selected from each cluster to compose the new population. At last, the complete algorithm of MaOEA/C is provided to show the details of other components.

A. Environmental Selection

In our design, the environmental selection operator follows the principle of diversity first and convergence second, aiming to select individuals with good convergence or diversity as the new population. The pseudo-code of our environmental selection is given in **Algorithm 3** with the inputs: $\boldsymbol{P}$ (the parent population), $\boldsymbol{Q}$ (the offspring population), N (the population size), and m (the number of objectives). In line 1, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are combined to get the union population $\boldsymbol{U}$ and then $\boldsymbol{P}$ is reset to an empty set. After that, all the individuals of $\boldsymbol{U}$ are normalized by (3) in line 2, and then their convergence indicator values are obtained by (6) in line 3.

Algorithm 3 *Environmental_Selection*($\boldsymbol{P}$, $\boldsymbol{Q}$, N, m)

```
1:  get the union population U = P ∪ Q, and set P = ∅
2:  normalize the individuals of U by (3)
3:  get the indicator values for the individuals of U by (6)
4:  (C_1^PCM, C_2^PCM, ..., C_m^PCM) = Partitional_Clustering (U, m)
5:  for i := 1 to m
6:    (F_1, F_2, ..., F_t) = Non-dominated-Sorting( C_i^PCM )
7:    set S = ∅ and j=1
8:    while j ≤ t
9:       S = S ∪ F_j and j = j+1
10:      if |S| ≥ k  break; // k=N/m
11:   end while
12:   (C_1^HCM, C_2^HCM, ..., C_k^HCM) = Hierarchical_Clustering (S, k)
13:   find the cluster C_d^HCM (d ∈ (1,k)) closest to C_i^PCM by (4)
14:   find the solution p closest to c_i^PCM from C_d^HCM by (2)
15:   add p into P and remove C_d^HCM from C_1^HCM, C_2^HCM, ..., C_k^HCM
16:   find one individual with the best convergence indicator
      value by (6) from each of the remaining clusters into P
17: end for
18: output P
```

$$c(u_i) = \sum\nolimits_{j=1}^{m} f_j^{'}(u_i) \quad (6)$$

where u_i is the ith individual from $\boldsymbol{U}$ and $f_j^{'}(u_i)$ indicates the normalized value of the jth objective for u_i by (3). This indicator only prefers the individuals' convergence in the same cluster, while the diversity is mainly maintained by the adopted clustering methods (PCM and HCM). In line 4, PCM (**Algorithm 1**) is first used to separate $\boldsymbol{U}$ into m main clusters C_i^{PCM} $(i = 1, 2, \ldots, m)$, which have $2k$ individuals in one cluster ($k = N/m$). For each cluster C_i^{PCM}, non-dominated sorting [4] is used to rank its solutions into different t convergence layers (i.e., F_1, F_2, …, F_t) in line 6, and then a solution set $\boldsymbol{S}$ is obtained by selecting no less than k individuals with the better Pareto-based rankings, as shown in lines 7-11. Then, in line 12, HCM (**Algorithm 2**) is further applied on $\boldsymbol{S}$ to get k clusters $C_1^{HCM}, C_2^{HCM}, \ldots, C_k^{HCM}$. In line 13, the cluster C_d^{HCM} $(d \in (1, k))$ closest to the cluster C_i^{PCM} is found based on the vector angles among their centroids by (4), and then the solution p closest to the centroid c_i^{PCM} is selected from C_d^{HCM} by (2) in line 14. In line 15, this solution p is added into $\boldsymbol{P}$ and the cluster C_d^{HCM} is removed from $C_1^{HCM}, C_2^{HCM}, \ldots, C_k^{HCM}$, as a representative solution p from C_d^{HCM} has been selected into $\boldsymbol{P}$. For each of the remaining clusters (i.e., $C_1^{HCM}, C_2^{HCM}, \ldots, C_k^{HCM}$ excluding C_d^{HCM}), one individual with the best indicator value on convergence by (6) is selected into $\boldsymbol{P}$ in line 16.

The adopted clustering methods help to maintain diversity, using the vector angles of individuals to estimate the similarities of clusters in high dimensional objective space. To emphasize diversity on each objective, the solution closest to the centroid c_i^{PCM} $(i = 1, 2, \ldots, m)$ (also the axis vector) is first selected from C_d^{HCM} in line 14. For each of the remaining clusters, only one individual is chosen with the best convergence indicator value as defined by (6), which aims to maintain convergence. An alternative method is presented in [44] to select the extreme solutions first and then consider the rest ones with good convergence, which will affect the clustering result and will slightly worsen the performance according to

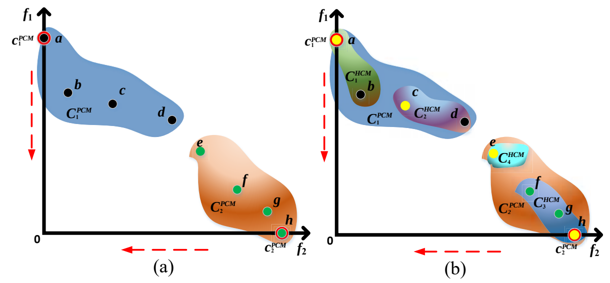


Fig. 1 A simple example with bi-objective optimization to illustrate the environmental selection process, using PCM and HCM.

our empirical studies. Please note that although a simple and effective convergence indicator in (6) is used, other convergence indicators in [47] can be also employed in MaOEA/C.

In order to facilitate the understanding of our environmental selection, a simple example is illustrated in Fig. 1. The union population includes eight individuals (*a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*) regarding the normalized bi-objective space. First, as shown in Fig. 1(a), PCM (**Algorithm 1**) is used to divide the population into two clusters (C_1^{PCM} and C_2^{PCM}) using the vector angles of each individual to the two centroids c_1^{PCM} (1, 0) and c_2^{PCM} (0, 1). Then, the individuals *a*, *b*, *c*, *d* belong to C_1^{PCM} as they are close to c_1^{PCM} by (2), while the individuals *e*, *f*, *g*, *h* are classified into C_2^{PCM} in the same manner. Second, in Fig. 1(b), HCM (**Algorithm 2**) is run on each of C_1^{PCM} and C_2^{PCM} , which divides C_1^{PCM} into two clusters (C_1^{HCM} and C_2^{HCM}) and C_2^{PCM} into two clusters (C_3^{HCM} and C_4^{HCM}). Finally, C_1^{HCM} has *a* and *b*, C_2^{HCM} includes *c* and *d*, C_3^{HCM} owns *f*, *g*, and *h*, while C_4^{HCM} contains *e*. After that, *a* is selected from C_1^{HCM} as it is closest to c_1^{PCM} (1, 0), and *h* is selected from C_3^{HCM} as it is closest to c_2^{PCM} (0, 1). *c* is chosen from C_2^{HCM} as it has the smallest convergence indicator value by (6) in C_2^{HCM} , and *e* is chosen from C_4^{HCM} in the same manner. These four selected individuals are marked with yellow color in Fig. 1(b).

B. *The Complete Algorithm of MaOEA/C*

In the above subsection, the environmental selection of our algorithm has been introduced, while our evolutionary operators are simulated binary crossover (SBX) and polynomial-based mutation [62] like in other MaOEAs [19]-[33], with a similarity-based mating selection. Here, the pseudo-code of the complete algorithm MaOEA/C is provided in **Algorithm 4**. In line 1, the values of m (the number of objectives), N (the population size), ε (a parameter to control the similarity-based mating selection) and G_{max} (the pre-set maximal generations) are initialized. Then, an initial population $\boldsymbol{P}$ is generated randomly in decision space Ω , and an offspring population $\boldsymbol{Q}$ is initialized as an empty set. The generation counter G is set to 1 in line 2. In line 4, the population $\boldsymbol{P}$ is classified into m clusters $C_1^{PCM}, C_2^{PCM}, ..., C_m^{PCM}$ using **Algorithm 1** with the inputs: $\boldsymbol{P}$ and m (details are provided in **Section II.A**). For each individual in cluster C_i^{PCM} ($i = 1, 2, ..., m$), the similarity-based mating selection is run such that two parents (p_1 and p_2) are randomly selected from the same cluster C_i^{PCM} with a probability of ε and from the entire population with a probability of 1- ε , as shown in lines 7-12. Afterwards, SBX is run for the parents p_1 and p_2 to get an intermediate solution u in

Algorithm 4 General Framework of MaOEA/C

1: initialize N, m, ε, G_{max}, $\boldsymbol{P}$ and $\boldsymbol{Q}$
2: $G = 1$
3: **while** $G \le G_{max}$
4: ($C_1^{PCM}, C_2^{PCM}, ..., C_m^{PCM}$) = *Partitional_Clustering* ($\boldsymbol{P}$, m)
5: **for** i := 1 to m
6: **for** j := 1 to $|C_i^{PCM}|$
7: generate a random number r in (0,1)
8: **if** $r < \varepsilon$
9: randomly select two parents (p_1, p_2) from C_i^{PCM}
10: **else**
11: randomly select two parents (p_1, p_2) from $\boldsymbol{P}$
12: **end if**
13: run SBX on p_1 and p_2 to get u
14: run polynomial-based mutation on u to get v
15: add v to $\boldsymbol{Q}$
16: **end for**
17: **end for**
18: $\boldsymbol{P}$ = *Environmental_Selection*($\boldsymbol{P}$, $\boldsymbol{Q}$, N, m)
19: set $\boldsymbol{Q}$ as an empty set and G++
20: **end while**
21: **return** P

line 13 and polynomial-based mutation is further applied to u to get a new solution v in line 14. This offspring v is then added into $\boldsymbol{Q}$ in line 15. After all the offspring are generated in lines 5-17 and collected into $\boldsymbol{Q}$, **Algorithm 3** is run in line 18 with the inputs: $\boldsymbol{P}$, $\boldsymbol{Q}$, N and m (details are provided in **Section III.A**), which selects the population for the next generation. At last, the offspring population $\boldsymbol{Q}$ is reset to an empty set and the generation counter G is increased by 1 in line 19. While G is smaller than G_{max}, the above procedures in lines 4-19 will be run iteratively. Otherwise, $\boldsymbol{P}$ is the output produced in the last generation in line 21, as the final approximation set.

