



A practical regularity model based evolutionary algorithm for multiobjective optimization

Wanpeng Zhang^a, Shuai Wang^{b,*}, Aimin Zhou^b, Hu Zhang^{c,*}

^a College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, Hunan, China

^b Shanghai Key Laboratory of Multidimensional Information Processing, School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

^c Beijing Electro-mechanical Engineering Institute, Beijing 100074, China

ARTICLE INFO

Article history:

Received 8 September 2021

Received in revised form 8 August 2022

Accepted 29 August 2022

Available online 7 September 2022

Keywords:

Multiobjective optimization

Evolutionary algorithm

Regularity model

Offspring generation

ABSTRACT

It is well known that domain knowledge helps design efficient problem solvers. The regularity model based multiobjective estimation of distribution algorithm (RM-MEDA) is such a method that uses the regularity property of continuous multiobjective optimization problems (MOPs). However, RM-MEDA may fail to work when dealing with complicated MOPs. This paper aims to propose some practical strategies to improve the performance of RM-MEDA. We empirically study the modeling and sampling components of RM-MEDA that influence its performance. After that, some new components, including the population partition, modeling, and offspring generation procedures, are designed and embedded in the regularity model. The experimental study suggests that the new components are more efficient than those in RM-MEDA when using the regularity model. The improved version has also been verified on various complicated benchmark problems, and the experimental results have shown that the new version outperforms five state-of-the-art multiobjective evolutionary algorithms.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Without loss of generality, in this work, the following box-constrained continuous multiobjective optimization problem (MOP) is considered [1]:

$$\begin{aligned} &\text{minimize} && F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ &\text{s.t.} && \mathbf{x} \in \Omega \end{aligned} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is a decision variable vector, $\Omega \subset \mathbb{R}^n$ is the decision (or variable) space; $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ denotes the mapping from the decision space to the objective space, which has m optimization objectives $f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})$. Multiobjective optimization is known from that it obtains the best trade-offs among these objectives, namely, the Pareto optimal solution set (PS), and the mapping of PS in the objective space is the Pareto front (PF) [2].

Since multiobjective evolutionary algorithms (MOEAs) can approximate the PF of a MOP in a single run, they have flourished in the last decades [3]. As far as MOEAs are concerned, both environment selection operator and offspring reproduction operator are two principal components in MOEAs [3–5]. The former

operator selects promising solutions as the parents of next generation, while the latter operator generates new trial solutions as the offspring of the current population. However, most MOEAs focus on environmental selections [4,6]. Generally, in terms of the selection operators, MOEAs can be classified into three categories, including the Pareto dominance based MOEAs [7–9], performance indicators based MOEAs [10–12] and decomposition based MOEAs [5,13–15].

Offspring generation operators are crucial to the performance of MOEAs as well [4]. With the development of evolutionary multiobjective optimization (EMO), different offspring generation operators have been used, e.g., differential evolution (DE) [16–19], particle swarm optimization (PSO) [20–22], estimation of distribution algorithm (EDA) [23,24]. In fact, the above offspring generation operators designed for single-objective optimization problems are usually embedded into EMO approaches without any modification, and they may not have a good performance for solving MOPs [4,25–27].

To design an efficient offspring generation operator, a general principle is that domain knowledge should be considered. In practice, two issues are critical to the use of such knowledge. The first one is which knowledge to use, and the second one is how to use it. For the first one, it has been proved that under mild conditions, the PS (PF) of a continuous MOP forms a piecewise $(m-1)$ -dimensional manifold in the decision space (the objective space), where m is the number of the objectives. This manifold

* Corresponding authors.

E-mail addresses: wangshuai515658@163.com (S. Wang), jxzhangu@126.com (H. Zhang).

property is called regularity property [28]. For the second one, a conceptual method, called regularity model based multiobjective estimation of distribution algorithm (RM-MEDA), is proposed to adopt the regularity property for offspring generation in EMO [4]. In each generation, RM-MEDA utilizes the local principal component analysis (local PCA) [29] method to partition the current population, and then builds $(m - 1)$ -D regularity models for each cluster to approximate the manifold of PSs explicitly. New trial solutions are sampled from the so built regularity model together with Gaussian noise vectors. RM-MEDA has been verified that it performs well on MOPs with variable linkages.

The regularity model based approaches have been well studied and applied in EMO. In [30], a regularity model with a reducing redundant cluster operator is proposed to build more precise models during the evolution. As suggested in [31], the regularity model is embedded into existing MOEAs to deal with noisy MOPs. Moreover, the regularity model is also used to solve different kinds of problems, such as many objective, multi-modal and dynamic, etc [32–34]. In fact, some manifold learning based offspring generation and latent space generation paradigms are proposed [35–38], which adopt the similar idea of RM-MEDA. What is more, a few mating restriction strategies are designed to choose parent solutions by extracting the regularity property [6,26,27,39–41].

However, in the past few years, it has been realized that there exist some drawbacks in RM-MEDA, which mainly influence the efficiency of RM-MEDA and are related to population modeling and offspring generation, i.e., how to use the regularity property practically.

- (1) **Manifold property may not be satisfied:** Although the Pareto optimal set of a continuous MOP defines a piecewise $(m - 1)$ -D manifold structure in the decision (or objective) space, in the early stages, the population solutions scatter in search space and do not show regularity. In fact, this case might happen to all MOEAs [6,26], but RM-MEDA always assumes that the population forms a $(m - 1)$ -D manifold along the whole evolutionary process.
- (2) **Regularity model may fail to capture the PS structure:** RM-MEDA builds K linear models with Gaussian perturbations to approximate the PS structure. The quality of the models depends on the distribution of the current solutions and the clustering method [31]. In the case of the solutions scattered chaotically or the clustering method failing, the learned regularity models cannot capture the population structure [30].
- (3) **Individual location information is ignored:** It has been shown that both the population distribution and individual information are useful in offspring generation [6,42]. However, RM-MEDA only utilizes the population distribution information, which may mislead the search when the probabilistic model fails.

What is more, the sensitivity of control parameters and high computational overhead also limit the efficiency of RM-MEDA [26]. Considering the above limitations, this paper proposes a practical approach, named regularity model based multiobjective evolutionary algorithm (RMEA), for continuous MOPs. Like RM-MEDA, RMEA is based on the regularity property. The main contributions focus on the offspring generation, in which the population is partitioned into different disjoint clusters and each parent solution for modeling will generate trial solutions by sampling from the regularity model. Unlike RM-MEDA, in RMEA, only the nondominated solutions are chosen to build a regularity model for each partition. Instead of using the Gaussian perturbation for sampled trial vectors, a permutation based on the genetic reproduction procedure is applied in RMEA.

The rest of this paper is organized as follows: Section 2 reviews and analyzes the regularity property in RM-MEDA. Section 3 presents the details of RMEA, including the population partition, the modeling and the offspring generation procedures. Section 4 empirically studies the generation operator in RMEA. Section 5 gives a systematic comparative study of RMEA with some state-of-the-art MOEAs on several test suits. Finally, Section 6 concludes the paper with some remarks for future work.

2. Regularity property in RM-MEDA

In this section, we empirically study the use of the regularity property in RM-MEDA. The following two test functions, F_1 and F_2 , are used in the study, where F_1 is the GLT1 test instance in [43] and F_2 is the LZ1 test instance in [5].

$$F_1 : \begin{cases} f_1 = (1 + g(x))x_1 \\ f_2 = (1 + g(x))(2 - x_1 - \text{sign}(\cos(2\pi x_1))) \end{cases} \quad (2)$$

$$F_2 : \begin{cases} f_1 = x_1 + \frac{2}{J_1} \sum_{j \in J_1} \left(x_j - x_1^{0.5(1 + \frac{3(j-2)}{n-2})} \right)^2 \\ f_2 = 1 - \sqrt{x_1} + \frac{2}{J_2} \sum_{j \in J_2} \left(x_j - x_1^{0.5(1 + \frac{3(j-2)}{n-2})} \right)^2 \end{cases} \quad (3)$$

For F_1 , $x = (x_1, x_2, \dots, x_n) \in [0, 1] \times [-1, 1]^{n-1}$, $n = 10$, $g(x) = \sum_{i=2}^n (x_i - \sin(2\pi x_1 + \frac{i}{n}\pi))^2$. For F_2 , $x = (x_1, x_2, \dots, x_n) \in [0, 1]$, $n = 10$, J_1 is odd and J_2 is even. Both F_1 and F_2 show complicated characteristics, e.g., F_1 has disconnected PF in the objective space and there are non-linear linkages among variables, and F_2 has complicated PS shape in the decision space.

RM-MEDA implements the regularity property by using an offspring generator, in which a new solution y is generated as:

$$y = \zeta + \varepsilon \quad (4)$$

where ζ is a trial vector sampled from the built regularity model, ε is a Gaussian perturbation vector that $\varepsilon \sim N(0, \sigma I)$, and the deviation σ represents the perturbed level.

To build regularity models, a local PCA method is used to partition the current population into clusters in RM-MEDA. In each cluster, an $(m - 1)$ -D linear model is built by the PCA approach, and the perturbation parameter σ is calculated based on the residual error. When generating an offspring solution, a cluster is chosen, a trial vector ζ is sampled from the regularity model, and then a perturbation ε is added to the trial vector to form an offspring solution y . More details are referred to [4].

Our analysis considers both the modeling and sampling processes in RM-MEDA. In the study, the population size is 100, and RM-MEDA terminates after 300 generations (30000 function evaluations). The other parameters of RM-MEDA are the same as in its original paper.

- **Modeling process:** We execute RM-MEDA on F_1 and F_2 , and record the models and generated offspring solutions. Fig. 1 shows the results in the decision space. It is clear that in the early evolution stage, the 30th and 60th generations in our cases, the population solutions are far from the real PSs and do not show regularities. Consequently, some learned regularity models fail to approximate the real PSs and cannot sample high-quality offspring solutions. What is more, as shown in Fig. 1(a), an inaccurate partition result also influences the accuracy of models in RM-MEDA.
- **Sampling process:** Fig. 2 shows the final populations obtained by RM-MEDA on F_1 and F_2 in the objective space. In Figs. 2(a) and (b), RM-MEDA is executed without the

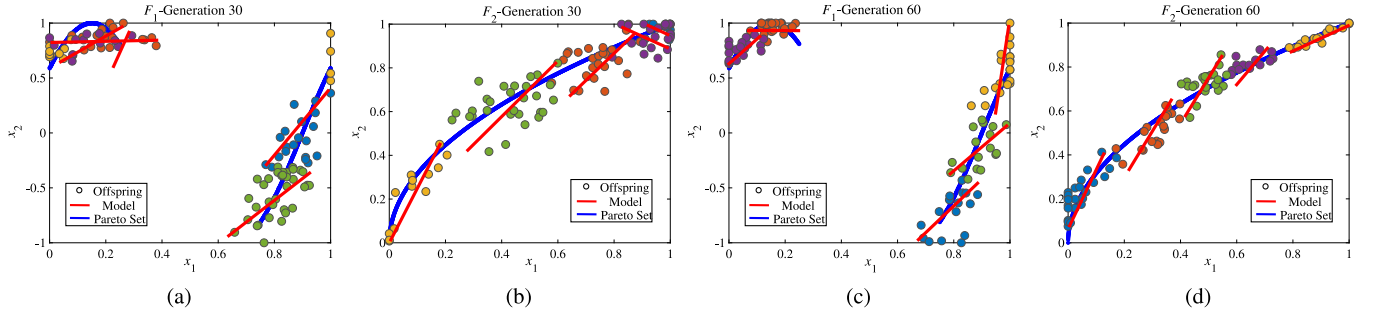


Fig. 1. Regularity models and offspring solutions obtained by RM-MEDA on F_1 and F_2 in the 30th and 60th generations in the decision space. The solutions with different colors indicate different partitions. The Models in figures are obtained by the $x_1 - x_2$ on F_1 and F_2 .

