A K-Means Clustering-Based Hybrid Offspring Generation Mechanism in Evolutionary Multi-Objective Optimization

YUHONG WANG and YI ZHANG
School of Mechanical Engineering, Changzhou University, Changzhou 213164, China
Corresponding author: Yi Zhang (zy@cczu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51875053, and in part by the China Ministry of Science and Technology Key Research and Development Program under Grant 2018YFC1903101.

ABSTRACT Model-based recombination operators ignore individual quality information, while genetic-based differential evolution (DE) operators lack the extraction and use of global information. This makes it impossible for the single offspring generation method to always achieve excellent performance on various optimization problems. In order to solve the above problems, the K-means clustering-based hybrid offspring generation mechanism multi-objective evolutionary algorithm (KMDEA) is proposed. KMDEA performs K-means clustering on the population, and builds a multivariate Gaussian model based on the clustering results to discover the global information (the regularity property) of the population. For realizing the fusion of global and individual information, this paper designs a new hybrid offspring generation mechanism (KMD mechanism) to extract and use local individual information. Compared with a variety of mainstream multi-objective evolutionary algorithms (MOEAs), the results show that KMDEA has obvious advantages in solving multi-objective optimization problems (MOPs) with complex characteristics.

INDEX TERMS Evolutionary algorithm, multi-objective optimization, clustering algorithm, hybrid recombination operator, information fusion.

I. INTRODUCTION

This paper considers the following continuous multi-objective optimization problems (MOPs) [28]:

\[
\begin{aligned}
\min F(x) &= (f_1(x), \ldots, f_m(x))^T \\
\text{s.t. } x &= (x_1, \ldots, x_n)^T \in \Omega
\end{aligned}
\]

Among them, \(x = (x_1, \ldots, x_n)^T\) is \(n\) dimensional decision variable, and \(\Omega\) is the space of decision (variable). The decision vector \(x\) is mapped from the decision space to the target space by the mapping function \(F : \Omega \rightarrow \mathbb{R}^m\), and the target vector \(y\) of the \(m\) dimension is obtained. \(m\) represents \(m\) optimization goals.

There are a large number of optimization problems in scientific research and production applications, such as industrial design, job scheduling, and resource allocation. With regard to MOPs, since Schaffer [29] first applied an evolutionary algorithm to solve MOPs, the studies on multi-objective evolutionary algorithms (MOEAs) have been a hot topic in the field of optimization [30]. Furthermore, MOEAs can conveniently solve the complex optimization problems, which are difficult to be solved by traditional optimization methods.

According to the basic ideals adopted by MOEAs, MOEAs can be roughly divided into the following three categories [2]: (a) MOEAs based on Pareto dominance, such as the non-dominated sorting genetic Algorithm 2 (NSGA-II) [24] and improved strength Pareto evolutionary algorithm (SPEA2) [23]; (b) MOEAs based on evolutionary performance metrics, such as the indicator-based evolutionary algorithm (IBEA) [31] and S-metric selection evolutionary multi-objective optimization algorithm (SMS-EMOA) [17]; (c) MOEAs based on decomposition (MOEA/D), MOEA/D [32] is different from the above two kinds of algorithms, as it is not only a specific algorithm, but also has a general algorithm framework that can be incorporated into evolutionary strategies [33].

Offspring recombination and environment selection are the two main operations of MOEAs. Offspring recombination aims to generate new individuals, and environment
selection determines which trial individuals can survive. These two operations are equally important in improving the performance of the algorithm. However, the existing MOEAs mainly focus on the design and analysis of environment selection operators, and pay little attention to recombination operators [3]. Currently, a large number of MOEAs directly apply recombination operators which are designed for single-objective optimization problems to produce new solutions [2]. In fact, the single-objective optimization problems usually obtain one or more global optimal offspring, while the multi-objective optimization problems (MOPs) usually have a set of optimal trade-off solutions (called Pareto front, referred to as PF). The corresponding solutions of the PF in the decision space are called the Pareto set (PS). It is worth mentioning that the PS of the continuous multi-objective optimization problem presents a (m-1)-dimensional manifold structure in the variable space [3]. Since there is an essential difference in the topology between single-objective optimization problems and multi-objective optimization problems, the direct application of single-objective optimization recombination operator to MOEAs cannot guarantee the best offspring effect [2].

In evolutionary algorithms, according to the way of generating new offsprings, recombination operators can be roughly divided into two categories, namely genetic-based recombination operators and model-based recombination operators [1]. The model-based recombination operator approximates the manifold structure of the Pareto solution set by establishing a probability model, and generates new solutions by sampling from the model [4]. It uses the global statistical information of the population to estimate the distribution of the solution set at a macro level, but it ignores the individual information [5]. The genetic-based differential evolution (DE) operator uses the information of individual differences to guide its further search, in which it lacks the extraction and use of global information [6]. Therefore, it has become a promising research direction to mix the two operators to generate new solutions.

In reference [7], part of the new offspring is sampled from the modified univariate histogram probability model, and the rest is generated by refining the parent individual without any functional evaluation of the cheap local search method. In 2016, Li et al. put forward MOEA/D-CMA [8], the algorithm decomposes a multi-objective optimization problem into multiple single-objective optimization problems. In the evolution process, only one selected sub-problem in each group is optimized by the probability model, while the other sub-problems are optimized by the DE operator. In 2017, Bing et al. [9] improved the sampling process of RM-MEDA, mapping the individuals on the model to the hidden space, and used the crossover mutation operation in DE operator to generate new individuals. In the existing hybrid new offspring generation method, sampling the model is helpful to improve the mining ability of the algorithm [10], [11]. The introduction of DE operator is beneficial to improve the exploration ability of the algorithm in offspring space [12], [18].