IV. Experimental Studies

A. *Benchmark Problems and Performance Measures*

In this study, the MaF1-MaF7 test problems [48] with complicated PFs and the WFG1-WFG9 [56] test problems with different scaled objectives were used. For each problem, the number of objectives m was varied from 5 to 15, i.e., $m \in \{5, 8, 10, 13, 15\}$. Moreover, the number of decision variables n in MaF1-MaF7 was set by $n = m+k-1$, where k was set to 10 for MaF1-MaF6 and to 20 for MaF7 as suggested in [48]. In WFG1-WFG9, the decision variables have k position-related parameters and l distance-related parameters. As recommended in [43], k was set to $2 \times (m-1)$ and l was set to 20. Due to page limitations, the main characteristics of the MaF and WFG test problems are summarized in Table A.I of the supplementary file.

The HV metric was used as our performance indicator to reflect the solutions' quality in terms of both convergence and maximum spread for different MaOEAs [18]-[29]. Due to page limitations, please refer to [8], [37] for details of HV. A larger HV value indicates a better approximation to the true PF. As pointed out in [70], the reference point should be carefully set for MaOPs with inverted triangular PFs to compute HV. To allow a fair comparison, the setting of reference point in [43], [63], [64] was used in this paper. The objective vectors in the final solution sets are normalized by $1.1 \times (f_1^{\max}, f_2^{\max}, ..., f_m^{\max})$,

TABLE I
PARAMETERS SETTINGS OF ALL THE COMPARED ALGORITHMS

Algorithm	Parameters settings
NSGA-III	p_c =1.0, p_m =1/n, η_c =30, η_m =20
MOEA/D-DU	p_c =1.0, p_m =1/n, η_c =20, η_m =20, T =20, δ =0.9, K=5
EFR-RR	p_c =1.0, p_m =1/n, η_c =30, η_m =20, K =2
θ-DEA	p_c =1.0, p_m =1/n, η_c =30, η_m =20, θ =5.0
SRA	p_c =1.0, p_m =0.1, η_c =15, η_m =15, p_c' =0.5
VaEA	p_c =1.0, p_m =1/n, η_c =30, η_m =20, $\sigma=\pi/2(N+1)$
MaOEA/C	p_c =1.0, p_m =1/n, η_c =30, η_m =20, ε=0.8

where $f_k^{\max}$ (k = 1,2,...,m) is the maximum value of the kth objective in the true PF, and then the reference point was set to (1.0,1.0,...,1.0) . Please note that the solutions that cannot dominate the reference point were not included to compute HV. The recently proposed walking fish group algorithm [37] was used to compute the exact HV values for test problems with no more than 10 objectives and to approximate HV by the Monte Carlo simulation [18] with 10^7 sampling points for test problems having 13 and 15 objectives.

B. Parameters Settings for the Compared Algorithms

In this study, six competitive MaOEAs, i.e., NSGA-III [26], MOEA/D-DU [21], EFR-RR [21], θ-DEA [22], SRA [43], and VaEA [44], were included for performance comparison. These algorithms cover all the main kinds of MaOEAs in Section I. The source code of NSGA-III was implemented by the authors of θ-DEA, while other source codes are provided by their authors. All the MaOEAs were implemented using Java codes and run on a personal computer having an Intel (R) Core (TM) i7-6700 CPU, 3.40GHz (processor), and 20 GB (RAM). The parameters settings of the compared algorithms are listed in Table I, as suggested in their references. All the competitors used the same evolutionary operators, i.e., SBX and polynomial-based mutation. p_c and p_m are the probabilities to run crossover and mutation respectively. η_c and η_m are respectively the distribution indexes of SBX and polynomial-based mutation. In MOEA/D-DU, T defines the neighborhood size among weight vectors, and δ indicates the probability to select parent solutions from T neighbors. K is used in MOEA/D-DU and EFR-RR to balance convergence and diversity, which is set to 5 for MOEA/D-DU and to 2 for EFR-RR. For θ-DEA, θ is a penalty parameter used in the dominance relation. Regarding SRA, p_c' is an inherent parameter for stochastic ranking, which controls the balance of two indicators. In VaEA, σ is a threshold to decide whether the solutions are deemed to search from a similar direction. Moreover, ε is set to 0.8 in line 8 of **Algorithm 4** for MaOEA/C.

The settings of population size for different numbers of objectives are listed in Table II. For test problems with 5, 8, 10, 13, and 15 objectives, the number of weight vectors was respectively set to 210, 240, 275, 182, and 240, using the two-layer generation method with the simplex-lattice design factor H in [33]. According to [20], the population size in NSGA-III should be set as the smallest multiple of four, which is slightly larger than the number of weight vectors in some cases. MOEA/D-DU, EFR-RR, and θ-DEA should use the same population size as NSGA-III, since they were designed based on NSGA-III. Thus, NSGA-III, MOEA/D-DU, EFR-RR, and θ-DEA adopted the population sizes of 210, 240, 276, 184, and 240 for 5-, 8-, 10-, 13-, and 15-objective problems respectively. For MaOEA/C, the population size should be set as a multiple of m. Thus, MaOEA/C, SRA and VaEA used the population sizes of 210, 240, 280, 182, and 240 for 5-, 8-, 10-, 13-, and 15-objective problems, respectively.

TABLE II
SETTINGS OF THE POPULATION SIZE AND EVALUATIONS

Objectives (m)	Divisions (H)	Population Size	G_{max}	Evaluations
5	6	210	500	105000
8	3,3	240	700	168000
10	3,2	276/280	700	196000
13	2,2	184/182	1000	182000
15	2,2	240	1000	240000

All the algorithms were run 30 times independently on each test problem. The mean HV values and the standard deviations (included in bracket after the mean HV results) from 30 runs were collected for comparison. All the algorithms were terminated when a predefined maximum number of generations G_{max} was reached. The settings of G_{max} for different numbers of objectives are listed in Table II. For each algorithm, the maximum function evaluations (*MFE*) can be easily determined by $MFE = N \cdot G_{max}$. To obtain a statistically sound conclusion, Wilcoxon rank sum test was run with a significance level $\alpha = 0.05$, showing the statistically significant differences on the results of MaOEA/C and other competitors. In the following tables, the symbols "+", "–", and "~" indicate that the results of other competitors are significantly better than, worse than, and similar to the ones of MaOEA/C using this statistical test, respectively.

C. Comparison with Six Competitive MaOEAs

1) Comparison Results on MaF1-MaF7

Table III provides a comparison of results in terms of HV on MaF1-MaF7 with 5 to 15 objectives. As observed from the second last row of Table III, MaOEA/C obtained the best results in 24 out of 35 cases, while NSGA-III, MOEA/D-DU, θ-DEA, SRA, VaEA, and EFR-RR performed respectively best in 6, 9, 2, 3, 4, and 4 cases, which validates the superiority of MaOEA/C on these MaF problems.

MaF1 was obtained by inverting the PF shape of DTLZ1 [65]. This way, the shape of the reference points cannot fit to the PF shape, and the performance of the reference point based MaOEAs (NSGA-III, MOEA/D-DU, EFR-RR, and θ-DEA) was worse than other MaOEAs that don't adopt reference points (SRA, VaEA and MaOEA/C) in the cases of 5, 8 and 10 objectives. However, for the cases of 13 and 15 objectives, NSGA-III, MOEA/D-DU, EFR-RR, and θ-DEA seemed more advantageous, as they had the smallest HV values larger than 0, while SRA, VaEA and MaOEA/C only showed the HV value as 0. This may be because the search directions of SRA, VaEA and MaOEA/C as guided by the vector angle was not so effective with a small population size in very high dimensional objective space, while NSGA-III, MOEA/D-DU, EFR-RR, and θ-DEA could still find some solutions to approximate the true PF and obtained very small HV values. Anyhow, MaOEA/C still showed a competitive performance in MaF1, as it was best in the instances having 5, 8 and 10 objectives.