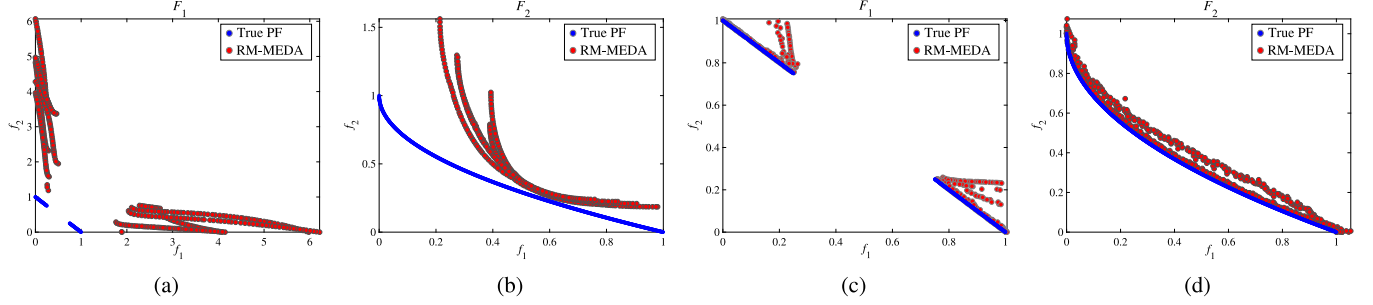


Fig. 2. The 5 final populations in the objective space obtained by RM-MEDA on F_1 and F_2 : (a) and (b) are without the Gaussian perturbation, and (c) and (d) are with the Gaussian perturbation.

Gaussian perturbation on the sampled trial solutions. It is clear that RM-MEDA converges prematurely. In Figs. 2(c) and (d), RM-MEDA is executed with the Gaussian perturbation, and RM-MEDA cannot ensure that all populations converge to the real PF. This comparison study suggests that (a) only sampling from the regularity models may lead to premature convergence, and (b) the Gaussian perturbation can improve the performance of RM-MEDA to some degree, but the results are not always satisfactory.

From the above study and analysis, we can conclude that RM-MEDA does not use the regularity property efficiently in both population modeling and offspring generation. It might be possible to improve the algorithm from the following three angles: (a) building more accurate regularity models, (b) using other methods to replace the noise perturbation to make regularity models work efficiently, and (c) combining with genetic reproduction to prevent misleading when models fail to capture the regularity property as suggested in [6,42,44].

3. RMEA: the proposed method

Section 2 suggests that the use of prior domain knowledge is crucial in algorithm design. This section aims to improve the performance of RM-MEDA, and RMEA mainly uses the following two practical strategies.

- On model building: we use a spectral clustering method [45] to partition the population and use only the nondominated solutions to build regularity models.
- On offspring generation: we use genetic reproduction, the differential vector and polynomial mutation (PM), to generate perturbation vectors for these trial vectors.

3.1. Algorithm framework

Fig. 3 gives the basic procedures of RMEA at each generation. In the figure, *Partition* denotes that the population is partitioned

into several clusters by a clustering method. In each cluster, the nondominated solutions are used to build a regularity model, and new trial vectors are sampled from the built regularity model in *Sampling*. In *Perturbation*, the parent solutions in the same cluster are adopted to generate perturbation vectors. In this way, the offspring solutions are generated in RMEA.

Algorithm 1: RMEA Framework

```

1 Randomly initialize  $Pop = \{x^1, \dots, x^N\}$ ;
2 for  $t = 1, \dots, T$  do
3   Sort  $Pop$ :  $\{B_1, \dots, B_L\} = NDS(Pop)$ ;
4   Partition  $Pop$  into  $K$  clusters  $C_k$  and build a model  $\psi^k$ 
     on solutions in  $C_k \cap B_1$  for each cluster  $C_k$ 
     ( $k = 1, \dots, K$ );
5   foreach  $x \in C_k$  ( $k = 1, \dots, K$ ) do
6     Generate offspring:  $y = Generate(x, C_k, \psi^k, B_1)$ ;
7     Evaluate  $y$  and update  $Pop$ :  $Pop = Select(Pop, y)$ ;
8   end
9 end
10 Return  $Pop$ .
```

The following notations will be used in the descriptions of RMEA:

- N : the population size;
- K : the number of clusters;
- T : the maximum number of generations.

The framework of the RMEA is given in Algorithm 1, and we have some comments on Algorithm 1 as follows:

- Initialization (line 1):** RMEA maintains a population Pop , where Pop is initialized by randomly generating a set of N solutions;
- Stopping condition (line 2):** The algorithm will be terminated when the maximum number of generations is reached.

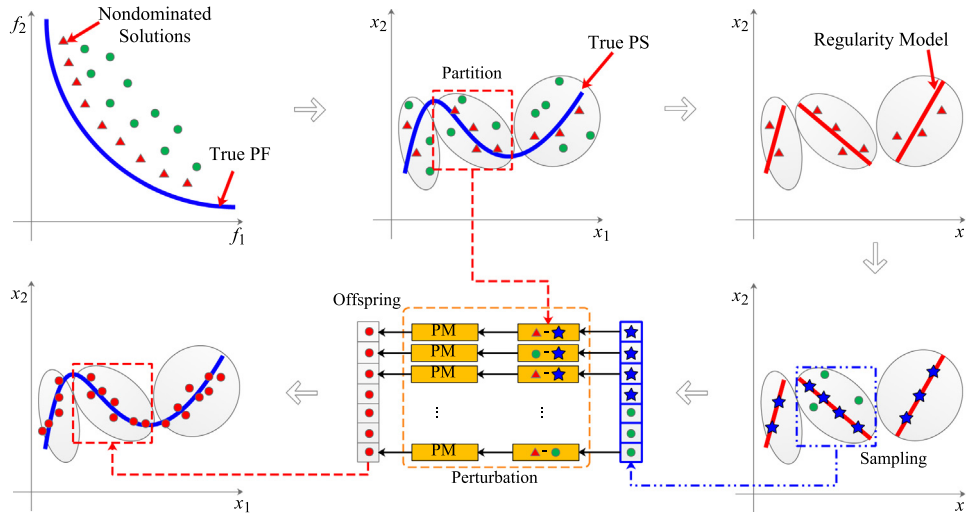


Fig. 3. The basic procedure of RMEA in one generation.

- (c) **Nondominated sorting (line 3):** The fast nondominated sorting method in NSGA-II [7] is applied to partition Pop into a set of fronts $\{B_1, \dots, B_L\}$, where B_i indicates the i th best front, and B_1 contains the nondominated solutions.
- (d) **Partition and regularity model building (line 4):** A spectral clustering method is employed to partition population Pop into several clusters in the decision space. The nondominated solutions within the cluster are chosen to build a regularity model by PCA for each cluster.
- (e) **Offspring generation (line 6):** For the nondominated solutions, new trial vectors are sampled from regularity models, and for the dominated solutions, they are directly chosen as the trial vectors. A perturbation is generated with the difference between the trial vector and its neighborhood population solution together with the PM operator. An offspring solution is generated by adding a trial vector and a perturbation vector.
- (f) **Environment selection (line 7):** The population is updated with the new trial solutions by the S-metric based selection [11].

In the following, we will indicate the modeling, partition, offspring generation, and selection procedures, respectively.

Algorithm 2: Spectral Clustering

Input :
Data set: P ;
The number of clusters: K ;

Output:
Clustering result: $C = \{C^1, C^2, \dots, C^K\}$;

- 1 Build similarity matrix W and degree matrix D , $W, D \in R^{n \times n}$;
- 2 Calculate the Laplacian matrix L : $L = D^{-1}(D - W)$;
- 3 Compute the first K eigenvectors v_1, \dots, v_K of L ;
- 4 Use the $V \in R^{n \times K}$ as the column vector to form the feature matrix U , $U = \{u_1, u_2, \dots, u_K\}$, $U \in R^{n \times K}$;
- 5 The normalization operation is performed on each row of the feature matrix U , $u^{ij} = u^{ij} / \sqrt{\sum_K u_{iK}^2}$;
- 6 The U is clustered by the K-means algorithm into clusters C^1, \dots, C^K ;
- 7 **Return** clustering results, $C = \{C^1, C^2, \dots, C^K\}$;

3.2. Partition and modeling

The modeling component in RM-MEDA can be regarded as a pattern of “clustering + PCA”, and it has been indicated that the clustering approach can help the population approximate the PS manifold [6]. It is argued that a wrong partition will mislead the modeling process [30]. In this work, spectral clustering is used to extract the population structure, which has been verified to deal with complex PS structures well in our previous work [46].

3.2.1. Spectral clustering

The spectral clustering is based on graph theory [45], which can partition data with any shape. The data points in high-dimensional space are reduced to a low-dimensional space, and then they are partitioned by another clustering algorithm, such as the K-means [47], in the low-dimensional space. Therefore, spectral clustering algorithms can deal with large-scale and complex data sets. A general spectral clustering method can be summarized into the following three main steps.

- Step 1: Define a measurement of the similarity between data points and construct a similarity matrix.
- Step 2: Calculate the eigenvectors corresponding to the first K (same as the cluster number) large eigenvalues of the similarity matrix and construct a new data set in the feature space with these eigenvectors.
- Step 3: Apply a classical clustering algorithm to the data points in the feature space and obtain a partition result $C = \{C_1, C_2, \dots, C_K\}$.

In this work, the spectral clustering based on random walk Laplacian matrix is used [48]. Random walk in graph theory is a random process that jumps between vertices. In spectral clustering, the division of the graph ensures that random walk stays in the same cluster, and rarely travels to other clusters. The random walk can effectively overcome the shortcomings of sensitive initial values in traditional clustering, which makes clustering more efficient and accurate. Algorithm 2 gives the framework of spectral clustering based on a random walk.