Researches have shown that in MOEAs, mating with similar individuals can improve the quality of new solutions and accelerate the convergence of algorithms [39], [40]. Therefore, some mating restriction strategies have been proposed. Zhang et al. [26] used a predefined mating restriction probability \( \delta \) to select parents from the same class, and selected parents from all solutions with a probability of 1-\( \delta \). In reference [34], based on the establishment of a predefined mating restriction probability to construct a mating pool, the value of this parameter \( \delta \) is adaptively adjusted according to the recombination utility of two parental sources. In addition, many mating restriction strategies [35], [14] have been proposed. It is not difficult to see that most of these algorithms currently use probability values to determine new solutions for global exploration or local exploitation. These mating restrictions pay little attention to individual quality information, they also require the setting of multiple control parameters, e.g., sharing radius [37], candidate size [38].

In the evolution process, each individual has different performances in convergence and diversity. Individuals with better quality need more mining; individuals with poor quality need more exploration [36]. Therefore, in order to improve the performance of the DE operator, we can use individual quality information to guide the construction of the mating pool.

Based on the above analysis, and inspired by the existing achievements, this paper proposes a K-means clustering-based hybrid offspring generation mechanism multi-objective evolutionary algorithm (KMDEA). The innovations of this paper are stated as follows:

- Compared with the traditional clustering-based MOEAs, KMDEA constructs a multivariate Gaussian model according to the clustering results of the initial population. The Gaussian model based on clustering algorithm is used to approximate the manifold structure and detect the regularity property.
- Compared with the existing hybrid recombination operators, the model sampling offspring is used as the mating parent to expand the offspring population generated by DE operator.
- In the process of mating pool construction, different from the previous mating restriction strategy in the form of probability, this paper directly uses deterministic information to construct mating pool.

The rest of the paper is organized as follows. Section II presents the proposed algorithm in details. Section III presents the experimental results, and section IV provides sensitivity to control parameters on KMDEA. Section V concludes the paper with some remarks for future work.

II. A K-MEANS CLUSTERING BASED HYBRID OFFSPRING GENERATION MECHANISM MOEA

A. THE FRAMEWORK OF KMDEA

Algorithm 1 gives the algorithm framework of KMDEA. In the initialization phase, we first get the initial population \( P \) (line 1). In the iterative process, the fast
non-dominated sorting scheme [24] is used to divide the population \( P \) into \( L \) different nondominated fronts, in which \( B_1 \) denotes the best front, and \( B_L \) indicates the worst front (line 3). In line 4, the K-means clustering is performed for the population \( P \) to obtain clustering results \( \{C^K\} \).

Lines 5-12 are the hybrid new solution generation mechanism, which is briefly described here. The above mechanism first constructs a mixture Gaussian model (line 6) for the individuals in the best dominant front in each class, and samples the model to obtain the model offspring \( y_{\text{mod}} \) (line 7). Then, the model sampling offspring is added to the mating pool as the mating parent to expand the mating pool (line 9). In this way, the sampled offsprings can participate in the generation of new offsprings as both offspring and mating parents, so as to enhance the local exploitation ability of the next neighbor-based mating. Finally, the mating pool \( Q \) is established for the parent individuals in each cluster, and the DE operator is used to generate a new solution \( y \) (lines 10-11). The result of fast non-dominated sorting is directly applied in the construction of mating pool \( Q \). The non-dominated solutions in each class are used for local exploitation, while the dominated solutions are used for global exploration.

### B. K-MEANS CLUSTERING

K-means clustering [13] is one of the most widely used clustering algorithms. It is a clustering method based on partition. The so-called partition based clustering algorithm is to divide the data object set into independent subsets according to the similarity between data objects. K-means clustering algorithm concretizes the similarity between data objects into the distance between each object and the pre selected clustering center. This clustering algorithm assigns each object to the nearest cluster center to form a cluster, and iteratively optimizes the square error criterion to improve the quality of each cluster.

In the hybrid offspring generation mechanism based on K-means clustering, K-means algorithm is used to explore the distribution structure (neighbor relationship) [14] of offsprings in the population. Based on the population distribution, a multivariate Gaussian model is constructed for the dominant individuals in each class to approximate the population structure, and the model offspring is generated by sampling.

### C. GENERATE NEW SOLUTIONS

Gauss model is one of the most widely used probabilistic models [2]. A multivariate Gauss distribution of random variables \( x = (x_1, x_1, \ldots, x_n)^T \) can be expressed as \( x \sim N(\mu, \Psi) \). Among them, \( \mu \) is the mean vector, \( \Psi \) is the covariance matrix. The corresponding probability density function of random variable is expressed as:

\[
p(x) = (2\pi)^{-\frac{n}{2}} |\Psi|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Psi^{-1} (x - \mu) \right\} \quad (2)
\]