TABLE III
COMPARISON OF RESULTS OF MaOEA/C AND SIX COMPETITIVE MaOEAs ON MaF1-MaF7 USING HV

Problem	*m*	NSGA-III	MOEA/D-DU	θ-DEA	SRA	VaEA	EFR-RR	MaOEA/C
MaF1	5	7.550E–03(5.78E–04)–	2.940E–03(4.66E–05)–	5.608E–03(2.22E–04)–	1.056E–02(1.24E–04)–	1.107E–02(1.79E–04)–	3.718E–03(6.09E–04)–	**1.287E-02(1.09E-04)**
	8	2.677E–05(1.27E–06)–	4.127E–06(1.29E–06)–	2.945E–05(1.27E–06)–	1.612E–05(8.68E–07)–	2.703E–05(1.31E–06)–	1.445E–05(4.40E–06)–	**4.127E-05(1.34E-06)**
	10	4.256E–07(1.57E–08)–	3.547E–08(1.28E–08)–	3.213E–07(4.90E–08)–	1.710E–07(8.04E–09)–	3.597E–07(2.07E–08)–	1.843E–07(3.78E–08)–	**5.646E-07(2.80E-08)**
	13	4.128E–10(3.61E–11)+	1.592E–11(4.89E–12)+	**4.363E–10(8.29E–11)+**	0.000E+00(0.00E+00)~	0.000E+00(0.00E+00)~	1.208E–10(5.68E–11)+	0.000E+00(0.00E+00)
	15	4.714E–12(8.02E–13)+	1.300E–13(2.58E–14)+	**5.707E–12(1.03E–12)+**	0.000E+00(0.00E+00)~	0.000E+00(0.00E+00)~	1.342E–12(5.26E–13)+	0.000E+00(0.00E+00)
MaF2	5	2.438E–01(2.16E–03)–	2.420E–01(1.11E–03)–	2.340E–01(3.56E–03)–	2.526E–01(1.16E–03)–	2.357E–01(3.60E–03)–	2.277E–01(3.43E–03)–	**2.609E-01(9.24E-04)**
	8	2.046E–01(5.42E–03)–	1.957E–01(1.76E–03)–	1.768E–01(1.10E–02)–	2.255E–01(1.93E–03)–	2.023E–01(5.33E–03)–	1.862E–01(4.57E–03)–	**2.406E-01(2.62E-03)**
	10	2.131E–01(3.60E–03)–	1.914E–01(1.72E–03)–	1.949E–01(9.50E–03)–	2.136E–01(2.18E–03)–	2.010E–01(7.53E–03)–	1.952E–01(5.44E–03)–	**2.436E-01(2.52E-03)**
	13	1.420E–01(6.99E–03)–	1.339E–01(3.40E–03)–	1.340E–01(1.42E–02)–	1.898E–01(4.65E–03)–	1.890E–01(5.30E–03)–	1.159E–01(1.30E–02)–	**2.306E-01(5.22E-03)**
	15	1.325E–01(9.24E–03)–	1.369E–01(2.92E–03)–	1.248E–01(1.54E–02)–	1.941E–01(3.67E–03)–	1.902E–01(5.13E–03)–	1.021E–01(1.51E–02)–	**2.324E-01(5.90E-03)**
MaF3	5	9.254E–01(2.52E–01)–	**9.993E–01(5.59E–05)+**	9.912E–01(2.46E–03)+	9.932E–01(2.59E–03)+	9.967E–01(1.20E–03)+	0.000E+00(0.00E+00)–	9.857E-01(4.31E-03)
	8	9.954E–01(1.18E–02)+	**1.000E+00(6.76E–07)+**	9.914E–01(5.86E–03)+	9.983E–01(1.08E–03)+	9.330E–01(2.54E–01)–	6.687E–01(4.13E–01)–	9.896E-01(9.54E-03)
	10	9.220E–01(2.53E–01)–	**1.000E+00(5.06E–09)+**	9.902E–01(4.67E–03)~	9.994E–01(4.40E–04)+	5.624E–01(4.97E–01)–	1.695E–01(2.70E–01)–	9.919E-01(7.00E-03)
	13	4.786E–01(4.90E–01)–	**9.999E–01(2.02E–04)+**	8.463E–01(1.96E–01)–	9.996E–01(3.57E–04)+	7.274E–01(3.70E–01)–	8.842E–01(2.10E–01)–	9.211E-01(9.94E-02)
	15	4.877E–01(4.49E–01)–	**9.999E–01(1.35E–04)+**	7.220E–01(2.83E–01)–	9.998E–01(1.92E–04)+	7.354E–01(3.95E–01)–	6.536E–01(2.44E–01)–	9.387E-01(1.86E-02)
MaF4	5	8.132E–02(6.15E–03)–	1.421E–04(7.76E–04)–	7.196E–02(9.07E–03)–	8.776E–02(7.31E–03)–	1.162E–01(3.89E–03)–	8.277E–03(1.30E–02)–	**1.171E-01(8.09E-03)**
	8	2.159E–03(3.07E–04)–	0.000E+00(0.00E+00)–	1.418E–03(5.27E–04)–	2.368E–04(6.10E–05)–	2.356E–03(2.69E–04)–	2.522E–03(3.39E–04)–	**5.181E-03(4.35E-04)**
	10	2.329E–04(2.48E–05)–	0.000E+00(0.00E+00)–	2.282E–04(2.25E–05)–	1.815E–06(7.59E–07)–	1.383E–04(1.68E–05)–	2.441E–04(4.19E–05)–	**4.716E-04(3.56E-05)**
	13	3.120E–06(2.89E–07)–	0.000E+00(0.00E+00)–	2.359E–06(2.76E–07)–	7.230E–10(4.80E–10)–	2.037E–06(4.77E–07)–	3.590E–06(8.94E–07)–	**4.134E-06(8.55E-07)**
	15	2.024E–07(1.46E–08)–	0.000E+00(0.00E+00)–	1.420E–07(1.60E–08)–	7.233E–12(3.76E–12)–	8.160E–08(6.47E–08)–	2.749E–07(4.55E–08)–	**2.751E-07(5.24E-08)**
MaF5	5	**9.998E–01(4.45E–06)~**	9.993E–01(6.24E–06)–	9.967E–01(9.71E–04)–	9.950E–01(2.41E–03)–	9.996E–01(2.78E–05)~	9.997E–01(1.29E–05)~	9.993E-01(2.35E-04)
	8	**1.000E+00(4.15E–07)~**	**1.000E+00(7.08E–07)~**	9.923E–01(1.03E–03)–	9.980E–01(1.56E–03)–	**1.000E+00(1.92E–06)~**	**1.000E+00(1.59E–07)~**	**1.000E+00(1.46E-05)**
	10	**1.000E+00(1.90E–08)+**	**1.000E+00(1.77E–07)~**	9.891E–01(8.84E–04)–	9.989E–01(1.50E–03)–	9.994E–01(3.11E–03)–	**1.000E+00(6.62E–07)~**	**1.000E+00(4.41E-06)**
	13	**1.000E+00(0.00E+00)+**	**1.000E+00(2.54E–07)~**	9.970E–01(8.78E–04)–	9.999E–01(1.07E–04)~	**1.000E+00(5.86E–14)+**	**1.000E+00(2.58E–07)~**	**1.000E+00(5.15E-06)**
	15	**1.000E+00(0.00E+00)+**	**1.000E+00(1.83E–07)~**	9.983E–01(5.09E–04)–	**1.000E+00(3.13E–05)~**	**1.000E+00(9.73E–16)+**	**1.000E+00(1.24E–16)+**	**1.000E+00(1.43E-06)**
MaF6	5	1.235E–01(2.23E–03)–	1.200E–01(2.95E–04)–	1.176E–01(2.70E–03)–	1.294E–01(1.15E–04)–	**1.300E–01(5.37E–05)~**	0.000E+00(0.00E+00)–	**1.300E-01(1.97E-04)**
	8	1.048E–01(8.30E–04)–	9.985E–02(1.77E–04)–	8.230E–02(3.63E–02)–	1.061E–01(4.24E–05)~	9.779E–02(2.80E–02)–	0.000E+00(0.00E+00)–	**1.063E-01(2.12E-04)**
	10	1.241E–02(2.38E–02)–	9.409E–02(2.76E–04)–	1.947E–02(2.80E–02)–	9.533E–02(1.86E–02)–	5.190E–02(4.20E–02)–	0.000E+00(0.00E+00)–	**1.008E-01(1.89E-04)**
	13	1.471E–02(2.27E–02)–	9.406E–02(2.06E–04)–	6.825E–02(1.73E–02)–	8.074E–03(1.47E–02)–	8.360E–02(7.58E–03)–	1.551E–03(7.23E–03)–	**9.647E-02(9.78E-04)**
	15	1.653E–03(7.34E–03)–	9.286E–02(1.79E–04)–	5.325E–02(2.73E–02)–	3.194E–03(7.65E–03)–	7.740E–02(8.48E–03)–	8.723E–04(3.79E–03)–	**9.447E-02(1.22E-03)**
MaF7	5	3.055E–01(2.55E–03)~	1.246E–01(5.95E–02)–	2.776E–01(9.50E–03)–	**3.241E–01(1.52E–03)+**	3.025E–01(2.78E–03)~	1.699E–01(1.66E–02)–	3.053E-01(4.06E-03)
	8	2.581E–01(2.59E–03)+	2.629E–03(1.16E–02)–	2.187E–01(2.79E–02)–	**2.670E–01(5.67E–03)+**	2.240E–01(4.17E–03)–	2.159E–01(1.41E–02)–	2.404E-01(4.19E-03)
	10	**2.349E–01(3.53E–03)+**	6.613E–05(6.98E–05)–	2.241E–01(1.77E–02)+	1.942E–01(1.59E–02)–	1.814E–01(1.05E–02)–	1.389E–01(1.40E–02)–	2.006E-01(1.07E-02)
	13	1.332E–01(7.81E–02)–	1.243E–02(1.97E–02)–	1.249E–01(1.61E–02)–	9.387E–02(2.55E–02)–	1.384E–01(2.79E–03)–	1.436E–01(2.47E–02)–	**1.715E-01(1.25E-02)**
	15	1.048E–01(6.35E–02)–	6.431E–03(1.24E–02)–	1.167E–01(1.74E–02)–	7.881E–02(2.35E–02)–	1.306E–01(2.49E–03)–	1.277E–01(1.77E–02)–	**1.595E-01(1.20E-02)**
best/all		6/35	9/35	2/35	3/35	4/35	4/35	24/35
+/–/~		8/24/3	7/24/4	5/29/1	7/22/6	3/26/6	3/28/4	– –

MaF2 was obtained from DTLZ2 by raising the difficulty of convergence, and all the objectives of MaF2 need to be optimized simultaneously in order to reach the true PF. MaOEA/C showed the best performance on MaF2 with all the objectives. Regarding MaF3, it is characterized with a convex PF and a large number of local PFs. Only MOEA/D-DU and SRA can consistently solve this problem well. Other algorithms may step into the local PFs, as observed from the large values of standard deviation. MaF4 was derived from DTLZ3 by inverting its PF shape, and our algorithm performed best in all the instances of MaF4. On MaF5 which has a highly biased distribution on PS and a badly scaled PF, all the algorithms solved it very well as their HV values were close to 1.0. Regarding MaF6 with a degenerate PF, MaOEA/C showed to be advantageous as it obtained the best results in all cases. At last, on MaF7 with a disconnected PF, MaOEA/C had a competitive performance, as it obtained the best results in the instances having 13 and 15 objectives, while it was outperformed by SRA in the case with 5 objectives, by NSGA-III and SRA in the case of 8 objectives, and by NSGA-III and θ-DEA in the case of 10 objectives.

