

RESEARCH ARTICLE

A Multi-Objective Evolutionary Algorithm With Hierarchical Clustering-Based Selection

SHENGHAO ZHOU¹, ZE CHEN², QI LI¹, MENGJUN GU³, ZHOUCHEG BAO¹, WENDA HE¹,
AND WEIGUO SHENG¹, (Member, IEEE)

¹School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China

²Engineering Research Center of Mobile Health Management System, Hangzhou Normal University, Hangzhou 311121, China

³China Telecom Corporation Ltd., Zhejiang Branch, Hangzhou 310022, China

Corresponding author: Weiguo Sheng (w.sheng@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873082 and Grant 62003121, in part by the Scientific Research Fund of Zhejiang Provincial Education Department, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ20F030014.

ABSTRACT This paper proposes an evolutionary algorithm with hierarchical clustering based selection for multi-objective optimization. In the proposed algorithm, a hierarchical clustering is employed to design the environmental and mating selections, named local coverage selection and local area selection, respectively, for multi-objective evolutionary algorithm. The local coverage selection strategy aims to preserve well-distributed individuals with good convergence. While, the local area selection strategy is devised to deliver a balanced evolutionary search. This is achieved by encouraging individuals for exploration or exploitation according to the $I_{\epsilon}+$ indicator. In both strategies, a hierarchical clustering method is employed to discover the population structure. Based on the clustering results, in local coverage selection, the individuals of different clusters will be retained according to their coverage areas and crowding distances, such that distributing as evenly as possible in the Pareto front. In local area selection, the individual(s) with the best value of $I_{\epsilon}+$ indicator in each cluster will be selected to perform mating, with the purpose of achieving a balanced exploration and exploitation. The proposed algorithm has been evaluated on 26 bench-mark problems and compared with related methods. The results clearly show the significance of the proposed method.


INDEX TERMS Multi-objective evolutionary algorithm, multi-objective optimization problem, hierarchical clustering.

I. INTRODUCTION

Multi-objective optimization problems (MOPs) refer to the problems with more than two conflicting objectives. They are commonly seen in the real world, e.g., electrical engineering [1], scheduling [2], engineering modeling [3] and transport engineering [4]. The MOPs can be formulated as:

$$\begin{cases} \text{minimize } F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{subject to } x \in X \end{cases} \quad (1)$$

where x is the decision vector, X is the decision space and m is the number of objectives [5]. Due to the conflicting nature of different objectives in MOPs, there is no single solution

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato .

that can optimize all the objectives simultaneously. Instead, a set of tradeoff solutions, which cannot be improved in any objective without degenerating at least one other objective, can be obtained. These tradeoff solutions are called Pareto optimal solutions. The collection of all Pareto optimal solutions is called Pareto optimal set (PS), and the image of projection of PS in the objective space is called Pareto optimal front (PF).

Over the last decades, a number of multi-objective evolutionary algorithms (MOEAs) have been proposed, which have demonstrated good performance in dealing with MOPs [6]. Generally, MOEAs can be classified into three categories. The first category is called Pareto-based MOEAs [7], which utilizes the non-dominated sorting mechanism along with a diversity maintenance scheme to select candidates.

In this approach, when the proportion of non-dominated solutions increases, the Pareto dominance could fail to provide sufficient selection pressure, resulting in poor population diversity and convergence. In recent years, a few domination principles have also been proposed to relax the comparison criteria, so that feasible solutions can be compared with each other. Examples methods include ϵ -domination principle [8] and fuzzy Pareto dominance [9]. It is worth noting that the performance of these methods depends on their parameter settings. Consequently, although the diversity and convergence can be improved, the final non-dominated solutions provided by these methods may not cover the entire PF uniformly [10].

The indicator-based MOEAs fall in the second category. It adopts an indicator (e.g., hypervolume (HV) [11], inverted generational distance (IGD) [12] and R2 [13]) to measure both convergence and diversity. These indicators, however, may prefer some specific regions of PF, causing certain optimal solutions to disappear during evolution. As a result, final solutions may not be evenly distributed along the PF. For example, in the HV indicator based MOEAs [14], [15], the candidate solutions could be biased to distribute in the middle of a convex/concave PF [16].

The decomposition-based MOEAs tend to decompose the original MOP into a number of single-objective optimization problems or simpler MOPs to be solved in a collaborative manner [17], [18]. In these methods, since the predefined weight vectors are uniformly sampled on the unit hyperplane, the distribution of candidate solutions in the middle of convex/concave PFs will be more/less crowded than those on the border. The situation could become even worse when the PF has a sharp peak or long tail [19]. Further, as the distribution of candidate solutions is mostly determined by the predefined weight vectors, the difference between PF shape and distribution of weight vectors could also lead to a substantial deterioration of the performance of decomposition-based MOEAs [20].

To alleviate the above issues, a clustering-based environmental selection (named local coverage selection) along with a clustering-based mating selection (named local area selection) have been proposed and incorporated into an evolutionary algorithm for multi-objective optimization. The main contributions of this study are two-fold:

- A local coverage selection strategy, which is designed to preserve a group of well-distributed individuals with good convergence. In this strategy, a non-dominated fast sorting is first employed to eliminate individuals, which are far away from the Pareto front. Then, the population is divided into clusters by a hierarchical clustering method. The number of individuals in each cluster, which will be retained, is determined based on the coverage area of each cluster. These individuals are subsequently selected based on the crowding distance, so that the individuals in the same cluster can be evenly distributed. Consequently, the local coverage selection

strategy can be used to preserve evenly distributed individuals in Pareto front.

- A local area selection strategy, which is designed to deliver a balanced evolutionary search. In this strategy, the population is divided into clusters by a hierarchical clustering method. The individuals are selected for exploration or exploitation according to the $I_\epsilon+$ indicator, with the purpose of achieving a balanced exploration and exploitation.

The proposed algorithm has been evaluated on widely used bench-mark problems and compared with related methods. The experimental results clearly show the significance of our devised strategies. The results also show that the proposed algorithm generally outperforms the related methods to be compared.

The remainder of this paper is structured as follows. Section II introduces related work. Section III describes the proposed algorithm including a detailed explanation of the proposed local coverage selection and local area selection strategies. The experimental setting and results are reported and analyzed in Section IV. Section V provides our conclusion.

II. RELATED WORK

In literature, many MOEAs have been proposed. For example, an adaptive reference vector generation approach for inverse model-based MOEA [21] was developed for problems with degenerated and disconnected Pareto fronts. In this method, at the early stage of evolution, a reference vector in the crowded area is replaced by a randomly generated vector. Since the vector for replacement is created randomly, this method may have difficulty to guarantee an even distribution of the obtained solutions. A variant of MOEA/D that adjusts weight vectors, termed as MOEA-ABD, was proposed in [22] for bi-objective optimization problems with discontinuous Pareto fronts. This method detects the discontinuous part of Pareto front by calculating the deviation between the weight vector and its corresponding normalized objective function vector, and adjusts the number of weight vectors according to the length of each continuous part. Some other variants of MOEA/D have also been proposed, such as an improved MOEA/D with a two-phase strategy and a niche-guided scheme, called MOEA/D-TPN [23], MOEA/D with sorting and selection [24], AMOEA/D [25] with an auto-switching strategy based on the aggregation function enhancement, and angle-based adaptive penalty (AAP) scheme for MOEA/D [26]. These methods, however, still subject to drawback due to employing a fixed weight vector set. A fast method for pruning non-dominated solutions in many-objective problems, known as MOEA/D-AWA, was introduced in [19]. This method first calculates the sparsity level of each individual using a vicinity distance. Then, overcrowded subproblems are deleted while new subproblems are added into the sparse regions. This method is helpful to reduce invalid subproblems at the cost of introducing extra

computation time and cannot guarantee an even distribution of the solutions. In [27], a reference vector guided evolutionary algorithm (RVEA) [27] was proposed. In this method, predefined reference vectors are adjusted at certain generation using the Hadamard product [28] to adapt the shape of front. In [27], a variant of RVEA, termed as RVEA*, was proposed. In this method, if no individual is associated with a reference vector, it will be replaced by a randomly generated vector. Consequently, it is difficult to guarantee an even distribution of the vectors. In [29], an improved version of A-NSGA-III [30], termed as A²-NSGA-III [29], was introduced to address several limitations of A-NSGA-III. As in A-NSGA-III, reference points, which have no individual to be associated will be deleted and relocated around a reference point associated with more than two individuals. This method also requires relocation of reference points, which may lead to an uneven distribution of the solutions. In MOEA/D-SCS [31], the evolutionary process is separated into three stages, where different indicators are utilized to screen elite and inferior solutions. However, it is difficult to distinguish early, middle, and late stages with fixed values for different types of MOPs.

Clustering algorithms have been previously employed to improved MOEAs [32], [33]. These methods generally employ clustering to adapt reference vectors or reference points. For example, the clustering-ranking method for many-objective optimization (termed as crEA) [32] employed a series of predefined reference lines to cluster individuals into subregions and the individual nearest to the Pareto front in each cluster will be selected. Since the reference lines are fixed, by employing clustering approach, it is difficult to adapt to MOPs with irregular Pareto fronts. A clustering-based MOEA (termed as CLUMOE) [34] tended to adopt the k -means clustering algorithm to divide individuals of population into clusters and individuals in the same subproblem are allowed to perform crossover, thereby accelerating the convergence. However, the limitations of k -means, such as sensitivity to initial cluster centers and assuming spherical data distribution, could reduce the performance of CLUMOE. An evolutionary many-objective optimization algorithm with clustering-based selection (termed as EMYO/C) [33] tried to first calculate the sum of Euclidean distance between the individuals of two clusters and then merge those clusters, which have the minimum sum value.

III. PROPOSED ALGORITHM

In this section, we propose an evolutionary algorithm with hierarchical clustering-based selection (HCCA) for multi-objective optimization. During the process of method, firstly, two populations, DP and PP , with N individuals are randomly sampled from the search space to form initial populations. Then, individuals are randomly selected from the DP to produce offspring, in which the differential evolution (DE) [35] and polynomial-based mutation (PM) [36] are adopted. For PP , in the proposed local area selection (LAS) strategy, simulated binary crossover (SBX) [37] and polynomial-based

Algorithm 1 Evolutionary Algorithm With Hierarchical Clustering Based Selection

Input: Population size N , total number of function evaluations max_{ev} , the number of objectives m and the number of variables D

Output: Final population P

```

1:  $[DP, PP, W, Z] \leftarrow initialization(N, D, m)$ 
2:  $ev \leftarrow 0, K \leftarrow 0.5N, M \leftarrow \{\}$ 
3: while  $ev < max_{ev}$  do
4:    $M \leftarrow$  randomly select  $N - K$  solutions from  $DP$ 
5:   for  $i \leftarrow 1$  to  $|M_{DP}|$  do
6:     Offspring  $\leftarrow$  crossover and mutation ( $M_i, DP$ )
7:      $Q_1 \leftarrow Q_1 \cup$  Offspring
8:   end for
9:    $[K, Q_2] \leftarrow$  local area selection( $[PP \cup Q_1], K$ )
10:   $PP \leftarrow PP \cup Q_1 \cup Q_2$ 
11:   $DP \leftarrow DP \cup Q_1 \cup Q_2$ 
12:   $[DP, Z] \leftarrow$  environment selection ( $DP, W, Z$ )
13:   $PP \leftarrow$  local coverage selection( $PP$ )
14: end while
15:  $P \leftarrow$  crowding degree strategy( $PP, DP$ )

```

mutation (PM) are employed to generate offspring. Subsequently, a decomposition-based criterion and the proposed local coverage selection (LCS) strategy are performed to select N individuals for DP and PP , respectively. It should be noted that the decomposition-based criterion used in our method adopts the Tchebycheff method [17]. Finally, at the end of evolution, the crowding degree strategy, as presented in [38], will be implemented to select a group of individuals from DP and PP as the final output P . The main loop of HCCA is shown in Algorithm 1.

In the following subsections, we shall describe the details of proposed LAS and LCS strategies in the proposed algorithm.

A. LOCAL AREA SELECTION STRATEGY

Mating selection aims at selecting a group of parents for producing offspring. Typically, parents with high-quality performance would have high chances to produce offspring. The traditional Pareto dominance criterion, which have been adopted in existing methods, may fail to discriminate the convergence degrees of individuals. In addition, mating strategies in existing methods generally emphasize convergence and ignore population diversity [39]. To improve the situation, here, a local area selection strategy is proposed.

In LAS, a convergence fitness $Fitness(x)$ based on a convergence indicator (indicator $I_{\epsilon+}$ in IBEA [40]) is introduced to measure the convergence performance of an individual, which is defined as:

$$Fitness(x_1) = \sum_{x_2 \in PP \setminus \{x_1\}} -e^{I_{\epsilon+}(x_2, x_1)/0.05} \quad (2)$$

where x_1 and x_2 are individuals from the population PP and m is the number of objectives. The convergence fitness

Algorithm 2 Local Area Selection Strategy

Input: Population PP , number of individuals to be generated K and the size of population N

Output: Offspring pool $OFFP$

- 1: $Q \leftarrow \emptyset, M \leftarrow \emptyset$
- 2: $(F_1, F_2, \dots, F_t) \leftarrow \text{non-dominated-sorting}(PP)$
- 3: **for** $j \leftarrow 1$ to t **do**
- 4: $Q \leftarrow Q \cup F_j$
- 5: **if** $|Q| \geq N$ **then**
- 6: **break**
- 7: **end if**
- 8: **end for**
- 9: $AF \leftarrow$ calculate the convergence fitness of each individual in the population PP according to Eq. 2
- 10: $C_1, C_2, \dots, C_K \leftarrow$ divide individuals in population Q into K clusters by a hierarchical clustering algorithm
- 11: $OFFP \leftarrow \emptyset$
- 12: $OP \leftarrow \emptyset$
- 13: **for** $i \leftarrow 1$ to K **do**
- 14: $OP \leftarrow$ select the individual with the best convergence fitness from C_i
- 15: **if** $|C_i| \geq 2$ **then**
- 16: $[x_1, x_2] \leftarrow$ select two individuals with the best convergence fitness from C_i
- 17: Offspring \leftarrow crossover and mutation(x_1, x_2)
- 18: $OFFP \leftarrow OFFP \cup$ Offspring
- 19: **end if**
- 20: **end for**
- 21: **if** $|OFFP| < K$ **then**
- 22: **for** $i \leftarrow 1$ to $K - |OFFP|$ **do**
- 23: $[x_1, x_2] \leftarrow$ randomly select two individuals from OP
- 24: Offspring \leftarrow crossover and mutation(x_1, x_2)
- 25: $OFFP \leftarrow OFFP \cup$ Offspring
- 26: **end for**
- 27: **end if**

value can reflect the convergence performance of each individual. A smaller convergence fitness value implies a better convergence performance. More importantly, a hierarchical clustering is used to divide the population into clusters, and parents are selected from different clusters to maintain diversity.

A detailed process of the proposed LAS can be summarized as follows. Firstly, the population PP is sorted into different fronts (F_1, F_2, \dots, F_t) based on the principle of non-dominated sorting [7]. Then, the fronts are moved one by one into population Q from the lowest level to the highest level until a front F_t is encountered. Subsequently, assign the convergence fitness to each individuals in the population PP . After that, a hierarchical clustering method is employed to divide the population into K clusters. Here, the parameter K is dynamically adjusted according to the fitness improvement value of PP and DP (i.e., FI_{PP} and FI_{DP}) at each generation

as follows:

$$\begin{cases} FI_{PP} = \sum \Delta f^{o_1} / K' & o_1 \in Off_{PP} \\ FI_{DP} = \sum \Delta f^{o_2} / (N - K') & o_2 \in Off_{DP} \end{cases} \quad (3)$$

where Off_{PP} and Off_{DP} denote a group of offspring generated from PP and DP , respectively. While, o_1 and o_2 stand for the offspring in Off_{PP} and Off_{DP} , respectively. N is the population size and K' is the value of K at previous generation. $\Delta f^i(x, y | \lambda^i, z^*)$ is the enhancement brought by the new offspring y associated to the parent x , which is defined as:

$$\Delta f^i(x, y | \lambda_i, z^*) = g^{tch}(x | \lambda_i, z^*) - g^{tch}(y | \lambda_i, z^*) \quad (4)$$

where $g^{tch}(x | \lambda_i, z^*)$ is the fitness value, which is assigned using the following TCH decomposition function:

$$g^{tch}(x | W, Z^*) = \max_{1 \leq j \leq m} \left\{ \lambda_j * |f_j(x) - z_j^*| \right\} \quad (5)$$

where $W = (\lambda_1, \lambda_2, \dots, \lambda_n)$ is a set of weight vectors, $f_j(x)$ represents the j^{th} objective value. $Z^* = (z_1^*, z_2^*, \dots, z_m^*)$ is the ideal vector for m objectives, which is approximated by the minimum value of each objective in the current population, i.e., for all $j \in \{1, \dots, m\}$

$$z_j^* = \min_{1 \leq x \leq m} \{f_j(x) | x \in DP\} \quad (6)$$

The normalized fitness improvement value (FI'_{PP} and FI'_{DP}) can be obtained by:

$$\begin{cases} FI'_{DP} = FI_{DP} / (FI_{PP} + FI_{DP}) \\ FI'_{PP} = FI_{PP} / (FI_{PP} + FI_{DP}) \end{cases} \quad (7)$$

where $0 \leq FI'_{PP}, FI'_{DP} \leq 1$. Based on FI'_{PP} and FI'_{DP} , the parameter K can be finally calculated as:

$$K = \max(\min([FI'_{PP} * N], N - 3), 3) \quad (8)$$

where $\max()$ and $\min()$ return the maximum and minimum values, respectively. It is worth noting that the minimum K is set to be 3, which ensures that the population PP has at least three individuals to be selected into the mating pool.

After obtaining the clusters, the individuals with the best convergence performance within each cluster will be selected and added into the mating pool. If the number of individuals in the cluster is greater than or equal to two, two individuals with the best convergence fitness value will be selected to produce offspring. Finally, when the number of individuals in the offspring pool $OFFP$ is less than the number of individuals that need to be generated, the parent will be randomly selected from the OP , and $K - OFFP$ offspring will be generated and added to the $OFFP$. The pseudo-code of LAS is shown in Algorithm 2.

To help understand the LAS strategy further, an example is given in Fig. 1 to illustrate its process. In the Fig. 1, eight solutions are allocated to five clusters. For the clusters C_2 and C_3 , in which the number of solutions is greater than or equal to two, two solutions with the best convergence fitness value will be selected, i.e., solutions D and E from C_2 and solutions F and H from C_3 will be selected for mating and

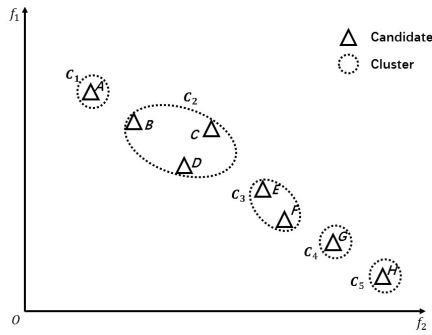


FIGURE 1. Example of illustration of LAS strategy.

the generated offspring will be inserted into offspring pool *OFF*. At the same time, the solution with the best convergence fitness value in C_1 to C_5 will be added into the mating pool. If the number of generated offspring is less than the number of offspring to be generated, K -*OFF* solutions will be selected from mating pool to generate offspring and added to *OFF*.

B. LOCAL COVERAGE SELECTION STRATEGY

The main purpose of environment selection is to preserve a group of individuals from the union of current population and their offspring to create a new population for the next generation. These surviving individuals should be well-distributed and have good convergence. In case that the PF has complicated geometrics, the diversity maintenance mechanism based on Pareto-based MOEAs could have difficulty to guarantee the individuals to be distributed evenly in the entire PF, thus reducing their performance. To alleviate this issue, we propose to employ a hierarchical clustering to divide the population into clusters for discovering population structure. The number of individuals to be retained is determined according to the coverage area of each cluster, while the crowding distance is used to determine the final retained individuals. This allows individuals to be distributed as evenly as possible throughout the entire PF, thus maintaining population diversity.