#### Algorithm 1 Framework of KMDEA

**Input:**
- \( K \): Number of cluster categories;
- \( N \): Population size;
- \( T \): Maximum evolutionary algebra;

**Output:**
- Population \( P \);
- Population \( P \);
- For \( t = 1 : T \) do
- Fast non-dominated sorting for \( P \), obtain results \( B = \{B_1, \ldots, B_L\} \);
- K-means clustering is performed on \( P \), to obtain the clustering results \( C = \{C_1, \ldots, C^K\} \);
- for \( k = 1 : K \) do
- \( (\mu, \Psi) = \text{modeling}(C^k \cap B_1) \);
- \( y_{\text{mod}} = \text{GaussianSample}(\mu, \Psi) \);
- for \( i = 1 : |C^k| \) do
- Expanding mating pool with sampling results:
- \( Q = Q \cup y_{\text{mod}} \);
- Set mating pool \( Q \):
- \( Q = \begin{cases} C^k & \text{if } x \in C^k \cap B_1 \\ M & \text{otherwise} \end{cases} \)
- \( M \) represents a global subpopulation consisting of individuals randomly selected from each category.
- \( y = \text{SolGen}(x^i, Q_1, \circ \text{DE}) \);
- End
- End
- Environment selection: \( P = \text{Select}(P, y) \);
- End

For a given set of data \( x^1, x^2, \ldots, x^K \), The mean vector and covariance matrix [15] are estimated as follows:

\[
\mu = \frac{1}{K} \sum_{k=1}^{K} x^k \quad (3)
\]

\[
\Psi = \frac{1}{K-1} \sum_{k=1}^{K} (x^k - \mu)(x^k - \mu)^T \quad (4)
\]

The process of sampling a new point \( x = (x_1, x_1, \ldots, x_n)^T \) by the multivariate Gaussian probability model is as follows:

#### Algorithm 2 GaussianSample

1. A lower triangular matrix \( A \) is obtained by Cholesky decomposition covariance matrix, which satisfies \( \Psi = AA^T \);
2. Generate a single factor Gaussian distribution vector \( y = (y_1, y_2, \ldots, y_n)^T \), where \( y_j \sim N(0, 1), j = 1, 2, \ldots, n \) obeys the standard normal distribution.
3. Command \( x = \mu + Ay \).

DE operator [16] is a way to generate new offsprings for KMDEA. Its specific details are shown in Algorithm 3. In the process of generating the new offspring, the DE operator generates the initial offspring, and then the polynomial mutation operator is used to mutate the new offspring. In order to make the new offspring feasible, the new offspring is repaired.
according to the situation before and after mutation operation. In Algorithm 3, F and CR are the parameters of DE operator, \( p_m \) is the probability of variation, \( \eta_m \) is the distribution index of the mutation operator.

\[
\text{Algorithm 3 } y^i = \text{SolGen}(x^i, Q^i, @DE)
\]

**Input:**
- \( Q \): Mating pool
- \( x \): Current individual

**Output:**
- New individual: \( y = (y_1, y_2, ..., y_n)^T \)
  1. Two parental individuals \( x^{i1} \) and \( x^{i2} \) were randomly selected from mating pool \( Q^i \);
  2. Generate a new initial offspring \( y^i = (y_1^i, y_2^i, ..., y_n^i)^T \), where \( i = 1, ..., n \)
     \[
y_i^i = \begin{cases} 
    x_i + F \times (x_{i1}^{i1} - x_{i2}^{i2}) & \text{if } \text{rand}() < CR \\
    x_i & \text{otherwise}
  \end{cases}
\]
  3. Repair the offspring \( y_i^i \)
     \[
y_i^r = \begin{cases} 
    a_i^r & \text{if } y_i^i < a_i \\
    b_i^r & \text{if } y_i^i > b_i \\
    y_i^i & \text{otherwise}
  \end{cases}
\]
  4. Mutate the offspring \( y_i^r \)
     \[
y_i = \begin{cases} 
    y_i^r + \delta_i \times (b_i^r - a_i^r) & \text{if } \text{rand}() < p_m \\
    y_i^r & \text{otherwise}
  \end{cases}
\]
     \[
\delta_i = \begin{cases} 
    2(1 - r) \frac{y_i^r - a_i^r}{b_i^r - a_i^r} & \text{if } r < 0.5 \\
    1 - 2(1 - r) \left( \frac{y_i^r - a_i^r}{b_i^r - a_i^r} \right)^{1/3} & \text{otherwise}
  \end{cases}
\]
     where \( r = \text{rand}() \) is a random number.
  5. Return the new offspring \( y \).

**D. ENVIRONMENT SELECTION**

The purpose of environment selection operation is to protect the effective offspring. KMDEA adopts an environment selection method based on the super volume index in SMS-EMOA algorithm [17]. Algorithm 4 gives the details of the environment selection method adopted by KMDEA.

In line 1, a new solution and the current population are combined, and the fast non-dominated sorting scheme is used to divide the population \( P \) into \( L \) different non-dominated fronts, in which \( B_1 \) denotes the best front, \( B_L \) indicates the worst front. Then, the individual with the least amount of hypervolume in the worst front \( B_L \) is found for removal (lines 2-3). Details of the calculation of hypervolume can be found in [17].

**E. TIME COMPLEXITY**

At each generation, the operations of the developed algorithm include 1) a K-mean clustering procedure; 2) hybrid offspring generation mechanism. It is known that the time complexity of K-means is \( O(NKnI) \), where \( I \) is the number of iterations for training. For the second operation, the time complexity is \( O(Nn) \). Thus in total, the time complexity is \( O(NKnT + Nn) \).

**Algorithm 4** \( P = \text{Select}(P, y^i) \)

**Input:**
- \( P \): Population
- \( y^i \): New generation of individuals

**Output:**
- the updated population \( P \)
  1. Fast non-dominated sorting for \( P \cup y^i \), obtain results \( B = \{B_1, ..., B_L\} \);
  2. Identifying the worst individuals:
     \[
x^* = \arg \max_{x \in P \cup y^i} \Delta \phi(x, B_L);
\]
  3. Delete the worst individuals \( P = P \setminus x^* \);
  4. Return \( P \).