In the last row of Table III, the one-by-one comparisons of MaOEA/C and other six competitors were summarized, where "+/–/~" indicate the numbers of test problems in which the competitors are respectively better than, worse than and similar to MaOEA/C. From these comparisons, MaOEA/C was better than NSGA-III, MOEA/D-DU, θ-DEA, SRA, VaEA, and EFR-RR, in 24, 24, 29, 22, 26, and 28 out of 35 cases, respectively. Conversely, it was worse than NSGA-III, MOEA/D-DU, θ-DEA, SRA, VaEA, and EFR-RR, in 8, 7, 5, 7, 3, and 3 cases, respectively. Therefore, it is reasonable to conclude that MaOEA/C showed a superior performance over its six competitors, in most instances of MaF1-MaF7.

2) *Comparison Results on WFG1-WFG9*

Table IV collects the HV comparison results of all the algorithms on WFG1-WFG9 with 5 to 15 objectives. On these results, MaOEA/C also showed a superior performance, as it was best in about half of the test problems, i.e., in 23 out of 45 cases. MOEA/D-DU, θ-DEA, SRA, VaEA, and EFR-RR were respectively best in 3, 3, 2, 2, and 13 cases, while NSGA-III was unable to perform best on any test problem. These results were summarized in the second last row of Table IV.

Regarding WFG1 with a convex, mixed and biased PF, MaOEA/C performed best in the cases of 5 and 8 objectives, while θ-DEA was best in the case of 10 objectives and VaEA obtained the best results in the cases of 13 and 15 objectives. For WFG2 with a disconnected and mixed PF, MaOEA/C only gave a median performance among all the compared MaOEAs.

TABLE IV
COMPARISON OF RESULTS OF MaOEA/C AND SIX COMPETITIVE MOEAS ON WFG1-WFG9 USING HV

Problem	*m*	NSGA-III	MOEA/D-DU	θ-DEA	SRA	VaEA	EFR-RR	MaOEA/C
WFG1	5	3.643E-01(2.96E-02)–	5.951E-01(3.25E-02)–	5.391E-01(2.72E-02)–	4.917E-01(2.20E-02)–	3.411E-01(3.21E-02)–	3.590E-01(2.04E-02)–	**6.416E-01(1.58E-02)**
	8	4.353E-01(4.05E-02)–	5.636E-01(2.84E-02)–	7.711E-01(2.12E-02)–	6.126E-01(2.26E-02)–	5.694E-01(2.70E-02)–	6.206E-01(7.83E-02)–	**7.782E-01(8.24E-03)**
	10	6.531E-01(4.01E-02)–	5.676E-01(3.78E-02)–	**8.547E-01(9.43E-03)+**	6.466E-01(1.95E-02)–	7.106E-01(3.14E-02)–	7.607E-01(3.39E-02)–	8.233E-01(1.29E-02)
	13	8.425E-01(2.50E-02)–	8.580E-01(3.15E-02)+	8.696E-01(2.02E-02)+	7.686E-01(2.14E-02)–	**8.715E-01(1.20E-02)+**	7.653E-01(4.47E-02)–	8.450E-01(9.98E-03)
	15	8.610E-01(1.75E-02)+	8.526E-01(2.45E-02)~	8.652E-01(1.04E-02)+	7.691E-01(2.30E-02)–	**8.679E-01(7.93E-03)+**	7.590E-01(5.18E-02)–	8.527E-01(1.25E-02)
WFG2	5	9.576E-01(5.23E-02)–	9.746E-01(5.34E-02)+	9.482E-01(6.54E-02)–	9.725E-01(3.19E-03)+	9.651E-01(3.11E-02)~	**9.861E-01(2.36E-03)+**	9.664E-01(4.35E-02)
	8	9.323E-01(7.84E-02)–	9.841E-01(3.23E-02)+	8.963E-01(8.51E-02)–	9.764E-01(3.16E-02)~	9.683E-01(4.38E-02)–	**9.952E-01(1.36E-03)+**	9.731E-01(6.03E-02)
	10	9.599E-01(5.99E-02)–	**9.971E-01(1.05E-03)+**	9.076E-01(8.50E-02)–	9.798E-01(3.22E-02)~	9.836E-01(3.39E-03)~	9.601E-01(5.82E-02)–	9.818E-01(3.23E-02)
	13	9.394E-01(7.37E-02)–	9.536E-01(5.82E-02)~	7.821E-01(6.05E-02)–	**9.755E-01(4.45E-02)+**	9.693E-01(5.38E-02)+	9.521E-01(7.06E-02)~	9.520E-01(7.96E-02)
	15	9.362E-01(7.62E-02)–	9.791E-01(9.16E-03)+	7.897E-01(5.10E-02)–	9.841E-01(3.23E-02)+	9.803E-01(5.41E-02)+	**9.844E-01(3.28E-02)+**	9.695E-01(5.80E-02)
WFG3	5	6.164E-01(7.72E-03)–	6.231E-01(7.88E-03)–	6.329E-01(7.32E-03)–	6.245E-01(5.38E-03)–	5.666E-01(1.34E-02)–	5.903E-01(8.18E-03)–	**6.382E-01(6.52E-03)**
	8	5.938E-01(1.48E-02)–	4.795E-01(1.14E-02)–	5.722E-01(3.07E-02)–	6.020E-01(1.33E-02)–	5.759E-01(1.51E-02)–	6.161E-01(9.11E-03)–	**6.593E-01(4.26E-03)**
	10	6.216E-01(2.75E-02)–	4.522E-01(8.20E-03)–	6.053E-01(1.97E-02)–	5.898E-01(1.44E-02)–	5.393E-01(3.53E-02)–	6.081E-01(1.55E-02)–	**6.684E-01(6.14E-03)**
	13	6.222E-01(2.80E-02)–	4.177E-01(1.40E-02)–	5.746E-01(2.54E-02)–	5.545E-01(1.90E-02)–	5.675E-01(4.30E-02)–	5.485E-01(2.55E-02)–	**6.610E-01(6.41E-03)**
	15	6.455E-01(1.86E-02)–	4.194E-01(1.40E-02)–	5.896E-01(2.69E-02)–	5.582E-01(1.84E-02)–	5.753E-01(4.52E-02)–	5.640E-01(1.98E-02)–	**6.729E-01(5.40E-03)**
WFG4	5	7.441E-01(6.46E-03)–	**7.620E-01(7.38E-03)+**	7.525E-01(4.61E-03)–	7.165E-01(4.58E-03)–	7.034E-01(6.37E-03)–	7.379E-01(5.01E-03)–	7.556E-01(5.06E-03)
	8	7.675E-01(1.74E-02)–	8.513E-01(9.31E-03)–	7.931E-01(1.47E-02)–	7.701E-01(8.35E-03)–	8.043E-01(1.17E-02)–	**8.975E-01(5.42E-03)~**	8.957E-01(5.49E-03)
	10	8.523E-01(9.95E-03)–	9.149E-01(5.64E-03)–	8.696E-01(9.14E-03)–	7.836E-01(1.04E-02)–	8.158E-01(1.00E-02)–	8.419E-01(1.16E-02)–	**9.258E-01(5.83E-03)**
	13	7.386E-01(2.56E-02)–	7.793E-01(1.93E-02)–	8.074E-01(1.83E-02)–	7.507E-01(1.42E-02)–	8.201E-01(1.66E-02)–	8.858E-01(2.73E-02)–	**9.054E-01(1.40E-02)**
	15	7.483E-01(3.08E-02)–	7.986E-01(2.10E-02)–	8.298E-01(1.58E-02)–	7.607E-01(1.59E-02)–	8.133E-01(1.39E-02)–	8.481E-01(2.59E-02)–	**9.291E-01(9.10E-03)**
WFG5	5	7.249E-01(2.45E-03)–	7.280E-01(3.98E-03)+	7.277E-01(2.89E-03)~	6.943E-01(4.26E-03)–	6.905E-01(6.20E-03)–	**7.336E-01(5.42E-03)+**	7.269E-01(4.26E-03)
	8	7.847E-01(6.55E-03)–	8.181E-01(3.16E-03)–	7.897E-01(5.73E-03)–	7.512E-01(7.74E-03)–	7.908E-01(8.06E-03)–	**8.588E-01(4.99E-03)+**	8.398E-01(5.06E-03)
	10	8.494E-01(4.15E-03)–	8.716E-01(3.09E-03)–	8.534E-01(3.59E-03)–	7.650E-01(8.58E-03)–	8.050E-01(7.65E-03)–	8.103E-01(7.11E-03)–	**8.806E-01(3.75E-03)**
	13	7.598E-01(2.09E-02)–	7.286E-01(1.49E-02)–	8.181E-01(1.18E-02)–	7.540E-01(1.84E-02)–	8.186E-01(8.60E-03)–	8.488E-01(1.00E-02)–	**8.517E-01(1.75E-02)**
	15	7.527E-01(2.67E-02)–	7.543E-01(1.19E-02)–	8.101E-01(1.39E-02)–	7.718E-01(1.94E-02)–	8.181E-01(7.66E-03)–	8.555E-01(7.80E-03)–	**8.660E-01(6.38E-03)**
WFG6	5	7.331E-01(6.32E-03)+	7.017E-01(1.18E-02)–	**7.379E-01(6.50E-03)+**	6.768E-01(9.52E-03)–	6.925E-01(7.33E-03)–	7.349E-01(7.37E-03)–	7.302E-01(6.88E-03)
	8	8.106E-01(1.01E-02)–	7.847E-01(1.14E-02)–	8.142E-01(8.97E-03)–	7.181E-01(1.30E-02)–	8.187E-01(1.21E-02)–	**8.682E-01(8.18E-03)+**	8.450E-01(9.33E-03)
	10	8.737E-01(7.46E-03)–	8.547E-01(1.00E-02)–	8.769E-01(7.78E-03)–	7.234E-01(1.37E-02)–	8.331E-01(8.48E-03)–	8.389E-01(9.72E-03)–	**8.825E-01(7.64E-03)**
	13	8.676E-01(1.43E-02)–	7.366E-01(1.44E-02)–	8.714E-01(1.10E-02)–	7.158E-01(3.58E-02)–	8.511E-01(1.38E-02)–	**8.841E-01(8.24E-03)+**	8.814E-01(1.19E-03)
	15	8.670E-01(1.19E-02)–	7.687E-01(1.61E-02)–	8.776E-01(1.07E-02)–	7.385E-01(3.12E-02)–	8.542E-01(8.93E-03)–	**8.969E-01(9.29E-03)~**	8.958E-01(1.00E-03)
WFG7	5	7.775E-01(2.94E-03)–	7.671E-01(6.39E-03)–	7.893E-01(2.71E-03)~	7.390E-01(3.73E-03)–	7.406E-01(5.88E-03)–	**7.924E-01(1.83E-03)+**	7.852E-01(2.41E-03)
	8	8.264E-01(1.55E-02)–	8.605E-01(7.00E-03)–	8.488E-01(8.55E-03)–	8.030E-01(1.07E-02)–	8.714E-01(6.32E-03)–	**9.147E-01(1.79E-03)~**	**9.147E-01(2.69E-03)**
	10	9.104E-01(5.55E-03)–	9.308E-01(2.48E-03)–	9.187E-01(3.99E-03)–	8.145E-01(7.50E-03)–	8.885E-01(7.35E-03)–	9.021E-01(9.21E-03)–	**9.549E-01(1.74E-03)**
	13	8.481E-01(2.00E-02)–	7.757E-01(2.05E-02)–	8.595E-01(1.70E-02)–	7.554E-01(1.39E-02)–	8.996E-01(9.90E-03)–	9.341E-01(8.27E-03)–	**9.486E-01(1.73E-02)**
	15	8.449E-01(2.65E-02)–	7.912E-01(2.30E-02)–	8.959E-01(1.41E-02)–	7.661E-01(1.45E-02)–	9.025E-01(6.82E-03)–	9.336E-01(8.94E-03)–	**9.602E-01(1.26E-02)**
WFG8	5	6.528E-01(4.89E-03)–	**6.767E-01(6.53E-03)+**	6.574E-01(4.63E-03)~	6.156E-01(5.29E-03)–	5.896E-01(8.08E-03)–	6.332E-01(4.71E-03)–	6.561E-01(3.67E-03)
	8	6.549E-01(1.85E-02)–	7.474E-01(2.46E-02)–	6.580E-01(1.90E-02)–	6.569E-01(1.14E-02)–	6.285E-01(1.15E-02)–	7.599E-01(6.88E-03)–	**7.712E-01(1.01E-02)**
	10	7.713E-01(1.42E-02)–	8.308E-01(2.15E-02)–	7.764E-01(1.23E-02)–	6.595E-01(1.44E-02)–	6.271E-01(3.34E-02)–	7.142E-01(1.84E-02)–	**8.435E-01(1.01E-02)**
	13	6.182E-01(1.00E-01)–	6.789E-01(4.71E-02)–	7.579E-01(4.63E-02)–	6.169E-01(2.53E-02)–	7.212E-01(2.60E-02)–	8.072E-01(1.28E-02)–	**8.736E-01(4.87E-03)**
	15	7.377E-01(1.10E-01)–	7.300E-01(5.42E-02)–	8.384E-01(2.11E-02)–	6.302E-01(2.19E-02)–	7.705E-01(2.98E-02)–	8.411E-01(1.23E-02)–	**9.083E-01(5.86E-03)**
WFG9	5	6.546E-01(1.48E-02)–	6.563E-01(2.32E-02)–	6.648E-01(1.77E-02)–	**6.754E-01(2.07E-02)+**	6.346E-01(5.36E-03)–	6.438E-01(7.99E-03)–	6.668E-01(2.57E-02)
	8	6.632E-01(2.84E-02)–	7.265E-01(2.13E-02)–	7.040E-01(2.72E-02)–	7.359E-01(2.00E-02)~	6.980E-01(1.71E-02)–	**7.667E-01(2.89E-02)+**	7.349E-01(2.47E-02)
	10	7.264E-01(1.23E-02)–	7.322E-01(2.96E-02)–	**7.507E-01(2.67E-02)+**	7.438E-01(1.65E-02)~	6.990E-01(1.20E-02)–	7.382E-01(2.41E-02)–	7.449E-01(2.69E-02)
	13	6.680E-01(4.01E-02)–	5.799E-01(5.04E-02)–	6.911E-01(2.25E-02)–	7.191E-01(1.77E-02)–	6.761E-01(1.88E-02)–	**7.495E-01(2.74E-02)+**	7.400E-01(3.90E-02)
	15	6.980E-01(3.00E-02)–	5.776E-01(5.79E-02)–	7.172E-01(2.25E-02)–	7.330E-01(2.24E-02)–	6.802E-01(1.85E-02)–	7.501E-01(2.31E-02)–	**7.535E-01(4.08E-02)**
Best/All		0/45	3/45	3/45	2/45	2/45	13/45	23/45
+/–/~		2/43/0	8/35/2	5/37/3	4/37/4	4/39/2	10/31/4	– –