In Algorithm 2, the similarity matrix W is defined as Eq. (5), w_{ij} denotes Gaussian similarity, σ is the Gaussian similarity bandwidth, which is usually set to be $\sigma = 1$. D is the diagonal matrix, and the value on the diagonal $d(i, i)$ is defined as Eq. (6). More details are referred to [48].

$$W(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad i, j = 1, \dots, n \quad (5)$$

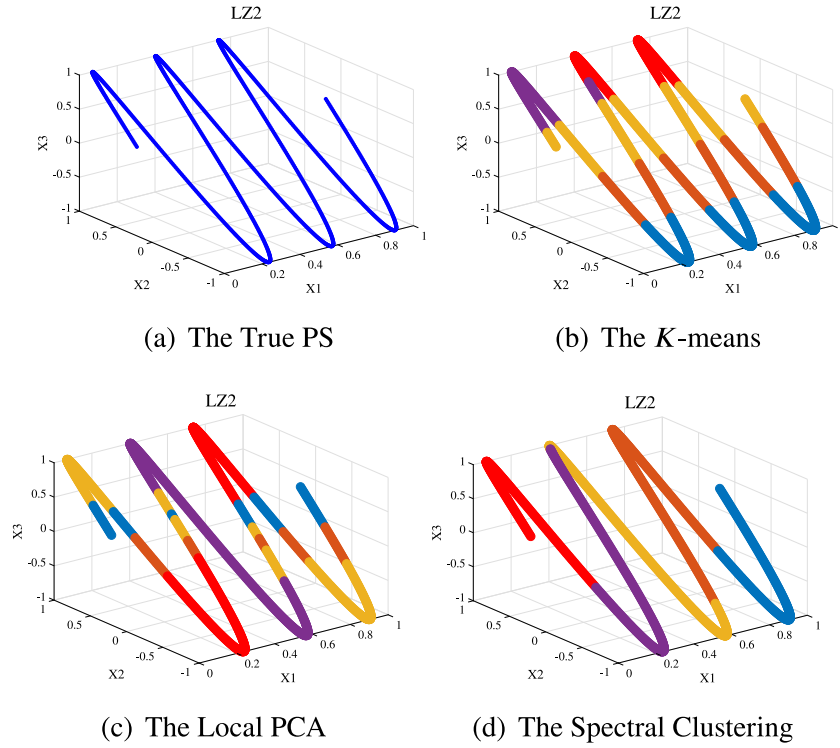


Fig. 4. The partitions obtained by K-means, Local PCA and spectral clustering on the LZ2 test instance.

$$d(i, i) = \sum_j^n w_{ij} \quad (6)$$

We apply the spectral clustering, the K-means and local PCA methods to the PS of LZ2 test instance [5]. The results are demonstrated in Fig. 4 with different colors indicating different partitions. It is clear that on LZ2, the spectral clustering method works much better than K-means and Local PCA. The spectral cluster method can find proper clusters, while K-means and local PCA fail to capture the structure. This suggests that the spectral clustering method is more suitable than K-means and local PCA methods to capture the PS structures in EMO.

3.2.2. Regularity model building

Based on the partition and nondominated sorting results, regularity models are built to approximate the PS. Specifically, an $(m-1)$ -D affine-space Ψ is defined in Eq. (7) for $C_k \cap B_1$, i.e., the nondominated solutions in the cluster.

$$\Psi^k = \{\zeta \in \mathbb{R}^n | \zeta = \bar{x} + \sum_{i=1}^{m-1} \theta_i U_i^k\}, \theta_i \in [a_i^k, b_i^k] \quad (7)$$

where \bar{x} is the centroid of $C_k \cap B_1$, and ζ is the sampled trial vector; $U_i^k (1 \leq i \leq m-1)$ are the $(m-1)$ principal components of Ψ^k , which are obtained by eigen decomposition on covariance matrix $Cov = \frac{1}{|C_k \cap B_1| - 1} \sum_{x \in C_k \cap B_1} (x - x_k)(x - x_k)^T$ of the partition $C_k \cap B_1$; θ_i is the coordinate of x in the i th principal component U_i^k ; a_i^k and b_i^k are the ranges which are defined as Eqs. (8) and (9), respectively. In addition, to make a better approximation of the PS, the range of U_i^k , $[a_i^k, b_i^k]$ are extended by 25% along each principal component. More details about regularity model are referred to [4].

$$a_i^k = \min_{x \in C_k \cap B_1} \{(x - \bar{x})^T U_i^k\} \quad (8)$$

$$b_i^k = \max_{x \in C_k \cap B_1} \{(x - \bar{x})^T U_i^k\} \quad (9)$$

By the above “clustering” and “modeling” steps, the current solutions are partitioned into disjoint clusters and regularity models are built. Moreover, the neighborhood relationship of each cluster is also obtained. Note that if the number of individuals in $C_k \cap B_1$ is less than three, it will be modeled using C_k instead of $C_k \cap B_1$ in our implementation.

3.3. Offspring solution generation

In offspring generation, both regularity model based sampling and genetic operation are used to generate offspring solutions in RMEA. The details of the procedure are given in Algorithm 3, and it is clear that the offspring generation contains two main components in RMEA.

- **Generate a trial vector:** as shown in lines 1–5 in Algorithm 3, if the current solution x is a nondominated one, a new trial vector ζ is sampled from the built regularity model Ψ^k (line 2); otherwise, the current solution is directly chosen as the trial vector (line 4).
- **Generate a perturbation vector:** for the trial vector ζ , the difference between the trial vector and neighborhood population solution in its cluster is employed to generate a perturbation vector ε together with the PM operator in lines 6–7.

Finally, an offspring solution is generated by combining the trial vector and the perturbation vector in line 8, and it is repaired if necessary in line 9. In terms of information fusion, RMEA generates offspring solutions on a chromosome level [42].

3.4. Environment selection

In RMEA, an environment selection based on the hypervolume (HV) metric in SMS-EMOA [11] is used to update the population.

Algorithm 3: $y = \text{Generate}(x, C_k, \Psi^k, B_1)$

Input :
 the parent solution x ;
 the cluster C_k with x ;
 the regularity model Ψ^k of C_k ;
 the nondominated solutions B_1 ;

Output:
 an offspring solution y ;

// Generate a trial vector

1 **if** $x \in B_1$ **then**

2 Sample a trial vector on Ψ^k : $\zeta = \bar{x} + \sum_{i=1}^{m-1} \alpha_i U_i^k$,
 where α_i is a random number from
 $[a_i^k - \frac{1}{4}(b_i^k - a_i^k), b_i^k + \frac{1}{4}(b_i^k - a_i^k)]$;

3 **else**

4 Set $\zeta = x$;

5 **end**

// Generate a perturbation vector

6 Randomly choose one parent x^{r1} from C_k ;

7 Generate a perturbation vector $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ by:

$$\varepsilon_i = \begin{cases} \beta \times (\zeta_i - x_i^{r1}) + \delta_i \times \Delta x_i & \text{if } \text{rand}() < p_m \\ \beta \times (\zeta_i - x_i^{r1}) & \text{otherwise} \end{cases}$$

where $\beta, \delta_i \in [-1, 1]$, is a random value from $[-1, 1]$,
 $i = 1, \dots, n$, $\Delta x_i = \hat{x}_i - \check{x}_i$, and

$$\delta_i = \begin{cases} (2 \times r)^{\frac{1}{m+1}} - 1 & \text{if } r < 0.5 \\ 1 - (2 \times r)^{\frac{1}{m+1}} & \text{otherwise} \end{cases} \quad \text{where}$$

$r = \text{rand}()$ generates a random number from $[0, 1]$;

// Generate an offspring solution

8 Set $y = \zeta + \varepsilon$;

// Repair y if necessary

9 Set $y_i = \begin{cases} \check{x}_i + \frac{1}{2} \text{rand}() (\hat{x}_i - \check{x}_i) & \text{if } y_i < \check{x}_i \\ \hat{x}_i - \frac{1}{2} \text{rand}() (\hat{x}_i - \check{x}_i) & \text{if } y_i > \hat{x}_i \\ y_i & \text{otherwise} \end{cases}$;

10 **Return** the offspring solution y .

When a new solution y is generated, the population is updated. The details of the update method are shown in Algorithm 4.

Algorithm 4: $\text{Pop} = \text{Select}(\text{Pop}, y)$

Input :
 the new solution y ;
 the population Pop ;

Output:
 the updated population Pop ;

1 Partition $\text{Pop} \cup \{y\}$ into L fronts by
 $\{B_1, \dots, B_L\} = \text{NDS}(\text{Pop} \cup \{y\})$.

2 Set $x^* = \arg \min_{x \in B_L} \Delta_\phi(x, B_L)$.

3 Set $\text{Pop} = \text{Pop} \cup \{y\} \setminus \{x^*\}$.

4 **Return** the updated Pop .

In line 1 of Algorithm 4, $\text{Pop} \cup \{y\}$ is divided into L different nondominated fronts by the fast nondominated sorting proposed in NSGA-II [7], where the B_1 is the best front and the B_L is the worst. Then, the individual with the least amount of hypervolume in the worst front B_L is removed (lines 2–3). The details of the calculation of hypervolume Δ_ϕ can be found in [11].

3.5. Remarks

RMEA is a practical version of RM-MEDA, and both of them share basic ideas and algorithm frameworks. Although RMEA

Table 1

Dissimilarities of offspring generation between RMEA and RM-MEDA.

	RM-MEDA	RMEA
Modeling solutions	Whole population	Nondominated solutions
Partition method	Local PCA	Spectral clustering
Perturbation	Gaussian noise	Genetic reproduction
Used information	Global information	Global and local information

Table 2

Number of nondominated solutions over 1000 offspring solutions, which are generated by three strategies.

Generation operator	F_1	F_2
RMEA	653	814
RM-MEDA	410	485
DE and PM	222	558

and RM-MEDA apply differential environment selection operators, their significant difference is in the offspring generation presented in Table 1. It should be noted that this work proposes a general strategy to combine model based and genetic operators to generate high-quality solutions in EMO. Different model based operators, e.g., Covariance matrix adaptation (CMA) [44], Gaussian Model (GM) [6], and genetic operators, e.g., the PSO, can also be used by following the basic idea of RMEA.

4. Efficiency of generation operator in RMEA

This section investigates the efficiency of the offspring generation operators. Three operators are empirically studied.

- The offspring generation operator in RMEA;
- The offspring generation operator in RM-MEDA;
- The DE/rand/1 and polynomial mutation operators.

To have a fair comparison, a population of 100 solutions is randomly sampled from the PSs of F_1 and F_2 respectively, which are ideal parent solutions. The above three offspring generation operators are applied to the parents, and 1000 offspring solutions are generated independently. The generated solutions are shown in Fig. 5.

Regarding the RMEA generation operator, the offspring solutions are around PS and close to the real PS. For the RM-MEDA offspring generation operator, although the offspring solutions are around the true PS, they are far from the PS compared to those of RMEA. It is clear that for the DE and PM operators, a lot of offspring solutions scatter in the middle of the search space or far away from the real PS, where there are no parent solutions. As discussed in Section 1, the reasons might be: (a) the regularity property is not taken into account in DE and PM operators that the current population is taken as parents selected from different Pareto set manifolds; (b) in RM-MEDA, although the PS structures are captured, the perturbation part introduces too much noise to the solutions. Only RMEA utilizes both global and local information for offspring generation, and it can capture the population structure and generate high-quality solutions that follow the hidden structure.

The offspring generation efficiency can also be seen in Table 2. For RMEA, there are more than 600 nondominated solutions among the offspring, e.g., 653 nondominated solutions for F_1 and 814 nondominated solutions for F_2 , which are much higher than those of the other two operators. This section demonstrates that the offspring generation efficiency of RMEA is higher than those of RM-MEDA and DE. The following section will investigate the performance of the proposed RMEA as a MOEA.