**III. EXPERIMENTAL STUDY**

**A. TEST PROBLEMS AND PERFORMANCE METRICS**

This paper selects GLT1-GLT6 and LZ1-LZ9, 15 target test functions to test the performance of KMDEA algorithm. Among them, the GLT test set [18] has complex PF frontiers, while the LZ test set [6] has a complex PS structure. More details show in Table 1. The performance measures are inverted generation distance (IGD) and hypervolume (HV).

1) \text{INVERTED GENERATION DISTANCE (IGD)}

The inverse generation distance evaluation index is a comprehensive performance evaluation index [21]. It mainly reflects the convergence and distribution performance of the algorithm by calculating the average value of the minimum distance from the uniform point on the Pareto surface to the non-dominated offspring set, it is defined as follows:

\[
\text{IGD } (P^*, P) = \frac{\sum x^* \in P^* d(x^*, P)}{|P^*|}
\]

where \( d(x^*, P) \) refers to the shortest Euclidean distance between the point \( x^* \) and the individual set \( P \) obtained by the algorithm. The \( |P^*| \) refers to the number of midpoint of the \( P^* \). The smaller the IGD value, the better the comprehensive performance (convergence and diversity) of the algorithm.

2) \text{HYPERVOLUME METRIC (HV)}

The hypervolume represents the volume of the hypercube surrounded by the obtained Pareto offspring set and the reference point, it is an index to evaluate the comprehensive performance of the algorithm [20]. It is defined as follows:

\[
\text{HV } (P, r) = \text{VOL} \left( \bigcup_{x \in P} \left[ x_1, r_1 \right] \times \cdots \times \bigcup_{x \in P} \left[ x_n, r_n \right] \right)
\]

where \( r = (r_1, ..., r_n) \) represents the reference point of the Pareto optimal offspring in the target space. The \( \left[ x \right] \) represents the volume of hypercube composed from reference points and non-dominated individuals. The \( \text{VOL}() \) represents Lebesgue measure, and reference [22]
TABLE 1. Properties and settings of the test suites. Obj and Var: dimensions of the objective space and the search space.

| Instance | Obj | Var | Characteristics                                      |
|----------|-----|-----|------------------------------------------------------|
| GLT1     | 2   | 10  | Disconnected, Nonlinear variable linkage             |
| GLT2     | 2   | 10  | Convex, Nonlinear variable linkage                   |
| GLT3     | 2   | 10  | Convex, Nonlinear variable linkage                   |
| GLT4     | 2   | 10  | Convex, Disconnected, Nonlinear variable linkage     |
| GLT5     | 3   | 10  | Convex, Nonlinear variable linkage                   |
| GLT6     | 3   | 10  | Convex, Disconnected, Nonlinear variable linkage     |
| LZ1      | 2   | 30  | Convex, Complex PS                                   |
| LZ2      | 2   | 30  | Convex, Complex PS                                   |
| LZ3      | 2   | 30  | Convex, Complex PS                                   |
| LZ4      | 2   | 30  | Convex, Complex PS                                   |
| LZ5      | 2   | 30  | Convex, Complex PS                                   |
| LZ6      | 3   | 10  | Convex, Complex PS                                   |
| LZ7      | 2   | 10  | Convex, Multiple-peak, Complex PS                    |
| LZ8      | 2   | 10  | Convex, Multiple-peak, Complex PS                    |
| LZ9      | 2   | 30  | Concave, Complex PS                                  |

points out the specific calculation method of Lebesgue measure. The larger the HV value, the better the performance of the algorithm.

B. COMPARISON OF KMD MECHANISM WITH MULTIPLE ENVIRONMENT SELECTION OPERATORS

In order to verify the validity of the K-means clustering-based hybrid offspring generation mechanism, we combine the above mechanisms into 3 different environment selection operators, and get KMDSP, KMDNS and KMDEA. The three environment selection operators are respectively from the SPEA2 [23], NSGA-II [24] and SMS-EMOA [17]. This paper compares the above six algorithms. The parameter settings of all algorithms are as follows, where the DE parameters and mutation parameters are set as recommended in [6].

**Common parameter settings**
- Population size: $N = 100$;
- maximum evolutionary generation: $T = 300$;
- DE parameters: $F = 0.5$, $CR = 1$;
- mutation parameters: $p_m = \frac{1}{n}$, $\eta_m = 20$.

**KMD mechanism combines algorithm parameter settings**
- Number of clusters: $K = 5$.

Among them, SPEA2, NSGA-II and SMS-EMOA do not need to set special parameters except public parameters.

From Table 2, we can see that for the GLT test set with complex PF characteristics, the combination algorithm of KMD mechanism obtains the best IGD value on GLT1-GLT5 test questions. Although the GLT6 test question did not get the best value, the KMD mechanism combined algorithm also has a good performance. This is because SPEA2, NSGA-II and SMS-EMOA ignore the regularity property, they are not suitable for solving complex PS or PF test suites. In the three KMD mechanism combination algorithm, the KMDEA has the best performance, and obtains the optimal value of the three test questions of GLT1, GLT2 and GLT3. This is because the hypervolume indicator is known to be “Pareto compliant” [21]. It has been shown that the hypervolume indicator based environmental selection approach has exhibited superior performance against those based on dominance and decomposition [27]. This shows that the K-means clustering-based hybrid offspring generation mechanism is effective in enhancing the performance of MOEAs.

C. COMPARISON BETWEEN KMDEA AND VARIOUS ALGORITHMS

In order to verify the performance of KMDEA, this paper chooses IMMOEA [25], MOEA/D-CMA [8], RM-MEDA [3] and SMEA [26] as the comparison algorithm. All comparison algorithms use the best parameters in the original literature. All algorithms are programmed in MATLAB. The specific parameters of each algorithm are set as follows:

**Public parameters**
- population size: $N = 100$;
- maximum evolutionary generation: $T = 300$;
- DE parameters: $F = 0.5$, $CR = 1$;
- mutation parameters: $p_m = \frac{1}{n}$, $\eta_m = 20$.

**Parameters of IM-MOEA**
- number of reference vectors: $K = 5$;
- size of random group: $L = 3$.

**Parameters of MOEA/D-CMA**
- Number of Gaussian models: $K = 5$. 

167646
TABLE 2. Statistics (mean(standard dev.)) of IGD metric values of final populations obtained by SPEA2, KMDSP, NSGA-II, KMDNS, SMSEMOA and KMDEA algorithms with 20 independent runs on GLT test suites.

| Instance | SPEA2 | KMDSP | NSGA-II | KMDNS | SMSEMOA | KMDEA |
|----------|-------|-------|---------|-------|---------|-------|
| GLT1     | 1.2147e-2 (1.08e-2) | 2.4738e-3 (8.59e-5) | 9.9372e-3 (6.54e-3) | 2.9852e-3 (1.44e-4) | 1.4708e-2 (1.24e-2) | 1.8765e-3 (8.11e-5) |
| GLT2     | 3.3258e-2 (1.57e-3) | 3.0099e-2 (1.21e-3) | 3.4154e-2 (1.73e-3) | 3.2947e-2 (1.05e-3) | 3.4111e-2 (1.57e-3) | 2.8705e-2 (6.57e-4) |
| GLT3     | 4.6174e-2 (2.73e-3) | 7.2446e-3 (5.52e-3) | 1.8753e-2 (2.23e-2) | 5.8137e-3 (3.98e-4) | 5.2647e-2 (1.81e-2) | 6.6860e-3 (5.51e-3) |
| GLT4     | 4.2621e-2 (5.11e-3) | 1.2964e-2 (2.19e-2) | 5.1287e-2 (5.83e-2) | 3.0737e-2 (7.91e-2) | 4.8046e-2 (5.25e-2) | 5.1055e-3 (3.56e-4) |
| GLT5     | 4.2194e-1 (1.06e-3) | 4.1588e-2 (1.41e-3) | 4.9984e-2 (2.35e-3) | 4.7461e-2 (2.40e-3) | 9.7301e-2 (2.35e-2) | 4.8570e-2 (2.31e-3) |
| GLT6     | 4.8264e-2 (1.90e-2) | 9.2183e-2 (6.63e-2) | 5.2711e-2 (1.03e-2) | 1.0261e-1 (6.26e-2) | 1.1372e-1 (5.10e-2) | 1.1111e-1 (6.25e-2) |

TABLE 3. Statistics (mean(standard dev.)) of IGD metric values of final populations obtained by IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and KMDEA algorithms with 30 independent runs on GLT and LZ test suites.

| Instance | IM-MOEA | MOEA/D-CMA | RM-MEDA | SMEA | KMDEA |
|----------|---------|------------|---------|------|-------|
| GLT1     | 1.9794e-2 (2.68e-3) | 3.6289e-3 (1.97e-5) | 1.8295e-2 (2.72e-2) | 5.6584e-3 (4.93e-3) | 1.8422e-3 (1.37e-5) |
| GLT2     | 4.3127e-1 (2.02e-1) | 3.0270e-1 (5.66e-2) | 3.0181e-2 (6.42e-4) | 3.0877e-2 (1.09e-3) | 2.8752e-2 (9.19e-4) |
| GLT3     | 1.0807e-1 (2.68e-2) | 2.4594e-2 (7.49e-3) | 1.7026e-2 (1.80e-2) | 2.5628e-2 (1.94e-2) | 6.9963e-3 (8.06e-3) |
| GLT4     | 2.0823e-2 (5.03e-3) | 2.7036e-2 (5.42e-2) | 2.8402e-2 (5.36e-2) | 2.4374e-2 (4.76e-2) | 1.0993e-2 (3.31e-2) |
| GLT5     | 7.2539e-2 (5.73e-3) | 6.3258e-2 (2.40e-3) | 4.9774e-2 (2.36e-3) | 8.5097e-2 (1.93e-2) | 4.7444e-2 (2.64e-3) |
| GLT6     | 7.8509e-2 (2.27e-2) | 8.2069e-2 (4.32e-2) | 5.4242e-2 (5.66e-3) | 1.0248e-1 (3.88e-2) | 5.5230e-2 (2.25e-2) |
| LZ1      | 7.4994e-3 (3.47e-4) | 4.0388e-3 (4.00e-5) | 4.3985e-3 (1.36e-4) | 3.6890e-3 (2.50e-5) | 3.6215e-3 (1.19e-5) |
| LZ2      | 6.9035e-2 (1.35e-2) | 4.0952e-2 (7.62e-3) | 6.8680e-2 (1.40e-2) | 3.7282e-2 (8.50e-3) | 3.5512e-2 (1.60e-2) |
| LZ3      | 3.7288e-2 (5.81e-3) | 2.5004e-2 (2.04e-2) | 3.1987e-2 (7.48e-3) | 2.7566e-2 (3.66e-3) | 2.4931e-2 (1.22e-2) |
| LZ4      | 3.4461e-2 (1.18e-2) | 3.4617e-2 (2.38e-2) | 3.3047e-2 (2.52e-3) | 2.8869e-2 (4.06e-3) | 2.6927e-2 (2.06e-2) |
| LZ5      | 3.6473e-2 (1.48e-2) | 2.4991e-2 (4.02e-3) | 2.6224e-2 (5.92e-3) | 2.7969e-2 (4.93e-3) | 2.2420e-2 (6.11e-3) |
| LZ6      | 1.5724e-2 (4.52e-2) | 1.2411e-1 (1.89e-1) | 2.2556e-1 (7.23e-2) | 2.7424e-1 (8.18e-2) | 5.3255e-1 (1.81e-1) |
| LZ7      | 1.3689e-3 (2.88e-4) | 4.5504e-3 (3.39e-4) | 2.6172e-1 (9.07e-2) | 1.4105e-1 (8.35e-2) | 2.4684e-2 (3.32e-2) |
| LZ8      | 1.7237e-2 (2.39e-2) | 2.1611e-2 (1.18e-2) | 2.2712e-1 (8.18e-2) | 1.6598e-1 (5.13e-2) | 9.3775e-2 (2.90e-2) |
| LZ9      | 6.9941e-2 (1.54e-2) | 7.0784e-2 (5.94e-3) | 8.6924e-2 (1.38e-2) | 4.7048e-2 (1.10e-2) | 3.5944e-2 (1.63e-2) |

+/–= 1/14/0  2/12/1  1/13/1  1/13/1

1) STATISTICAL ANALYSIS

| Parameters of RM-MEDA |
|-----------------------|
| cluster number in local PCA: $K = 5$; |
| extension rate of sampling: $\phi = 0.25$. |

| Parameters of SMEA |
|-------------------|
| initial learning rate: $\tau = 0.7$; |
| size of neighborhood mating pools: $H = 5$; |
| probability of mating restriction: $\beta = 0.9$. |

| Parameters of KMDEA |
|-------------------|
| Number of clusters: $K = 5$. |

For the results of Wilcoxon rank sum test, the + labeled in back of a result denotes that the compared algorithm is better than KMDEA; in contrast, the - means that KMDEA is outperformed by the compared algorithm; while the = means that there is no statistically significant difference between the results obtained by KMDEA and the compared algorithm. The best statistical results of the above test problems are highlighted.

Table 3 shows the analysis results of IGD, among which IMMOEA, MOEA/D-CMA, RM-MEDA, SMEA and KMDEA algorithm get 1, 2, 1, 1 and 10 best indicators respectively. Table 4 shows the analysis results of the HV index. Each algorithm obtains 1, 4, 1, 1 and 8 best values respectively. In the 30 comparison results of all other comparison algorithms, for the performance of Wilcoxon rank sum test.
TABLE 4. Statistics (mean(std. dev.)) of HV metric values of final populations obtained by IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and KMDEA algorithms with 30 independent runs on GLT and LZ test suites.

| Instance | IM-MOEA | MOEA/D-CMA | RM-MEDA | SMEA | KMDEA |
|----------|---------|------------|---------|------|-------|
| GLT1     | 4.2064e-1 (9.25e-3) | **4.8116e-1 (3.99e-5)** | 4.4289e-1 (4.01e-2) | 4.6092e-1 (1.67e-2) | 4.8082e-1 (4.01e-4) |
| GLT2     | 7.4683e-1 (2.59e-2) | 7.9018e-1 (4.86e-3) | 8.1451e-1 (3.90e-4) | 8.1796e-1 (1.57e-4) | **8.1933e-1 (3.20e-5)** |
| GLT3     | 9.4622e-1 (1.71e-3) | 9.5366e-1 (8.89e-4) | 9.5543e-1 (1.43e-3) | 9.5574e-1 (1.68e-3) | **9.5763e-1 (7.73e-4)** |
| GLT4     | 5.6937e-1 (3.62e-3) | 5.6274e-1 (6.07e-2) | 5.7623e-1 (1.47e-2) | 5.6518e-1 (5.20e-2) | **5.7711e-1 (3.63e-2)** |
| GLT5     | 9.5305e-1 (1.78e-3) | **9.6922e-1 (6.70e-4)** | 9.5434e-1 (2.41e-3) | 9.2811e-1 (1.46e-2) | 9.5714e-1 (2.31e-3) |
| GLT6     | 9.3023e-1 (2.22e-2) | 9.4230e-1 (2.97e-2) | 9.4420e-1 (4.66e-3) | 9.1072e-1 (3.47e-2) | **9.4545e-1 (2.19e-2)** |
| LZ1      | 7.1392e-1 (5.55e-4) | 7.1943e-1 (1.07e-4) | 7.1873e-1 (2.68e-4) | 7.2036e-1 (5.97e-5) | **7.2065e-1 (1.27e-5)** |
| LZ2      | 6.2492e-1 (2.01e-2) | 6.6318e-1 (1.36e-2) | 6.2498e-1 (2.07e-2) | 6.6955e-1 (1.29e-2) | **6.7348e-1 (2.08e-2)** |
| LZ3      | 6.8132e-1 (4.34e-3) | **6.9398e-1 (1.25e-2)** | 6.8759e-1 (6.27e-3) | 6.8933e-1 (4.26e-3) | 6.9328e-1 (7.72e-3) |
| LZ4      | 6.8775e-1 (6.35e-3) | 6.8319e-1 (1.93e-2) | 6.8566e-1 (3.03e-3) | 6.8848e-1 (4.82e-3) | 6.9290e-1 (1.45e-2) |
| LZ5      | 6.8459e-1 (7.48e-3) | 6.9279e-1 (3.93e-3) | 6.9390e-1 (5.18e-3) | 6.9056e-1 (4.35e-3) | **6.9598e-1 (6.80e-3)** |
| LZ6      | 3.8493e-1 (2.80e-2) | **4.6403e-1 (1.04e-2)** | 2.8445e-1 (8.26e-2) | 2.1689e-1 (8.01e-2) | 7.7252e-2 (5.82e-2) |
| LZ7      | 5.2269e-1 (4.82e-2) | **7.1834e-1 (4.80e-4)** | 3.5461e-1 (1.26e-1) | 5.4660e-1 (1.03e-1) | 6.9120e-1 (4.23e-2) |
| LZ8      | 4.7029e-1 (3.34e-2) | **6.9312e-1 (1.76e-2)** | 3.9950e-1 (1.03e-1) | 4.5952e-1 (6.61e-2) | 5.8312e-1 (4.22e-2) |
| LZ9      | 3.4754e-1 (2.09e-2) | 3.4116e-1 (7.61e-3) | 3.4126e-1 (1.65e-2) | 3.8419e-1 (1.36e-2) | **4.0073e-1 (2.36e-2)** |

+/−/ = 1/14/0 4/9/2 1/13/1 1/13/1

FIGURE 1. Evolution of the best IGD values and corresponding standard deviations obtained by other 5 algorithms and KMDEA with 30 independent runs on GLT.

test at 5% significant level, KMDEA algorithm obtained 28, 21, 26 and 26 significant advantages IGD and HV indicators.

In general, compared with IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA, KMDEA algorithm has certain advantages in GLT test suite with complex PF and LZ test suite with complex PS. Specifically, for IM-MOEA, RM-MOEA and SMEA, they use the single offspring generation mechanism. These algorithms only obtained optimal values on a few test problems, while KMDEA showed advantages on almost the entire test suite. For MOEA/D-CMA, although the algorithm also uses DE operator and covariance matrix adaptive strategies to generate new solutions,
the performance of MOEA/D-CMA is much lower than that of KMDEA. The main reason for this result is that the MOEA/D-CMA fails to take full account of the connectivity of PS and the regular property of MOPs.

2) SEARCH EFFICIENCY

In order to analyze the search efficiency of KMDEA algorithm, figure 1 shows the evolution curve of the optimal IGD index after all 30 independent operations of GLT test.

It can be seen from figure 1 that KMDEA has a very clear advantage for GLT3 and GLT4. For the remaining GLT test problems, although there are comparison algorithms with similar performance to KMDEA, it is undeniable that KMDEA can still obtain the smallest IGD index value at the fastest speed. The comparison results show that for the GLT
and builds a Gaussian mixture model based on the clustering results. Then, the sampling results of the model are used to expand the mating pool to make up for the shortcomings of the single offspring generation mechanism. Furthermore, in order to balance the local exploitation and global exploration, this paper uses individual quality information to guide the construction of the mating pool. Finally, the K-means clustering-based hybrid offspring generation mechanism multi-objective evolutionary algorithm (KMDEA) is proposed.

In this paper, the K-means clustering-based hybrid offspring generation mechanism is integrated into the environment selection operators of SMS-EMOA, SPEA2 and NSGA-II to verify the effectiveness of the mechanism. After that, comparing KMDEA with IMMOEA, MOEA/D-CMA, RMMEDA, SMEA and other mainstream multi-objective evolutionary algorithms, the results show that KMDEA has obvious advantages in solving MOPs with complex characteristics. Finally, parametric sensitivity analysis is carried out, and the results show that KMDEA has good robustness.

The follow-up work will focus on the following points: (a) improvement of clustering algorithm; (b) combining other multi-objective optimization algorithms in the KMDEA framework, such as particle swarm optimization, ant colony optimization, etc.

REFERENCES

[1] Y. Li, A. Zhou, and G. Zhang, “An MOEA/D with multiple differential evolution mutation operators,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jul. 2014, pp. 397–404.
[2] A. M. Zhou, Q. F. Zhang, and G. X. Zhang, “Multi-objective evolutionary algorithm based on mixture Gaussian models,” J. Softw., vol. 25, no. 5, pp. 913–928, 2014.
[3] Q. Zhang, A. Zhou, and Y. Jin, “RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm,” IEEE Trans. Evol. Comput., vol. 12, no. 1, pp. 41–63, Feb. 2008.
[4] P. Larraaga and J. A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, New York, NY, USA: Springer, 2002.
[5] E. Bengoetxea, P. Larrañaga, I. Bloch, and A. Perchant, “Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems,” in Energy Minimization Methods in Computer Vision and Pattern Recognition, Berlin, Germany: Springer, Sep. 2001.
[6] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 284–302, Apr. 2009.
[7] A. Zhou, J. Sun, and Q. Zhang, “An estimation of distribution algorithm with cheap and expensive local search methods,” IEEE Trans. Evol. Comput., vol. 19, no. 6, pp. 807–822, Dec. 2015.
[8] H. Li, Q. Zhang, and J. Deng, “Biased multiobjective optimization and decomposition algorithm,” IEEE Trans. Cybern., vol. 47, no. 1, pp. 1–15, Jan. 2016.
[9] B. Dong, A. Zhou, and G. Zhang, “Sampling in latent space for a multiobjective estimation of distribution algorithm,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jul. 2016, pp. 3027–3034.
[10] 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, Oct. 2009.
[11] J. Sun, Q. Zhang, and E. Tsang, “DE/EDA: A new evolutionary algorithm for global optimization,” Inf. Sci., vol. 169, nos. 3–4, pp. 249–262, Feb. 2005.
[12] J. Sun, H. Zhang, A. Zhou, Q. Zhang, and K. Zhang, “A new learning-based adaptive multi-objective evolutionary algorithm,” Swarm Evol. Comput., vol. 44, pp. 304–319, Feb. 2019.
[13] J. B. Macqueen, “Some methods for classification and analysis of multivariate observations,” in Proc. 5th Berkeley Symp. Math. Statist. Probability, 1967, pp. 281–297.