MOEA/D-DU and SRA performed best in the cases of 10 objectives and 13 objectives, respectively, while EFR-RR was best in all the other cases. Concerning WFG3 with linear and unimodal PF, MaOEA/C performed best in all cases. For WFG4-WFG8 with concave PFs, MaOEA/C showed a superior performance over its competitors, as it performed best in more than half of all cases. Regarding WFG9 with a multi-modal and deceptive PF, MaOEA/C only obtained the best result in the case with 15 objectives, while SRA and θ-DEA were respectively best in the cases of 5 and 10 objectives. EFR-RR showed the best performance in all the other cases.

From the one-by-one comparisons in the last row of Table IV, MaOEA/C performed better than NSGA-III, MOEA/D-DU, θ-DEA, SRA, VaEA and EFR-RR in 43, 35, 37, 37, 39, and 31 out of 45 cases, respectively, while it was only outperformed by NSGA-III, MOEA/D-DU, θ-DEA, SRA, VaEA and EFR-RR in 2, 8, 5, 4, 4, and 10 cases, respectively. Thus, MaOEA/C was found to present a superior performance over its six competitors, in most instances of WFG1-WFG9.

3) *A Further Discussion and Analysis on MaOEA/C*

To further study the evolutionary behavior of all the compared MaOEAs, Fig. A.1 plots their evolutionary curves in the supplementary file due to page limitations, using the average HV values in all the 10-objective MaF and WFG test problems. The average HV values were recorded at an interval of 20 generations. The subfigures in Fig. A.1 confirm the advantages

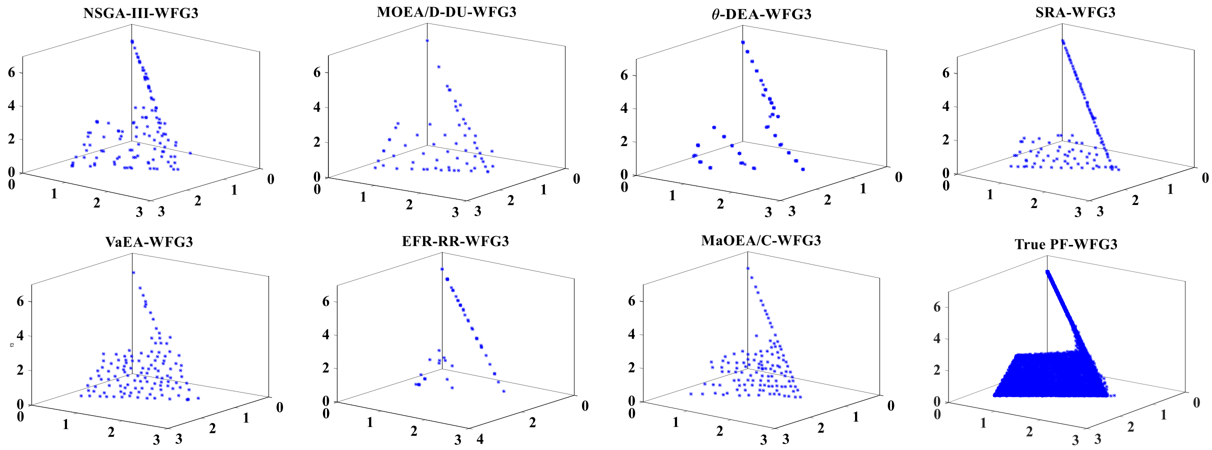


Fig. 3 The final solution sets achieved by seven MaOEAs and the true PF on 3-objective WFG3 problem

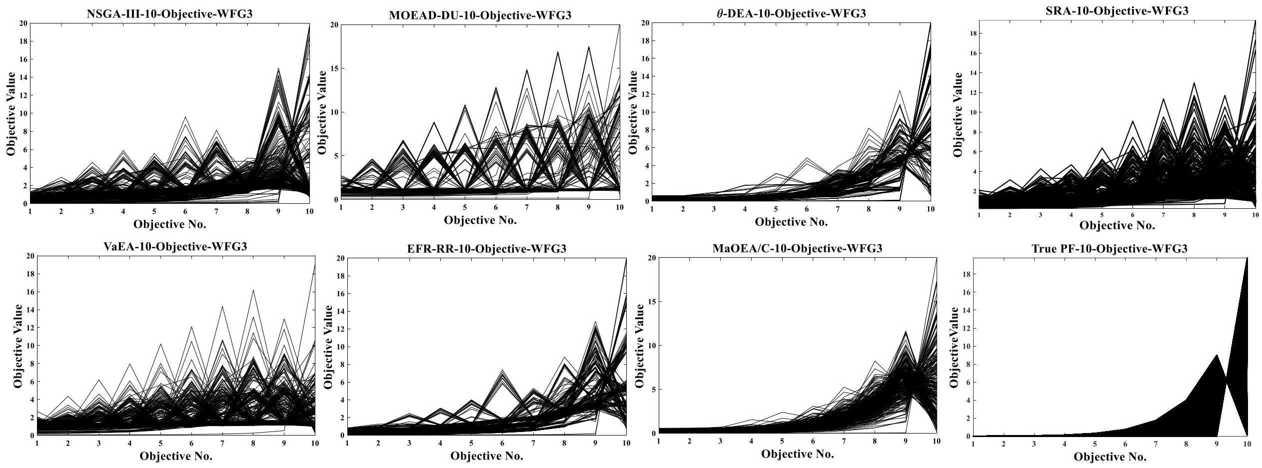


Fig. 4 The final solution sets achieved by seven MaOEAs and the true PF on 10-objective WFG3 problem, shown by parallel coordinates