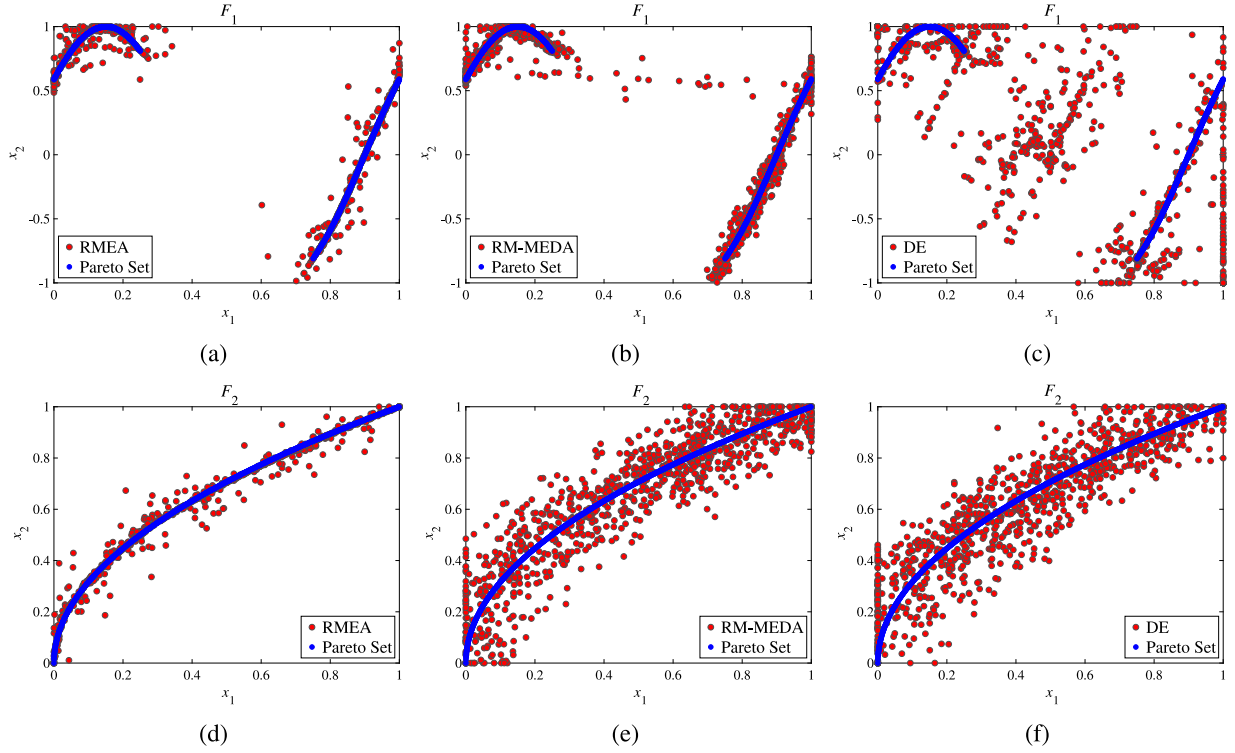


Fig. 5. 1000 offspring solutions on F_1 and F_2 obtained by the RMEA generation operator in (a) and (b), by the RM-MEDA generation operator in (c) and (d), and by the DE and PM operators in (e) and (f).

5. Comparison study

Section 4 demonstrates that the offspring generation efficiency of RMEA is higher than those of RM-MEDA and DE and PM operators. What is the performance of the proposed method in an MOEA? This section investigates its performance. For this reason, five state-of-the-art MOEAs are adopted for comparisons, which also focus on the offspring generation.

- (1) AMEA: one hybrid MOEA combined the Gaussian mixture model and the DE operator in a probabilistic way [27].
- (2) IM-MOEA: an inverse Gaussian model based MOEA [49], which samples trial solutions from the objective space and then maps them into the decision space.
- (3) MOEA/D-CMA: a decomposition based MOEA that uses the Covariance matrix adaptation (CMA) evolution strategy [50] and the DE operator to generate offspring solutions [44].
- (4) RM-MEDA: a kind of multiobjective estimation of distribution algorithm based on regularity model [4], the predecessor of the proposed RMEA.
- (5) SMEA: one learning-based MOEA, which performs a general mating restriction strategy by extracting the regularity property [26].

5.1. Test suites and performance metrics

To investigate the performance of the RMEA, 15 test instances, GLT1-GLT6 (10 dimension variables) [43] and LZ1-LZ9 (30 dimension variables for LZ1-LZ5, LZ7-LZ9, and 10 dimension variables for LZ6) [5], with different complex characteristics, are used for this purpose. GLT1-GLT6 have complicated PFs and LZ1-LZ9 have complicated PSs. Among these test instances, GLT1-GLT4, LZ1-LZ5, and LZ7-LZ9 are bi-objective MOPs, GLT5-GLT6 and LZ6 have three objectives.

We used the weakly Pareto-compliant inverted generational distance (IGD^+) [51] and HV [52] metrics as performance indicators for assessing the performance of the compared MOEAs. They are used to measure the convergence and diversity of the solutions found by MOEAs. In terms of the HV metric, since the PFs of these test suites are known, we set the reference point in HV as $1.1 * \max PF[f_1, \dots, f_m]$.

5.2. Experimental settings

Parameter settings have a significant influence on the performance of the algorithms. Therefore, we choose the best parameters in the original literature for each comparison algorithm. All the experiments are implemented on the PlatEMO to create a fair environment for comparison [53]. In terms of GLT and LZ test suites, the population size of all the six algorithms is set to 100, and the stopping condition is determined by the total number of generations which is 600 in each run of six algorithms. What is more, the DE control parameters: $F = 0.5$; and the PM control parameters: $p_m = 1/n$, $\eta_m = 20$. To ensure the fairness of the experiment, the parameter $CR = 1$ is set for all algorithms with the DE operator. The specific parameters of each algorithm are set as follows.

- Parameters of AMEA

- number of clusters: $K = 5$.
- initial control probabilities : $\beta_0 = 0.9$, $\alpha_0 = 0.05$.

- Parameters of IM-MOEA

- size of random group: $L = 3$;
- number of reference vectors: $K = 10$.

- Parameters of MOEA/D-CMA

- number of groups: $K = 5$.

- Parameters of RM-MEDA

Table 3

Statistical results (mean(std.dev.)(rank)) obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RMEA over 30 independent runs on the GLT and LZ test suites in terms of the IGD⁺ and HV metrics.

Instances	AMEA	IM-MOEA	MOEA/D-CMA	RM-MEDA	SMEA	RMEA
IGD ⁺						
GLT1	2.4805e-4(7.82e-5)[2]-	4.8460e-3(1.76e-3)[6]-	8.6516e-4(1.25e-4)[3]-	2.3764e-3(5.40e-4)[5]-	9.4734e-4(1.83e-4)[4]-	1.8121e-4(3.34e-5)[1]
GLT2	2.4616e-3(5.46e-4)[2]-	3.9542e-2(1.72e-2)[6]-	3.1701e-2(2.27e-2)[5]-	1.0755e-2(1.96e-3)[4]-	3.9128e-3(7.39e-4)[3]-	1.2263e-3(7.17e-4)[1]
GLT3	2.3440e-4(4.22e-5)[2]=	1.4404e-2(1.61e-2)[6]-	2.0342e-3(8.25e-4)[5]-	1.0695e-3(2.51e-4)[4]-	2.9257e-4(3.79e-5)[3]-	2.2061e-4(3.49e-5)[1]
GLT4	2.1874e-3(1.04e-2)[3]-	1.2242e-2(2.19e-2)[6]-	1.1652e-2(2.86e-2)[5]-	2.7035e-3(2.23e-3)[4]-	9.1116e-4(1.37e-4)[2]-	2.7318e-4(2.12e-4)[1]
GLT5	3.4168e-2(5.13e-3)[3]-	4.6159e-1(1.33e-1)[6]-	1.3939e-2(2.51e-3)[2]-	4.2578e-2(4.79e-3)[4]-	4.7976e-2(4.59e-3)[5]-	6.6828e-3(6.59e-4)[1]
GLT6	4.2108e-2(8.87e-3)[3]-	4.0058e-1(3.28e-1)[6]-	6.7154e-3(6.62e-3)[2]=	5.6358e-2(8.02e-3)[5]-	5.2826e-2(7.20e-3)[4]-	5.7452e-3(7.09e-4)[1]
LZ1	3.8956e-4(5.54e-5)[2]-	8.6542e-3(4.12e-3)[6]-	1.3963e-3(1.36e-4)[4]-	2.2685e-3(9.86e-4)[5]-	6.7540e-4(8.55e-5)[3]-	2.4235e-4(2.71e-5)[1]
LZ2	5.6404e-2(5.46e-2)[5]-	3.0731e-2(1.52e-2)[2]=	4.7221e-2(8.28e-3)[4]-	6.7262e-2(3.64e-2)[6]-	3.7246e-2(9.44e-3)[3]-	2.9127e-2(1.30e-2)[1]
LZ3	1.4272e-2(1.06e-2)[5]-	1.1734e-2(4.07e-3)[3]-	1.3904e-2(3.45e-3)[4]-	1.0485e-2(2.21e-3)[2]-	2.2727e-2(6.46e-3)[6]-	4.9963e-3(1.38e-3)[1]
LZ4	4.0237e-2(4.52e-2)[6]-	1.5798e-2(6.15e-3)[2]-	3.6983e-2(2.76e-2)[5]-	2.9712e-2(3.68e-2)[4]-	2.7257e-2(1.73e-2)[3]-	1.1042e-2(1.71e-2)[1]
LZ5	1.2880e-2(3.46e-3)[4]-	1.0645e-2(3.13e-3)[3]-	1.5183e-2(2.61e-3)[5]-	9.8584e-3(2.37e-3)[2]-	2.4210e-2(4.80e-3)[6]-	5.5870e-3(1.54e-3)[1]
LZ6	4.0404e+0(1.73e+0)[6]-	5.7949e-1(1.83e-1)[3]-	1.8129e-1(9.30e-2)[2]-	2.1696e+0(8.09e-1)[4]-	2.2553e+0(1.41e+0)[5]-	5.5293e-3(8.13e-4)[1]
LZ7	2.7469e-2(3.56e-2)[4]-	1.9691e-1(8.64e-2)[5]-	1.5274e-3(2.21e-4)[2]-	4.3559e-1(3.28e-1)[6]-	1.0169e-2(3.15e-3)[3]-	1.2269e-3(1.57e-4)[1]
LZ8	1.4570e-1(1.36e-1)[4]-	2.7879e-1(9.14e-2)[6]-	5.5817e-3(6.10e-3)[1]+	1.6292e-1(1.15e-1)[5]-	1.9240e-2(9.72e-3)[2]+	7.5905e-2(6.14e-2)[3]
LZ9	4.5025e-2(4.08e-2)[4]-	4.3620e-2(4.10e-2)[3]=	5.2475e-2(1.05e-2)[5]-	6.7110e-2(8.20e-2)[6]-	3.6127e-2(8.22e-3)[2]-	2.7545e-2(6.75e-3)[1]
HV						
GLT1	4.7910e-1(7.43e-3)[3]-	4.2140e-1(9.78e-3)[6]-	4.8077e-1(8.76e-5)[2]-	4.5123e-1(4.41e-2)[5]-	4.7521e-1(6.03e-3)[4]-	4.8101e-1(6.40e-4)[1]
GLT2	8.1919e-1(4.59e-5)[2]-	7.5712e-1(2.04e-2)[6]-	7.9100e-1(3.77e-3)[5]-	8.1377e-1(3.82e-4)[4]-	8.1790e-1(1.97e-4)[3]-	8.1939e-1(2.00e-5)[1]
GLT3	9.5750e-1(7.42e-4)[2]-	9.4716e-1(2.05e-3)[6]-	9.5394e-1(4.36e-4)[5]-	9.5555e-1(1.16e-3)[4]-	9.5733e-1(4.69e-4)[3]-	9.5776e-1(1.24e-4)[1]
GLT4	5.7587e-1(2.83e-2)[4]-	5.6891e-1(3.44e-3)[5]-	5.5606e-1(6.88e-2)[6]-	5.7813e-1(1.22e-2)[3]-	5.8315e-1(2.74e-4)[2]-	5.8376e-1(2.40e-3)[1]
GLT5	9.5903e-1(1.86e-3)[3]-	9.5290e-1(1.83e-3)[5]-	9.6919e-1(4.16e-4)[2]-	9.5347e-1(2.51e-3)[4]-	9.5033e-1(2.17e-3)[6]-	9.7655e-1(5.09e-5)[1]
GLT6	8.7969e-1(6.00e-2)[6]-	9.3189e-1(2.15e-2)[4]-	9.2614e-1(3.27e-2)[5]-	9.4716e-1(2.49e-3)[2]-	9.4685e-1(2.93e-3)[3]-	9.6845e-1(2.75e-3)[1]
LZ1	7.2028e-1(6.88e-5)[2]-	7.1403e-1(4.90e-4)[6]-	7.1876e-1(1.37e-4)[4]-	7.1703e-1(9.86e-4)[5]-	7.1990e-1(1.02e-4)[3]-	7.2048e-1(3.35e-5)[1]
LZ2	6.4205e-1(1.65e-2)[4]-	6.2150e-1(2.66e-2)[5]-	6.6074e-1(1.05e-2)[1]+	6.1590e-1(2.21e-2)[6]-	6.4885e-1(1.40e-2)[3]-	6.5161e-1(1.40e-2)[2]
LZ3	6.9521e-1(6.41e-3)[3]-	6.7888e-1(9.87e-3)[5]-	6.9932e-1(3.89e-3)[2]-	6.9091e-1(4.66e-3)[4]-	6.7786e-1(5.50e-3)[6]-	7.0562e-1(3.37e-3)[1]
LZ4	6.8186e-1(1.35e-2)[5]-	6.8811e-1(1.15e-2)[3]-	6.8989e-1(8.57e-3)[2]-	6.8551e-1(3.77e-3)[4]-	6.7899e-1(4.44e-3)[6]-	7.0064e-1(3.82e-3)[1]
LZ5	7.0063e-1(2.03e-3)[2]-	6.8610e-1(8.08e-3)[5]-	6.9734e-1(3.89e-3)[3]-	6.9317e-1(4.68e-3)[4]-	6.8386e-1(3.59e-3)[6]-	7.0312e-1(2.63e-3)[1]
LZ6	1.5506e-1(9.98e-2)[6]-	3.8402e-1(3.10e-2)[3]-	4.6331e-1(7.85e-3)[2]-	3.1100e-1(4.56e-2)[4]-	2.1962e-1(9.56e-2)[5]-	5.5642e-1(1.23e-2)[1]
LZ7	6.6349e-1(5.60e-2)[4]-	5.2870e-1(4.15e-2)[5]-	7.1834e-1(3.17e-4)[2]-	3.6901e-1(1.30e-1)[6]-	6.8540e-1(5.71e-2)[3]-	7.1920e-1(2.01e-4)[1]
LZ8	5.5005e-1(6.08e-2)[3]-	4.6864e-1(2.34e-2)[5]-	7.1191e-1(1.39e-2)[1]+	4.4103e-1(1.16e-1)[6]-	5.4428e-1(5.65e-2)[4]-	5.7450e-1(5.23e-2)[2]
LZ9	3.5426e-1(4.13e-2)[4]-	3.5379e-1(1.78e-2)[5]-	3.6316e-1(1.17e-2)[3]-	3.4325e-1(1.88e-2)[6]-	3.6756e-1(1.52e-2)[2]=	3.7301e-1(1.43e-2)[1]
Mean Rank +/-/≈	3.6000 0/28/2	4.7667 0/28/2	3.3000 3/26/1	4.4333 0/30/0	3.7667 1/27/2	1.1333

- 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: $\beta = 0.7$.

• Parameters of RMEA

- number of clusters: $K = 5$.

5.3. Comparison results

5.3.1. Statistical results

The mean and standard deviations are recorded, and all statistical results are shown in Table 3. The mean IGD⁺ and HV values for each instance are sorted in ascending order, and the numbers in the square brackets of Table 3 are their ranks. The Wilcoxon's rank sum test at a 5% significance level is performed in terms of the metrics obtained by each pair of algorithms. "+", "-", and "≈" in Table 3 denote the performance of the comparison algorithm is better than, worse than, and similar to that of RMEA according to Wilcoxon's rank sum test, respectively. The bold data with gray background in the table are the best mean metric values yielded by the algorithms for each instance.

In terms of each test instance, Table 3 shows that for GLT and LZ test suites, among the 30 comparisons, RMEA achieves 27 best mean metric values. RMEA gets the best results for all GLT test instances. For the GLT and LZ test suites, according to Wilcoxon's rank sum test, RMEA gets 28, 28, 26, 30, 27 better, 0, 0, 3, 0, 1 worse, and 2, 2, 1, 0, 2 similar metric values than AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA, respectively.

Considering all instances, the overall rank of the six algorithms is RMEA, MOEA/D-CMA, AMEA, SMEA, RM-MEDA and IM-MOEA. The mean rank indicates that RMEA achieves the best statistical results on the GLT and LZ test suites.

5.3.2. Convergence analysis

To explore the search efficiency of algorithms, Fig. 6 shows the evolution of the mean IGD⁺ values obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RMEA over 30 independent runs on GLT1-GLT6. It can be observed in Fig. 6 that RMEA could reach the lowest mean IGD⁺ metric values for all GLT test instances within 600 generations, which denotes that RMEA has the overall fastest convergence speed for this test suite. Specifically, RMEA always converges faster than AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA. The reason for the good performance of RMEA on convergence speed may be the global exploration by regularity modeling sampling and local exploitation by mating with neighborhood solutions.

5.3.3. Visual comparison

Fig. 7 shows the representative approximation fronts (AFs) obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA, and RMEA, respectively, associated with median IGD⁺ values of GLT1-GLT4. The figures indicate that IM-MOEA cannot converge to the PFs of GLT1-GLT2 and GLT4 successfully, and MOEA/D-CMA and RM-MEDA should be improved in both convergence and diversity on GLT1-GLT4. Meanwhile, some parts of the PF of GLT3 do not cover the objective points yielded by SMEA. For RMEA, well approximated and evenly distributed AFs are achieved for all GLT1-GLT4 test instances.

5.4. Parameter settings

5.4.1. Sensitivity to the number of clusters: K

In spectral clustering, K denotes the number of clusters that will be produced. In RMEA, K also denotes the number of PCA analysis. RMEA with $K = 1, 5, 8, 10$ and 15 are tested on the GLT test suite. Fig. 8 shows that expect for $K = 1$, for GLT test suite, RMEA with different K values obtain similar results. When the number of clusters is small, it is difficult to model the regularity property accurately, and the neighborhood relationship of solutions may not make sense.

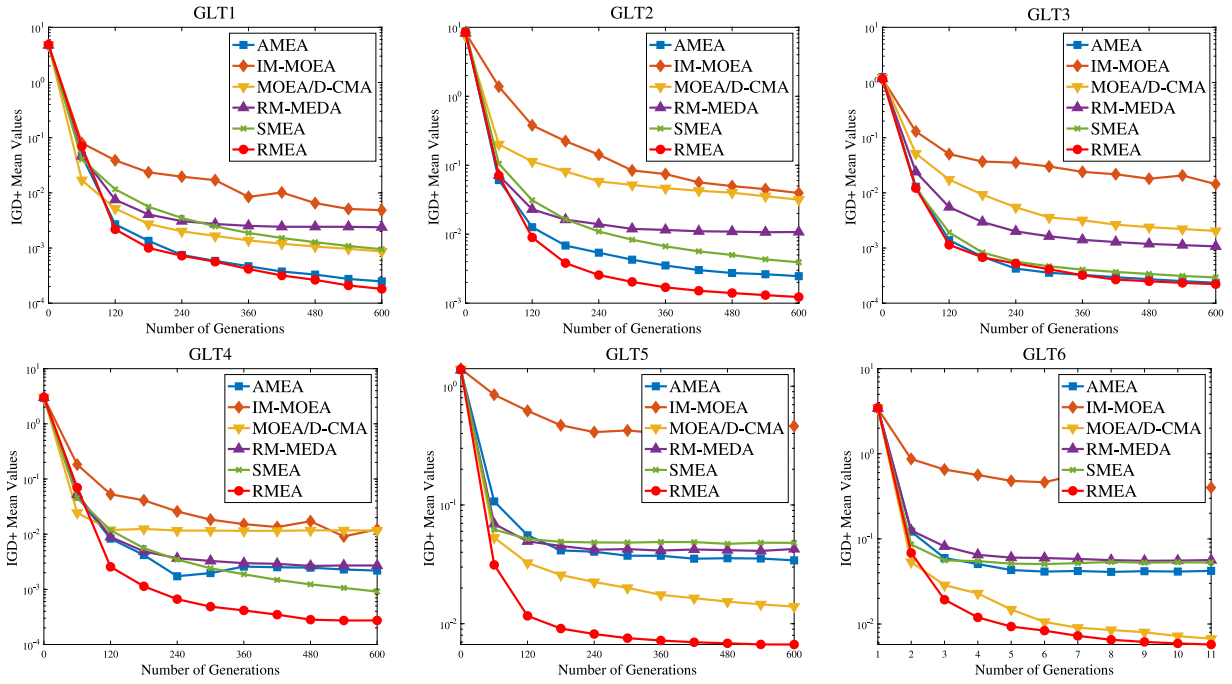


Fig. 6. Evolution of the mean IGD⁺ values obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RMEA over 30 independent runs on GLT1-GLT6.

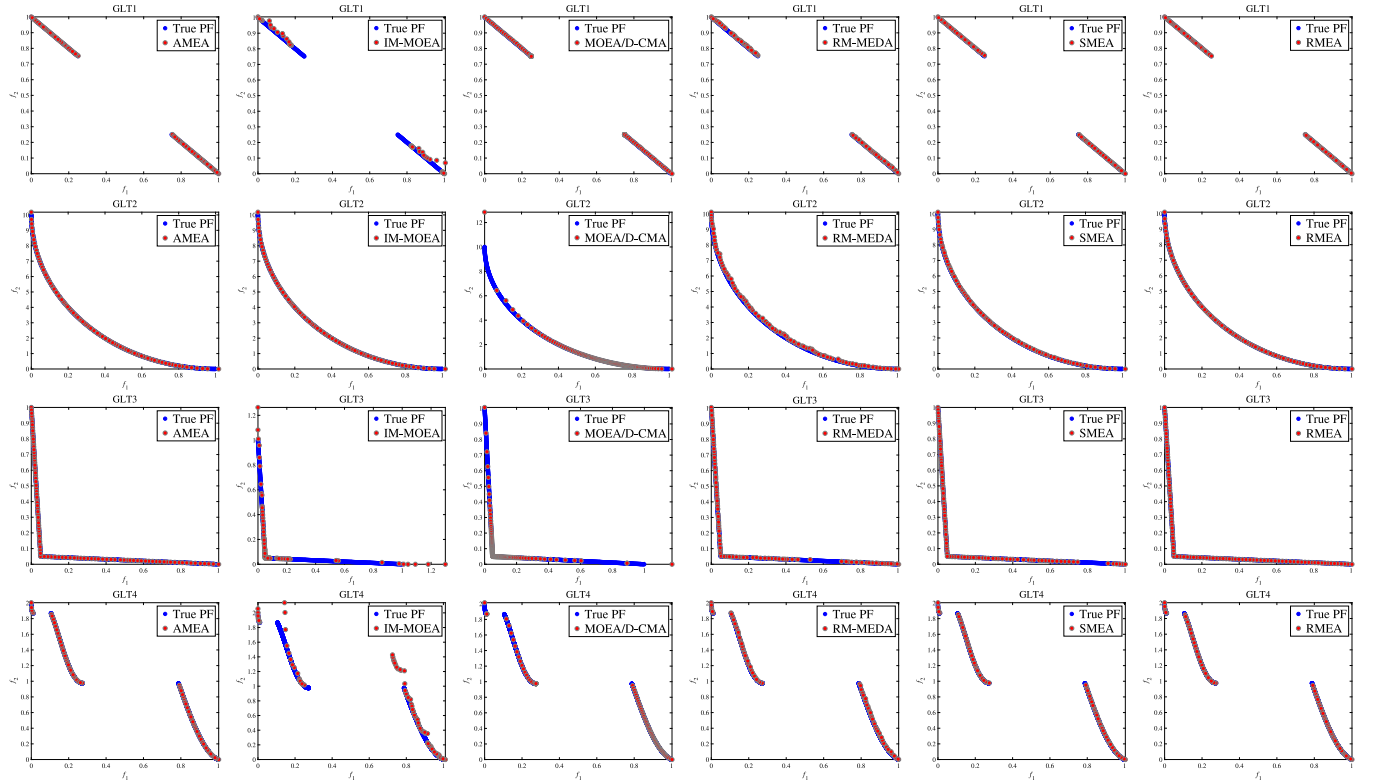


Fig. 7. The representative AFs obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RMEA on GLT1-GLT4 test instances associated with median IGD⁺ metric values over 30 runs in the objective space.

5.4.2. Sensitivity to the perturbation scaling factor: β

In terms of the offspring generation of RMEA, the perturbation vector is influenced by the scaling factor β . In order to investigate the influence of the perturbation scaling factor on the performance of the proposed algorithm, the RMEA is compared with the following four RMEA variants.

- RMEA1: $\beta \in \{-0.5, 0.5\}$, where β is -0.5 or 0.5 with equal probability;
- RMEA2: $\beta \in \{-1, 1\}$, where β is -1 or 1 with equal probability;
- RMEA3: $\beta \in [-0.5, 0.5]$, β is a random value from $[-0.5, 0.5]$;
- RMEA4: $\beta \in [-1, 1]$, β is a random value from $[-1, 1]$.

Table 4

Statistical results (mean(std.dev.))[rank] obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RMEA over 30 independent runs on the WFG and ZZJ test suites in terms of the IGD⁺ and HV metrics.

Instance	AMEA	IM-MOEA	MOEA/D-CMA	RM-MEDA	SMEA	RMEA
IGD ⁺						
WFG1	8.7797e-1(7.57e-2)[3]-	9.8553e-1(1.07e-1)[4]-	8.3449e-1(8.54e-2)[2]-	1.2672e+0(1.08e-2)[6]-	1.2086e+0(4.93e-2)[5]-	6.1874e-1(1.38e-1)[1]
WFG2	1.3328e-2(1.03e-2)[5]-	5.1065e-3(7.34e-4)[4]-	3.1717e-3(7.38e-4)[2]-	4.9045e-3(1.97e-3)[3]-	1.6575e-2(2.11e-3)[6]-	2.3168e-3(5.42e-4)[1]
WFG3	2.5377e-3(1.16e-3)[2]=	9.1486e-3(6.87e-4)[5]-	7.1759e-3(7.68e-4)[4]-	1.1348e-2(2.68e-3)[6]-	3.4354e-3(3.37e-4)[3]-	1.9841e-3(6.50e-4)[1]
WFG4	5.0343e-2(8.76e-3)[4]-	1.5554e-2(2.77e-3)[2]-	1.9104e-2(4.38e-3)[3]-	5.8843e-2(2.33e-3)[6]-	5.4177e-2(8.82e-3)[5]-	1.1945e-2(3.79e-3)[1]
WFG5	6.4432e-2(2.24e-4)[5]=	6.3035e-2(1.66e-3)[3]+	6.0061e-2(9.83e-4)[1]+	6.6787e-2(1.07e-3)[6]-	6.2544e-2(4.37e-3)[2]+	6.4388e-2(2.19e-4)[4]
WFG6	2.1035e-2(1.45e-2)[4]-	1.8126e-2(5.65e-3)[3]-	7.5583e-2(1.01e-1)[5]-	8.7219e-3(1.70e-3)[2]-	1.0127e-1(1.15e-1)[6]-	2.1983e-3(1.36e-3)[1]
WFG7	7.0542e-4(2.16e-4)[1]+	1.2784e-2(1.30e-3)[6]-	2.4263e-3(4.84e-4)[4]-	6.1597e-3(9.21e-4)[5]-	1.9734e-3(2.18e-4)[3]-	8.9285e-4(2.13e-4)[2]
WFG8	9.3620e-2(8.69e-3)[3]-	1.0946e-1(1.21e-3)[6]-	1.0154e-1(3.30e-3)[4]-	1.0336e-1(1.01e-2)[5]-	8.7152e-2(1.40e-3)[2]-	8.3708e-2(3.91e-3)[1]
WFG9	1.2448e-2(6.72e-3)[2]-	1.4397e-2(2.50e-3)[3]-	9.3040e-2(1.02e-1)[5]-	2.0864e-2(9.40e-4)[4]-	2.0652e-1(7.46e-2)[6]-	8.9434e-3(1.82e-3)[1]
HV						
WFG1	2.2074e-1(2.06e-2)[3]-	2.0082e-1(2.44e-2)[4]-	2.9626e-1(3.30e-2)[2]=	2.9410e-3(7.06e-3)[6]-	4.7479e-2(2.18e-2)[5]-	3.0618e-1(4.16e-2)[1]
WFG2	6.2596e-1(6.14e-3)[5]-	6.2984e-1(5.29e-4)[4]-	6.3050e-1(5.71e-4)[3]-	6.3097e-1(1.15e-3)[2]-	6.2423e-1(1.23e-3)[6]-	6.3261e-1(3.30e-4)[1]
WFG3	5.8127e-1(6.08e-4)[2]=	5.7615e-1(3.66e-4)[6]-	5.7807e-1(4.13e-4)[4]-	5.7619e-1(1.17e-3)[5]-	5.8078e-1(1.73e-4)[3]-	5.8157e-1(3.31e-4)[1]
WFG4	3.2091e-1(4.46e-3)[4]-	3.3675e-1(1.40e-3)[2]-	3.3445e-1(2.51e-3)[3]-	3.1770e-1(1.22e-3)[6]-	3.1958e-1(4.21e-3)[5]-	3.4072e-1(2.27e-3)[1]
WFG5	3.1131e-1(2.01e-3)[3]-	3.0921e-1(1.08e-3)[5]-	3.0999e-1(2.43e-3)[4]-	3.0890e-1(1.93e-3)[6]-	3.1361e-1(2.59e-3)[1]-	3.1234e-1(1.55e-3)[2]
WFG6	3.3677e-1(7.65e-3)[3]-	3.3527e-1(2.96e-3)[4]-	3.0492e-1(5.53e-2)[5]-	3.4281e-1(8.49e-4)[2]-	2.9587e-1(5.92e-2)[6]-	3.4677e-1(7.09e-4)[1]
WFG7	3.4756e-1(1.19e-4)[1]+	3.3701e-1(8.62e-4)[6]-	3.4517e-1(3.32e-4)[4]-	3.4414e-1(5.41e-4)[5]-	3.4689e-1(1.14e-4)[3]-	3.4750e-1(1.07e-4)[2]
WFG8	2.9386e-1(3.32e-3)[3]-	2.8689e-1(5.93e-4)[6]-	2.9202e-1(2.02e-3)[4]-	2.8942e-1(4.66e-3)[5]-	2.9559e-1(2.72e-4)[2]-	2.9730e-1(1.42e-3)[1]
WFG9	3.3778e-1(4.30e-3)[2]-	3.3648e-1(1.68e-3)[3]-	2.9214e-1(5.51e-2)[5]-	3.3465e-1(1.31e-3)[4]-	2.4062e-1(3.78e-2)[6]-	3.4148e-1(2.06e-3)[1]
Mean Rank	3.0556	4.2222	3.5556	4.6667	4.1667	1.3333
+ / - / ≈	2/13/3	1/17/0	1/16/1	0/18/0	1/16/1	
IGD ⁺						
ZZJ1	2.7298e-4(4.65e-5)[3]-	1.6199e-3(1.81e-4)[5]-	7.0629e-4(5.74e-5)[4]-	1.6988e-3(4.36e-4)[6]-	1.0451e-4(1.68e-5)[2]-	3.3067e-5(4.28e-6)[1]
ZZJ2	3.2937e-4(5.51e-5)[3]-	2.2918e-3(1.38e-3)[5]-	5.7307e-4(6.55e-5)[4]-	2.4872e-3(3.53e-4)[6]-	1.3390e-4(2.59e-5)[2]-	3.7424e-5(5.85e-6)[1]
ZZJ3	5.3159e-1(8.07e-2)[4]-	4.9514e-1(1.38e-1)[3]-	6.7548e-3(9.97e-3)[1]+	1.8437e+0(1.25e+0)[6]-	8.7574e-1(2.42e-1)[5]-	4.2007e-1(7.11e-2)[2]
ZZJ4	5.1901e-2(1.66e-2)[4]-	3.3512e-1(1.17e-1)[6]-	4.3541e-3(3.23e-4)[2]-	3.2915e-2(4.44e-3)[3]-	5.3167e-2(5.68e-3)[5]-	3.1517e-3(1.88e-4)[1]
ZZJ5	5.1974e-4(7.54e-5)[2]-	3.0860e-3(1.60e-3)[6]-	1.5448e-3(1.47e-4)[4]-	1.6736e-3(3.28e-4)[5]-	1.0643e-3(1.87e-4)[3]-	3.0670e-4(4.26e-5)[1]
ZZJ6	6.9156e-4(1.03e-4)[2]-	2.3785e-3(4.79e-4)[5]-	2.0305e-3(3.63e-4)[4]-	2.4536e-3(6.84e-4)[6]-	1.8957e-3(3.46e-4)[3]-	4.3769e-4(3.88e-5)[1]
ZZJ7	3.5712e-1(6.16e-2)[2]+	3.6799e-1(7.15e-2)[3]+	1.2752e-1(5.02e-2)[1]+	2.0517e+0(1.42e+0)[6]-	8.6909e-1(1.86e-1)[5]-	5.0570e-1(5.43e-2)[4]
ZZJ8	5.2240e-2(5.52e-2)[4]-	5.4620e-1(1.37e-1)[6]-	1.2599e-2(4.14e-3)[2]-	3.7768e-2(3.39e-2)[3]-	5.5948e-2(3.43e-2)[5]-	5.8283e-3(4.05e-4)[1]
ZZJ9	3.4876e-3(6.22e-3)[3]-	3.2147e-3(2.19e-3)[2]=	4.2587e-3(2.67e-3)[4]-	6.8867e-3(4.99e-3)[5]-	1.3769e-2(1.19e-2)[6]-	2.9484e-3(2.18e-3)[1]
ZZJ10	1.1639e+1(1.59e+1)[3]-	4.9072e-4(1.43e-4)[1]+	1.4304e+1(1.05e+1)[4]-	5.2586e+1(2.51e+1)[6]-	2.9826e+1(2.93e+1)[5]-	6.5854e-1(1.91e+0)[2]
HV						
ZZJ1	7.2043e-1(5.75e-5)[3]-	7.1726e-1(6.15e-4)[6]-	7.1952e-1(6.58e-5)[4]-	7.1769e-1(5.17e-4)[5]-	7.2064e-1(2.42e-5)[2]-	7.2075e-1(5.38e-6)[1]
ZZJ2	4.4494e-1(6.81e-5)[3]-	4.4050e-1(1.04e-3)[6]-	4.4430e-1(8.43e-5)[4]-	4.4127e-1(5.01e-4)[5]-	4.4518e-1(3.58e-5)[2]-	4.4533e-1(8.95e-6)[1]
ZZJ3	1.0717e-1(2.36e-2)[3]-	3.5777e-3(6.89e-3)[6]-	3.8267e-1(1.12e-2)[1]+	8.0960e-2(6.69e-2)[5]-	9.8476e-2(7.15e-3)[4]-	1.3372e-1(1.98e-2)[2]
ZZJ4	4.9186e-1(1.73e-2)[5]-	4.1808e-1(1.32e-2)[6]-	5.5609e-1(5.15e-4)[2]-	5.1044e-1(6.30e-3)[3]-	4.9483e-1(5.43e-3)[4]-	5.6898e-1(1.65e-4)[1]
ZZJ5	7.2011e-1(9.52e-5)[2]-	7.0849e-1(2.28e-3)[6]-	7.1862e-1(1.71e-4)[4]-	7.1705e-1(4.39e-4)[5]-	7.1939e-1(2.30e-4)[3]-	7.2040e-1(5.26e-5)[1]
ZZJ6	4.4449e-1(1.25e-4)[2]-	3.5570e-1(1.34e-2)[6]-	4.4233e-1(4.94e-4)[4]-	4.4133e-1(9.03e-4)[5]-	4.4299e-1(4.23e-4)[3]-	4.4483e-1(4.83e-5)[1]
ZZJ7	1.9456e-2(2.85e-2)[6]-	2.1462e-2(2.46e-2)[5]-	2.2271e-1(4.74e-2)[1]+	6.1746e-2(5.07e-2)[4]-	9.0927e-2(5.58e-5)[3]-	1.0644e-1(1.93e-2)[2]
ZZJ8	4.4569e-1(4.08e-2)[5]-	3.8911e-1(1.00e-2)[6]-	5.3876e-1(1.23e-2)[2]-	4.8576e-1(3.80e-2)[3]-	4.5580e-1(3.36e-2)[4]-	5.6426e-1(6.10e-4)[1]
ZZJ9	7.1599e-1(7.93e-3)[3]-	7.1658e-1(1.54e-3)[1]-	7.1403e-1(3.64e-3)[4]-	7.1084e-1(6.63e-3)[5]-	6.9786e-1(1.69e-2)[6]-	7.1644e-1(2.86e-3)[2]
ZZJ10	2.9849e-2(5.22e-2)[4]-	1.2755e-1(7.81e-4)[1]+	6.2340e-2(9.31e-2)[3]=	0.0000e+0(0.00e+0)[6]-	1.5871e-2(4.19e-2)[5]-	1.0079e-1(4.64e-2)[2]
Mean Rank	3.3000	4.5500	2.9500	4.9000	3.8500	1.4500
+ / - / ≈	1/18/1	3/15/2	4/15/1	0/20/0	0/20/0	
Overall MR	3.1842	4.3947	3.2369	4.7895	4.0000	1.3947
+ / - / ≈	3/31/4	4/32/2	5/31/4	0/38/0	1/36/1	

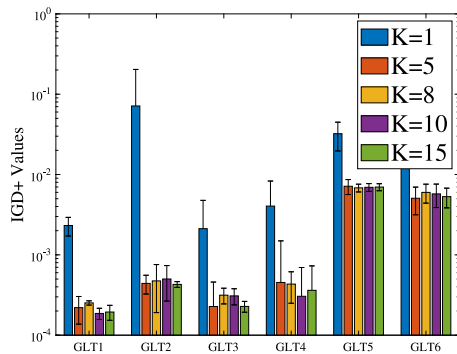


Fig. 8. Mean and standard deviations of IGD⁺ values obtained by RMEA with different K values and 20 independent runs on GLT1-GLT6.

For these four RMEA variants, the GLT1-GLT4 test instances are conducted for comparison, the parameter settings are set to be the same as used in Section 5.2. Fig. 9 shows the evolution of the mean IGD⁺ metric values of populations obtained by RMEA with different β on GLT1-GLT4.

From Fig. 9, we can find that on GLT1-GLT4, different β have different influences on the performance of RMEA, and random β values within $[-1, 1]$ have the best overall convergence efficiency. Moreover, a fixed β value degenerates the search of RMEA on GLT2 and GLT4. It is also worthwhile to note that the optimized β is still problem-dependent. In short, the proposed algorithm has good robustness.

5.5. Performance on more test suites

To further verify the performance of the proposed RMEA, experimental comparisons between RMEA and other five state-of-the-art algorithms are conducted on WFG [54] and ZZJ [4] test suites with different properties. The experimental settings are the same as those in Section 5.2. Each algorithm was independently run 31 times on each instance with 600 generations. The statistical results of IGD⁺ and HV metric values for each instance are presented in Table 4.

5.5.1. Performance on WFG test suite

We have conducted experiments on the WFG test instances [54], which have complicated PF shapes. AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA are tested on these WFG instances (two objectives with 11 variables).

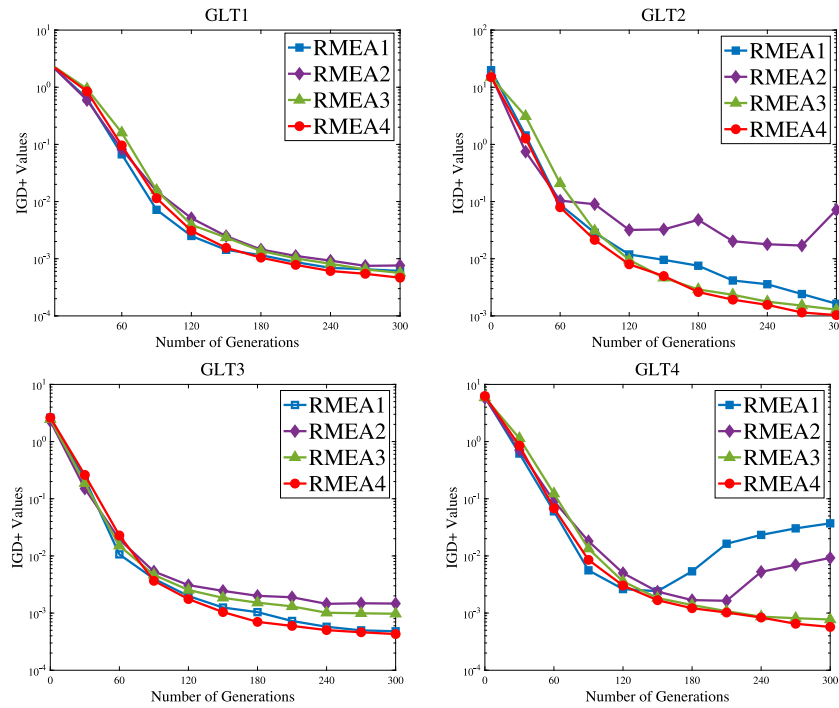


Fig. 9. Evolution of the mean IGD^+ values obtained by four RMEA variants over 30 independent runs on GLT1-GLT4.

Table 4 shows that RMEA performs best on 14 instances overall 18 comparisons on WFG test suite. Meanwhile, compared with AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, and SMEA, RMEA obtains 13, 17, 16, 18 and 16 significantly better mean metric values in 18 comparisons, respectively. In addition, the overall rank of the six algorithms is RMEA, AMEA, MOEA/D-CMA, SMEA, RM-MEDA, and IM-MOEA according to the mean ranks. In a word, the results mentioned above indicate that RMEA has good performance in handling the WFG test suite.

5.5.2. Performance on ZZJ test suite

To further verify the performance of RMEA, we have tested RMEA on the ZZJ [4] test suite. The test instances, ZZJ1-ZZJ4, are variants of ZDT1, ZDT2, ZDT6, and DTLZ2 with linear variable linkages. ZZJ5 and ZZJ10 have nonlinear variable linkages, where the PSs of ZZJ5-ZZJ7 are bounded continuous curves, and the ZZJ8 test instance has a 2-D bounded continuous surface. Furthermore, ZZJ9 and ZZJ10 have many local Pareto fronts.

Table 4 shows that, for the WFG test suite, RMEA achieves 18, 15, 15, 20, 20 better, 1, 3, 4, 0, 0 worse and 1, 2, 1, 0, 0 similar metric values than AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, and SMEA, respectively. In terms of ranking, we see that RMEA performs significantly better on the WFG test suite than the compared algorithms in terms of ranking.

In conclusion, the experimental results on WFG and ZZJ test suites indicate that the superiority of RMEA is significant in these three test suites.

5.6. More discussions with statistical verifications

In the above sections, the mean and standard deviations of metrics (i.e., IGD^+ and HV) of six algorithms have been represented in Tables 3 and 4. Although the Wilcoxon's rank sum test is involved in these tables which is adopted to verify the performance difference between each comparison algorithm and RMEA on each instance, the groups of differences between these methods are not indicated. Therefore, we aim to further indicate the performance of the compared algorithms by ranking them by

Table 5

Average ranks and p -value of Friedman test procedure for the compared algorithms on GLT, LZ, WFG and ZZJ test suites with IGD^+ metric.

Algorithm	Friedman
AMEA	3.3529
IM-MOEA	4.3235
MOEA/D-CMA	3.3529
RM-MEDA	4.7353
SMEA	3.9118
RMEA	1.3235
p -values	1.3048e-13

some multiple comparisons procedures with the IGD^+ statistical results summarized in Tables 3 and 4.

Two multiple comparison procedures, the Friedman test [55] and a post-hoc method (i.e., the Bonferroni corrected procedure [56]), are adopted in this section. As suggested in [57], the Friedman test is regarded as a nonparametric statistical procedure which is commonly adopted to detect significant differences between the behavior of two or more algorithms whether the algorithms follow rank distributions with different medians. The average rank of each algorithm is computed by the Friedman test procedure, and then returns a scalar p -value to determine whether the null-hypothesis with a threshold $\alpha = 0.05$ is rejected ($p > \alpha$) or accepted ($p < \alpha$). Moreover, a post-hoc procedure with RMEA as the control method is conducted to obtain the Bonferroni corrected p -values. More details of the Friedman test and Bonferroni correction could be found in [57], and the results of statistical analysis are shown in Tables 5 and 6, respectively.

From the statistical results based on the Friedman test and Bonferroni correction procedures on GLT, LZ, WFG and ZZJ test suites with the IGD^+ metric shown in Table 5, we can observe that the proposed RMEA achieves the best average rank and significantly outperforms other compared algorithms by using the ranks computed by the Friedman test. Taking the proposed RMEA as a control method, the Bonferroni corrected p -values are shown

Table 6

Bonferroni corrected p -values for the Friedman test (RMEA is the control method).

vs.	Friedman	
	p -value	Bonferroni
AMEA	2.6758e-07	1.3379e-06
IM-MOEA	1.5712e-06	7.8560e-06
MOEA/D-CMA	8.2357e-06	4.1179e-05
RM-MEDA	5.5112e-09	2.7556e-08
SMEA	2.6758e-07	1.3379e-06

in Table 6, and it indicates a significant improvement of RMEA over AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA for the post-hoc Bonferroni corrected procedure. All the statistical analysis results support the conclusion that the performances of RMEA are statistically better than other algorithms for the employed experimental design.

6. Conclusion

This paper proposes a practical regularity model based multiobjective evolutionary algorithm, called RMEA, as an improved and practical version of the regularity model based multiobjective estimation of distribution algorithm (RM-MEDA). Basically, RMEA follows the idea and framework of RM-MEDA. The drawbacks of RM-MEDA are empirically studied, and some practical improved strategies are proposed, including effective clustering and modeling methods, and a hybrid offspring generation method with genetic perturbations.

To evaluate the performance of RMEA, comprehensive experiments are performed for this purpose, including the efficiency of offspring generation operator and comparison experiments with five state-of-the-art MOEAs, e.g., AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA, on 15 complicated test instances. Parameter sensitivity studies, performance on more test suites, and RMEA variant analysis are also given. The results have demonstrated that RMEA has a good performance for dealing with continuous MOPs.

There is still much work worthwhile to be investigated in the future. Some interesting and important research issues include (a) using online approaches (such as online clustering and PCA) to reduce the overhead of calculation, (b) generalizing the proposed idea to other meta-heuristic approaches, and (c) applying RMEA to real-world optimization problems. What is more, the proposed method does not perform well on ZZJ test instances with local PFs (ZZJ9 and ZZJ10). One possible improvement is to perform clustering in the objective space to deal with local PFs characteristics, and we will do further research in our future work.

CRediT authorship contribution statement

Wanpeng Zhang: Writing – review & editing, Methodology, Software. **Shuai Wang:** Writing – original draft, Methodology, Software. **Aimin Zhou:** Methodology, Software, Visualization. **Hu Zhang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the Science and Technology Commission of Shanghai Municipality, China under Grant No. 19511120601, the Scientific and Technological Innovation 2030 Major Projects under Grant No. 2018AAA0100902, and the National Natural Science Foundation of China under Grant No. 61731009, 61907015.

References

- [1] Kaisa Miettinen, Nonlinear Multiobjective Optimization, Vol. 12, Springer Science & Business Media, 2012.
- [2] Kalyanmoy Deb, Multi-Objective Optimization using Evolutionary Algorithms, Vol. 16, John Wiley & Sons, 2001.
- [3] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagarathnam Suganthan, Qingfu Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm Evol. Comput.* 1 (1) (2011) 32–49.
- [4] Qingfu Zhang, Aimin Zhou, Yaochu Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, *IEEE Trans. Evol. Comput.* 12 (1) (2008) 41–63.
- [5] Hui Li, Qingfu Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evol. Comput.* 13 (2) (2008) 284–302.
- [6] Jianyong Sun, Hu Zhang, Aimin Zhou, Qingfu Zhang, Ke Zhang, A new learning-based adaptive multi-objective evolutionary algorithm, *Swarm Evol. Comput.* 44 (2019) 304–319.
- [7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, TAMT Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
- [8] Eckart Zitzler, Marco Laumanns, Lothar Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-Rep.* 103 (2001).
- [9] Juan Zou, Qite Yang, Shengxiang Yang, Jinhua Zheng, Ra-dominance: A new dominance relationship for preference-based evolutionary multiobjective optimization, *Appl. Soft Comput.* 90 (2020) 106192.
- [10] Eckart Zitzler, Simon Künzli, Indicator-based selection in multiobjective search, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 2004, pp. 832–842.
- [11] Nicola Beume, Boris Naujoks, Michael Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, *European J. Oper. Res.* 181 (3) (2007) 1653–1669.
- [12] Trinadh Pamulapati, Rammohan Mallipeddi, Ponnuthurai Nagarathnam Suganthan, I_{SDE+} —An indicator for multi and many-objective optimization, *IEEE Trans. Evol. Comput.* 23 (2) (2018) 346–352.
- [13] Qingfu Zhang, Hui Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 11 (6) (2007) 712–731.
- [14] Mengyuan Wu, Ke Li, Sam Kwong, Qingfu Zhang, Jun Zhang, Learning to decompose: A paradigm for decomposition-based multiobjective optimization, *IEEE Trans. Evol. Comput.* 23 (3) (2018) 376–390.
- [15] Leilei Cao, Lihong Xu, Erik D. Goodman, Hui Li, Decomposition-based evolutionary dynamic multiobjective optimization using a difference model, *Appl. Soft Comput.* 76 (2019) 473–490.
- [16] Rainer Storn, Kenneth Price, Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.* 11 (4) (1997) 341–359.
- [17] Swagatam Das, Ponnuthurai Nagarathnam Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE Trans. Evol. Comput.* 15 (1) (2010) 4–31.
- [18] Swagatam Das, Sankha Subhra Mullick, Ponnuthurai N Suganthan, Recent advances in differential evolution—an updated survey, *Swarm Evol. Comput.* 27 (2016) 1–30.
- [19] Jianye Wu, Wenyin Gong, Ling Wang, A clustering-based differential evolution with different crowding factors for nonlinear equations system, *Appl. Soft Comput.* 98 (2021) 106733.
- [20] James Kennedy, Russell Eberhart, Particle swarm optimization, in: *Proceedings of ICNN'95—International Conference on Neural Networks*, Vol. 4, IEEE, 1995, pp. 1942–1948.
- [21] Jing J Liang, A Kai Qin, Ponnuthurai N Suganthan, S Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.* 10 (3) (2006) 281–295.
- [22] Manoela Kohler, Marley M.B.R. Vellasco, Ricardo Tanscheit, PSO+: A new particle swarm optimization algorithm for constrained problems, *Appl. Soft Comput.* 85 (2019) 105865.
- [23] Pedro Larrañaga, Jose A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Vol. 2, Springer Science & Business Media, 2001.

- [24] Heinz Mühlenbein, Jürgen Bendisch, H.-M. Voigt, From recombination of genes to the estimation of distributions II. Continuous parameters, in: *International Conference on Parallel Problem Solving from Nature*, Springer, 1996, pp. 188–197.
- [25] David H. Wolpert, William G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [26] Hu Zhang, Aimin Zhou, Shenmin Song, Qingfu Zhang, Xiao-Zhi Gao, Jun Zhang, A self-organizing multiobjective evolutionary algorithm, *IEEE Trans. Evol. Comput.* 20 (5) (2016) 792–806.
- [27] Jianyong Sun, Hu Zhang, Aimin Zhou, Qingfu Zhang, Ke Zhang, Zhenbiao Tu, Kai Ye, Learning from a stream of nonstationary and dependent data in multiobjective evolutionary optimization, *IEEE Trans. Evol. Comput.* 23 (4) (2018) 541–555.
- [28] Claus Hillermeier, et al., *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*, Vol. 135, Springer Science & Business Media, 2001.
- [29] Nandakishore Kambhatla, Todd K. Leen, Dimension reduction by local principal component analysis, *Neural Comput.* 9 (7) (1997) 1493–1516.
- [30] Yong Wang, Jian Xiang, Zixing Cai, A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator, *Appl. Soft Comput.* 12 (11) (2012) 3526–3538.
- [31] Handing Wang, Qingfu Zhang, Licheng Jiao, Xin Yao, Regularity model for noisy multiobjective optimization, *IEEE Trans. Cybern.* 46 (9) (2015) 1997–2009.
- [32] Yanan Sun, Gary G. Yen, Zhang Yi, Improved regularity model-based EDA for many-objective optimization, *IEEE Trans. Evol. Comput.* 22 (5) (2018) 662–678.
- [33] Aimin Zhou, Qingfu Zhang, Yaochu 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.* 13 (5) (2009) 1167–1189.
- [34] Aimin Zhou, Yaochu Jin, Qingfu Zhang, A population prediction strategy for evolutionary dynamic multiobjective optimization, *IEEE Trans. Cybern.* 44 (1) (2013) 40–53.
- [35] Ke Li, Sam Kwong, A general framework for evolutionary multiobjective optimization via manifold learning, *Neurocomputing* 146 (2014) 65–74.
- [36] Qiong Yuan, Guangming Dai, Yuzhen Zhang, A novel multi-objective evolutionary algorithm based on LLE manifold learning, *Eng. Comput.* 33 (2) (2017) 293–305.
- [37] Bing Dong, Aimin Zhou, Guixu Zhang, Sampling in latent space for a multiobjective estimation of distribution algorithm, in: *2016 IEEE Congress on Evolutionary Computation, CEC, IEEE*, 2016, pp. 3027–3034.
- [38] Rui Wang, Nan-Jiang Dong, Dun-Wei Gong, Zhong-Bao Zhou, Shi Cheng, Guo-Hua Wu, Ling Wang, PCA-assisted reproduction for continuous multi-objective optimization with complicated Pareto optimal set, *Swarm Evol. Comput.* 60 (2021) 100795.
- [39] Linqiang Pan, Lianghao Li, Ran Cheng, Cheng He, Kay Chen Tan, Manifold learning-inspired mating restriction for evolutionary multiobjective optimization with complicated Pareto sets, *IEEE Trans. Cybern.* (2019).
- [40] Xin Li, Hu Zhang, Shenmin Song, MOEA/D with the online agglomerative clustering based self-adaptive mating restriction strategy, *Neurocomputing* 339 (2019) 77–93.
- [41] Hu Zhang, Xiujie Zhang, Xiao-Zhi Gao, Shenmin Song, Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble, *Neurocomputing* 173 (2016) 1868–1884.
- [42] Hui Fang, Aimin Zhou, Hu Zhang, Information fusion in offspring generation: A case study in DE and EDA, *Swarm Evol. Comput.* 42 (2018) 99–108.
- [43] Fangqing Gu, Hai-Lin Liu, Kay Chen Tan, A multiobjective evolutionary algorithm using dynamic weight design method, *Int. J. Innovative Comput. Inf. Control* 8 (5 (B)) (2012) 3677–3688.
- [44] Hui Li, Qingfu Zhang, Jingda Deng, Biased multiobjective optimization and decomposition algorithm, *IEEE Trans. Cybern.* 47 (1) (2016) 52–66.
- [45] Ulrike Von Luxburg, A tutorial on spectral clustering, *Stat. Comput.* 17 (4) (2007) 395–416.
- [46] Shuai Wang, Hu Zhang, Yi Zhang, Aimin Zhou, Adaptive population structure learning in evolutionary multi-objective optimization, *Soft Comput.* (2019) 1–18.
- [47] Anil K. Jain, Data clustering: 50 years beyond K-means, *Pattern Recognit. Lett.* 31 (8) (2010) 651–666.
- [48] Andrew Y. Ng, Michael I. Jordan, Yair Weiss, On spectral clustering: Analysis and an algorithm, in: *Advances in Neural Information Processing Systems*, 2002, pp. 849–856.
- [49] Ran Cheng, Yaochu Jin, Kaname Narukawa, Bernhard Sendhoff, A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling, *IEEE Trans. Evol. Comput.* 19 (6) (2015) 838–856.
- [50] Nikolaus Hansen, Sibylle D. Müller, Petros Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evol. Comput.* 11 (1) (2003) 1–18.
- [51] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, Yusuke Nojima, Modified distance calculation in generational distance and inverted generational distance, in: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 2015, pp. 110–125.
- [52] Eckart Zitzler, Lothar Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Trans. Evol. Comput.* 3 (4) (1999) 257–271.
- [53] Ye Tian, Ran Cheng, Xingyi Zhang, Yaochu Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], *IEEE Comput. Intell. Mag.* 12 (4) (2017) 73–87.
- [54] Simon Huband, Luigi Barone, Lyndon While, Phil Hingston, A scalable multi-objective test problem toolkit, in: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 2005, pp. 280–295.
- [55] Milton Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *J. Amer. Statist. Assoc.* 32 (200) (1937) 675–701.
- [56] Olive Jean Dunn, Multiple comparisons among means, *J. Amer. Statist. Assoc.* 56 (293) (1961) 52–64.
- [57] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (1) (2011) 3–18.