The proposed local coverage selection works as follows. Firstly, the efficient non-dominated-sorting method is used to maintain the convergence of population. Then, the crowding distance is applied to reflect the crowding degree among individuals. The crowding distance $dis(X)$ can be defined as:

$$dis(X) = \sum_{k=1}^m \frac{f^k(X_{n+1}) - f^k(X_{n-1})}{f_{max}^k - f_{min}^k} \quad (9)$$

where m is the number of objectives, f_{max}^k and f_{min}^k denote the maximum and minimum, respectively, of the k^{th} objective value. $f^k(X_{n+1})$ and $f^k(X_{n-1})$ are two nearest solutions on either side of the member along the k^{th} objective. X_{n-1} and X_{n+1} are two closest solutions to X_n on both sides of the membership of the k^{th} objective. For the non-dominated members, which have maximum or minimum value for any objective, the crowding distance will be assigned with an infinite value.

Algorithm 3 Local Coverage Selection Strategy

Input: Population PP and the size of population N

Output: Final population Q

```

1:  $Q \leftarrow \emptyset$ 
2:  $(F_1, F_2, \dots, F_t) \leftarrow \text{non-dominated-sorting}(PP)$ 
3: while  $|Q \cup F_t| < N$  do
4:    $Q \leftarrow Q \cup F_t$ 
5:    $i \leftarrow i + 1$ 
6: end while
7: if  $|Q| > N$  then
8:    $AD \leftarrow$  calculate the crowding distance of each individual in population  $Q$  according to Eq. 9
9:   Normalize objective values of individuals in population  $Q$  according to Eq. 10
10:   $C_1, C_2, \dots, C_N \leftarrow$  divide individuals in population  $Q$  into  $N$  clusters by a hierarchical clustering algorithm
11:   $U \leftarrow \emptyset, j \leftarrow 1$ 
12:  for  $i \leftarrow 1$  to  $|Q|$  do
13:    if the individual in  $x_i$  is not in  $U$  then
14:       $C' \leftarrow$  the cluster to which  $x_i$  belongs
15:       $U_j \leftarrow C'$ 
16:      if  $|U_j| = 1$  then
17:        while  $x_i$ 's nearest individual  $X$  is a scattered individual do
18:           $U_j \leftarrow U_j \cup X$ 
19:        end while
20:      end if
21:       $U \leftarrow U \cup U_j$ 
22:       $j \leftarrow j + 1$ 
23:    end if
24:  end for
25:   $CA \leftarrow$  calculate the size of coverage area according to Eq.11
26:  for  $i \leftarrow 1$  to  $|U|$  do
27:    keep  $Num(C)$  individuals with the best crowding distance
28:  end for
29:   $Q \leftarrow U$ 
30: end if

```

A bigger crowding distance value for the individual implies a better diversity performance.

Since Euclidean distance is used in hierarchical clustering, objective values will be normalized to the range of $[0, 1]$ for robust clustering. The objective values of each individual in population Q are scaled as:

$$f^{k'}(X) = \frac{f^k(X) - f_{min}^k}{f_{max}^k - f_{min}^k} \quad (10)$$

Based on the normalized values, a hierarchical clustering is then used to divide population into N clusters. As too many singular clusters are not helpful in recovering the population structure, the following steps have been employed to reduce the number of singular clusters. First, identify the nearest

neighbor, X'_i , of the individual X_i from a singular cluster. If X'_i is also a singular cluster, individuals X_i and X'_i will be merged to form a new cluster. This process, termed as cluster recombination, will be repeated until the nearest neighbor X'_i is not a singular cluster. After obtaining the clusters, the coverage area of each cluster is subsequently calculated and the number of individuals that need to be retained in each cluster is determined according to:

$$area(C) = \prod_{k=1}^m f_{max}^k(C) - f_{min}^k(C) \quad (11)$$

$$Num(C) = \min(1, floor(\frac{area(C)}{\sum_1^T area(C_i)} * (N - sN))) \quad (12)$$

where C represents a cluster, $f_{max}^k(C)$ and $f_{min}^k(C)$ denote the maximum and minimum, respectively, of the k^{th} objective value in the cluster C . T and sN is the number of clusters and the number of singular clusters, respectively. The function, $floor()$, returns the nearest integer in the direction of negative infinity, and function, $min()$, returns the minimum value. Finally, each cluster retains $Num(C)$ individuals with the best crowding distance to form a new population, so that the distribution of individuals within each cluster to be as uniformly as possible over the entire PF. The pseudo-code of the proposed LCS strategy is shown in Algorithm 3.

In order to facilitate the understanding, a simple example of implementing the LCS has been illustrated in Fig. 2. In Fig. 2, suppose the individuals A, B, C, D, E, F, G and H to be partitioned into four clusters C_1, C_2, C_3 and C_4 according to a hierarchical clustering method. The number of retained individuals $Num(C)$ will then be calculated based on the coverage area of clusters C_1, C_2, C_3 and C_4 , respectively. Assuming the calculated values of $Num(C)$ for clusters C_1 to C_4 to be 1, 2, 2 and 1, respectively. Also, assuming the crowding distance to be $B > D > C, E > F$ and $G > H$. Then, for cluster C_1 , the individual A will be retained. For cluster C_2 , two individuals with the best crowding distance (i.e., B and D) will be preserved. Similarly, the individuals E, F and G will also be kept. Finally, based on the procedure of LCS, a set of solutions containing A, B, D, E, F and G will be obtained for evolution of next generation.

IV. EXPERIMENTS

In this section, we first evaluate the significance of LAS and LCS strategies in the proposed algorithm. Then, we compare the proposed method with state-of-the-art multi-objective evolutionary algorithms. Unless otherwise stated, the median and corresponding interquartile range (IQR) over 30 independent trials of each algorithm are reported and formatted as (median \pm IQR). For each row in the table, we highlight the best value in bold. To obtain a statistically sound conclusion, Wilcoxon rank sum test is performed at a significance level $\alpha = 0.05$. In the tables, the symbols “+”, “-”, and “ \approx ” indicate that the results of methods to be compared are significantly better, worse and similar, respectively, to our method. The experiments are carried out on a machine with

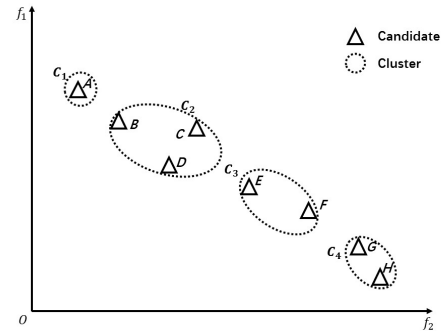


FIGURE 2. Example of illustration of LCS strategy.

TABLE 1. Characteristics of test problems and parameter settings.

Problem	M	D	Characteristics	N	max_{ev}
WFG1	2	22	Mixed, Biased	200	1.0e5
WFG2	2	22	Convex, Discontinuous, Nonseparable	200	1.0e5
WFG3	2	22	Linear, Degenerate, Nonseparable	200	1.0e5
WFG4	2	22	Concave, Multimodal	200	1.0e5
WFG5	2	22	Concave, Deceptive	200	1.0e5
WFG6	2	22	Concave, Nonseparable	200	1.0e5
WFG7	2	22	Concave, Biased	200	1.0e5
WFG8	2	22	Concave, Nonseparable, Biased	200	1.0e5
WFG9	2	22	Concave, Nonsepar., Deceptive, Biased	200	1.0e5
DTLZ1	3	7	Linear, Multimodal	500	1.0e5
DTLZ2	3	12	Concave, Sphere PF	500	1.0e5
DTLZ3	3	12	Concave, Sphere PF, Multimodal	500	1.0e5
DTLZ4	3	12	Concave, Biased	500	1.0e5
DTLZ5	3	12	Concave, Degenerate	500	1.0e5
DTLZ6	3	12	Concave, Degenerate, Biased	500	1.0e5
DTLZ7	3	22	Mixed, Discontinuous, Multimodal	500	1.0e5
UF1	2	30	Convex, Complex PS	300	3.0e5
UF2	2	30	Convex, Complex PS	300	3.0e5
UF3	2	30	Convex, Complex PS	300	3.0e5
UF4	2	30	Concave, Complex PS	300	3.0e5
UF5	2	30	Linear, Discrete, Complex PS	300	3.0e5
UF6	2	30	Linear, Discontinuous, Complex PS	300	3.0e5
UF7	2	30	Linear, Complex PS	300	3.0e5
UF8	3	30	Concave, Complex PS	600	3.0e5
UF9	3	30	Linear, Discontinuous, Complex PS	600	3.0e5
UF10	3	30	Concave, Complex PS	600	3.0e5

Microsoft Window 10 Pro, Intel Core i5-5200 2.40 GHz and 16GB RAM.

A. TEST PROBLEMS AND PARAMETER SETTINGS

The MOPs to be tested include WFG [41], DTLZ [42] and UF [43] problems. The characteristics and parameter settings of these problems are shown in Table 1. These parameter settings are the same as recommended in [44].

We compare our proposed method HCCA with seven related algorithms including MOEA/D-PaS [45], EAG-MOEA/D [46], CA-MOEA [47], SPEA/R [48], DEANG [49], A-NSGA-III [30] and EMyO/C [33]. The parameters of these algorithm are listed in Table 2, which are specified or chosen according to the original setting with the best performance. Here, P_c is the crossover probability, P_m is the mutation probability, η_c and η_m are the distribution indexes of SBX and PM, respectively. For the DE operator, CR and

TABLE 2. Parameter settings of the methods to be compared.

Algorithm	Parameter Settings
MOEA/D-PaS	$CR = 0.9, F = 0.5, P_m = 1/n, \eta = 20, \delta = 0.9, T = 0.1N, n_r = 0.1T$
EAG-MOEA/D	$N = 100, CR = 1, F = 0.5, P_m = 1/n, \eta_m = T = 20$
CA-MOEA	$P_c = 1, P_m = 1/n, \eta_m = \eta_c = 20$
EMyO/C	$P_m = 1/n, \eta_m = 20, CR = 0.15$
SPEA/R	$P_c = 1, P_m = 1/N, \eta_m = \eta_c = 20$
DEAGNG	$ A_S = MN, \epsilon = 0.05\pi, HP_{max} = 2 A_S , age_{max} = N, \lambda = 0.2N, \epsilon_a = 0.2, \epsilon_{nb} = 0.01, \alpha = 0.5, \beta = 0.9$
A-NSGA-III	$P_c = 1, P_m = 1/N, \eta_m = 20, \eta_c = 30$
HCCA	$P_c = 1, P_m = 1/n, \eta_m = \eta_c = 20, \delta = 0.9, n_r = 2, CR = 1, F = 0.5$

TABLE 3. Comparing HCCA with HCCA-RAND in terms of IGD and HV metrics.

Problem	IGD		HV	
	HCCA-RANDOM	HCCA	HCCA-RANDOM	HCCA
WFG1	(1.4±0.27)e-2	(1.3±0.52)e-2	(6.72±0.013)	(6.72±0.035)
WFG2	(7.1±0.24)e-3	(7.1±0.032)e-3	(6.13±0.002)	(6.13±0.001)
WFG3	(6.6±0.25)e-3	(6.6±0.14)e-3	(5.647±0.002)	(5.647±0.001)
WFG4	(7.05±0.52)e-3	(8.48±0.43)e-3	(3.360±0.005)	(3.360±0.001)
WFG5	(6.2±0.047)e-2	(6.2±0.017)e-2	(3.04±0.006)	(3.05±0.002)
WFG6	(2.5±1.8)e-2	(2.4±0.63)e-2	(3.24±0.102)	(3.25±0.033)
WFG7	(6.6±0.092)e-3	(6.6±0.068)e-3	(3.376±0.0004)	(3.376±0.0003)
WFG8	(5.5±0.18)e-2	(5.5±0.13)e-2	(3.082±0.009)	(3.082±0.007)
WFG9	(1.1±0.19)e-2	(1.1±0.32)e-2	(3.331±0.009)	(3.331±0.021)
DTLZ1	(9.3±0.11)e-3	(9.3±0.0098)e-3	(1.42±0.002)e-1	(1.42±0.001)e-1
DTLZ2	(2.4±0.035)e-2	(2.4±0.027)e-2	(7.75±0.022)e-1	(7.75±0.008)e-1
DTLZ3	(2.5±0.17)e-2	(2.5±0.24)e-2	(7.70±0.048)e-1	(7.70±0.050)e-1
DTLZ4	(2.4±0.033)e-2	(2.4±0.001)e-2	(7.76±0.015)e-1	(7.77±0.0095)e-1
DTLZ5	(9.9±0.18)e-4	(9.7±0.18)e-4	(4.41±0.0009)e-1	(4.41±0.0001)e-1
DTLZ6	(8.51±0.51)e-4	(8.51±0.50)e-4	(4.41±0.0011)e-1	(4.41±0.0018)e-1
DTLZ7	(3.2±0.028)e-2	(3.2±0.020)e-2	(2.65±0.019)e-1	(2.65±0.003)e-1
UF1	(2.1±0.22)e-3	(2.0±0.17)e-3	(8.723±0.003)e-1	(8.727±0.004)e-1
UF2	(5.4±2.1)e-3	(5.8±1.4)e-3	(8.67±0.018)e-1	(8.67±0.031)e-1
UF3	(2.1±2.3)e-2	(9.4±1.6)e-3	(8.52±0.36)e-1	(8.52±0.27)e-1
UF4	(3.4±0.12)e-2	(3.5±0.13)e-2	(4.856±0.006)e-1	(4.860±0.0069)e-1
UF5	(2.5±1.9)e-1	(2.5±1.5)e-1	(2.0±2.7)e-1	(2.4±2.0)e-1
UF6	(9.7±1.5)e-2	(9.7±1.05)e-2	(4.21±1.1)e-1	(4.4±0.49)e-1
UF7	(2.4±0.87)e-3	(2.2±0.023)e-3	(7.052±0.032)e-1	(7.057±0.014)e-1
UF8	(8.8±0.46)e-2	(8.1±0.46)e-2	(6.41±0.10)e-1	(6.38±0.10)e-1
UF9	(1.9±1.2)e-1	(7.7±1.2)e-2	(8.51±1.3)e-1	(9.7±1.4)e-1
UF10	(2.6±0.47)e-1	(2.3±0.09)e-1	(4.07±1.7)e-1	(4.51±0.32)e-1

TABLE 4. Comparing HCCA with HCCA-CROWD in terms of IGD and HV metrics.

Problem	IGD		HV	
	HCCA-CROWD	HCCA	HCCA-CROWD	HCCA
WFG1	(1.5±0.23)e-2	(1.3±0.52)e-2	(6.70±0.022)	(6.72±0.035)
WFG2	(5.4±0.011)e-3	(7.1±0.032)e-3	(6.13±0.001)	(6.13±0.001)
WFG3	(6.7±0.16)e-3	(6.6±0.14)e-3	(5.647±0.001)	(5.647±0.001)
WFG4	(7.7±0.69)e-3	(8.48±0.43)e-3	(3.361±0.006)	(3.360±0.001)
WFG5	(6.2±0.019)e-2	(6.2±0.017)e-2	(3.05±0.002)	(3.05±0.002)
WFG6	(2.7±8.8)e-2	(2.4±0.63)e-2	(3.24±0.033)	(3.25±0.033)
WFG7	(6.6±0.079)e-3	(6.6±0.068)e-3	(3.376±0.0005)	(3.376±0.0003)
WFG8	(5.6±0.54)e-2	(5.5±0.13)e-2	(3.082±0.009)	(3.082±0.007)
WFG9	(1.2±0.20)e-2	(1.1±0.32)e-2	(3.331±0.011)	(3.331±0.021)
DTLZ1	(9.3±0.0084)e-3	(9.3±0.0098)e-3	(1.42±0.008)e-1	(1.42±0.001)e-1
DTLZ2	(2.4±0.011)e-2	(2.4±0.027)e-2	(7.74±0.015)e-1	(7.75±0.008)e-1
DTLZ3	(2.6±0.21)e-2	(2.5±0.24)e-2	(7.68±0.044)e-1	(7.70±0.050)e-1
DTLZ4	(2.4±0.003)e-2	(2.4±0.001)e-2	(7.76±0.011)e-1	(7.77±0.0095)e-1
DTLZ5	(9.7±0.31)e-4	(9.7±0.18)e-4	(4.41±0.0002)e-1	(4.41±0.0001)e-1
DTLZ6	(8.51±0.58)e-4	(8.51±0.50)e-4	(4.41±0.0019)e-1	(4.41±0.0018)e-1
DTLZ7	(3.3±0.010)e-2	(3.2±0.020)e-2	(2.65±0.003)e-1	(2.65±0.003)e-1
UF1	(2.1±0.84)e-3	(2.0±0.17)e-3	(8.727±0.005)e-1	(8.727±0.004)e-1
UF2	(5.8±1.6)e-3	(5.8±1.4)e-3	(8.67±0.045)e-1	(8.67±0.031)e-1
UF3	(4.5±1.5)e-3	(9.4±1.6)e-3	(8.66±0.26)e-1	(8.52±0.27)e-1
UF4	(3.5±0.37)e-2	(3.5±0.13)e-2	(4.855±0.0014)e-1	(4.860±0.0069)e-1
UF5	(2.3±0.55)e-1	(2.5±1.5)e-1	(2.4±1.3)e-1	(2.4±2.0)e-1
UF6	(9.8±1.7)e-2	(9.7±1.05)e-2	(4.3±1.4)e-1	(4.4±0.49)e-1
UF7	(2.4±0.048)e-3	(2.2±0.023)e-3	(7.056±0.014)e-1	(7.057±0.014)e-1
UF8	(9.1±0.61)e-2	(8.1±0.46)e-2	(6.39±0.044)e-1	(6.38±0.10)e-1
UF9	(1.2±1.6)e-1	(7.7±1.2)e-2	(8.5±1.6)e-1	(9.7±1.4)e-1
UF10	(2.3±0.56)e-1	(2.3±0.09)e-1	(4.51±0.20)e-1	(4.51±0.32)e-1

F are the crossover rate and scaling factor, respectively, T denotes the size of neighborhood for weight vectors, δ is the probability to select the parents from T neighbors, and n_r is the maximum number of parent solutions to be updated by each offspring solution. $|A_S|$ and HP_{max} are the signal archive

and maximum, respectively, of Hit point of node. age_{max} is the maximum cluster age and λ represents the cycle for topology reconstruction. ϵ_a stands for learning coefficient and ϵ_{nb} denotes learning coefficient of neighbor. α indicates

TABLE 5. Comparing the performance of HCCA with different initial values of parameter K in terms of IGD and HV metrics.

	IGD					HV				
	K=0	K=0.25N	K=0.5N	K=0.75N	K=N	K=0	K=0.25N	K=0.5N	K=0.75N	K=N
WFG1	(1.3±0.45)e-2	(1.3±0.43)e-2	(1.3±0.52)e-2	(1.3±0.45)e-2	(1.3±0.27)e-2	(6.72±0.036)	(6.72±0.024)	(6.72±0.035)	(6.72±0.027)	(6.72±0.030)
WFG2	(7.1±0.035)e-3	(7.1±0.036)e-3	(7.1±0.032)e-3	(7.1±0.039)e-3	(7.1±0.033)e-3	(6.13±0.001)	(6.13±0.001)	(6.13±0.001)	(6.13±0.001)	(6.13±0.001)
WFG3	(6.6±0.16)e-3	(6.6±0.13)e-3	(6.6±0.14)e-3	(6.6±0.23)e-3	(6.6±0.15)e-3	(5.647±0.001)	(5.647±0.001)	(5.647±0.001)	(5.647±0.002)	(5.647±0.001)
WFG4	(8.25±0.80)e-3	(7.88±0.58)e-3	(8.48±0.43)e-3	(7.85±0.70)e-3	(7.76±0.58)e-3	(3.357±0.005)	(3.359±0.005)	(3.360±0.001)	(3.359±0.004)	(3.359±0.005)
WFG5	(6.2±0.065)e-2	(6.2±0.020)e-2	(6.2±0.017)e-2	(6.2±0.079)e-2	(6.2±0.017)e-2	(3.05±0.009)	(3.05±0.002)	(3.05±0.002)	(3.05±0.009)	(3.05±0.003)
WFG6	(2.3±1.20)e-2	(2.5±1.05)e-2	(2.4±0.63)e-2	(2.5±0.56)e-2	(2.5±0.48)e-2	(3.25±0.067)	(3.24±0.057)	(3.25±0.033)	(3.25±0.031)	(3.24±0.025)
WFG7	(6.6±0.066)e-3	(6.6±0.066)e-3	(6.6±0.068)e-3	(6.6±0.064)e-3	(6.6±0.079)e-3	(3.376±0.0005)	(3.376±0.0005)	(3.376±0.0003)	(3.376±0.0007)	(3.376±0.0005)
WFG8	(5.6±0.03)e-2	(5.5±0.09)e-2	(5.5±0.13)e-2	(5.5±0.010)e-2	(5.6±0.078)e-2	(3.081±0.0031)	(3.081±0.0060)	(3.082±0.007)	(3.082±0.0049)	(3.081±0.0028)
WFG9	(1.2±0.17)e-2	(1.2±0.27)e-2	(1.1±0.32)e-2	(1.1±0.29)e-2	(1.2±0.25)e-2	(3.324±0.013)	(3.324±0.019)	(3.331±0.021)	(3.330±0.017)	(3.324±0.013)
DTLZ1	(9.3±0.013)e-3	(9.3±0.014)e-3	(9.3±0.0098)e-3	(9.3±0.063)e-3	(9.3±0.091)e-3	(1.42±0.001)e-1	(1.42±0.001)e-1	(1.42±0.001)e-1	(1.42±0.001)e-1	(1.42±0.001)e-1
DTLZ2	(2.4±0.031)e-2	(2.4±0.036)e-2	(2.4±0.027)e-2	(2.4±0.036)e-2	(2.4±0.034)e-2	(7.74±0.009)e-1	(7.74±0.015)e-1	(7.74±0.008)e-1	(7.74±0.009)e-1	(7.74±0.011)e-1
DTLZ3	(2.5±0.76)e-2	(2.5±0.24)e-2	(2.5±0.24)e-2	(2.5±0.16)e-2	(2.5±0.16)e-2	(7.70±0.026)e-1	(7.70±0.048)e-1	(7.69±0.050)e-1	(7.70±0.042)e-1	(7.70±0.037)e-1
DTLZ4	(2.4±0.0028)e-2	(2.4±0.0028)e-2	(2.4±0.001)e-2	(2.4±0.0026)e-2	(2.4±0.0029)e-2	(7.77±0.012)e-1	(7.77±0.012)e-1	(7.77±0.0095)e-1	(7.77±0.012)e-1	(7.77±0.008)e-1
DTLZ5	(9.8±0.24)e-4	(9.6±0.22)e-4	(9.7±0.18)e-4	(9.9±0.03)e-4	(9.8±0.01)e-4	(4.41±0.0001)e-1	(4.41±0.0002)e-1	(4.41±0.0001)e-1	(4.41±0.0001)e-1	(4.41±0.0001)e-1
DTLZ6	(8.77±0.62)e-4	(8.46±0.55)e-4	(8.51±0.50)e-4	(8.72±0.05)e-4	(8.77±0.06)e-4	(4.41±0.0024)e-1	(4.41±0.0009)e-1	(4.41±0.0001)e-1	(4.41±0.0001)e-1	(4.41±0.002)e-1
DTLZ7	(3.3±0.18)e-2	(3.2±0.14)e-2	(3.2±0.020)e-2	(3.2±0.13)e-2	(3.2±0.20)e-2	(2.65±0.018)e-1	(2.65±0.023)e-1	(2.65±0.003)e-1	(2.65±0.023)e-1	(2.65±0.032)e-1
UF1	(1.9±0.10)e-3	(1.9±0.15)e-3	(1.8±0.17)e-3	(1.9±0.20)e-3	(2.0±0.13)e-3	(8.727±0.004)e-1	(8.727±0.003)e-1	(8.729±0.004)e-1	(8.727±0.005)e-1	(8.726±0.005)e-1
UF2	(6.0±1.4)e-3	(5.4±1.4)e-3	(5.6±1.4)e-3	(6.8±2.5)e-3	(5.0±1.4)e-3	(8.66±0.030)e-1	(8.67±0.024)e-1	(8.67±0.031)e-1	(8.66±0.039)e-1	(8.67±0.020)e-1
UF3	(4.3±1.3)e-3	(9.4±3.3)e-3	(9.4±1.6)e-3	(5.9±2.2)e-3	(1.2±2.8)e-2	(8.68±0.19)e-1	(8.52±0.49)e-1	(8.52±0.27)e-1	(8.66±0.35)e-1	(8.52±0.43)e-1
UF4	(3.4±0.10)e-2	(3.4±0.12)e-2	(3.5±0.13)e-2	(3.4±0.14)e-2	(3.4±0.10)e-2	(4.858±0.0063)e-1	(4.857±0.0044)e-1	(4.860±0.0069)e-1	(4.857±0.0087)e-1	(4.857±0.0085)e-1
UF5	(2.5±0.84)e-1	(2.6±1.18)e-1	(2.5±1.5)e-1	(2.7±1.22)e-1	(1.7±1.56)e-1	(2.6±1.2)e-1	(2.3±1.5)e-1	(2.4±2.0)e-1	(2.3±1.6)e-1	(3.8±2.7)e-1
UF6	(9.5±0.28)e-2	(9.5±0.71)e-2	(9.7±1.05)e-2	(9.8±1.91)e-1	(1.0±0.16)e-1	(4.4±0.40)e-1	(4.4±0.57)e-1	(4.4±0.49)e-1	(4.3±2.0)e-1	(4.3±2.0)e-1
UF7	(2.4±0.023)e-3	(2.4±0.027)e-3	(2.2±2.3)e-3	(2.5±0.052)e-3	(2.4±0.038)e-3	(7.056±0.0052)e-1	(7.055±0.079)e-1	(7.057±0.014)e-1	(7.053±0.010)e-1	(7.054±0.007)e-1
UF8	(9.0±0.86)e-2	(8.2±1.2)e-2	(8.1±0.46)e-2	(8.8±1.1)e-2	(8.7±1.4)e-2	(6.38±0.05)e-1	(6.38±0.11)e-1	(6.38±0.10)e-1	(6.38±0.04)e-1	(6.38±0.06)e-1
UF9	(1.9±1.3)e-1	(6.6±2.0)e-2	(7.7±1.2)e-2	(8.3±1.2)e-2	(8.8±1.2)e-2	(8.4±1.4)e-1	(9.8±1.5)e-1	(9.7±1.4)e-1	(9.6±1.3)e-1	(9.6±1.3)e-1
UF10	(2.3±0.05)e-1	(2.5±0.05)e-1	(2.3±0.09)e-1	(2.3±0.12)e-1	(2.4±0.07)e-1	(4.49±0.11)e-1	(4.19±0.15)e-1	(4.51±0.32)e-1	(4.51±0.56)e-1	(4.15±0.17)e-1

nodes $r1_{max}$ and $r2_{max}$ error reduction constant, while β is error reduction coefficient.

B. PERFORMANCE METRICS

Two widely employed performance metrics, inverted generational distance (IGD) [12] and HV [11], have been adopted for evaluation. IGD can reflect both convergence and diversity. Let P^* denote a subset of PS that is distributed evenly along the PF and P indicate the approximate set obtained by an algorithm. The IGD value of P to P^* is calculated as:

$$IGD(P^*, P) = \frac{\sum_{x \in P^*} d(x, P)}{|P^*|} \tag{13}$$

where $d(x, P)$ is the minimum Euclidean distance from the point x to P and $|P^*|$ returns the size of P^* . Generally, a smaller IGD value means that the approximate set is closer to the PF and distributed more evenly.

The HV [11] is also believed to be able to account for both convergence and diversity. To calculate the value of HV for

the final solution set, a reference point which is dominated by all Pareto optimal solutions should be predefined. Here, as suggested by [44], the reference points for computing HV are set to be 1.1 times the nadir point of true PF, i.e., $1.1 \times (0.5, 0.5, 0.5)$ for DTLZ1, $1.1 \times (1.0, \dots, 1.0)$ for DTLZ2- DTLZ6 and UF problems, $1.1 \times (1.0, 1.0, 2.0 \times m)$ for DTLZ7, $1.1 \times (2.0, 2.0 \times m)$ for WFG1-WFG9 (where m is the number of objectives). Note that a larger HV value indicates a better performance.

C. RESULTS

We first examine the merit of LAS strategy by comparing HCCA and its variant (i.e., HCCA-RAND). In HCCA-RAND, rather than the proposed local area selection strategy, the random selection is employed as mating selection. Table 3 shows the comparison results of HCCA and HCCA-RAND on the test problems. The results show that HCCA could deliver significantly better performance on the problems based on both metrics. Using IGD metric,

TABLE 6. Comparing HCCA with related methods in term of IGD metric.

IGD	MOEA/D-PaS	EAG-MOEA/D	CA-MOEA	SPEA/R	DEAGNG	A-NSGA-III	EMyO/C	HCCA
WFG1	(3.5±0.91)e-1	(5.4±1.2)e-1	(1.0±0.10)e-1	(8.13±6.22)e-2	(1.0±0.21)e-1	(1.09±0.073)e-1	(8.93±0.67)e-1	(1.3±0.52)e-2
WFG2	(1.1±0.11)e-2	(5.2±5.6)e-2	(6.5±0.24)e-3	(1.05±0.069)e-2	(6.1±0.39)e-3	(7.53±0.44)e-3	(1.15±0.14)e-2	(7.1±0.032)e-3
WFG3	(8.1±0.82)e-3	(1.0±0.083)e-2	(7.7±0.03)e-3	(8.20±0.59)e-3	(6.7±0.45)e-3	(9.16±0.77)e-3	(1.68±0.19)e-2	(6.6±0.14)e-3
WFG4	(6.25±0.96)e-2	(2.5±0.45)e-2	(8.1±0.90)e-3	(7.41±0.39)e-3	(7.5±0.018)e-1	(9.36±1.35)e-3	(1.79±0.20)e-2	(8.48±0.43)e-3
WFG5	(6.7±0.45)e-2	(6.6±0.0025)e-2	(6.2±0.058)e-2	(6.26±0.096)e-2	(6.2±0.058)e-2	(6.78±0.51)e-2	(6.62±0.40)e-2	(6.2±0.017)e-2
WFG6	(1.1±0.022)e-2	(4.9±6.3)e-2	(4.9±0.71)e-2	(4.63±0.98)e-2	(5.0±0.56)e-2	(5.02±0.67)e-2	(1.12±0.63)e-1	(2.4±0.63)e-2
WFG7	(7.3±0.12)e-3	(8.8±0.53)e-3	(7.6±0.091)e-3	(6.96±0.27)e-3	(7.08±0.18)e-3	(8.40±0.26)e-3	(9.31±0.63)e-3	(6.6±0.068)e-3
WFG8	(5.9±0.26)e-2	(1.3±0.067)e-1	(5.9±0.21)e-2	(5.85±0.28)e-2	(5.8±0.31)e-2	(6.03±0.20)e-2	(7.78±0.37)e-2	(5.5±0.13)e-2
WFG9	(2.9±8.5)e-2	(7.6±11)e-2	(1.2±5.1)e-2	(1.36±0.32)e-2	(1.1±0.15)e-2	(1.54±5.59)e-2	(1.12±0.0025)e-1	(1.1±0.32)e-2
+/-	1/0/8	0/0/9	2/2/5	1/1/7	1/3/5	0/0/9	0/1/8	
DTLZ1	(9.8±0.0010)e-3	(1.1±0.035)e-2	(9.0±0.024)e-3	(8.81±0.22)e-3	(1.0±0.12)e-2	(9.36±1.14)e-3	(9.00±0.021)e-3	(9.3±0.0098)e-3
DTLZ2	(2.5±0.023)e-2	(2.8±0.074)e-2	(2.5±0.013)e-2	(2.32±0.033)e-2	(2.3±0.021)e-2	(2.30±0.090)e-2	(2.39±0.019)e-2	(2.4±0.027)e-2
DTLZ3	(2.6±0.21)e-2	(2.9±0.09)e-2	(2.7±0.43)e-2	(7.97±5.16)e-2	(3.3±0.95)e-2	(5.02±3.73)e-2	(2.39±0.026)e-2	(2.5±0.24)e-2
DTLZ4	(2.9±0.17)e-2	(3.1±0.37)e-2	(2.4±0.014)e-2	(2.33±0.052)e-2	(2.3±0.025)e-2	(2.26±0.028)e-2	(2.38±0.016)e-2	(2.4±0.001)e-2
DTLZ5	(4.2±0.16)e-3	(1.1±0.05)e-3	(1.0±0.017)e-3	(1.39±0.094)e-2	(9.3±0.014)e-3	(1.86±0.27)e-3	(9.28±0.083)e-4	(9.7±0.18)e-4
DTLZ6	(4.1±0.014)e-3	(1.0±0.082)e-3	(8.9±0.083)e-4	(1.44±0.19)e-2	(9.7±0.003)e-4	(2.5±0.005)e-3	(9.19±0.081)e-4	(8.51±0.50)e-4
DTLZ7	(7.9±3.44)e-1	(3.3±0.19)e-2	(2.4±0.001)e-2	(3.59±0.17)e-2	(2.5±0.006)e-2	(2.87±0.054)e-2	(2.63±0.063)e-2	(3.2±0.020)e-2
+/-	0/4/3	0/1/6	1/3/3	2/1/4	2/2/3	2/2/3	3/3/1	
UF1	(2.2±0.18)e-3	(3.1±2.0)e-1	(1.0±0.096)e-2	(5.55±1.38)e-2	(9.3±1.6)e-2	(8.20±1.25)e-2	(2.26±0.62)e-2	(1.8±0.17)e-3
UF2	(8.1±2.0)e-3	(1.4±21.6)e-2	(2.8±0.60)e-2	(1.81±0.80)e-2	(2.7±1.0)e-2	(2.25±0.65)e-2	(1.20±0.22)e-2	(5.6±1.4)e-3
UF3	(3.1±8.6)e-3	(2.7±2.1)e-2	(2.1±1.4)e-1	(1.21±0.38)e-1	(2.0±0.86)e-1	(1.50±0.51)e-1	(3.04±0.48)e-1	(9.4±1.6)e-3
UF4	(6.6±0.54)e-2	(7.2±0.54)e-2	(4.0±0.11)e-2	(3.92±0.11)e-2	(4.0±0.057)e-2	(4.07±0.052)e-2	(3.83±0.11)e-2	(3.5±0.13)e-2
UF5	(3.1±1.0)e-1	(3.9±1.8)e-1	(2.7±0.10)e-1	(1.93±0.68)e-1	(2.1±0.29)e-1	(2.11±0.58)e-1	(2.51±0.93)e-1	(2.5±1.5)e-1
UF6	(2.3±2.9)e-1	(4.0±2.1)e-1	(1.2±0.10)e-1	(9.88±2.04)e-2	(1.1±0.21)e-1	(1.17±0.20)e-1	(9.54±7.08)e-2	(9.7±1.05)e-2
UF7	(3.8±6.3)e-3	(9.0±10)e-3	(3.3±1.01)e-2	(1.93±0.53)e-2	(3.8±2.3)e-2	(3.50±1.17)e-2	(1.42±0.92)e-2	(2.2±2.3)e-3
UF8	(1.2±0.10)e-1	(7.0±2.3)e-2	(2.4±0.0025)e-1	(8.90±8.94)e-2	(2.3±0.021)e-1	(4.95±0.79)e-1	(8.78±4.46)e-2	(8.1±0.46)e-2
UF9	(2.3±0.17)e-2	(1.7±1.0)e-1	(1.7±1.9)e-1	(9.76±2.47)e-2	(1.8±0.78)e-1	(1.94±0.95)e-1	(1.13±0.14)e-1	(7.7±1.2)e-2
UF10	(4.8±0.69)e-1	(4.6±0.65)e-1	(3.2±0.87)e-1	(2.67±0.26)e-1	(3.2±0.48)e-1	(3.31±0.76)e-1	(4.02±2.80)e-1	(2.3±0.09)e-1
+/-	2/0/8	1/0/9	0/0/10	1/1/8	1/0/9	1/0/9	1/1/8	
Total +/-	3/4/19	1/1/24	3/5/18	4/3/19	4/5/17	3/2/21	4/5/17	

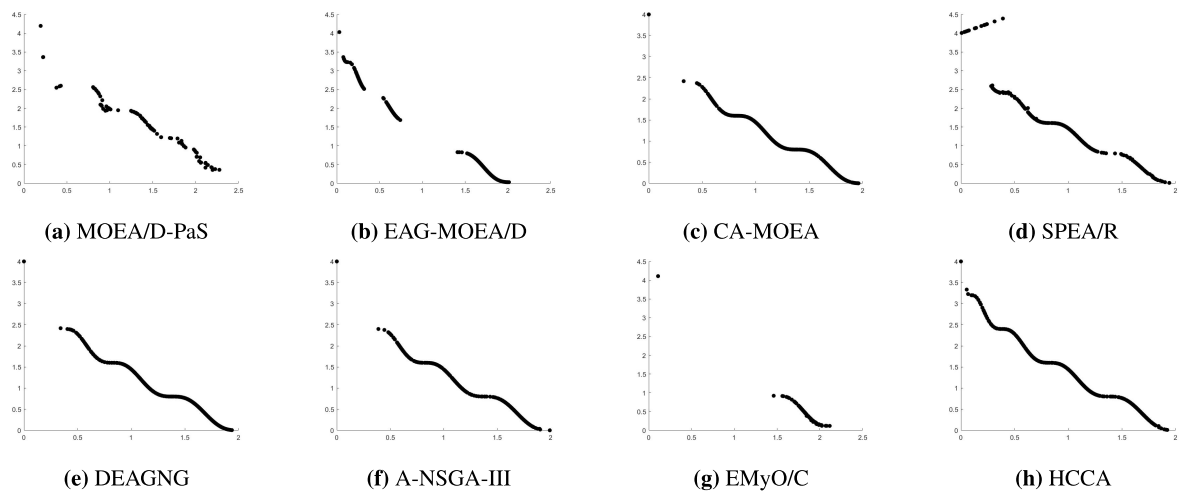


FIGURE 3. Final populations obtained by a typical run of various methods on WFG1 test problem.

HCCA outperforms HCCA-RAND on 20 out of 26 cases. Similar results can also be found in term of HV metric. Based on the results, we can conclude that the proposed

LAS strategy can help select a suitable group of individuals for reproduction, thereby improving the performance of the algorithm.

TABLE 7. Comparing HCCA with related methods in term of HV metric.

HV	MOEA/D-PaS	EAG-MOEA/D	CA-MOEA	SPEA/R	DEANG	A-NSGA-III	EMyO/C	HCCA
WFG1	(5.06±0.33)	(4.01±0.4)	(6.38±0.14)	(6.48±0.242)	(6.41±0.18)	(6.34±0.10)	(2.71±0.47)	(6.72±0.035)
WFG2	(6.11±0.008)	(5.93±0.006)	(6.13±0.31)	(6.10±0.007)	(6.13±0.0043)	(6.12±0.0038)	(6.09±0.006)	(6.13±0.001)
WFG3	(5.633±0.005)	(5.623±0.003)	(5.640±0.0040)	(5.634±0.003)	(5.647±0.0046)	(5.63±0.0070)	(5.584±0.010)	(5.647±0.001)
WFG4	(3.158±0.030)	(3.28±0.015)	(3.364±0.040)	(3.364±0.004)	(3.37±0.0018)	(3.35±0.0040)	(3.329±0.015)	(3.360±0.001)
WFG5	(3.006±0.033)	(2.994±0.0006)	(3.049±0.0007)	(3.04±0.016)	(3.05±0.0029)	(2.99±0.044)	(3.004±0.026)	(3.05±0.002)
WFG6	(2.785±0.0006)	(3.167±0.11)	(3.117±0.038)	(3.13±0.063)	(3.113±0.030)	(3.11±0.037)	(2.78±0.350)	(3.25±0.033)
WFG7	(3.365±0.0012)	(3.365±0.002)	(3.369±0.00042)	(3.371±0.0018)	(3.379±0.00054)	(3.36±0.0042)	(3.359±0.0013)	(3.376±0.0003)
WFG8	(3.065±0.014)	(2.699±0.02)	(3.081±0.0053)	(3.067±0.017)	(3.083±0.0082)	(3.07±0.0083)	(2.99±0.016)	(3.082±0.007)
WFG9	(3.202±0.432)	(3.156±0.49)	(3.32±0.27)	(3.318±0.023)	(3.34±0.015)	(3.30±0.31)	(2.778±0.0034)	(3.331±0.021)
+/ \approx /-	0/0/9	0/0/9	0/5/4	2/0/7	2/5/2	0/0/9	0/0/9	
DTLZ1	(1.42±0.0002)e-1	(1.42±0.004)e-1	(1.43±0.0018)e-1	(1.43±0.001)e-1	(1.42±0.0035)e-1	(1.42±0.0037)e-1	(1.43±0.002)e-1	(1.42±0.001)e-1
DTLZ2	(7.75±0.006)e-1	(7.66±0.019)e-1	(7.73±0.0067)e-1	(7.78±0.005)e-1	(7.75±0.013)e-1	(7.78±0.019)e-1	(7.78±0.003)e-1	(7.74±0.008)e-1
DTLZ3	(7.73±0.032)e-1	(7.64±0.021)e-1	(7.67±0.047)e-1	(6.53±1.04)e-1	(7.59±0.18)e-1	(7.35±0.61)e-1	(7.79±0.003)e-1	(7.69±0.050)e-1
DTLZ4	(7.74±0.012)e-1	(7.69±0.014)e-1	(7.74±0.0094)e-1	(7.78±0.010)e-1	(7.77±0.015)e-1	(7.79±0.0072)e-1	(7.78±0.004)e-1	(7.77±0.0095)e-1
DTLZ5	(4.38±0.002)e-1	(4.41±0.0004)e-1	(4.41±0.0002)e-1	(4.28±0.0113)e-1	(4.41±0.0001)e-1	(4.40±0.0023)e-1	(4.41±0.0001)e-1	(4.41±0.0001)e-1
DTLZ6	(4.38±0.001)e-1	(4.41±0.0002)e-1	(4.41±0.0005)e-1	(4.29±0.0292)e-1	(4.41±0.0002)e-1	(4.40±0.0019)e-1	(4.41±0.0009)e-1	(4.41±0.0001)e-1
DTLZ7	(1.64±0.383)e-1	(2.64±0.0057)	(2.64±0.002)	(2.62±0.032)e-1	(2.65±0.0045)	(2.64±0.0026)	(2.65±0.002)e-1	(2.65±0.003)e-1
+/ \approx /-	0/4/3	0/3/4	1/5/1	2/1/4	0/6/1	1/5/1	3/4/0	
UF1	(8.72±0.003)e-1	(8.718±0.01)e-1	(7.14±0.33)e-1	(7.718±0.022)e-1	(7.33±0.15)e-1	(7.47±0.20)e-1	(8.412±0.108)e-1	(8.729±0.004)e-1
UF2	(8.62±0.035)e-1	(8.59±0.09)e-1	(8.35±0.10)e-1	(8.48±0.069)e-1	(8.34±0.10)e-1	(8.39±0.081)e-1	(8.57±0.031)e-1	(8.67±0.031)e-1
UF3	(8.67±0.166)e-1	(8.317±0.23)e-1	(5.81±1.5)e-1	(6.72±0.47)e-1	(5.64±1.03)e-1	(6.31±0.54)e-1	(4.10±0.36)e-1	(8.52±0.27)e-1
UF4	(4.28±0.054)e-1	(4.222±0.07)e-1	(4.76±0.0081)e-1	(4.738±0.018)e-1	(4.77±0.013)e-1	(4.74±0.013)e-1	(4.775±0.010)e-1	(4.860±0.0069)e-1
UF5	(2.1±1.6)e-1	(2.42±1.3)e-1	(2.72±1.0)e-1	(3.2±1.3)e-1	(3.2±0.77)e-1	(3.69±0.95)e-1	(2.1±1.3)e-1	(2.4±2.0)e-1
UF6	(2.5±2.9)e-1	(3.327±2.2)e-1	(4.62±0.35)e-1	(4.3±0.73)e-1	(4.21±0.62)e-1	(4.38±0.51)e-1	(4.5±0.89)e-1	(4.4±0.49)e-1
UF7	(7.01±0.147)e-1	(7.025±0.13)e-1	(6.53±0.13)e-1	(6.72±0.108)e-1	(6.44±0.33)e-1	(6.52±0.17)e-1	(6.85±0.150)e-1	(7.057±0.014)e-1
UF8	(6.42±0.070)e-1	(6.606±0.38)e-1	(4.44±0.020)e-1	(6.106±0.95)e-1	(4.45±0.015)e-1	(4.63±0.0031)e-1	(6.119±0.76)e-1	(6.38±0.10)e-1
UF9	(8.89±0.023)e-1	(8.282±0.9)e-1	(8.21±2.4)e-1	(9.38±0.534)e-1	(7.71±0.80)e-1	(7.91±1.11)e-1	(9.02±0.31)e-1	(9.7±1.4)e-1
UF10	(1.38±0.47)e-1	(1.471±0.2)e-1	(3.13±1.9)e-1	(4.242±1.31)e-1	(2.75±1.1)e-1	(3.80±2.33)e-1	(1.08±2.42)e-1	(4.51±0.32)e-1
+/ \approx /-	2/1/7	1/1/8	2/0/8	1/0/9	1/0/9	1/0/9	1/0/9	
Total +/ \approx /-	2/5/19	1/4/21	3/10/13	5/1/20	3/11/12	2/5/19	4/4/18	

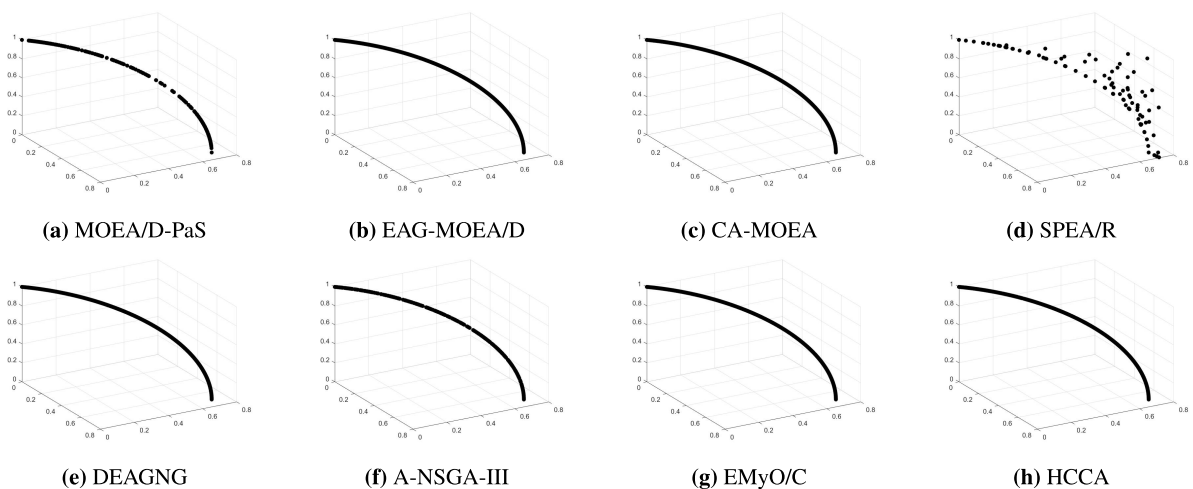


FIGURE 4. Final populations obtained by a typical run of various methods on DTLZ6 test problem.

Then, the effectiveness of proposed LCS strategy in HCCA is accessed. For this purpose, we compare the HCCA with its variant HCCA-CROWD, in which the proposed LCS strategy

is replaced by a crowding distance as the environmental selection. The results are reported in Table 4. From the results, we can see that HCCA can deliver better results on

most problem cases in terms of both HV and IGD metrics. This is mainly due to the proposed LCS strategy can effectively retain a set of individuals with good convergence and diversity during evolution, thus leading to the performance improvement.

Subsequently, the sensitivity of parameter K in LAS strategy is evaluated. Five different initial values of K (i.e., $K = 0, 0.25N, 0.5N, 0.75N, N$) have been used for evaluation. The results are reported in Table 5. The results show that, by setting the parameter K with different initial values, the HCCA could deliver comparable performance. The results thus indicate the HCCA could be robust to initial values of parameter K . This is mainly due to the value of K is set to be dynamically adjusted during the evolution.

Finally, we compare the performance of HCCA with related methods. The results are shown in Tables 6 and 7. The final populations obtained by our algorithm and related methods on two representative problems have also been plotted in Figs. 3 and 4. From the table 6, we can find that our method delivers the best solutions on 13 out of 26 test problems in term of IGD metric. While, MOEA/D-PaS, EAG-MOEA/D, CA-MOEA, SPEA/R, DEAGNG, A-NSGA-III and EMyO/C give the best solutions on 1, 1, 2, 2, 3, 1 and 3 out of 26, respectively. By examining the results of our algorithm with each of the methods to be compared, the Wilcoxon rank-sum test results show our method can achieve better or comparable performance than MOEA/D-PaS, EAG-MOEA/D, CA-MOEA, SPEA/R, DEAGNG, A-NSGA-III and EMyO/C on 23, 25, 23, 22, 22, 23 and 22, respectively, out of 26 test problems. Similar results can also be found in term of HV metric. From Fig. 3, we can find that the final solutions obtained by HCCA could well approximate the PF on WFG1 problem, while MOEA/D-PaS, EAG-MOEA/D, CA-MOEA, SPEA/R, DEAGNG, A-NSGA-III and EMyO/C show relatively poor distribution of the final population in certain region of PF. Fig. 4 shows the final populations obtained by HCCA, DEAGNG and EMyO/C could be more evenly distributed along the PF than those obtained by the rest methods. Based on the results, clearly, our method is the best alternative and could significantly outperform the related algorithms to be compared.

V. CONCLUSION

This paper proposes and implements a multi-objective evolutionary algorithm with hierarchical clustering based selection. In the proposed method, the hierarchical clustering method is employed to design both environmental selection (termed as local coverage selection) and mating selection (termed as local area selection). The local coverage selection strategy is employed to preserve a group of well-distributed with good convergence during evolution, thus appropriately searching the PF. While, the local area selection strategy tends to deliver a balanced evolutionary. The significance of the two proposed strategies has been clearly shown in the results. The results also show that our algorithm could greatly outperform related methods to be compared.

The proposed method can be extended in several directions. Firstly, it is interesting to employ region partitioning technique such as zoning search [50] to design the LAS and LCS. In this regard, their performance can be compared. Secondly, it is desirable to employ clustering with crowding degree to extract the solutions from multiple populations as the final output. Additionally, employing the proposed algorithm to address problems including image segmentation [51], parameter estimation [52], [53] and nonlinear system control [54], [55], [56], [57] can also be investigated.

ACKNOWLEDGMENT

(Shenghao Zhou and Ze Chen contributed equally to this work.)

REFERENCES

- [1] F. Rivas-Dávalos and M. R. Irving, "An approach based on the strength Pareto evolutionary algorithm 2 for power distribution system planning," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2005, pp. 707–720.
- [2] Q. Fan, W. Wang, and X. Yan, "Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator for chemical and biochemical dynamic optimization problems," *Appl. Soft Comput.*, vol. 59, pp. 33–44, Oct. 2017.
- [3] B. R. Campomanes-Álvarez, O. Cordon, and S. Damas, "Evolutionary multi-objective optimization for mesh simplification of 3D open models," *Integr. Comput.-Aided Eng.*, vol. 20, no. 4, pp. 375–390, Aug. 2013.
- [4] Q. Fan, Y. Zhang, and N. Li, "An autoselection strategy of multiobjective evolutionary algorithms based on performance indicator and its application," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 2422–2436, Jul. 2022.
- [5] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.
- [6] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.
- [7] 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, Apr. 2000.
- [8] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.
- [9] 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, Apr. 2014.
- [10] W. Huang, Y. Zhang, and L. Li, "Survey on multi-objective evolutionary algorithms," *J. Phys., Conf. Ser.*, vol. 1288, no. 1, Aug. 2019, Art. no. 012057.
- [11] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.
- [12] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 892–899.
- [13] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," in *Proc. Int. Conf. Learn. Intell. Optim.* Cham, Switzerland: Springer, 2013, pp. 70–74.
- [14] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.
- [15] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [16] Y. Tian, X. Zhang, R. Cheng, C. He, and Y. Jin, "Guiding evolutionary multiobjective optimization with generic front modeling," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1106–1119, Mar. 2020.

- [17] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [18] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Apr. 2013.
- [19] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, Jun. 2014.
- [20] 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, Apr. 2016.
- [21] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected Pareto fronts," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Cham, Switzerland: Springer, 2015, pp. 127–140.
- [22] C. Zhang, K. C. Tan, L. H. Lee, and L. Gao, "Adjust weight vectors in MOEA/D for bi-objective optimization problems with discontinuous Pareto fronts," *Soft Comput.*, vol. 22, no. 12, pp. 3997–4012, Jun. 2018.
- [23] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.
- [24] X. Cai, Z. Yang, Z. Fan, and Q. Zhang, "Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2824–2837, Sep. 2017.
- [25] H. Zhou and J. Qiao, "Multiobjective optimal control for wastewater treatment process using adaptive MOEA/D," *Appl. Intell.*, vol. 49, no. 3, pp. 1098–1126, Mar. 2019.
- [26] J. Qiao, H. Zhou, C. Yang, and S. Yang, "A decomposition-based multiobjective evolutionary algorithm with angle-based adaptive penalty," *Appl. Soft Comput.*, vol. 74, pp. 190–205, Jan. 2019.
- [27] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [28] T. Ando, "Majorization relations for Hadamard products," *Linear Algebra Appl.*, vols. 223–224, pp. 57–64, Jul. 1995.
- [29] H. Jain and K. Deb, "An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Cham, Switzerland: Springer, 2013, pp. 307–321.
- [30] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
- [31] Y. Zhu, Y. Qin, D. Yang, H. Xu, and H. Zhou, "An enhanced decomposition-based multi-objective evolutionary algorithm with a self-organizing collaborative scheme," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 118915.
- [32] L. Cai, S. Qu, Y. Yuan, and X. Yao, "A clustering-ranking method for many-objective optimization," *Appl. Soft Comput.*, vol. 35, pp. 681–694, Oct. 2015.
- [33] R. Denysiuk, L. Costa, and I. E. Santo, "Clustering-based selection for evolutionary many-objective optimization," in *Proc. Int. Conf. Parallel Problem Solving Nature.* Cham, Switzerland: Springer, 2014, pp. 538–547.
- [34] H. Zhang, S. Song, A. Zhou, and X.-Z. Gao, "A clustering based multi-objective evolutionary algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 723–730.
- [35] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [36] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems.* New York, NY, USA: Springer, 2007, vol. 5.
- [37] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, p. 115–148, Apr. 1995.
- [38] 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, Oct. 2016.
- [39] S. Wang, H. Zhang, Y. Zhang, A. Zhou, and P. Wu, "A spectral clustering-based multi-source mating selection strategy in evolutionary multi-objective optimization," *IEEE Access*, vol. 7, pp. 131851–131864, 2019.
- [40] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nature.* Berlin, Germany: Springer, 2004, pp. 832–842.
- [41] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2005, pp. 280–295.
- [42] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1, May 2002, pp. 825–830.
- [43] Q. Zhang, S. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Univ. Essex, Colchester, U.K. Nanyang Technol. Univ., Singapore, Tech. Rep., 264, 2008, pp. 1–30.
- [44] W. Wang, S. Yang, Q. Lin, Q. Zhang, K.-C. Wong, C. A. C. Coello, and J. Chen, "An effective ensemble framework for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 645–659, Aug. 2019.
- [45] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, Dec. 2016.
- [46] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, Aug. 2015.
- [47] Y. Hua, Y. Jin, and K. Hao, "A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2758–2770, Jul. 2019.
- [48] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, Jun. 2017.
- [49] Y. Liu, H. Ishibuchi, N. Masuyama, and Y. Nojima, "Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 439–453, Jun. 2020.
- [50] Q. Fan and X. Yan, "Solving multimodal multiobjective problems through zoning search," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 8, pp. 4836–4847, Aug. 2021.
- [51] N. Zeng, H. Li, Z. Wang, W. Liu, S. Liu, F. E. Alsaadi, and X. Liu, "Deep-reinforcement-learning-based images segmentation for quantitative analysis of gold immunochromatographic strip," *Neurocomputing*, vol. 425, pp. 173–180, Feb. 2021.
- [52] B. Shen, Z. Wang, and H. Qiao, "Event-triggered state estimation for discrete-time multidelayed neural networks with stochastic parameters and incomplete measurements," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1152–1163, May 2017.
- [53] X. Ge, Q.-L. Han, and Z. Wang, "A dynamic event-triggered transmission scheme for distributed set-membership estimation over wireless sensor networks," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 171–183, Jan. 2019.
- [54] Y. Yuan, H. Yuan, Z. Wang, L. Guo, and H. Yang, "Optimal control for networked control systems with disturbances: A delta operator approach," *IET Control Theory Appl.*, vol. 11, no. 9, pp. 1325–1332, Jun. 2017.
- [55] E. Tian, Z. Wang, L. Zou, and D. Yue, "Chance-constrained H_∞ control for a class of time-varying systems with stochastic nonlinearities: The finite-horizon case," *Automatica*, vol. 107, pp. 296–305, Jan. 2019.
- [56] Q. Li, B. Shen, Z. Wang, T. Huang, and J. Luo, "Synchronization control for a class of discrete time-delay complex dynamical networks: A dynamic event-triggered approach," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1979–1986, May 2019.
- [57] M. Wang, Z. Wang, H. Dong, and Q.-L. Han, "A novel framework for backstepping-based control of discrete-time strict-feedback nonlinear systems with multiplicative noises," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1484–1496, Apr. 2021.



SHENGHAO ZHOU received the B.Sc. degree in computer science from the Jiaxing University Nanhui College. He is currently pursuing the M.Sc. degree in computer science with Hangzhou Normal University. His research interests include evolutionary computation, optimization, and machine learning.



ZE CHEN received the master’s degree in project management from the University of Quebec, Canada, in 2015. He is currently working as an Experimenter with Hangzhou Normal University. His research interests include data mining, machine learning, and intelligent health.



ZHOUCHEG BAO is currently pursuing the M.Sc. degree in electronic information with Hangzhou Normal University. His research interests include meta-heuristic algorithms and data mining.



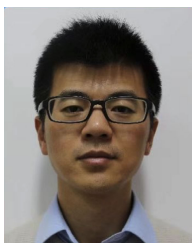
QI LI received the B.Eng. degree in electrical engineering and automation from the Jiangsu University of Technology, Changzhou, China, in 2013, and the Ph.D. degree in control science and engineering from Donghua University, Shanghai, China, in 2018. From June 2016 to July 2016, she was a Research Assistant with the Department of Mathematics, Texas A&M University at Qatar, Qatar. From November 2016 to November 2017, she was a Visiting Ph.D. Student with the Department of Computer Science, Brunel University London, U.K. She is currently an Associate Professor with Hangzhou Normal University, Hangzhou, China. Her current research interests include network communication, complex networks, and sensor networks. She is a very active reviewer for many international journals.



WENDA HE is currently pursuing the M.Sc. degree in electronic information with Hangzhou Normal University. His research interests include machine learning and semi-supervised algorithms.



MENGJUN GU received the B.Sc. degree in information technology from Zhejiang Sci-Tech University, in 2006, and the M.B.A. degree from the Zhejiang University of Technology, in 2009. He is currently working with China Telecom Zhejiang Branch. His research interests include neural networks and digital system construction.



WEIGUO SHENG (Member, IEEE) received the M.Sc. degree in information technology from the University of Nottingham, U.K., in 2002, and the Ph.D. degree in computer science from Brunel University, U.K., in 2005. Then, he worked as a Researcher with the University of Kent, U.K., and Royal Holloway, University of London, U.K. He is currently a Professor with Hangzhou Normal University. His research interests include evolutionary computation, data mining/clustering, pattern recognition, and machine learning.

...