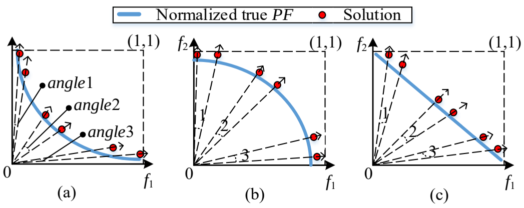
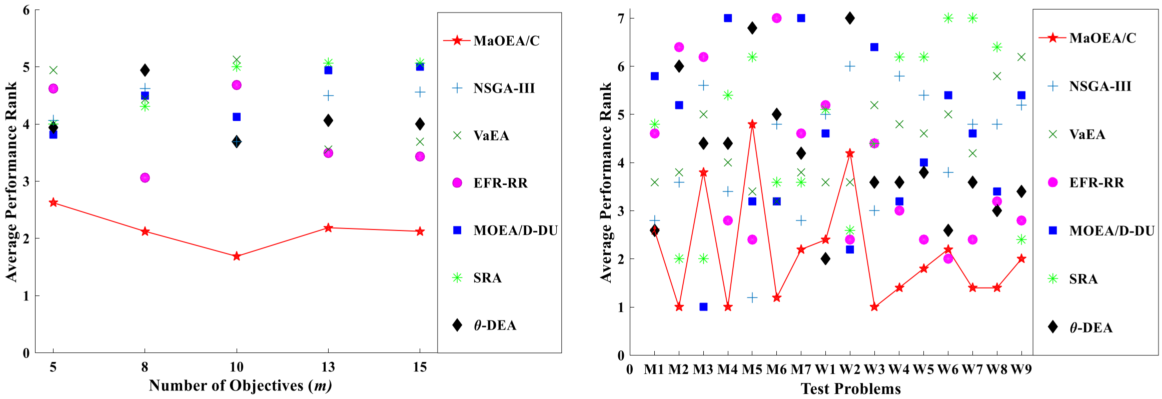


Fig. 2 The illustration of the angle between the boundary solutions and the intermediate solutions for the (a) convex (b) concave (c) linear PFs in the normalized bi-objective space

of MaOEA/C in providing a strong selection pressure towards the true PFs for problems with concave PFs, such as MaF2, MaF4, and WFG4-WFG9. However, our algorithm showed a slightly poor performance for problems with convex PFs, such as MaF3, MaF5, and WFG1-WFG2. Moreover, in problems with linear, degenerate or disconnected PFs, such as MaF1, MaF6, MaF7 and WFG3, MaOEA/C obtained a very promising performance.

Based on the above summary of comparisons, MaOEA/C performed better in problems with concave, degenerate or disconnected PFs, but was worse in problems with convex PFs. This phenomenon is mainly induced by the use of the vector angle to distinguish the similarities of individuals in the clusters using PCM and HCM. For problems with convex PFs, the vector angle seems not so fair to reflect the similarities of the individuals near the boundaries. To justify the above statement, three cases to compute the vector angles are provided in Fig. 2 with six individuals, respectively for problems with convex, concave and linear PFs. In Fig. 2(a) with a convex PF, *angle1* and *angle3* are much smaller than *angle2*. That is to say, the vector angles of two boundary individuals are much smaller than that of two intermediate individuals, even though they show a similarly good distribution on the PF. However, in Figs. 2(b) and 2(c) respectively with concave and linear PFs, the vector angle is very fair to estimate the distributions, regardless of the boundary individuals or the intermediate individuals. Therefore, the slightly worse performance of MaOEA/C in problems with convex PFs can be properly explained. On the other hand, for problems with degenerate, inverted or disconnected PFs, such as MaF1, MaF4, MaF6, MaF7 and WFG3, the reference point based MaOEAs (NSGA-III, MOEA/D-DU, θ-DEA, and EFR-RR) performed relatively poorly, since the pre-set reference points cannot properly match their PFs [13]. To visually show and support the above discussion results, some final solution sets with the median HV values from 30 runs were plotted in Figs. 3-4, respectively for WFG3 with a particularly irregular PF [67] in the cases of 3 objectives to better visualize the performance and of 10 objectives to show the solutions' distribution in high dimensional objective space. Due to page limitations, Figs. A.2 to A.9 are further provided in the supplementary file, respectively for MaF1 with an



(a) (b)

Fig. 5 (a) Average performance rank over all test problems with different number of objectives. (b) Average performance rank in terms of all objective dimensions for different test problems, namely MaF1-MaF7 (M1-M7) and WFG1-WFG9 (W1-W9)

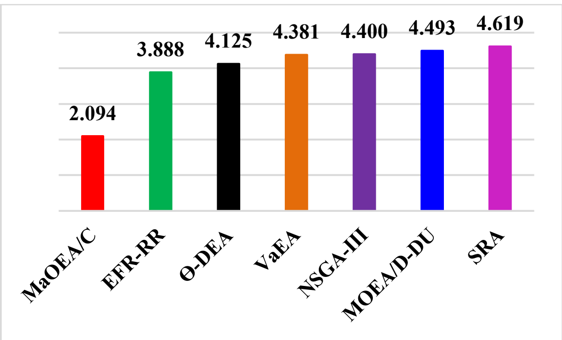


Fig. 6 Average ranking of Friedman's test for the compared MaOEAs

inverted PF, MaF5 with a badly scaled convex PF, MaF6 with a degenerated PF, and MaF7 with a disconnected PF in the cases of 3 objectives and of 10 objectives. It is worth noting that the same parameters settings from Section IV.B were used in these experiments, except that the population size was set to 120 and the maximum number of generations was set to 400 for the 3-objective problems.

Moreover, to quantify how well each algorithm performs overall, Friedman's test from the software tool KEEL [68] was used to rank the compared algorithms in all the test problems. Fig. 5 summarizes the average performance rank for different number of objectives in (a) and different test problems in (b). It is noted that the ranks of MaOEA/C are connected by a red line to easily observe the values. In Fig. 5(a), our algorithm showed obvious advantages on all the test problems with different numbers of objectives. It can also be confirmed from Fig. 5(b) that our algorithm achieved median performance in four test problems (MaF3, MaF5, WFG1 and WFG6) and the best performance in the other cases. Using Friedman's test, Fig. 6 further provides the average performance ranks in all the 80 test problems, which shows that the average performance rank (2.094) of MaOEA/C is much smaller than that of other competitors and thus confirms the superiority of MaOEA/C over other competitors when considering all the test problems. In addition, to show the significant differences of results among these compared algorithms, *p*-values obtained from Bonferroni-Dunn's and Holm's post hoc procedure in the software tool KEEL are listed in Table A.II of the supplementary file due to page limitations, where a *p*-value closer to 0 means the more significant differences on the results. In Table A.II, most *p*-values are very close to 0, indicating that MaOEA/C had a better performance with statistical significance when compared to its competitors.

At the review process of our paper, a number of competitive MaOEAs were published and thus more experimental studies were conducted by comparing MaOEA/C with four recently proposed MaOEAs, i.e., SPEA/R [27], MOEA/D-LWS [32], RVEA [33] and 1by1EA [47]. The parameters of these compared MaOEAs were set according to their references, while other experimental settings are the same as given in Section IV.B. Due to page limitations, their HV comparison results on tackling the WFG and MaF test problems with 5, 10, and 15 objectives are given in Table A.III of the supplementary file. From Table A.III, MaOEA/C showed a superior performance on most cases, as it was best on 27 out of 45 cases, while SPEA/R, MOEA/D-LWS, RVEA, and 1by1EA were respectively best on 8, 3, 4, and 6 cases. Moreover, two engineering problems from [26], [44] (Crashworthiness design of vehicles and modified Car design of side-impact) were used to validate the effectiveness of MaOEA/C on coping with real-life MaOPs, which are provided in Table A.IV and Section 2 of the supplementary file due to page limitations.

Overall, the superior performances of MaOEA/C on various test MaOPs and real-life MaOPs are validated. As introduced in Section III, its advantages are mainly brought by the used clustering methods (PCM and HCM) to maintain diversity and the convergence indicator in (6) to ensure convergence.

D. More Discussions about the Clustering-based MaOEAs

1) Comparison Results on Different Clustering Methods

Here, in order to verify the efficiency of our two-step clustering strategy, two MaOEA/C variants were designed by only embedding a single clustering method, i.e., the HCM-based MaOEA (MaOEA/C-H) and the density clustering method (DCM) [69] based MaOEA (MaOEA/C-D). MaOEA/C-H and MaOEA/C-D respectively used HCM and DCM to classify the union population into N clusters in their environmental selection mechanism. Due to page limitations, the details of DCM are provided in Section 3 of the supplementary file. To have a fair comparison, other components of MaOEA/C-H and

TABLE V
SUMMARY OF THE SIGNIFICANCE TEST BETWEEN MAOEA/C AND OTHER TWO CLUSTERING-BASED MAOEAS

MaOEA/C	vs.	MaOEA/C-H	MaOEA/C-D
On MaF Problems (m=5,10,15)	*Better*	7	11
	Worse	6	3
	Similar	8	7
On WFG Problems (m=5,10,15)	*Better*	6	18
	Worse	10	6
	Similar	11	3

MaOEA/C-D were kept the same as in the original MaOEA/C, also running 30 times with the same settings in Tables I and II.