[14] X. Li, H. Zhang, and S. Song, “A self-adaptive mating restriction strategy based on survival length for evolutionary multiobjective optimization,” Swarm Evol. Comput., vol. 43, pp. 31–49, Dec. 2018.

[15] N. Hansen, “The CMA evolution strategy: A tutorial,” Tech. Rep., 2005.[Online]. Available: https://www研究gate.net/publication/248797670

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

[17] 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, Sep. 2007.

[18] A. Zhou, Q. Zhang, Y. Jin, and B. Sendhoff, “Combination of EDA and DE for continuous biobjective optimization,” in Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.), Jun. 2008, pp. 1447–1454.

[19] F. Gu, H. L. Liu, and K. C. Tan, “A multiobjective evolutionary algorithm using dynamic weight design method,” Int. J. Innov. Comput. Inf. Control, vol. 8, no. 5B, pp. 3677–3688, 2012.

[20] E. Zitzler and L. Thiele, “Multiobjective optimization using evolutionary algorithms—A comparative case study,” in Proc. Int. Conf. Parallel Problem Solving from Nature—PPSN VIII, vol. 1, no. 1, pp. 32–49, 2011.

[21] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. D. Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” IEEE Trans. Evol. Comput., vol. 7, no. 2, pp. 117–132, Apr. 2003.

[22] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithms,” Comput. Eng. New. Lab., Dept. Elect. Eng., Swiss Federal Inst. Technol. (ETH) Zurich, Zurich, Switzerland, Tech. Rep., 2001, pp. 95–100, vol. 3242, no. 103.

[23] 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. 2002.

[24] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, “A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling,” IEEE Trans. Evol. Comput., vol. 19, no. 6, pp. 838–856, Dec. 2015.

[25] H. Zhang, A. Zhou, S. Song, Q. Zhang, X.-Z. Gao, and J. Zhang, “A self-organizing multiobjective evolutionary algorithm,” IEEE Trans. Evol. Comput., vol. 20, no. 5, pp. 792–806, Oct. 2016.

[26] J. Bader and E. Zitzler, “HypE: An algorithm for fast hypervolume-based many-objective optimization,” Evol. Comput., vol. 19, no. 4, pp. 45–76, Mar. 2011.

[27] S. Mardle and K. M. Miettinnen, “Nonlinear multiobjective optimization,” J. Oper. Res. Soc., vol. 51, no. 2, p. 246, 1999.

[28] D. J. Schaffer, “Multiple objective optimization with vector evaluated genetic algorithms,” in Proc. 1st Int. Conf. Genetic Algorithms. Hillsdale, NJ, USA: L. Erlbaum, 1985, pp. 93–100.

[29] 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.

[30] E. Zitzler and S. Künzli, “Indicator-based selection in multi-objective search,” in Parallel Problem Solving from Nature—PPSN VIII, in Lecture Notes in Computer Science, vol. 3242. Berlin, Germany: Springer, 2004, pp. 832–842.

[31] Q. Zhang and H. Li, “MOEA/D: A multi-objective evolutionary algorithm based on decomposition,” in Proc. IEEE Int. Conf. Adv. Learn. Technol., Dec. 2005, pp. 712–731.

[32] H.-L. Liu, F. Gu, and Q. Zhang, “Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems,” IEEE Trans. Evol. Comput., vol. 18, no. 3, pp. 450–455, Jun. 2014.

[33] H. Zhang, X. Zhang, S. Song, and X.-Z. Gao, “An affinity propagation-based multiobjective evolutionary algorithm for selecting optimal aiming points of missiles,” Soft Comput., vol. 21, pp. 3013–3031, Jun. 2017.

[34] 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. 851–864, 2019.

[35] X. Li, S. Song, and H. Zhang, “Evolutionary multiobjective optimization with clustering-based self-adaptive mating restriction strategy,” Soft Comput., vol. 23, no. 3, pp. 3–25, 2018.

[36] J. Horn, N. Nafpliotis, and D. E. Goldberg, “A niched Pareto genetic algorithm for multiobjective optimization,” in Proc. 1st IEEE Conf. Evol. Comput. IEEE World Congr. Comput. Intell., Jun. 1994, pp. 82–87.

[37] H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, “An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization,” Eur. J. Oper. Res., vol. 188, no. 1, pp. 57–75, 2008.

[38] J. Gao, “WBMOIGA: Weight-based multiobjective immune genetic algorithm and its application,” in Proc. Int. Conf. Intell. Autom., Jun. 2009, pp. 1–6.

[39] H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, “An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization,” Eur. J. Oper. Res., vol. 188, no. 1, pp. 57–75, 2008.