The detailed HV results are listed in Tables A.V-A.VI of the supplementary file. Here, Table V provides a summary of the significance test on the HV results for all the MaF and WFG test problems with m = 5, 10, 15, where "*Better*", "*Worse*" and "*Similar*" respectively indicate the number of test problems in which the performance of MaOEA/C is better than, worse than, and similar to that of its competitors. Fig. 7 further shows their average running times (in seconds: s) in all the test problems with 10 objectives. When compared to MaOEA/C-D in Table V, MaOEA/C was better in more than half of the test problems adopted, which confirms that the adopted clustering methods (PCM and HCM) are more effective for tackling MaOPs. Also, Fig. 7 shows the superiority of MaOEA/C over MaOEA/C-D regarding the running time. When compared to MaOEA/C-H in Table V, MaOEA/C obtained statistically similar results in 8 instances of the MaF problems and in 11 instances of the WFG problems. Moreover, MaOEA/C was slightly better than MaOEA/C-H in the MaF test problems, but slightly worse than MaOEA/C-H in the WFG test problems. These comparisons indicate that even the use of a single clustering method such as HCM adopted in the environmental selection mechanism is effective to maintain the population's diversity, so as to provide an overall competitive performance. However, when considering the running times in Fig. 7, MaOEA/C has a much lower computational complexity, mainly due to the use of a two-step clustering strategy (PCM and HCM).

To summarize, when considering the performance and computational complexity of clustering-based MaOEAs, it is advisable to run a two-step clustering strategy (PCM and HCM) in our algorithm.

2) Computational Complexity Analysis of MaOEA/C

The computational complexity of MaOEA/C in one generation is mainly dominated by the environmental selection that is described in **Algorithm 3**. **Algorithm 3** requires a time complexity of $O(mN)$ (m is the number of objectives and N is the population size) to obtain the union population $\boldsymbol{U}$ in line 1, to normalize each solution of $\boldsymbol{U}$ in line 2 and to get the convergence indicator value for each solution of $\boldsymbol{U}$ in line 3. In line 4, it needs a time complexity of $O(m^2N)$ to run **Algorithm 1**. In the loop for each main cluster C_i^{PCM} in lines 6-23, the time complexity is mainly determined by the non-dominated sorting procedure in line 7 and the procedures of **Algorithm 2** in line 12. As it requires a time complexity of $O(N\log^{m-2}k)$ [64] to run the non-dominated sorting for $2k$ individuals ($k = N/m$) in line 7 and an approximate time complexity of $O(m^3k^2)$ to

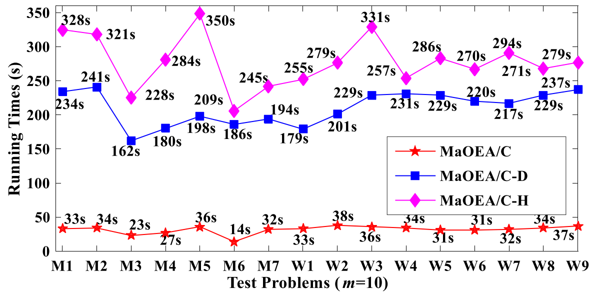


Fig. 7 The average running times of three clustering-based MaOEAs on MaF1-MaF7 (M1-M7) and WFG1-WFG9 (W1-W9) with 10 objectives

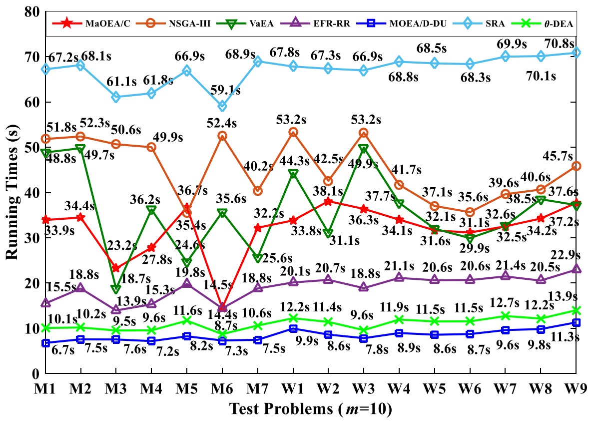


Fig. 8 The average running times of the selected seven MaOEAs on MaF1-MaF7 (M1-M7) and WFG1-WFG9 (W1-W9) with 10 objectives

run **Algorithm 2** in line 12, the overall worst time complexity of MaOEA/C is $\max\{O(N\log^{m-2}k), O(m^3k^2)\}$ $(k = N/m)$ in one generation, i.e., $O(mN^2)$, which shows comparable time complexities with most MaOEAs [44].

To evaluate the actual runtime of all the compared MaOEAs, their average running times (in seconds: s) from 30 runs are plotted in Fig. 8, for all the MaF and WFG test problems with 10 objectives. Obviously, in Fig. 8, MOEA/D-DU showed the fastest speed due to the use of a simple decomposition framework, while SRA had the slowest speed due to the sorting of two performance indicators. Although the two clustering methods (PCM and HCM) were used in MaOEA/C, it was still faster than NSGA-III and mostly similar to VaEA. θ-DEA and EFR-RR also seemed faster than MaOEA/C. Therefore, the adopted clustering methods will not significantly slow down the running time of MaOEA/C, but shows a comparable performance in their actual running times.

E. More Discussions about the Similarity-based Mating Selection and the Convergence Indicators

Here, in order to study the impact of ε (the probability to select the parents in the same cluster) in the mating selection mechanism, Table VI gives average rank of Friedman's test for MaOEA/C with different ε values from {0.0, 0.2, 0.5, 0.8, 1.0}, while the details of their HV results are provided in Tables A.VII-A.VIII of the supplementary file due to page limitations. From Table VI, it is confirmed that ε=0.8 in our setting is reasonable and more effective, as it obtained the overall best performance when considering all the test prob-

TABLE VI
AVERAGE RANKING FOR MAOEA/C WITH DIFFERENT ε VALUES

Objectives	ε=0.0	ε=0.2	ε=0.5	ε=0.8	ε=1.0
m=5	3.15	3.0188	2.925	**2.775**	3.1312
m=10	3.6562	3.4688	3.125	**1.875**	2.875
m=15	3.875	3.4375	2.6562	2.6875	**2.3438**
All	3.3958	3.3438	2.875	**2.5625**	2.8229

TABLE VII
SUMMARY OF THE SIGNIFICANCE TEST FOR MAOEA/C
WITH DIFFERENT CONVERGENCE INDICATORS

MaOEA/C vs.	MaOEA/C-	EdI	EdN	CdI	ASF
On MaF Problems (*m*=5,10,15)	*Better*	7	6	8	5
	Worse	3	3	5	6
	Similar	11	12	8	10
On WFG Problems (*m*=5,10,15)	*Better*	23	13	24	19
	Worse	2	3	1	1
	Similar	3	11	2	7

lems adopted. Similarly, in [27], it was claimed that a proper similarity-based mating selection can benefit reproduction to enhance performance of MaOEAs.

Moreover, to study the impact of the convergence indicator in our algorithm, four indicators, i.e., the Euclidean distance to an ideal point (EdI) [47], the Euclidean distance to the Nadir point (EdN) [47], the Chebyshev distance to the ideal point (CdI) [47], and the modified achievement scalarizing function (ASF) [46], were used in MaOEA/C to compare with the indicator in (6). Table VII gives the summary of the significance test for MaOEA/C with different convergence indicators, while the details of their HV results are provided in Tables A.IX-A.X of the supplementary file due to page limitations. As observed from Table VII, MaOEA/C with different convergence indicators performed competitively on most of MaF test problems, while the indicator in (6) showed some advantages on most of the WFG test problems. Thus, the use of (6) as our convergence indicator in MaOEA/C is reasonable and more effective.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a novel clustering-based evolutionary algorithm was presented for MaOPs, i.e., MaOEA/C. This algorithm mainly relies on the use of clustering methods (PCM and HCM) to classify N clusters, which helps to portray the population's distribution. Then, one individual from each of the clusters that are respectively close to the m axis vector is selected to maintain diversity, while one individual from each of the remaining clusters is chosen using a convergence indicator. This way, diversity and convergence can be properly balanced in MaOEA/C. Moreover, in the mating selection mechanism, the parent population is also divided into m clusters using PCM and then the parents for applying recombination are selected from the same cluster with a high probability, aiming to encourage exploitation. When compared to six competitive MaOEAs (NSGA-III, MOEA/D-DU, EFR-RR, θ-DEA, SRA, and VaEA), MaOEA/C showed its superiority, especially in some problems with incomplete and irregular PFs. Moreover, the experiments validated the effectiveness and efficiency of the two-step clustering strategy in MaOEA/C, which has a comparable time complexity with most of the compared MaOEAs, even though the execution of two clustering methods is often costly. At last, the effect of parameter ε in mating selection and the effectiveness of other convergence indicators in environmental selection were also experimentally studied in this paper.

In our future work, the use of vector angles to reflect the similarities of individuals will be further studied, with the aim of alleviating the difficulty in tackling problems with convex PFs. Moreover, the application of MaOEA/C to some real-life problems will be conducted as part of our future work.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, Ma, USA: Kluwer Academic, 1999.

[2] 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, JUNE. 2017.

[3] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: objective space reduction and diversity improvement," *IEEE Trans. Evol. Comput.,* vol. 20, no. 1, pp. 145-160, 2016

[4] 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.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des.*, *Optimisation Control.*, pp. 95–100, 2002.

[6] 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.

[7] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," *Parallel Problem Solving from Nature - PPSN VIII*, *Lecture Notes in Computer Science*, vol. 3242, pp. 832–842, 2004.

[8] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

[9] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimization: an exploratory analysis," in Proc. of the 2003 *IEEE Congress on Evolutionary Computation*, Canberra, Australia, pp. 2066-2073, 2003.

[10] E. Hughes, "Evolutionary many-objective optimization: many once or one many?" in Proc. of the 2005 *IEEE Congress on Evolutionary Computation*, Edinburgh, UK, pp. 222-227, 2005.

[11] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: a study in differential evolution," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 400–413, 2015.

[12] Y. Tian, X. Zhang, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, 2015.

[13] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective Knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, 2015.

[14] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, pp. 13:1–13:35, Sep 2015.

[15] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169-190, 2017.

[16] M. Li, S. Yang, and X. Liu, "Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, 2016.

[17] S. Jiang, J. Zhang, "A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm," *IEEE Trans. Cybernetics.,* vol. 45, no. 10, pp. 2202-2213, 2015.

[18] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume based many-objective optimization," *Evolutionary Computation*, vol.19, no. 1, pp. 45-76, 2011.

[19] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, 2014.

[20] H. Wang, Y. Yao "Corner Sort for Pareto-Based Many-Objective Optimization," *IEEE Trans. Evol. Cybernetics.*, vol. 44, no. 1, pp. 92-102, 2014.

[21] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180-198, April 2016.

[22] Y. Yuan, H. Xu, B. Wang, and X. Yao, “A new dominance relation based evolutionary algorithm for many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16-37, February 2016.

[23] M. Elarbi, S. Bechikh, A. Gupta, L. Ben, and Y.S. Ong, “A New Decomposition-Based NSGA-II for Many-Objective Optimization,” *IEEE Trans. on Systems, Man, and Cyberentics: Systems*, in press, **doi:** 10.1109/TSMC.2017.2654301, 2017.

[24] C. Zhu, L. Xu, and E. D. Goodman, “Generalization of Pareto-optimality for many-objective evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 299-315, 2016.

[25] M. Li, S. Yang, X. Liu, “Shift-based density estimation for Pareto-based algorithms in many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[26] 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.

[27] S. Y. Jiang, S. X. Yang, “A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multi-objective and Many-objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329-346, 2017.

[28] K. Li, K. Deb, Q.F. Zhang, and S. Kwong, “An evolutionary many-objective optimization algorithm based on dominance and decomposition,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, 2015.

[29] M. Asafuddoula, T. Ray, and R. Sarker “A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 19, no.3, pp.445 - 460, 2015.

[30] H. Sato, “Chain-reaction solution update in MOEA/D and its effects on multi- and many-objective optimization,” *Soft Comput.,* vol. 20, pp. 3803-3820, 2016.

[31] X. Cai, Z. X. Yang, Z. Fan, Q. F. Zhang, “Decomposition-Based-Sorting and Angle-Based-Selection for Evolutionary Multiobjective and Many-Objective Optimization,” *IEEE Trans. Evol. Cybernetics.*, vol. 47, no. 9, pp. 2824-2837, 2017.

[32] 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.

[33] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, “A Reference Vector Guided Evolutionary Algorithm for Many-objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773-791, 2016.

[34] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, “A Surrogate-assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129-142, 2018.

[35] F. Q. Gu, Y. M. Cheung, “Self-organizing Map-based Weight Design for Decomposition-based Many-objective Evolutionary Algorithm,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 211-225, 2018.

[36] K. Bringmann and T. Friedrich, “An efficient algorithm for computing hypervolume contributions,” *Evol. Comput.*, vol. 18, no. 3, pp. 383–402, 2010.

[37] L. While, L. Bradstreet, and L. Barone, “A fast way of calculating exact hypervolumes,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, Feb. 2012.

[38] L. M. Russo and A. P. Francisco, “Quick hypervolume,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 481–502, Aug. 2014.

[39] D. Brockhoff, T. Wagner, and H. Trautmann, “On the properties of the R2 indicator,” in Proc. 14th *Annu. Conf. Genet. Evol. Comput.*, Philadelphia, PA, USA, pp. 465–472, 2012.

[40] R. H. Gomez and C. A. C. Coello, “MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator,” in Proc. *IEEE Congr. Evol. Comput.*, Cancún, Mexico, pp. 2488–2495, 2013.

[41] M. Wagner and F. Neumann, “A fast approximation-guided evolutionary multi-objective algorithm,” in Proc. 15th *Annu. Conf. Genet. Evol. Comput. Conf.*, Amsterdam, The Netherlands, pp. 687–694, 2013.

[42] H. Wang, L. Jiao and X. Yao, “Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization”, *IEEE Trans. Evol. Comput.*, vol. 19, no. 4,pp. 524–541, 2015.

[43] B.D. Li, K. Tang, J.L. Li, and X. Yao, “Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators”, *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924-938, 2016.

[44] Y. Xiang, Y. R. Zhou, M. Q. Li, “A Vector Angle based Evolutionary Algorithm for Unconstrained Many-Objective Optimization,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131-152, 2017.

[45] J. Cheng, G. Yen, G. Zhang, “A Many-Objective Evolutionary Algorithm With Enhanced Mating and Environmental Selections,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 592-605, 2015.

[46] Z. N. He, G. G. Yen, “Many-Objective Evolutionary Algorithm Based on Coordinated Selection Strategy,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220-233, 2017.

[47] Y. Liu, D. Gong, J. Sun, and Y. Jin, “A Many-Objective Evolutionary Algorithm Using A One-by-One Selection Strategy,” *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689-2702, 2017.

[48] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, X. Yao, “A benchmark test suite for evolutionary many-objective optimization,” *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 67-81, 2017.

[49] E. Zitzler, and T. Lothar, " Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

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

[51] H. Zhang, S. Song, A. Zhou, and X.-Z. Gao, “A clustering based multiobjective evolutionary algorithm,” *IEEE Congress on Evolutionary Computation* (CEC), pp. 723–730, 2014.

[52] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong， “A multiobjective evolutionary algorithm based on decision variable analyses for multi-objective optimization problems with large scale variables,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, 2016.

[53] Y. Hua, Y. Jin, K. Hao, “A Clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts,” *IEEE Trans. Evol. Cybernetics.*, doi: 10.1109/TCYB.2018.2834466, in press, 2018.

[54] R. Denysiuk, C. Lino, and E. S. Isabel, "Clustering-based selection for evolutionary many-objective optimization." *International Conference on Parallel Problem Solving from Nature*. Springer, Cham, 2014.

[55] O. R. Castro, A. Pozo, J. A. Lozano, and R. Santana, "An investigation of clustering strategies in many-objective optimization: the I-Multi algorithm as a case study." *Swarm Intelligence*, vol.11, no. 2, pp. 101-130, 2017.

[56] S. Huband, L. Barone, R. while, and P. Hingston, “A scalable multi-objective test problem tookit,” in *Proc. 3rd Conf. Evol .Multi-Criterion Optimiz., Lecture Notes in Computer Science*, vol. 3410, pp. 280–295, 2005.

[57] M. J. Hossain, M. S. Alouini, and V. Bhargava, “Two-User Opportunistic Scheduling Using Hierarchical Modulations in Wireless Networks with Heterogenous Average Link Gains,” *IEEE Trans. Evol. Comput.*, vol. 58, no. 3, pp. 880-889, 2010.

[58] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, no. 14, pp. 281–297,1967.

[59] A. J. Hartigan, “Clustering Algorithms,” *Applied Statistics*, 23(6), pp.38-41, 1975.

[60] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[61] M. Makrehchi, “Hierarchical Agglomerative Clustering Using Common Neighbours Similarity,” *IEEE/WIC/ACM International Conference on Web Intelligence*, doi: 10.1109/WI.2016.0093, pp. 546-551, 2016.

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

[63] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, “Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point,” in Proc. 10th *ACM SIGEVO Workshop Found. Genet. Algorithms*, Orlando, FL, USA, pp. 87–102, 2009.

[64] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, “Manyobjective test problems to visually examine the behavior of multiobjective evolution in a decision space,” in Proc. *Int. Conf. Parallel Prob. Solv. Nat., Krakow, Poland*, pp. 91–100, 2010.

[65] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” *Evolutionary Multiobjective Optimization, ser. Advanced Information and Knowledge Processing*, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer London, pp. 105–145, 2005.

[66] M. T. Jensen, “Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 5, pp. 503-515, October 2003.

[67] H. Ishibuchi, H. Masuda, Y. Nojima, “Pareto Fronts of Many-Objective Degenerate Test Problems,” *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 807-813, 2016.

[68] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrel, J.Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F. Herrera, “KEEL: a software tool to assess evolutionary algorithms for data mining problems,” *Soft Comput*., vol. 13, pp. 307–318, 2009.

[69] A. Rodriguez, A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492-1496, 2014.

[70] H. Ishibuchi, R. Imada, Y. Setoguchu, and Y. Nojima, “Reference point specification in hypervolume calculation for fair comparison and efficient search,” *Genetic and Evolutionary Computation Conference*, pp. 585-592, 2017.

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 a Lecturer 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.

Songbai Liu received the B.S. degree from ChangSha University, ChangSha, China, in 2012, and worked for ShenZhen TVT Digital Technology Co., Ltd as a software QA engineer from 2013 to 2015. He received the M. S. degree from Shenzhen University, China, in 2018.

He is currently a Research Assistant in College of Computer Science and Software Engineering, Shenzhen University. His current research interests include nature-inspired computation, many-objective optimization, and machine learning.

Ka-Chun Wong received his B.Eng. in Computer Engineering from The Chinese University of Hong Kong in 2008. He has also received his M.Phil. degree at the same university in 2010. He received his PhD degree from the Department of Computer Science, University of Toronto in 2015. After that, he assumed his duty as assistant professor at City University of Hong Kong. His research interests include Bioinformatics, Computational Biology, Evolutionary Computation, Data Mining, Machine Learning, and Interdisciplinary Research.

Maoguo Gong (M'07-SM'14) received the B.S. degree and Ph.D. degree 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 as a Full Professor, respectively, both with exceptive admission. His research interests are in the area of computational intelligence with applications to optimization, learning, data mining and image understanding.

Dr. Gong received 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 the associate editor of *IEEE Trans. Evolutionary Computation.*

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.

Jianyong Chen is a professor in College of Computer Science and Software Engineering, Shenzhen University. He got his PhD from City University of Hong Kong, Hong Kong, China, in 2003. He is interested in Artificial Intelligence and Information Security.

He worked for ZTE Corporation as senior engineer of network technology from 2003 to 2006. After that, he joined the Shenzhen University. He was vice-chairman of International Telecommunication Union-Telecommunication (ITU-T) SG17 from 2004 to 2012, and editor of three recommendations developed in ITU-T SG17. He has published more than 30 papers and got more than 30 patents in the field of Artificial Intelligence and Information Security.

Jun Zhang (M'02–SM'08–F'17) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2002.

He is currently a Changjiang Chair Professor with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Guangzhou, China. His current research interests include computational intelligence, cloud computing, data mining, and power electronic circuits. He has published over 200 technical papers in the above areas.

Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS.