Swarm and Evolutionary Computation 78 (2023) 101261

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

A regularity augmented evolutionary algorithm with dual-space search for multiobjective optimization

Shuai Wang, Bingdong Li, Aimin Zhou *

Shanghai Institute of AI for Education, School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

ARTICLE INFO

Keywords:
Multiobjective optimization
Evolutionary algorithm
Convergence acceleration
Dual-space search (DSS) strategy

ABSTRACT

The well-known regularity property allows the Pareto optimal solutions of a multiobjective optimization problem (MOP) to be embedded in some latent spaces by the manifold structure of the Pareto optimal set. This paper proposes an efficient regularity augmented evolutionary algorithm (RAEA) for multiobjective optimization, which aims to adopt such latent spaces for use with offspring generations. In this algorithm, a singular value decomposition method is adopted at each iteration to extract such latent space from the population with the regularity property. The enhancements of regularity in RAEA include: (1) an acceleration of convergence by incrementally projecting the dominated solutions into the latent space built with elite solutions (i.e., the non-dominated solutions); and (2) a novel dual-space search strategy (DSS) is developed for the generation of promising offspring solutions by searching in both latent and decision space. The developed algorithm was empirically compared with five well-known multiobjective evolutionary algorithms on several complicated MOP test suites. Experimental results suggest that RAEA outperforms the compared algorithms on these test instances in terms of two commonly-used indicators. Both the effectiveness of the convergence acceleration scheme and the developed dual-space searching strategy are also validated.

1. Introduction

In real-world applications, many optimization problems are subjected to multiple conflicting objectives, termed multiobjective optimization problems (MOPs) [1,2]. Due to the conflicts among the objectives, there does not exist a single solution that can optimize all objectives simultaneously. Therefore, Pareto optimal solutions, the best-possible tradeoffs among the objective functions, are usually expected by decision-makers. The set of all the Pareto optimal solutions in the decision space is called the Pareto optimal set (PS), and its mapping in the objective space is denoted as the Pareto front (PF).

Since multiobjective evolutionary algorithms (MOEAs) can approximate the Pareto optimal solutions of a MOP in a single run, they have flourished in the last two decades [3]. In MOEAs, a population of promising solutions is driven to approximate the Pareto optimal solutions. The basic iterative steps of a general MOEA can be defined as follows:

$$\begin{cases} Pop_0 \\ Pop_{t+1} = \mathbf{Sel}(Pop_t, \mathbf{Gen}(Pop_t)) \end{cases} \tag{1}$$

where Pop_0 is the initial solution set given randomly. The offspring generation operator $\mathbf{Gen}(\cdot)$ is performed to generate new trail solutions in MOEAs, while promising or high-quality solutions are selected for the next generation by the environment selection operator $\mathbf{Sel}(\cdot)$. The above two operators are executed in a loop until the algorithm converges and obtains a set of well-approximated Pareto optimal solutions.

In fact, many studies suggest that key points to improve the performance of MOEAs are to design the efficient generation and selection operators [4,5]. However, as seen from the literature, most of the existing MOEAs mainly focus on designing and improving selection operators, and directly adopt the offspring generation operators developed for single objective optimization problems without any modifications [4,6]. In terms of the selection methods, MOEAs can be roughly classified into three categories based either on the Pareto dominance [7,8], performance indicators [9–11] or decomposition strategies [12–14].

In most cases, the existing MOEAs can drive the population to approximate the PF given enough computational resources with different generation operators, but their main difference is the search efficiency [13]. There is no doubt that effective offspring generators could accelerate the search process by generating high-quality solutions. However, the existing offspring generation operators, e.g., the differential evolution (DE) [15,16], particle swarm optimization (PSO) [17,18] and estimation of distribution algorithms (EDAs) [19], may not be always efficient when dealing with MOPs, due to the lack of consideration of the characteristics of MOPs [20,21].

* Corresponding author.
E-mail addresses: wangshuai515658@163.com (S. Wang), bdli@cs.ecnu.edu.cn (B. Li), amzhou@cs.ecnu.edu.cn (A. Zhou).

https://doi.org/10.1016/j.swevo.2023.101261
Received 22 May 2022; Received in revised form 23 December 2022; Accepted 3 February 2023
Available online 11 February 2023

In the case of continuous MOPs, it has been proved that under mild assumptions induced from the Karush-Kuhn–Tucker (KKT) condition, the Pareto optimal solutions of a continuous MOP form a piecewise $(m-1)$-dimensional continuous manifold in the decision space [22], where m is the number of the objectives. This manifold characteristic of continuous MOP is called regularity property [23,24]. Several regularity models (RMs) based offspring generations mainly benefit from the regularity property of MOPs [4,25–27], where the RM-based multiobjective EDA (hereafter RM-MEDA [4]) might be the first implementation of this method.

In terms of the existing regularity model based evolutionary approaches, some *latent spaces* are commonly involved in these methods, which are built to approximate the manifold structure of the population explicitly, according to the manifold property. For example, RM-MEDA aims to approximate the manifold structure by a set of hyperplanes indicated by principal component subspaces, e.g., a line segment for the bi-objective (i.e. $m = 2$), and a 2-D rectangle for tri-objective (i.e. $m = 3$), with a segmented principal component analysis (local PCA) approach. These hyperplanes (or principal component subspaces) could be regarded as *latent spaces*. The Latin hypercube sampling or uniform sampling procedures can be performed in such built latent spaces to generate offspring solutions in RM-MEDA, which have been verified that perform well on MOPs with variable lineages.

Inspired by the success of RM-MEDA, among a number of RM-MEDA improvements or variants, two specific methods attempt to search in the built latent space(s) with some genetic operators. In [28], instead of the Latin-like sampling method in RM-MEDA, the population solutions are projected into the latent spaces, and a DE mutation search (DES) strategy is performed among these latent spaces for RM-MEDA. Very recently, a PCA-assisted search approach has been proposed in [29], which builds a new principal component subspace (i.e. a new coordinate system) by the PCA method, and genetic reproduction is conducted for searching in the latent space. However, the number of latent spaces is hard to be determined in RM-MEDA/DES, which is problem-dependent. At the same time, the PCA-assisted search may fail to work in the early stage and has to combine with other generation operators. In a word, the studies of searching with the latent space for offspring generation are still very small and inefficient in evolutionary multiobjective optimization, and it is worth investigating further.

Given the above considerations, this study aims to make new contributions with the regularity latent space by proposing a regularity augmented evolutionary algorithm (short for RAEA) for multiobjective optimization. Instead of sampling from the latent space, a novel dual-space search (DSS) strategy is proposed in RAEA which adopts the genetic generation operators to search in both latent and decision spaces for offspring generation, that is also different from the above DES and PCA-assisted methods that only perform the genetic procedures in the latent space.

The proposed RAEA has the following new features:

- In RAEA, a singular value decomposition (SVD) approach is used to extract the regularity for the construction of latent space. To accelerate the convergence, at each generation, such latent spaces are built with the elite solutions (i.e. nondominated solutions) of the current population, while these dominated solutions are incrementally projected into the built latent space by the incremental computation procedures of SVD;
- To generate new trial solutions, the proposed DSS generation is performed with projected population points to search in the built latent space and then map back into the original decision space. Finally, these reconstructed trial vectors are adopted to generate new trial solutions with the parents sampled from the population. Moreover, the above generations adopt a simple neighborhood relationship obtained in the latent space by K-nearest-neighbors to choose neighbor parents.

The rest of this paper is organized as follows. In Section 2, we briefly provide the background and motivation of this work. The details of the proposed RAEA for continuous multiobjective optimization are presented in Section 3. Experimental analyses and comparisons of RAEA with some representative MOEAs on various benchmark problems are presented in Section 4. Conclusions are drawn in Section 5.

2. Background and motivation

2.1. Problem definition

Without loss of generality, this paper considers the following unconstrained MOP [2,3]:

$$\begin{aligned} &\text{minimize} \quad F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T \\ &\text{s.t.} \quad \boldsymbol{x} = (x_1, \ldots, x_n) \in \Omega \end{aligned} \tag{2}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a n-dimensional decision vector, $\Omega = \prod_{i=1}^{n}[\check{x}_i, \hat{x}_i]$ is the decision space, and $\check{x}_i$ and $\hat{x}_i$ are the lower and upper boundaries of the ith variable. $F(\boldsymbol{x})$ consists of m objective functions $f_i(\boldsymbol{x})$: $\Omega \to \mathbb{R}^m, i = 1, \ldots, m$, where $\mathbb{R}^m$ is the objective space.

Definition 1 (*Pareto Dominance*)**.**
Let $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \Omega$ be decision vectors, $\boldsymbol{x}_1$ Pareto-dominates $\boldsymbol{x}_2$ denoted by $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$ in minimization cases, if and only if

$$\begin{cases} f_i(\boldsymbol{x}_1) \le f_i(\boldsymbol{x}_2), & \forall i = 1, \ldots, m \\ f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2), & \exists i = 1, \ldots, m. \end{cases} \tag{3}$$

Definition 2 (*Pareto Optimal Solution*)**.**
A solution $\boldsymbol{x}^* \in \Omega$ is a Pareto-optimal solution if

$$\boldsymbol{x}^* \in \Omega \wedge \neg\exists \boldsymbol{x} \in \Omega : F(\boldsymbol{x}) \prec F(\boldsymbol{x}^*) \tag{4}$$

that the PS is the set of all Pareto-optimal solutions in Ω, and the PF is the set of all corresponding objective vectors in $\mathbb{R}^m$.

2.2. Theory of regularity property

Let the objectives $f_i(\boldsymbol{x}), i = 1, \ldots, m$ be continuously differentiable for the MOP defined in Eq. (2). If a solution $\boldsymbol{x}^* \in \Omega$ is Pareto optimal, then there exists a vector α that

$$\alpha \in \mathbb{R}^m \text{ with } \alpha_i \ge 0 \text{ and } \sum_{i=1}^{m} \alpha_i = 1,$$

such that

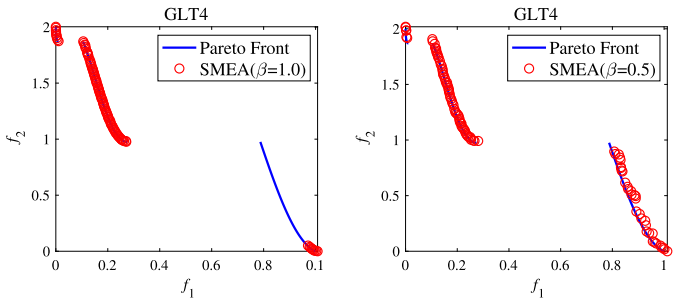
$$\sum_{i=1}^{m} \alpha_i \nabla f_i(\boldsymbol{x}^*) = 0 \tag{5}$$

where these Pareto optimal solutions are called KKT points which satisfy Eq. (5), while not all KKT points are (local) Pareto optimal solutions as the above theorem is not a sufficient condition. Thus, under certain smoothness conditions, the Pareto optimal solutions of a bi-objective optimization problem define a piece-wise continuous *curve*, while the Pareto optimal solutions of the three objective optimization problems define a piece-wise continuous *surface*[1]. More details of theoretical proofs can be found in [22].

2.3. Applications of regularity property in MOEAs

As mentioned in Section 1, the PS of a continuous MOP indicates a piece-wise $(m-1)$-D manifold structure in the decision space. This manifold structure of PS has shown great potential for offspring generation in MOEAs that follows two aspects.

[1] In some cases, we note that some complicated MOPs, the degenerated or equality constrained problems, may not meet the regularity property [30,31], but this paper mainly focuses on the continuous MOPs with the $(m-1)$-dimensional manifold structure.



(a) SMEA with $\beta = 1.0$ (b) SMEA with $\beta = 0.5$

Fig. 1. Final populations of SMEA on GLT4 test instance after 300 generations with different mating restriction probability, where the population size is 100. (a) SMEA cannot cover the entire PF if the parent is only allowed to select from the neighbors ($\beta = 1.0$); (b) SMEA still does not approximate PF well when sampling from neighbor or population in half.

On the one hand, several learning based MOEAs are proposed to design mating restriction mechanisms for local exploitation with the help of manifold structure. Generally, the clustering learning algorithms, such as k-means [32] and spectral clustering [33,34], are adopted to learn the manifold structure to select neighborhood parents for offspring generation. For instance, in [20], a self-organizing map learning method [35] is adopted to define the neighborhood relationship among the population in the SMEA. This algorithm suggests that parent solutions are selected from the neighboring solutions or the entire population with a predefined probability for each solution in the population. As one can expect, such fixed probability might be hard to balance the local and global search, which determines the exploitation and exploration of an algorithm. The results in Fig. 1 show different mating restriction probabilities significantly impact the performance of SMEA. Moreover, Sun and Zhang et al. [6] design some efficient improvement strategies, e.g., Gaussian-perturbed sampling and clustering reusing, to generate promising trial solutions in an adaptive evolutionary algorithm (AMEA). Then, several efforts focus on the adaption of probability in mating restrictions [36,37]. Furthermore, some incremental learning methods are designed to reduce the computational overhead for use with neighborhood mating in offspring generation [38,39].

On the other hand, some efforts aim to explicitly approximate the manifold of PS by estimated (or probability) models which are used to sample new trial solutions. The RM-MEDA is such a typical algorithm. Although RM-based evolutionary algorithms have been well-developed in the past years [25–27,40,41], most of these methods share the similar offspring generation procedures: building low dimensional latent space(s) to approximate the manifold and sampling among these spaces for generating offspring solutions. Recently, although some efforts have not explicitly discussed their method in the context of the regularity modeling procedures, they aim to extract this regularity property for help with the generation of offspring in large-scale multiobjective optimization [42,43].

2.4. Evolutionary search in latent space

Several studies have shown that a new rebuilt latent space may enhance the search for offspring generation, where the PCA method or its variants, is widely used for building latent space. Examples include *PCA-mutation* search for diversity maintenance [44], *PCA* or *kernel-PCA crossover* in new coordinate system [45,46], *PCA-projection* search with valley direction for constrained optimization [47] and others. However, we also notice that these searching approaches are designed for single-objective or global optimization. Moreover, their searches are all enhanced with *decreased correlations or linkages among variables* in the latent space seen in the literature.

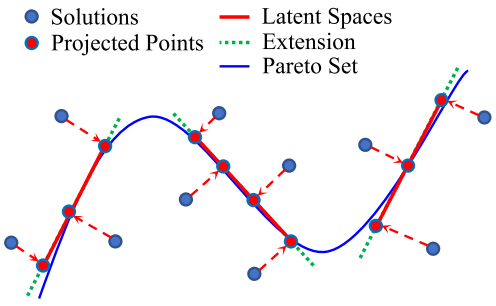
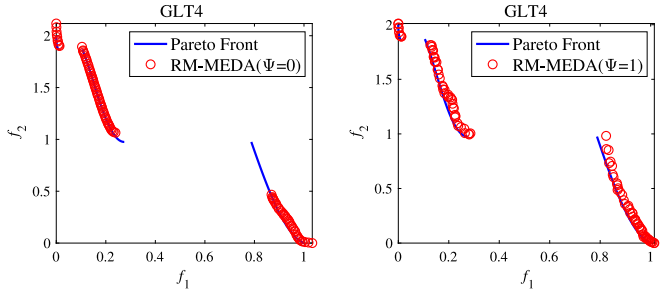


Fig. 2. Illustration of *Latent space* and *Projection* obtained by regularity models in RM-MEDA where the number of clusters is set to $K = 3$.



(a) RM-MEDA with no extension (b) RM-MEDA with 100% extension

Fig. 3. Final populations of RM-MEDA on GLT4 test instance after 300 generations with different extension rates of latent space, where the population size is 100. (a) no extension on the latent space ($\Psi = 0$) may result in a lack of global exploration by RM-MEDA; (b) a larger expansion ($\Psi = 100\%$) leads to the inefficiency of offspring generation.

As aforementioned, RM-MEDA is proposed for offspring generation by adopting this manifold property where the manifold latent spaces are extracted from the population by the PCA method. In RM-MEDA, at each generation, a set of $(m-1)$-D latent spaces is extracted from the current population by the local PCA approach. New trial solutions are sampled from these latent spaces together with some Gaussian white noise vectors.

Let S be a finite subset of R^n, such as a cluster. The latent space and projection procedures are performed as follows:

Step 1: Calculate mean value $\bar{S}$ and covariance matrix Cov:

$$\bar{S} = \frac{1}{|S|}\sum_{x\in S} x \tag{6}$$

$$Cov = \frac{1}{|S|-1}(S-\bar{S})(S-\bar{S})^T \tag{7}$$

where $|S|$ is the cardinality of S;

Step 2: Built the *latent space* $\mathcal{L}$ (r principal component subspace):

$$V = \{v_1, v_2, \ldots, v_r\} \tag{8}$$

where the ith principal component, $v_i (i = 1, \ldots, r)$, is a eigenvector associated with the ith largest eigenvalue of the matrix Cov;

Step 3: Project the population:

$$\Psi = (S-\bar{S})^T V \tag{9}$$

where $\Psi \in R^r$ in the r-dimensional principal component subspace of S.

In fact, RM-MEDA assumes a $(m-1)$-D manifold structure, hence the $(m-1)$-D $\mathcal{L}, \Psi \in R^{m-1}$ is required. An illustration of *latent space* and *projection* are shown in Fig. 2. In this way, the *latent space* $\mathcal{L}$ and

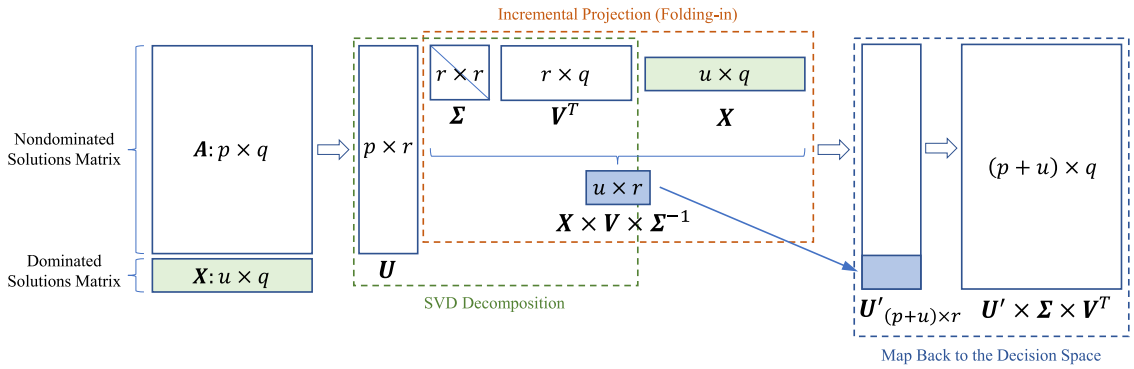


Fig. 4. Illustration of the incremental projection procedures of SVD adopted in RAEA for population convergence acceleration where the **A** denote the matrix of nondominated solutions and the **X** denote the matrix of dominated solutions.

projection Ψ are obtained, and offspring generation procedures can be further performed in these implementations.

1. In RM-MEDA, the range of latent space V is defined by the projection Ψ of the population and often extended along each principal component for better manifold approximation of the PS. In this way, trial solutions are sampled from thus extended latent spaces. As shown in Fig. 3, it is clear that RM-MEDA performs quite differently with different extensions of the latent space, which implies that the extension rate has a significant effect on the performance of RM-MEDA.
2. In terms of DES and PCA-assisted search strategies, they focus on the projection Ψ of the population for use with genetic operators for offspring generation. In fact, the genetic search in the latent space might remove the setting of the extension of latent space to some degree. However, such a latent space based search strategy may be inefficient and have to combine other generation procedures, such as the Gaussian noise in the DES and other converged populations obtained by other generation operators before the PCA-assisted search.

In a word, searching in the latent space extracted from the manifold structure can help generate promising solutions. However, we also note that only searching the latent space is inefficient. This is why the dual-space search strategy is proposed to search in both latent and decision space. Moreover, we suggest adopting the population projection to accelerate the convergence by the incremental projection method. This convergence acceleration scheme is efficient yet straightforward.

3. The proposed method

The basic idea of the developed algorithm in this work is to enhance the evolutionary multiobjective optimization (EMO) approach for use with the built latent space for population convergence acceleration and dual-space search of offspring generation. Therefore, this section introduces the acceleration for population convergence and the search in both latent and decision space for offspring generation in detail. Then, by integrating such newly developed strategies into the framework of MOEA, the framework of the proposed RAEA and the details of each step are presented. It should be noted here that all the described *latent spaces* are given in the context of the regularity property.

3.1. Population convergence acceleration

It is clear that the nondominated solutions are closer to the PF than the relative inferior solutions (i.e. dominated solutions). This motivates us to extract latent spaces from the nondominated solutions, and then project the dominated solutions into such built latent space incrementally, to accelerate the convergence. We acknowledge that the incremental projection technique of the SVD approach can achieve this goal.

3.1.1. Building latent space via SVD

SVD, as an important matrix factorization method in linear algebra and statistical analysis [48–51], has been developed with a complete theoretical system [50,52]. SVD is used to decompose the data matrix to extract the latent information of the data, where the latent semantic indexing (LSI) [53] and recommender systems (RS) [54] might be the most well-known applications of SVD.

In RAEA, we adopt the SVD method to extract the regularity property for building latent space. The SVD of the matrix indicates that for any non-zero p-by-q real matrix A (i.e. the decision variable matrix of nondominated solutions), $A \in \mathbb{R}^{p\times q}$, it can be expressed as the product of three matrices as follows:

$$A = U\Sigma V^T = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_p}] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \cdots & \\ & & & \sigma_j \end{bmatrix} \begin{bmatrix} \mathbf{v_1^T} \\ \mathbf{v_2^T} \\ \cdots \\ \mathbf{v_q^T} \end{bmatrix} \tag{10}$$

where U is a p-by-p orthogonal matrix, V is a q-by-q orthogonal matrix, and Σ is a diagonal matrix with $j = \min(p, q)$ non-negative diagonal elements in descending order that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_k > \sigma_{k+1} = \cdots = \sigma_j = 0, k = rank(A)$. The first r columns of U and V represent the orthogonal eigenvectors associated with the r nonzero eigenvalues of AA^T and A^TA, respectively. In other words, the r ($r < k$) columns of U corresponding to the nonzero singular values span the columns space.

Actually, these reduced columns spaces can be regarded as the *latent spaces* $\mathcal{L}$ in the proposed RAEA. What is more, the r-dimensional of latent space $\mathcal{L}$ in RAEA is defined with a threshold θ:

$$\sum_{i=1}^{r} \sigma_i \geq \theta \sum_{i=1}^{k} \sigma_i \tag{11}$$

where the threshold $0 \leq \theta \leq 1$. As seen from literature [26], this threshold is usually set as 0.9–0.99, while we set it as $\theta = 0.95$ in this paper.

3.1.2. Incremental projection for convergence acceleration

The incremental computation property of SVD could be allowed to project additional (or new) data into the built latent space. This incremental projection technique is known as *folding-in* procedures in SVD literature [55,56]. To project new solutions (i.e. the dominated solutions) into the reduced r latent space, we calculate the coordinates for those vectors on the basis U_r.

Let $\boldsymbol{X}$ be a thus new vector, a projection P with folding-in procedures that projects $\boldsymbol{X}$ onto the r-D latent space $\mathcal{L}$ computed as:

$$P = \boldsymbol{X} \times V_r \times \Sigma_r^{-1} \tag{12}$$

where the projection P is also allowed to map back into the decision space Ω as a trial vector $\boldsymbol{X}'$ defined by:

$$\boldsymbol{X}' = P \times \Sigma_r \times V_r^T \tag{13}$$

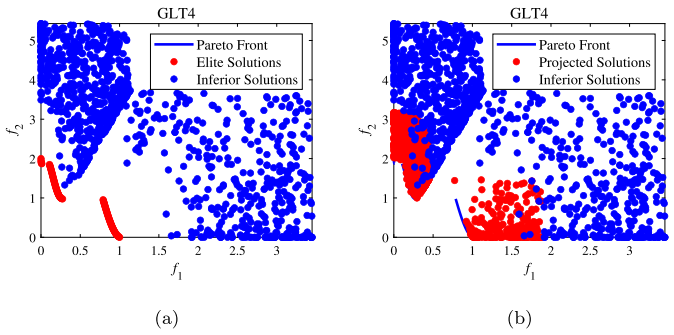


Fig. 5. 1000 new trial solutions on GLT4 obtained by the incremental projection and mapping back the decision space with 1000 random dominated solutions.

In this work, as shown in Fig. 4, we aim to extend the folding-in method to multiobjective evolutionary computation for convergence acceleration by incremental projection. Specifically, we first compute a latent space with the elite solutions, and then incrementally project the inferior solutions upon such built latent spaces. Note that, in Eqs. (12) and (13), both the folding-in dominated solutions onto the latent space and mapping back onto the decision space are based on the existing components (i.e. the orthogonal matrix V_k and diagonal matrix Σ_k) that new projections do not affect them (more details on the folding-in procedures can be seen in Refs. [55,56]). In practice, it is possible to refine these dominated solutions for convergence acceleration.

To facilitate understanding, we use the data representation in Fig. 5 as an example. In the figures, a well converged 100 nondominated solutions on GLT4 test instance is shown in Fig. 5 while has 1000 dominated solutions deviated from real PF. It is clear that these 1000 dominated solutions are driven to approximate the PF with the incremental projection procedures.

3.2. Dual-space search for offspring generation

The primary features of the developed dual-space search (DSS) strategy aim to perform an evolutionary search operator (i.e. DE operator) in both the above built latent space $\mathcal{L}$ and original decision space Ω. Based on the projections (coordinates) on the latent space, a simple neighborhood relationship is defined with K-nearest-neighbors in the latent space among the population. Each solution $\boldsymbol{x}$ has:

- a projected position in the latent space: $\boldsymbol{p} = (p_1, \ldots, p_r) \in \mathcal{L}$;
- K neighbor parents: $\boldsymbol{P}$ in the latent space $\mathcal{L}$ and corresponding $\boldsymbol{Q}$ in the decision space Ω;

In this way, to generate a new trial solution $\boldsymbol{y}$, a general paradigm of DSS can be summarized as:

$$DSS: \begin{cases} \boldsymbol{x} \in \Omega \rightarrow \boldsymbol{p} \in \mathcal{L} \\ \boldsymbol{p}' = \mathbf{Gen}(\boldsymbol{p}, \boldsymbol{P}) \\ \boldsymbol{p}' \in \mathcal{L} \rightarrow \boldsymbol{x}' \in \Omega \\ \boldsymbol{y} = \mathbf{Gen}(\boldsymbol{x}', \boldsymbol{Q}) \end{cases} \tag{14}$$

To implement the above idea, we use the DE operator as the generation operator to search the dual-spaces where the DE operator is performed with the current solution $\boldsymbol{x}$ and its mating pool $\boldsymbol{P}$ and $\boldsymbol{Q}$. Moreover, a polynomial mutation (PM) operator is also performed to mutate the trial vector in the decision space after DE operations, as most MOEAs do. The procedure details are given in Algorithm 1.

In Algorithm 1, first, DE is used to generate a trial r-dimensional vector $\boldsymbol{p}'$ by taking a projection $\boldsymbol{p}$ and its two neighbors $\boldsymbol{p}^{r1}$ and $\boldsymbol{p}^{r2}$ selected from the mating-pool $\boldsymbol{P}$ in the low-dimensional latent space (lines 1–2). Then, the new trial vector $\boldsymbol{p}'$ is mapped back into the decision space for further DE search procedure with its two parents $\boldsymbol{x}^{r1}$ and $\boldsymbol{x}^{r2}$ to generate a new trial solutions in the decision space (lines 3–5). Finally, a repair mechanism is employed to correct any component outside the search boundary (line 6). After repair, PM is applied to generate a new solution (line 7). If necessary, the new solution is repaired, and the final solution $\boldsymbol{y}$ is returned (line 9).

Algorithm 1: Offspring Generation with DSS

Input :
a projection in the latent space $\boldsymbol{p}$;
the neighborhood mating pool $\boldsymbol{P}$ and $\boldsymbol{Q}$;
the latent space matrix V_r and Σ_r;
Output:
an offspring solutions $\boldsymbol{y}$;

// Searching in the *latent space* $\mathcal{L}$

1 Randomly select two parents $\boldsymbol{p}^{r1}$ and $\boldsymbol{p}^{r2}$ form $\boldsymbol{P}$ in the latent space;

2 Generate a trial vector $\boldsymbol{p}' = (p'_1, \cdots, p'_r)$, $i = 1, \cdots, r$

$$p'_i = p_i + F \times (p_i^{r1} - p_i^{r2})$$

// Searching in the *decision space* Ω

3 Map back to the decision space as a trial solution $\boldsymbol{x}'$

$$\boldsymbol{x}' = \boldsymbol{p}' \times \Sigma_r \times V_r^T$$

4 Randomly select two parents $\boldsymbol{x}^{r1}$ and $\boldsymbol{x}^{r2}$ form $\boldsymbol{Q}$ in the decision space;

5 Generate a trial solution $\boldsymbol{y}' = (y'_1, \cdots, y'_n)$, $i = 1, \cdots, n$

$$y'_i = y_i + F \times (x_i^{r1} - x_i^{r2})$$

6 Repair the trial solution by $(i = 1, \cdots, n)$

$$y''_i = \begin{cases} \check{x}_i & \text{if } y'_i < \check{x}_i \\ \hat{x}_i & \text{elseif } y'_i > \hat{x}_i \\ y'_i & \text{otherwise} \end{cases}$$

7 Mutate the trial solution by $(i = 1, \cdots, n)$

$$y_i = \begin{cases} y''_i + \delta_i \times \Delta x_i & if\ rand() < p_m \\ y''_i & otherwise \end{cases}$$

where $i = 1, \cdots, n$, $\Delta x_i = \hat{x}_i - \check{x}_i$, $rand()$ generates a random number from $[0, 1]$, $r = rand()$, and

$$\delta_i = \begin{cases} \left[2r + (1-2r)\left(\frac{\hat{x}_i - y_i}{\hat{x}_i - \check{x}_i}\right)^{n_m}\right]^{\frac{1}{\eta_m}} - 1 & if\ r < 0.5 \\ 1 - \left[2 - 2r + (2r-1)\left(\frac{y_i - \check{x}_i}{\hat{x}_i - \check{x}_i}\right)^{\eta_m}\right]^{\frac{1}{\eta_m}} & otherwise \end{cases};$$

8 Repair $\boldsymbol{y}$ if necessary;

9 **Return** the offspring solution $\boldsymbol{y}$.

3.3. RAEA framework

In RAEA, the algorithmic parameters include the population size N and the maximum number of generations T. At each generation, the proposed RAEA maintains a population Pop with N solutions, where $Arch_{Nond}$ is the nondominated solution set, and $Arch_{Dom}$ contains the dominated solutions. The framework of RAEA is presented in Algorithm 2, and its components are explained as follows:

- **Population Initialization:** Pop is initialized by randomly generating a set of N solutions from the search space Ω (line 1);
- **Termination Condition:** if the termination condition, the maximum number of generations T, is met, stop and return all the final solutions in Pop (line 2);

Algorithm 2: RAEA Framework

Input :
the population size N;
the maximum number of generations T;
Output:
the population Pop;
// *Population Initialization*
1 Randomly initialize $Pop = \{x^1, x^2, \cdots, x^N\}$;
// *Termination Condition*
2 **for** $t = 1, \cdots, T$ **do**
// *Decomposing by SVD and Folding-in*
3 Perform SVD on $Arch_{Nond}$: $[U_r^{Nond}, \Sigma_r, V_r^T] = SVD(Arch_{Nond})$;
4 Incremental project $Arch_{Dom}$: $U_r^{Dom} = Folding-in(Arch_{Dom})$;
5 **foreach** $i \in 1, \cdots, N$ **do**
// *Generation with DSS*
6 Set the mating pool $\boldsymbol{P}$ and $\boldsymbol{Q}$ for $\boldsymbol{x}^i$;
7 Generate a trial solution $\boldsymbol{y}$ with Algorithm 1;
// *Selection*
8 Update the population Pop by $\boldsymbol{y}$;
9 **end**
10 **end**
11 **Return** Pop.

- **Latent Space and Projection:** to build the latent space, the SVD method is performed with the nondominated solutions $Arch_{Nond}$, while these dominated solutions $Arch_{Dom}$ are incrementally projected onto (or folding-in) thus built latent space (lines 3 and 4);
- **Generation:** in lines 6 and 7, with the projection of the latent space, each solution in the population searches in the latent space, and then is mapped back into the decision space for further offspring generation by the developed dual-space searching method indicated in Algorithm 1;
- **Selection:** Whenever offspring solutions are generated, a selection method based on the hypervolume (HV) indicator in SMS-EMOA [10] is used to update the population Pop with the new solution $\boldsymbol{y}$ (line 8).

Finally, RAEA returns the final population (line 11). The HV-based environment selection, computational complexity and some notes on the proposed RAEA are represented in the following.

3.4. Environment selection

Algorithm 3: Environment Selection

Input :
the new trial solution $\boldsymbol{y}$;
the population Pop;
Output:
the updated population Pop;
1 Partition $Pop \cup \{\boldsymbol{y}\}$ into L fronts by
$\{B_1, \cdots, B_L\} = NDS(Pop \cup \{\boldsymbol{y}\})$.
2 Set $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in B_L} \Delta_\phi(\boldsymbol{x}, B_L)$.
3 Set $Pop = Pop \cup \{\boldsymbol{y}\} \backslash \{\boldsymbol{x}^*\}$.
4 **Return** the updated Pop.

The environment selection operation aims to update the population Pop by removing the worst solution from $Pop \cup \{\boldsymbol{y}\}$. In this paper, we adopt the HV-based selection approach for RAEA, and the details of the update method are shown in Algorithm 3.

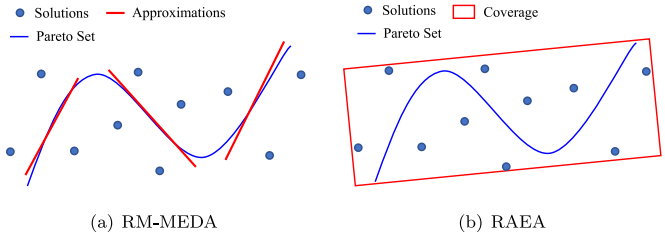


Fig. 6. Illustration of difference of the built latent spaces between RM-MEDA and PCA-assisted reproduction. (a) In RM-MEDA, the $(m-1)$-D latent spaces aim to approximate the manifold; (b) In RAEA, the latent space aims to cover the manifold of PS.

In line 1 of Algorithm 3, $Pop \cup \{\boldsymbol{y}\}$ is divided into L different non-dominated fronts by the fast non-dominated 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 [10].

3.5. Computational complexity

In this section, we consider the overall time complexity of RAEA at each generation, where the computational cost of RAEA is mainly due to the proposed dual-space search strategy. First, it is known the decomposition of SVD requires a run-time of $\mathcal{O}(p^3)$ for the construction of the latent space and the projections onto the built latent space cost $\mathcal{O}((N-p)nr)$ [55], where N is the size of population size, p is the number of the nondominated solutions, n is the dimension of variable vector and r is the dimension of latent space. For the search in both latent and decision spaces, the time complexity is $\mathcal{O}(Nn)$ with the DE operator to generate a population of offspring solutions. For the HV-based selection, the runtime is $\mathcal{O}(N^m)$ in our implementation. In a word, the time complexity of RAEA is $\mathcal{O}(p^3 + (N-p)nr + Nn + N^m)$ at each generation, while the runtime is $\mathcal{O}(N^3 + Nn + N^m)$ that all solutions in the population are non-dominated solutions.

3.6. Notes on RAEA

3.6.1. Connections between the PCA and SVD

Both PCA and SVD can be regarded as matrix decomposition approaches for the extraction of latent information, where PCA performs eigendecomposition on the covariance matrix (i.e. Eqs. (6), (7) and (8)) and SVD directly decomposes the original data matrix. Although the PCA method can also perform incremental projection, it may mislead the search of RAEA when reconstructing these projected solutions as the mean value has not changed. This is why we adopt the SVD to accelerate convergence in RAEA.

3.6.2. Difference between RAEA and RM-MEDA

Although both RAEA and RM-MEDA adopt the regularity property in the design of algorithms, RAEA allows the population to lie on any subspaces of the decision space instead of the strict $(m-1)$-D assumption in RM-MEDA. As shown in Fig. 6, an intuitive cognition is that the proposed RAEA aims to contain the manifold of PS, while the built latent spaces in RM-MEDA are proposed to approximate the PS explicitly. RAEA also benefits from the covered latent space that clustering is not required in manifold modeling.

3.6.3. Implementation details

In practice, we find that there may be few nondominated solutions in the population, especially in the initial population and the early stage of the algorithm. Although such cases are sporadic, we give a simple trick to handle this issue: the incremental projections are performed when the number of nondominated solutions is greater than or equal to 5. Moreover, the solutions in best front B_1 are set as the nondominated solutions to define the archive $Arch_{Nond}$ at each generation.

Table 1
Properties of the test instances.

Instance	Characteristics	Instance	Characteristics	Instance	Characteristics	Instance	Characteristics
GLT1	Disconnected, Nonlinear variable linkage	LZ6	Convex, Complex PS	IMF8	Concave, Nonlinear variable linkage	WFG1	Mixed, Biased
GLT2	Convex, Nonlinear variable linkage	LZ7	Convex, Local PF, Complex PS	IMF9	Convex , Multimodal, Nonlinear variable linkage	WFG2	Convex, Discontinuous, Non-separable
GLT3	Convex, Nonlinear variable linkage	LZ8	Convex, Local PF, Complex PS	IMF10	Convex, Multimodal, Nonlinear variable linkage	WFG3	Degenerate, Non-separable
GLT4	Convex, Disconnected, Nonlinear variable linkage	LZ9	Convex, Complex PS	IMOP1	Convex, Sharp-tailed	WFG4	Convex, Multimodal
GLT5	Convex, Nonlinear variable linkage	IMF1	Convex	IMOP2	Concave, Sharp-tailed	WFG5	Concave, Deceptive
GLT6	Convex, Disconnected, Nonlinear variable linkage	IMF2	Concave	IMOP3	Convex, Discontinuous	WFG6	Concave, Non-separable
LZ1	Convex, Complex PS	IMF3	Concave	IMOP4	Degenerate	WFG7	Concave, Biased
LZ2	Convex, Complex PS	IMF4	Concave	IMOP5	Disconnected	WFG8	Concave, Non-separable, Biased
LZ3	Convex, Complex PS	IMF5	Convex , Nonlinear variable linkage	IMOP6	Separable	WFG9	Concave, Non-separable, Deceptive, Biased
LZ4	Convex, Complex PS	IMF6	Concave, Nonlinear variable linkage	IMOP7	Concave		
LZ5	Convex, Complex PS	IMF7	Concave, Nonlinear variable linkage	IMOP8	Disconnected		

Table 2
Reference points of test instances.

Reference points	Instances
(1.1, 1.1)	GLT1, GLT3 LZ1-LZ5, LZ7-LZ9 IMF1-IMF3, IMF5-IMF7, IMF9-IMF10 IMOP1, IMOP2
(1.1, 11.0)	GLT2
(1.1, 2.2)	GLT4
(1.0, 1.3)	IMOP3
(1.1, 1.1, 1.1)	GLT5, GLT6 LZ6 IMF4, IMF8 IMOP4, IMOP6, IMOP7
(0.6, 0.6, 1.3)	IMOP5
(1.1, 1.1, 3.3)	IMOP8
(2.2, 4.4, 6.6)	WFG1-WFG9

4. Empirical studies

To empirically indicate the performance of the proposed RAEA in dealing with complicated MOPs, two parts of experiments have been performed:

- *Parameter Analysis and Ablation Study:* The role of the control parameters in RAEA are studied, e.g., the size of neighborhood mating pools K and the scaling factor F in the DE operator. Furthermore, some ablation studies are conducted to investigate the effect of different factors, convergence acceleration and dual-space searching strategies, in RAEA on its performance in MOPs.
- *Comparsion Study:* RAEA is compared with some representative MOEAs to indicate the performance of RAEA, including the statistical analysis, convergence speed and visual observation. Moreover, the scalability analysis of RAEA is also represented in the number of decision variables.

To be specific, five state-of-the-art MOEAs are involved in this work which also focus on offspring generation and cover the main categories of MOEA in the literature:

1. AMEA [38]: one hybrid MOEA combined Gaussian mixture model and DE operator in a probabilistic way;
2. IM-MOEA [5]: an inverse Gaussian model based MOEA, which samples trial solutions from the objective space and then maps them into the decision space;
3. MOEA/D-CMA [57]: a decomposition based MOEA that the Covariance matrix adaptation (CMA) evolution strategy and DE operator are used to generate offspring solutions;
4. RM-MEDA [4]: a kind of multiobjective estimation of distribution algorithm based on regularity model;
5. SMEA [20]: one learning-based MOEA, which performs a general mating restriction strategy by extracting the regularity property;

4.1. Experimental settings

4.1.1. Test problems

MOPs with complicated characteristics are focused in this paper. The GLT [58], LZ [13], IMF [5], IMOP [59] and WFG [60] test suites, including a total of 42 test instances, are adopted for comparisons. As summarized in Table 1, the characteristics of the selected problems are various, i.e., convex, concave, mixed and disconnection in PF, multimodal, biased and nonlinear variable linkage in PS, that challenge MOEAs.

4.1.2. Performance indicators

With the inverted generational distance (IGD) and hypervolume (HV) indicators, both indicators can measure the convergence and diversity of the solutions found by EMO approaches [61]. The details of the IGD and HV indicators are illustrated here.

For IGD, let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$, $\mathrm{IGD}_p(\mathcal{X}, \mathcal{Y})$ is defined by

$$\mathrm{IGD}_p(\mathcal{X}, \mathcal{Y}) = \left(\frac{1}{M} \sum_{i=1}^{M} \mathrm{dist}_p\left(\mathbf{y}_i, \mathcal{X}\right) \right)^{1/p}$$

where $\mathrm{dist}_p\left(\mathbf{y}_i, \mathcal{X}\right)$ is the minimal distance between the objective vector $\mathbf{y}_i$ and all vectors in the set $\mathcal{X}$, and we choose $p = 1$ in this paper. Let P^* be a set of uniformly distributed points in the theoretical PF, and P be an approximation to the PF, we computer $\mathrm{IGD}_1(P^*, P)$ to indicate its performance. A smaller IGD indicates better algorithmic performance.

Table 3
Statistical results (Mean(Std. Dev.)) obtained by five RAEA variants with different K values over 30 independent runs on the GLT test suite in terms of the IGD indicator.

Instances(m,n)	$K = 2$	$K = 4$	$K = 6$	$K = 8$	$K = 10$
GLT1(2,10)	2.2267e−2 (4.66e−2)	2.7378e−2 (5.94e−2)	2.9903e−2 (5.96e−2)	1.6374e−2 (4.55e−2)	1.7396e−2 (4.82e−2)
GLT2(2,10)	2.8818e−2 (1.03e−3)	2.7621e−2 (4.95e−4)	2.7683e−2 (5.70e−4)	2.7671e−2 (5.53e−4)	2.7918e−2 (7.37e−4)
GLT3(2,10)	6.1651e−3 (2.50e−3)	4.9951e−3 (1.29e−4)	5.1694e−3 (6.78e−4)	5.3142e−3 (1.34e−3)	5.1948e−3 (1.31e−4)
GLT4(2,10)	7.5238e−2 (1.02e−1)	4.5637e−2 (9.57e−2)	3.7232e−2 (9.58e−2)	2.7218e−2 (5.97e−2)	2.5704e−2 (8.50e−2)
GLT5(3,10)	2.9767e−2 (4.30e−4)	2.9321e−2 (2.48e−4)	2.9244e−2 (2.17e−4)	2.9303e−2 (2.26e−4)	2.9442e−2 (2.65e−4)
GLT6(3,10)	1.0020e−1 (7.33e−2)	1.0985e−1 (1.72e−1)	1.1291e−1 (1.80e−1)	6.4978e−2 (1.33e−1)	6.6938e−2 (1.63e−1)

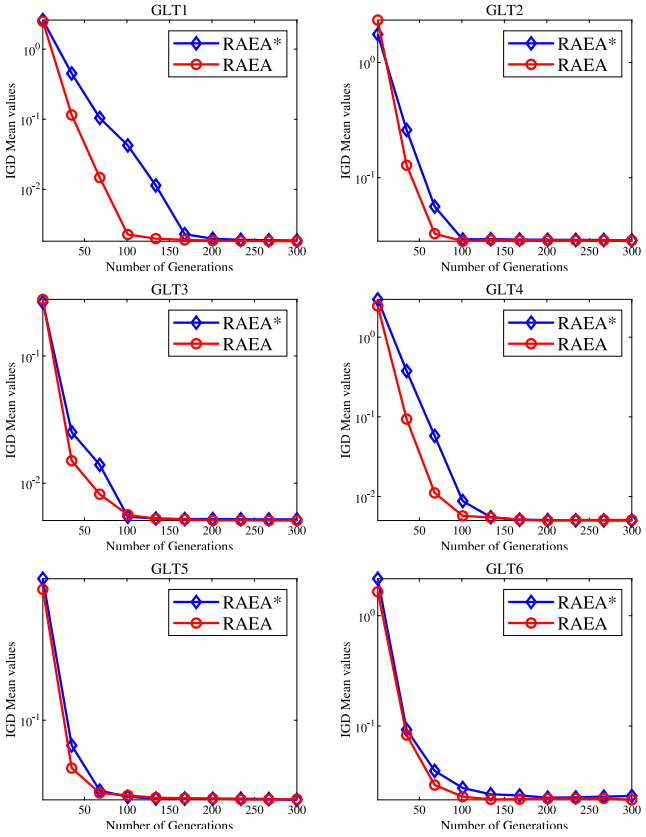


Fig. 7. The mean IGD indicator values versus numbers of generations for the RAEA with or without convergence acceleration mechanism over 30 independent runs on GLT1-GLT6.

For HV, the hypervolume indicator is defined as:

$$HV(P,r) = VOL\left\{\bigcup_{x\in P}\left[f_1(x),r_1\right]\times\cdots\left[f_m(x),r_m\right]\right\},$$

where a reference vector $r = (r_1,\ldots,r_m)$ is dominated by any objective vectors in the objective space, and $VOL(\cdot)$ is the Lebesgue measure. The larger the HV indicator value is, the better convergence and diversity of AFs corresponding to P. In terms of these test instances used in this paper, their reference points are shown in Table 2.

4.1.3. Parameter settings

Parameter settings have a significant influence on the performance of the algorithms. Therefore, we choose the same parameters in the original literature for each comparison algorithm. For each test instance, the population size of all six algorithms is set to 100. 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 DE operator. The specific parameters of each algorithm are set as follows.

- Parameters of AMEA
  - number of clusters: $K = 5$.
- 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
  - 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 RAEA
  - number of neighborhood mating pools: $K = 8$ recommended by the following experiments.

4.2. Algorithm analysis

In this subsection, we investigate three factors that may affect the performance of the proposed RAEA, i.e., the sensitivity of parameter K, the effect of the convergence acceleration, and the effect of the dual-space search strategy. The analysis and comparisons are conducted on 10-D GLT test instances with 300 generations.

4.2.1. Parameter sensitivity of the mating pool size K

In the proposed RAEA, the size of the neighborhood mating pool is determined by the parameter K that controls the mating parents in both the latent and decision spaces. To be specific, a larger K value with a bigger neighborhood size may prefer global exploration while a smaller K value helps local exploitation. Actually, this parameter might control the balance between the global and local search to some degree.

To learn about the sensitivity of the probability of the mating pool size K, $K = 2, 4, 6, 8, 10$ are separately tested. The GLT test suite is adopted for this analysis, and the stopping condition is determined by the total number of generations which is 300 in each run of five algorithms. The statistical IGD values obtained by RAEA with different settings of K are given in Table 3.

The experimental results indicate that the five compared variants have achieved similar results where RAEA with $K = 8$ achieves the best results, followed by algorithms with $K = 4$, $K = 10$, $K = 6$ and $K = 2$. A reason why RAEA does not work well with a smaller neighborhood mating pool might be that it reduces the capability of global search of RAEA. Therefore, we adopt the RAEA with $K = 8$ for the following analysis and comparisons. Moreover, we should point out that the appropriate size of a neighborhood mating pool is still problem-dependent.

Table 4
Statistical results (Mean(Std. Dev.)) obtained by four RAEA variants with different search strategies over 30 independent runs on the GLT test suite in terms of the IGD indicator.

Instances(m,n)	RAEA-A	RAEA-B	RAEA-C	RAEA-DSS
GLT1(2,10)	3.0856e−2 (6.06e−2) −	1.2462e−2 (4.11e−2) −	3.3975e−2 (6.09e−2) −	7.2641e−3 (3.03e−2)
GLT2(2,10)	2.8856e−2 (6.81e−4) −	2.7707e−2 (6.11e−4) =	2.7690e−2 (3.86e−4) =	2.7660e−2 (4.68e−4)
GLT3(2,10)	5.4089e−3 (1.82e−3) =	7.6994e−3 (1.33e−2) −	6.0746e−3 (4.94e−3) =	5.1099e−3 (1.07e−4)
GLT4(2,10)	5.2405e−2 (1.04e−1) −	3.7492e−2 (9.35e−2) −	5.9442e−2 (1.25e−1) −	7.6298e−3 (1.48e−2)
GLT5(3,10)	2.9203e−2 (2.22e−4) =	2.9383e−2 (2.54e−4) −	2.9485e−2 (2.93e−4) −	2.9237e−2 (2.33e−4)
GLT6(3,10)	1.1801e−1 (1.42e−1) =	6.9026e−2 (6.40e−2) =	8.4327e−2 (8.34e−2) −	5.1827e−2 (5.60e−2)
+/-/=	0/3/3	0/4/2	0/4/2	

Table 5
Statistical results (Mean(Std. Dev.)[rank]) obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RAEA over 30 independent runs on the GLT, LZ, IMF, IMOP and WFG test suites in terms of the IGD values.

Instances(m,n)	AMEA	IM-MOEA	MOEA/D-CMA	RM-MEDA	SMEA	RAEA
GLT1(2,20)	4.6709e−2(3.01e−2)[5]−	3.1807e−2(2.95e−3)[4]−	5.1341e−3(4.96e−4)[2]−	6.8693e−2(3.58e−2)[6]−	2.8969e−2(1.42e−2)[3]−	1.9360e−3(3.62e−5)[1]
GLT2(2,20)	3.2550e−2(7.59e−3)[2]−	7.1958e−1(2.46e−1)[6]−	2.8187e−1(8.45e−2)[5]−	4.6602e−2(1.97e−2)[3]−	4.9277e−2(5.78e−2)[4]−	2.7714e−2(4.46e−4)[1]
GLT3(2,20)	9.2540e−2(3.93e−2)[5]−	1.5361e−1(2.97e−2)[6]−	8.9923e−2(4.31e−2)[4]−	7.2058e−2(2.39e−2)[2]−	8.1398e−2(2.33e−2)[3]−	5.1273e−3(1.05e−4)[1]
GLT4(2,20)	1.5911e−1(1.07e−1)[6]−	5.8231e−2(1.39e−2)[2]−	8.4383e−2(8.69e−2)[3]−	1.0982e−1(7.59e−2)[5]−	1.0840e−1(6.44e−2)[4]−	5.0820e−3(3.54e−5)[1]
GLT5(3,20)	8.5032e−2(9.28e−3)[4]−	7.6442e−2(3.66e−3)[3]−	4.8817e−2(2.41e−3)[2]−	8.9084e−2(7.33e−3)[5]−	9.1870e−2(7.26e−3)[6]−	2.9636e−2(2.58e−4)[1]
GLT6(3,20)	1.4935e−1(1.27e−1)[6]−	1.1876e−1(4.62e−2)[4]−	1.4451e−1(5.35e−2)[5]−	9.6596e−2(2.33e−2)[3]−	8.5824e−2(3.49e−2)[2]−	3.2918e−2(5.63e−2)[1]
LZ1(2,30)	7.2440e−3(1.32e−3)[5]−	7.4160e−3(3.82e−4)[6]−	4.0779e−3(3.98e−5)[4]−	3.6368e−3(2.91e−5)[2]=	3.6909e−3(2.18e−5)[3]−	3.6268e−3(1.46e−5)[1]
LZ2(2,30)	1.2825e−1(2.98e−2)[6]−	7.1765e−2(1.36e−2)[5]−	4.7540e−2(6.70e−3)[2]=	7.0799e−2(1.45e−2)[4]−	5.2612e−2(5.64e−3)[3]−	4.6702e−2(6.35e−3)[1]
LZ3(2,30)	8.9006e−2(3.23e−2)[6]−	4.5555e−2(1.80e−2)[5]−	3.1216e−2(2.21e−2)[4]−	2.9759e−2(8.38e−3)[3]−	2.7211e−2(3.39e−3)[2]−	2.0089e−2(3.66e−3)[1]
LZ4(2,30)	1.0117e−1(3.47e−2)[6]−	3.2241e−2(1.49e−2)[3]−	3.4490e−2(1.42e−2)[5]−	3.3277e−2(6.45e−3)[4]−	2.9709e−2(5.27e−3)[2]−	2.1637e−2(2.78e−3)[1]
LZ5(2,30)	4.9718e−2(6.78e−3)[6]−	3.2024e−2(6.44e−3)[5]−	2.6795e−2(3.68e−3)[3]−	2.6780e−2(7.41e−3)[2]−	2.6953e−2(4.77e−3)[4]−	1.8926e−2(2.20e−3)[1]
LZ6(3,30)	4.4760e−1(9.54e−2)[5]−	2.6042e−1(2.70e−2)[2]−	2.8482e−1(7.83e−2)[3]=	3.3345e−1(6.63e−2)[4]−	5.7278e−1(1.61e−1)[6]−	2.5639e−1(1.00e−1)[1]
LZ7(2,30)	4.5687e−1(5.82e−2)[5]−	3.3939e−1(4.98e−2)[2]+	3.2211e−1(1.42e−1)[1]+	1.2081e+0(5.31e−1)[6]−	3.9481e−1(9.99e−2)[4]=	3.7129e−1(4.48e−2)[3]
LZ8(2,30)	2.7369e−1(3.07e−2)[6]−	1.0629e−1(1.47e−2)[5]−	4.6344e−2(2.94e−2)[1]+	8.5502e−2(5.10e−2)[4]=	6.5065e−2(2.95e−2)[2]=	8.3636e−2(4.36e−2)[3]
LZ9(2,30)	1.3945e−1(5.22e−2)[6]−	6.7757e−2(1.45e−2)[4]−	5.4153e−2(7.66e−3)[2]−	8.2963e−2(1.71e−2)[5]−	5.5288e−2(1.07e−2)[3]−	4.7015e−2(6.30e−3)[1]
IMF1(2,30)	6.7320e−2(2.54e−2)[6]−	6.1077e−3(1.82e−3)[5]−	4.0220e−3(5.42e−5)[3]−	3.8401e−3(3.25e−4)[2]=	4.4094e−3(1.88e−3)[4]−	3.7557e−3(2.95e−5)[1]
IMF2(2,30)	1.1838e−1(3.46e−2)[5]=	1.8775e−2(3.43e−2)[3]=	4.3967e−2(1.51e−1)[4]=	3.3620e−1(3.06e−1)[6]−	4.4333e−3(1.11e−4)[1]+	4.5215e−3(5.55e−4)[2]
IMF3(2,30)	3.0407e−1(6.37e−2)[6]−	5.3353e−3(1.79e−4)[4]−	3.3027e−3(4.72e−5)[3]−	5.9155e−2(6.40e−2)[5]−	3.0632e−3(2.76e−5)[2]−	3.0126e−3(1.24e−5)[1]
IMF4(3,30)	1.1157e+1(3.81e+0)[6]−	1.5338e−1(1.14e−2)[5]−	8.6978e−2(3.86e−2)[2]−	9.0553e−2(7.40e−3)[3]−	1.1371e−1(2.62e−2)[4]−	7.5553e−2(9.01e−4)[1]
IMF5(2,30)	2.3720e−2(1.24e−2)[6]−	5.3585e−3(1.54e−4)[4]−	4.9698e−3(1.71e−4)[3]−	1.0332e−2(5.68e−3)[5]−	4.7863e−3(3.73e−4)[2]=	4.7803e−3(2.08e−4)[1]
IMF6(2,30)	6.0223e−2(1.93e−2)[6]−	7.5529e−3(2.61e−4)[1]+	2.4937e−2(4.88e−3)[4]−	2.4917e−2(1.29e−2)[3]=	2.5649e−2(3.84e−3)[5]−	2.1292e−2(2.08e−3)[2]
IMF7(2,30)	2.3086e−1(2.15e−2)[6]−	7.2947e−3(3.82e−4)[4]−	4.9255e−3(1.03e−3)[2]=	5.9937e−2(4.76e−2)[5]−	3.6266e−3(5.49e−4)[1]+	5.5163e−3(2.34e−3)[3]
IMF8(3,30)	8.7962e−1(7.08e−2)[6]−	1.4391e−1(4.75e−3)[1]=	2.2987e−1(1.18e−1)[4]−	2.2402e−1(1.35e−1)[3]−	3.0275e−1(9.45e−2)[5]=	2.1573e−1(1.30e−1)[2]
IMF9(2,30)	1.1344e−1(3.21e−2)[5]−	6.0029e−3(1.01e−3)[2]−	2.3765e−2(2.54e−2)[3]−	1.1513e−1(3.24e−2)[6]−	2.6967e−2(2.81e−2)[4]−	4.0577e−3(9.27e−5)[1]
IMF10(2,30)	6.1405e+1(1.30e+1)[5]−	6.5029e+0(2.79e+0)[2]−	1.9692e+1(8.05e+0)[3]−	7.4964e+1(1.98e+1)[6]−	3.2355e+1(1.88e+1)[4]−	9.7130e−1(8.55e−1)[1]
IMOP1(2,10)	8.0540e−2(1.18e−2)[4]−	3.3705e−1(8.68e−3)[6]−	1.0407e−1(3.82e−3)[3]−	3.2058e−2(8.36e−3)[2]−	1.1768e−1(1.78e−2)[5]−	5.3107e−3(1.97e−4)[1]
IMOP2(2,10)	7.5403e−2(7.75e−3)[4]−	3.6076e−1(1.36e−2)[6]−	1.6638e−2(8.80e−3)[1]+	5.8971e−2(8.63e−3)[3]−	1.2478e−1(6.66e−3)[5]−	3.2596e−2(4.11e−3)[2]
IMOP3(2,10)	1.6359e−1(1.34e−1)[4]−	5.2823e−1(1.42e−2)[6]−	1.0536e−2(1.86e−3)[2]−	1.3699e−1(1.41e−1)[3]−	3.8581e−1(1.71e−1)[5]−	3.6853e−3(6.94e−5)[1]
IMOP4(3,10)	4.2831e−2(3.79e−3)[4]−	4.3849e−1(1.34e−2)[6]−	2.9473e−2(9.66e−4)[2]−	2.8387e−2(6.81e−3)[3]−	1.1481e−1(2.26e−2)[5]−	7.0857e−3(1.23e−4)[1]
IMOP5(3,10)	1.2168e−1(2.52e−2)[3]−	2.8312e−1(1.05e−2)[6]−	8.0075e−2(1.40e−2)[2]−	1.9991e−1(2.20e−2)[5]−	1.5024e−1(3.35e−2)[4]−	7.6193e−2(1.91e−2)[1]
IMOP6(3,10)	1.7618e−1(5.44e−2)[5]−	2.8529e−1(2.13e−2)[6]−	1.4505e−1(7.98e−2)[3]−	1.0849e−1(8.22e−2)[2]−	1.6118e−1(2.84e−2)[4]−	4.9250e−2(1.26e−3)[1]
IMOP7(3,10)	9.2107e−2(3.13e−2)[3]−	5.3092e−1(1.73e−2)[6]−	7.5662e−2(1.63e−2)[2]−	2.2713e−1(4.06e−2)[5]−	1.9213e−1(1.60e−2)[4]−	5.9675e−2(6.06e−3)[1]
IMOP8(3,10)	1.9500e−1(5.90e−2)[3]+	4.3795e−1(3.52e−2)[6]−	1.6404e−1(7.97e−3)[1]+	1.7994e−1(9.04e−2)[2]+	2.2806e−1(6.16e−2)[4]+	3.1868e−1(9.56e−2)[5]
WFG1(3,20)	1.6701e+0(3.05e−2)[4]+	1.4580e+0(2.62e−2)[1]+	1.5016e+0(4.51e−2)[2]+	2.0371e+0(8.60e−2)[6]−	1.6195e+0(2.25e−2)[3]+	1.8037e+0(2.21e−2)[5]
WFG2(3,20)	3.9121e−1(1.30e−1)[6]−	2.6183e−1(1.21e−2)[3]−	2.8240e−1(1.92e−2)[5]−	2.6019e−1(8.51e−3)[2]−	2.6331e−1(1.28e−2)[4]−	2.3709e−1(6.25e−3)[1]
WFG3(3,20)	2.2759e−1(5.05e−2)[5]−	2.1896e−1(1.07e−2)[4]−	1.8013e−1(2.19e−2)[2]−	3.4911e−1(2.40e−2)[6]−	2.1435e−1(3.34e−2)[3]−	5.3740e−2(1.79e−2)[1]
WFG4(3,20)	3.3689e−1(2.11e−2)[3]−	3.2668e−1(1.15e−2)[2]−	3.4265e−1(1.44e−2)[4]−	3.6046e−1(1.24e−2)[5]−	3.6766e−1(1.38e−2)[6]−	2.8826e−1(8.17e−3)[1]
WFG5(3,20)	3.5889e−1(3.90e−2)[6]−	3.3196e−1(8.91e−3)[5]−	3.0425e−1(9.46e−3)[3]−	3.2755e−1(2.24e−2)[4]−	3.0299e−1(1.69e−2)[2]−	2.8486e−1(7.23e−3)[1]
WFG6(3,20)	4.3299e−1(4.68e−2)[6]−	3.6605e−1(1.45e−2)[3]−	2.9040e−1(1.57e−2)[2]=	3.9317e−1(3.73e−2)[5]−	3.6713e−1(2.08e−2)[4]−	2.8479e−1(1.47e−2)[1]
WFG7(3,20)	3.9651e−1(6.04e−2)[6]−	3.5616e−1(1.41e−2)[3]−	3.1429e−1(7.37e−3)[2]−	3.6427e−1(1.05e−2)[4]−	3.9388e−1(2.03e−2)[5]−	2.8653e−1(4.71e−3)[1]
WFG8(3,20)	5.3124e−1(4.72e−2)[6]−	3.7985e−1(9.81e−3)[3]−	3.7059e−1(1.03e−2)[2]−	4.6083e−1(1.95e−2)[4]−	4.7704e−1(1.65e−2)[5]−	3.0219e−1(9.97e−3)[1]
WFG9(3,20)	3.3455e−1(3.57e−2)[3]−	3.4367e−1(1.01e−2)[4]−	3.2424e−1(1.75e−2)[2]−	3.5498e−1(1.49e−2)[6]−	3.4618e−1(1.06e−2)[5]−	2.7278e−1(1.71e−2)[1]
Ranking	5.0476	3.8095	2.9286	4.1190	3.5714	1.5238
+/−/=	2/39/1	3/37/2	5/32/5	2/36/4	4/34/4	

4.2.2. Effect of acceleration mechanism

In Algorithm 2 Steps 3–4, the latent space is constructed with the nondominated solution, while these dominated solutions are incrementally projected thus built space for the acceleration of population by the folding-in procedures of SVD method. In order to investigate the effects of the convergence acceleration mechanism on the performance of the proposed RAEA, we compare two variants of RAEA, one using the acceleration mechanism (same as in the above section), and the other not that the entire population is adopted to build the latent space (denoted as RAEA* hereafter).

As observed from Fig. 7, although the RAEA and RAEA* have similar IGD values after 300 generations, RAEA has the fastest convergence speed for overall GLT1-GLT6 test instances compared with the variant RAEA* without the acceleration mechanism, especially in the early evolutionary stages. It can be summarized that the proposed acceleration mechanism will accelerate the population convergences of RAEA in solving MOPs.

4.2.3. Effect of dual-space searching strategy

During the offspring generation, the dual-space search strategy is proposed to adopt the DE operator for searching in both latent and decision space to generate offspring solutions. To be specific, three different searching strategies can be used in RAEA to form the following variants to validate the effectiveness of RAEA with DSS strategy (called RAEA-DSS):

Table 6
Statistical results (Mean(Std. Dev.)[rank]) obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RAEA over 30 independent runs on the GLT, LZ, IMF, IMOP and WFG test suites in terms of the HV values.

Instances(m,n)	AMEA	IM-MOEA	MOEA/D-CMA	RM-MEDA	SMEA	RAEA
GLT1(2,20)	3.7250e−1(5.81e−2)[5]−	4.0781e−1(9.83e−3)[3]−	4.7880e−1(8.24e−4)[2]−	3.6410e−1(5.43e−2)[6]−	3.9334e−1(2.58e−2)[4]−	4.8116e−1(2.10e−4)[1]
GLT2(2,20)	8.1632e−1(3.56e−3)[2]−	6.9390e−1(2.48e−2)[6]−	7.6748e−1(1.56e−2)[5]−	8.0511e−1(6.90e−3)[4]−	8.0890e−1(2.07e−2)[3]−	8.1882e−1(1.27e−4)[1]
GLT3(2,20)	9.4966e−1(3.36e−3)[4]−	9.4196e−1(1.48e−3)[6]−	9.4879e−1(2.25e−3)[5]−	9.4978e−1(1.88e−3)[3]−	9.5062e−1(1.95e−3)[2]−	9.5776e−1(2.34e−5)[1]
GLT4(2,20)	4.9928e−1(9.89e−2)[5]−	5.5038e−1(7.12e−3)[2]−	4.9699e−1(9.94e−2)[6]−	5.4871e−1(2.33e−2)[3]−	5.2718e−1(6.68e−2)[4]−	5.8384e−1(9.16e−5)[1]
GLT5(3,20)	9.3477e−1(6.69e−3)[4]−	9.3785e−1(4.30e−3)[3]−	9.6321e−1(1.53e−3)[2]−	9.3161e−1(5.58e−3)[6]−	9.3218e−1(5.04e−3)[5]−	9.7585e−1(1.59e−4)[1]
GLT6(3,20)	8.3407e−1(1.34e−1)[6]−	8.9002e−1(2.84e−2)[5]−	8.9599e−1(3.50e−2)[4]−	8.9678e−1(2.93e−2)[3]−	9.1385e−1(4.28e−2)[2]−	9.5775e−1(6.23e−2)[1]
LZ1(2,30)	7.1305e−1(2.52e−3)[6]−	7.1409e−1(5.30e−4)[5]−	7.1931e−1(1.10e−4)[4]−	7.2061e−1(1.00e−4)[1]+	7.2034e−1(5.58e−5)[3]−	7.2060e−1(2.43e−5)[2]
LZ2(2,30)	5.4788e−1(3.41e−2)[6]−	6.2010e−1(1.67e−2)[5]−	6.5461e−1(1.02e−2)[2]=	6.2323e−1(1.96e−2)[4]−	6.4669e−1(9.42e−3)[3]−	6.5602e−1(1.05e−2)[1]
LZ3(2,30)	6.3926e−1(1.51e−2)[6]−	6.7740e−1(8.39e−3)[5]−	6.8595e−1(1.65e−2)[4]−	6.8997e−1(7.90e−3)[3]−	6.9000e−1(4.30e−3)[2]−	6.9871e−1(4.59e−3)[1]
LZ4(2,30)	6.3623e−1(1.85e−2)[6]−	6.8939e−1(8.71e−3)[2]−	6.8006e−1(1.52e−2)[5]−	6.8579e−1(6.50e−3)[4]−	6.8690e−1(6.57e−3)[3]−	6.9749e−1(3.14e−3)[1]
LZ5(2,30)	6.6460e−1(6.02e−3)[6]−	6.8734e−1(3.84e−3)[5]−	6.8930e−1(3.81e−3)[4]−	6.9506e−1(5.59e−3)[2]−	6.9137e−1(3.79e−3)[3]−	7.0004e−1(2.82e−3)[1]
LZ6(3,30)	1.3809e−1(7.07e−2)[5]−	3.1467e−1(5.93e−3)[3]−	3.2354e−1(3.94e−2)[2]−	1.7797e−1(7.04e−2)[4]−	4.6331e−2(4.13e−2)[6]−	4.2847e−1(5.96e−2)[1]
LZ7(2,30)	1.7366e−1(4.23e−2)[5]−	2.8056e−1(4.35e−2)[2]+	3.2587e−1(1.49e−1)[1]+	3.6166e−2(6.64e−2)[6]−	2.3658e−1(7.84e−2)[3]=	2.3183e−1(5.74e−2)[4]
LZ8(2,30)	3.5067e−1(2.68e−2)[6]−	5.5628e−1(2.55e−2)[5]−	6.5681e−1(3.64e−2)[1]+	6.0431e−1(6.51e−2)[4]=	6.3301e−1(3.83e−2)[2]=	6.1120e−1(5.67e−2)[3]
LZ9(2,30)	2.8113e−1(4.27e−2)[6]−	3.4818e−1(1.86e−2)[5]−	3.5891e−1(1.45e−2)[3]−	3.4990e−1(1.87e−2)[4]−	3.6979e−1(1.32e−2)[2]−	3.7781e−1(1.23e−2)[1]
IMF1(2,30)	6.1620e−1(3.16e−2)[6]−	7.1578e−1(2.27e−3)[5]−	7.1949e−1(1.41e−4)[3]−	7.1986e−1(8.39e−4)[2]=	7.1913e−1(2.75e−3)[4]−	7.2023e−1(1.01e−4)[1]
IMF2(2,30)	3.2686e−1(3.60e−2)[5]=	4.2361e−1(3.86e−2)[3]=	4.1908e−1(8.76e−2)[4]=	2.5042e−1(1.79e−1)[6]−	4.4481e−1(1.06e−4)[1]+	4.4421e−1(1.46e−3)[2]
IMF3(2,30)	1.2655e−1(4.81e−2)[6]−	3.8523e−1(2.95e−4)[4]−	3.8805e−1(1.28e−4)[3]−	3.3001e−1(6.43e−2)[5]−	3.8868e−1(9.42e−5)[2]−	3.8876e−1(4.78e−5)[1]
IMF4(3,30)	0.0000e+0(0.00e+0)[6]−	3.6531e−1(2.02e−2)[5]−	5.0128e−1(8.40e−2)[2]−	4.8085e−1(1.35e−2)[3]−	4.3897e−1(5.21e−2)[4]−	5.6460e−1(7.18e−4)[1]
IMF5(2,30)	6.8448e−1(2.17e−2)[6]−	7.1515e−1(4.98e−4)[4]−	7.1747e−1(2.92e−4)[2]=	7.0603e−1(1.24e−2)[5]−	7.1737e−1(1.28e−3)[3]=	7.1764e−1(3.66e−4)[1]
IMF6(2,30)	3.8955e−1(1.64e−2)[6]−	4.3863e−1(3.70e−4)[1]+	4.3442e−1(2.55e−3)[4]−	4.3071e−1(1.29e−2)[5]=	4.3641e−1(2.00e−3)[3]=	4.3693e−1(1.11e−3)[2]
IMF7(2,30)	1.9634e−1(1.41e−2)[6]−	3.8237e−1(4.66e−4)[4]−	3.8528e−1(1.41e−3)[2]=	3.3167e−1(4.08e−2)[5]−	3.8739e−1(9.03e−4)[1]+	3.8478e−1(2.84e−3)[3]
IMF8(3,30)	5.6527e−3(1.49e−2)[6]−	3.6631e−1(8.44e−3)[5]−	4.6649e−1(6.08e−2)[2]−	4.1606e−1(1.01e−1)[4]−	4.2431e−1(3.58e−2)[3]−	4.8602e−1(7.32e−2)[1]
IMF9(2,30)	5.7344e−1(3.77e−2)[5]−	7.1563e−1(2.15e−3)[2]−	6.9331e−1(2.93e−2)[3]−	5.7075e−1(3.98e−2)[6]−	6.9059e−1(3.17e−2)[4]−	7.1903e−1(2.37e−4)[1]
IMF10(2,30)	0.0000e+0(0.00e+0)[4]−	5.3863e−3(2.26e−2)[3]−	0.0000e+0(0.00e+0)[5]−	0.0000e+0(0.00e+0)[6]−	1.1131e−2(3.46e−2)[2]−	1.1143e−1(2.07e−2)[1]
IMOP1(2,10)	9.8373e−1(2.59e−3)[4]−	7.3032e−1(1.99e−2)[6]−	9.8500e−1(8.28e−5)[3]−	9.8774e−1(1.31e−4)[1]+	9.4550e−1(2.43e−2)[5]−	9.8722e−1(5.54e−5)[2]
IMOP2(2,10)	2.2694e−1(1.54e−3)[4]−	1.0392e−1(2.91e−3)[6]−	2.3099e−1(3.01e−4)[2]−	2.2864e−1(1.01e−3)[3]−	2.0984e−1(2.47e−3)[5]−	2.3226e−1(1.36e−4)[1]
IMOP3(2,10)	4.7837e−1(9.44e−2)[4]−	2.3645e−1(1.46e−2)[6]−	6.5634e−1(6.66e−4)[2]−	5.1673e−1(1.01e−1)[3]−	3.0640e−1(1.00e−1)[5]−	6.5881e−1(3.60e−5)[1]
IMOP4(3,10)	3.8591e−1(5.29e−3)[4]−	1.2142e−1(7.15e−3)[6]−	4.2423e−1(4.50e−4)[3]−	4.0926e−1(1.22e−2)[2]−	3.0024e−1(3.07e−2)[5]−	4.3489e−1(8.41e−5)[1]
IMOP5(3,10)	4.8725e−1(1.95e−2)[5]−	5.4953e−1(2.23e−2)[1]+	5.1210e−1(6.88e−3)[3]=	4.7360e−1(1.24e−2)[6]−	4.9886e−1(1.56e−2)[4]−	5.1270e−1(8.03e−3)[2]
IMOP6(3,10)	4.3167e−1(2.67e−2)[5]−	2.9290e−1(1.32e−2)[6]−	4.6428e−1(5.53e−2)[3]−	4.7750e−1(4.71e−2)[2]−	4.4590e−1(2.01e−2)[4]−	5.1596e−1(8.26e−4)[1]
IMOP7(3,10)	4.7697e−1(2.97e−2)[3]−	1.8094e−1(6.23e−3)[6]−	5.0579e−1(8.72e−3)[1]+	3.9590e−1(2.67e−2)[5]−	4.2378e−1(1.35e−2)[4]−	4.9447e−1(8.87e−3)[2]
IMOP8(3,10)	4.2138e−1(3.50e−2)[3]+	2.4558e−1(2.00e−2)[6]−	4.6556e−1(8.54e−3)[1]+	4.3109e−1(5.75e−2)[2]+	3.9658e−1(3.67e−2)[4]+	3.4758e−1(6.84e−2)[5]
WFG1(3,20)	1.9949e−1(1.39e−2)[4]+	3.0029e−1(1.42e−2)[1]+	2.8021e−1(1.19e−2)[2]+	5.7783e−2(2.95e−2)[4]−	2.5131e−1(5.11e−3)[4]+	1.3690e−1(1.32e−2)[5]
WFG2(3,20)	8.0987e−1(3.96e−2)[6]−	8.8229e−1(5.07e−3)[2]−	8.5931e−1(1.12e−2)[5]−	8.6945e−1(4.92e−3)[3]−	8.6630e−1(6.47e−3)[4]−	9.1229e−1(5.67e−3)[1]
WFG3(3,20)	3.3154e−1(1.79e−2)[5]−	3.3638e−1(5.25e−3)[4]−	3.4708e−1(1.20e−2)[2]−	2.7321e−1(7.94e−3)[6]−	3.3900e−1(1.19e−2)[3]−	3.9636e−1(8.09e−3)[1]
WFG4(3,20)	4.4761e−1(9.23e−3)[4]−	4.9984e−1(3.78e−3)[1]=	4.6483e−1(7.62e−3)[3]−	4.4112e−1(6.12e−3)[5]−	4.3310e−1(7.26e−3)[6]−	4.9959e−1(5.10e−3)[2]
WFG5(3,20)	4.1447e−1(2.02e−2)[6]−	4.7408e−1(7.29e−3)[2]−	4.6207e−1(9.32e−3)[3]−	4.4318e−1(9.40e−3)[5]−	4.5645e−1(1.17e−2)[4]−	5.1639e−1(2.79e−3)[1]
WFG6(3,20)	3.9194e−1(1.53e−2)[6]−	4.6707e−1(1.31e−2)[3]−	4.7009e−1(1.16e−2)[2]−	4.1939e−1(1.83e−2)[5]−	4.2930e−1(1.11e−2)[4]−	5.4305e−1(1.78e−2)[1]
WFG7(3,20)	4.2177e−1(2.38e−2)[5]−	4.9356e−1(3.98e−3)[2]−	4.9000e−1(9.94e−3)[3]−	4.3710e−1(6.35e−3)[4]−	4.1709e−1(1.01e−2)[6]−	5.5738e−1(1.05e−3)[1]
WFG8(3,20)	3.3438e−1(1.42e−2)[6]−	4.3933e−1(4.63e−3)[2]−	4.1281e−1(7.77e−3)[3]−	3.6500e−1(1.29e−2)[4]−	3.5256e−1(7.86e−3)[5]−	4.6271e−1(9.17e−3)[1]
WFG9(3,20)	4.3707e−1(1.66e−2)[4]−	4.8423e−1(6.46e−3)[2]−	4.4221e−1(2.67e−2)[3]−	4.1618e−1(9.72e−3)[6]−	4.2080e−1(5.07e−3)[5]−	5.3161e−1(2.35e−2)[1]
Ranking	5.2308	3.4359	3.0256	4.2051	3.3590	1.4103
+/−/=	2/39/1	4/36/2	5/32/5	3/36/3	5/33/4	

1. RAEA-A: the DE operator is only conducted in the decision space;
2. RAEA-B: the DE operator is only conducted in the latent space;
3. RAEA-C: the DE operator is conducted in both the latent and decision space, while the parents in the decision are sampled from the mapped trial vectors.

The statistical results are summarized in Table 4. From the table, we can see that RAEA-DSS achieves 5 best IGD indicator values out of 6. Moreover, RAEA-DSS yields 3, 0, 3 significantly better, worse and similar indicator values than RAEA-A, 4, 0, 2 significantly better, worse and similar indicator values than RAEA-B, and 4, 0, 2 better, worse and similar ones than RAEA-C, respectively. Moreover, a reason RAEA-C does not work well is that only searching by the latent space and corresponding mapped trial vectors may be inefficient and mislead the search without the current population solutions. We thus may conclude that the dual-space search strategy has a dramatic contribution to the performance of RAEA.

4.3. Comparisons with state-of-the-art MOEAs

In this section, we compare RAEA with AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA on 42 test instances that each instance will terminate after 600 generations over 30 independent runs. The statistics of IGD and HV results achieved by these compared algorithms are displayed in Table 5 and 6, respectively. In these tables, the mean and standard deviations are recorded. Moreover, the mean IGD and HV values for each instance are sorted in ascending order, and the numbers in the square brackets of Tables are their ranks. The Wilcoxon's rank sum test at a 5% significance level is performed in terms of the indicators obtained by each pair of algorithms. "+", "-", and "=" in Tables 5 and 6 denote the performance of the comparison algorithm is better than, worse than, and similar to that of RMDE according to Wilcoxon's rank sum test, respectively. The bold data with gray background in the table are the optimal mean indicator values yielded by the algorithms for each instance.

As can be seen in Table 5, in terms of the IGD indicator values, considering all problems, we observe that according to the ranking, the algorithms with the best to worst performance are RAEA, MOEA/D-CMA, SMEA, IM-MOEA, RM-MEDA and AMEA. Specifically, according to Wilcoxon's rank sum test, in the comparisons, RAEA achieves 39, 37, 32, 36 and 34 better, 2, 3, 5, 2 and 4 worse, and 1, 2, 5, 4 and 4 similar mean indicator values than AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA, respectively.

Taking each test suite individually, we see that for the GLT test suite, RAEA achieves the overall best IGD indicator values. For the LZ test suites, RAEA obtains 7 best IGD indicator values out of 9, and 6 best results out of 9 on the IMF test suites. For the IMOP and WFG test suites, RAEA obtains 12 best IGD indicator values out of 17 that IM-MOEA achieves the best result on WFG1 test instance, and MOEA/D-CMA achieves the best results on IMOP2 and IMOP8 test instances, respectively.

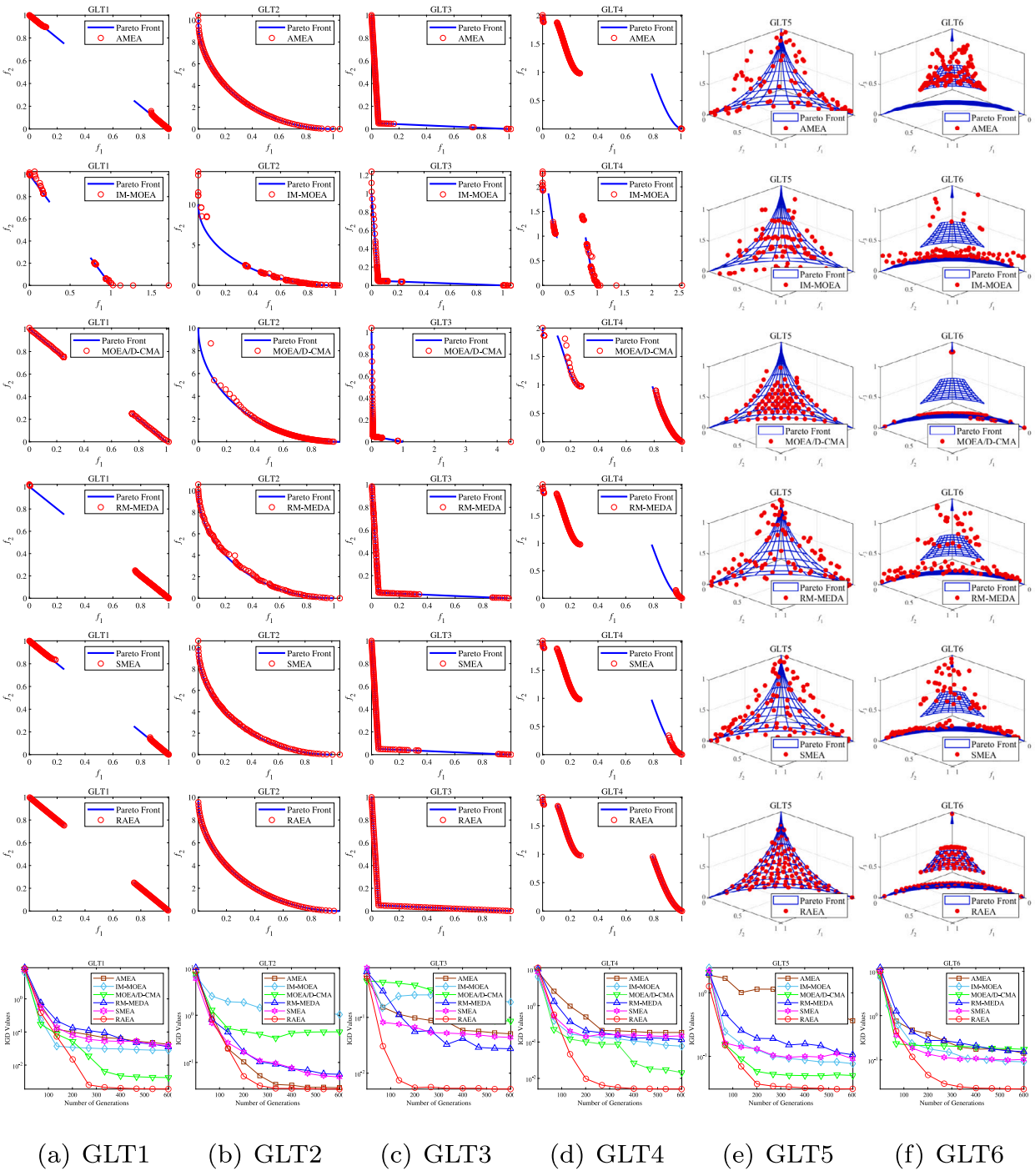


Fig. 8. The representative PFs and the evolution of IGD indicator value obtained by six compared algorithms on GLT1-GLT6 test instances associated with median IGD indicator values over 30 run.

Table 6 shows the statistics of the HV values obtained by the compared algorithms. As can be seen from the table, RAEA has significantly better HV values than AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA for 39, 36, 32, 36 and 33 out of the 42 test instances, respectively. Regarding the overall mean rankings, RAEA is best, followed by MOEA/D-CMA, SMEA, IM-MOEA, RM-MEDA and AMEA. In a word, the performance of RAEA in IGD and HV indicator is similar.

For visual observations, Fig. 8 shows the representative approximation fronts (AFs) and the evolution of IGD values obtained by the six compared algorithms, associated with median IGD values of GLT1-GLT6. RMDE performs well in convergence and diversity for GLT1-GLT6 test instances. Specifically, RAEA covers the whole PFs better for each instance than the other five compared algorithms. Furthermore, RAEA could reach the lowest mean IGD indicator values for GLT1-GLT6 within 600 generations, which denotes that RAEA has the fastest convergence speed for this test suite.

The scalability of RAEA to the number of decision variables is examined by varying the number of decision variables on the tri-objective WFG test instances. WFG1-WFG9 test instances with 50, 80, 100 and 200 dimensional variables are used in this work where the 50, 80 and 100-D instances will terminate after 1000 generations and 200-D instances will terminate after 2000 generations. The IGD results achieved by six compared MOEAs on 36 WFG test instances are given in Table 7, where RAEA achieves the 26 best results out of 36, followed

Table 7
Statistical results (Mean(Std. Dev.)[rank]) obtained by AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA, SMEA and RAEA over 30 independent runs on the WFG test suites with different variable dimensions in terms of the IGD indicator.

Instances	n	AMEA	IM-MOEA	MOEA/D-CMA	RM-MEDA	SMEA	RAEA
WFG1	50	1.6533e+0(4.16e−2)[4]+	1.4645e+0(1.83e−2)[1]+	1.5605e+0(3.00e−2)[2]+	2.1088e+0(5.68e−2)[6]−	1.6161e+0(2.76e−2)[3]+	1.8004e+0(1.82e−2)[5]
	80	1.6339e+0(4.04e−2)[4]+	1.4898e+0(1.60e−2)[1]+	1.5701e+0(2.60e−2)[2]+	2.1239e+0(7.81e−2)[6]−	1.6105e+0(2.45e−2)[3]+	1.7998e+0(3.06e−2)[5]
	100	1.6431e+0(4.14e−2)[4]+	1.5047e+0(1.68e−2)[1]+	1.5635e+0(3.05e−2)[2]+	2.1491e+0(7.13e−2)[6]−	1.6238e+0(3.18e−2)[3]+	1.8020e+0(2.28e−2)[5]
	200	1.5943e+0(3.21e−2)[3]+	1.4854e+0(1.60e−2)[1]+	1.5487e+0(2.32e−2)[2]+	2.1464e+0(7.17e−2)[6]−	1.6066e+0(3.30e−2)[4]+	1.7791e+0(2.39e−2)[5]
WFG2	50	4.1104e−1(1.02e−1)[6]−	2.6871e−1(7.40e−3)[2]−	2.8378e−1(1.77e−2)[3]−	3.1439e−1(1.84e−2)[5]−	3.1278e−1(1.65e−2)[4]−	2.4397e−1(5.20e−3)[1]
	80	4.4366e−1(1.11e−1)[6]−	2.8341e−1(9.28e−3)[2]−	3.0434e−1(1.50e−2)[3]−	3.9536e−1(2.56e−2)[5]−	3.5452e−1(1.60e−2)[4]−	2.4968e−1(5.65e−3)[1]
	100	4.3647e−1(7.10e−2)[5]−	2.9067e−1(9.28e−3)[2]−	3.1255e−1(1.60e−2)[3]−	4.6317e−1(4.64e−2)[6]−	3.7457e−1(1.66e−2)[4]−	2.5516e−1(7.79e−3)[1]
	200	3.9169e−1(4.98e−2)[5]−	2.9231e−1(1.29e−2)[2]−	3.4093e−1(1.50e−2)[3]−	4.9737e−1(4.15e−2)[6]−	3.8961e−1(1.35e−2)[4]−	2.5404e−1(4.95e−3)[1]
WFG3	50	3.4993e−1(3.26e−2)[5]−	2.4498e−1(1.30e−2)[2]−	2.9056e−1(2.30e−2)[3]−	5.0815e−1(2.33e−2)[6]−	3.4887e−1(3.85e−2)[4]−	1.1066e−1(1.91e−2)[1]
	80	3.7659e−1(3.12e−2)[4]−	2.6842e−1(1.47e−2)[2]−	3.3926e−1(2.47e−2)[3]−	5.8595e−1(2.36e−2)[6]−	4.1981e−1(2.43e−2)[5]−	1.4030e−1(1.33e−2)[1]
	100	3.9314e−1(4.01e−2)[4]−	2.8336e−1(1.69e−2)[2]−	3.5189e−1(1.85e−2)[3]−	6.1119e−1(2.54e−2)[6]−	4.3978e−1(2.64e−2)[5]−	1.5244e−1(1.33e−2)[1]
	200	3.6723e−1(3.29e−2)[3]−	2.8413e−1(2.11e−2)[2]−	3.9007e−1(1.20e−2)[4]−	6.4126e−1(1.79e−2)[6]−	4.6302e−1(2.30e−2)[5]−	1.6979e−1(9.35e−3)[1]
WFG4	50	3.5393e−1(2.17e−2)[4]−	3.1634e−1(9.13e−3)[2]−	3.5036e−1(7.39e−3)[3]−	4.0269e−1(1.47e−2)[6]−	3.8293e−1(1.11e−2)[5]−	3.0459e−1(7.26e−3)[1]
	80	3.5471e−1(2.43e−2)[3]−	3.1963e−1(6.75e−3)[2]−	3.6101e−1(1.19e−2)[4]−	4.6543e−1(4.31e−2)[6]−	3.8753e−1(1.14e−2)[5]−	3.1235e−1(8.93e−3)[1]
	100	3.5503e−1(2.32e−2)[3]−	3.2453e−1(9.82e−3)[2]−	3.6475e−1(1.07e−2)[4]−	5.1494e−1(5.38e−2)[6]−	3.9666e−1(1.19e−2)[5]−	3.1251e−1(9.04e−3)[1]
	200	3.5243e−1(2.26e−2)[3]−	3.2890e−1(9.48e−3)[2]−	3.6228e−1(1.08e−2)[4]−	5.4696e−1(6.23e−2)[6]−	3.9884e−1(1.15e−2)[5]−	3.1050e−1(8.50e−3)[1]
WFG5	50	3.7306e−1(4.68e−2)[5]−	3.2688e−1(9.16e−3)[4]−	3.1625e−1(4.34e−3)[3]−	3.9074e−1(2.68e−2)[6]−	2.9889e−1(1.72e−2)[2]−	2.7817e−1(6.34e−3)[1]
	80	3.5768e−1(5.76e−2)[5]−	3.3070e−1(7.84e−3)[4]−	3.1566e−1(4.96e−3)[3]−	4.3465e−1(2.18e−2)[6]−	2.9895e−1(1.93e−2)[2]−	2.8105e−1(7.44e−3)[1]
	100	3.5476e−1(5.50e−2)[5]−	3.4152e−1(1.14e−2)[4]−	3.1681e−1(3.71e−3)[3]−	4.4117e−1(2.43e−2)[6]−	3.0519e−1(1.64e−2)[2]−	2.7956e−1(6.59e−3)[1]
	200	3.1907e−1(4.02e−2)[3]−	3.3436e−1(1.07e−2)[5]−	3.1915e−1(4.27e−3)[4]−	4.6742e−1(2.99e−2)[6]−	2.9094e−1(1.86e−2)[2]−	2.8076e−1(7.09e−3)[1]
WFG6	50	4.2786e−1(4.10e−2)[6]−	3.4696e−1(1.25e−2)[4]−	2.4771e−1(1.63e−3)[1]+	3.6944e−1(3.16e−2)[5]−	3.1037e−1(2.85e−2)[3]=	3.0601e−1(1.30e−2)[2]
	80	4.3053e−1(5.79e−2)[6]−	3.5036e−1(1.00e−2)[4]−	2.5115e−1(1.81e−2)[1]+	3.5683e−1(3.73e−2)[5]−	2.9311e−1(2.35e−2)[2]=	3.0448e−1(1.18e−2)[3]
	100	3.9300e−1(3.38e−2)[6]−	3.5918e−1(1.50e−2)[4]−	2.5012e−1(1.63e−2)[1]+	3.8130e−1(4.35e−2)[5]−	2.8258e−1(3.17e−2)[2]+	3.0048e−1(1.13e−2)[3]
	200	3.5556e−1(3.08e−2)[5]−	3.4530e−1(9.88e−3)[4]−	2.4355e−1(8.59e−3)[1]+	4.1740e−1(7.02e−2)[6]−	2.7630e−1(2.44e−2)[2]+	3.0099e−1(1.02e−2)[3]
WFG7	50	4.5101e−1(5.49e−2)[6]−	3.5927e−1(1.27e−2)[3]−	3.1887e−1(9.22e−3)[2]−	4.2692e−1(1.87e−2)[4]−	4.3396e−1(1.49e−2)[5]−	2.7654e−1(5.54e−3)[1]
	80	4.5457e−1(5.06e−2)[4]−	3.6891e−1(1.08e−2)[3]−	3.3360e−1(1.15e−2)[2]−	4.7740e−1(2.07e−2)[6]−	4.6243e−1(1.49e−2)[5]−	2.7953e−1(6.07e−3)[1]
	100	4.5504e−1(5.38e−2)[4]−	3.8214e−1(1.73e−2)[3]−	3.4723e−1(1.46e−2)[2]−	5.0766e−1(2.01e−2)[6]−	4.6853e−1(1.42e−2)[5]−	2.8460e−1(5.22e−3)[1]
	200	4.6246e−1(5.21e−2)[4]−	3.8295e−1(1.42e−2)[3]−	3.6672e−1(9.16e−3)[2]−	5.3541e−1(2.93e−2)[6]−	4.7754e−1(1.65e−2)[5]−	2.9062e−1(5.50e−3)[1]
WFG8	50	5.5087e−1(5.77e−2)[6]−	3.5211e−1(1.05e−2)[2]−	4.2527e−1(1.19e−2)[3]−	5.1994e−1(1.97e−2)[5]−	5.1846e−1(1.79e−2)[4]−	3.2093e−1(8.46e−3)[1]
	80	4.8561e−1(5.40e−2)[4]−	3.4544e−1(7.95e−3)[2]=	4.4934e−1(1.02e−2)[3]−	5.5763e−1(2.81e−2)[6]−	5.3003e−1(1.45e−2)[5]−	3.4288e−1(8.73e−3)[1]
	100	4.5568e−1(6.31e−2)[3]−	3.4525e−1(8.41e−3)[1]+	4.5751e−1(1.34e−2)[4]−	5.7758e−1(2.89e−2)[6]−	5.4060e−1(1.58e−2)[5]−	3.5402e−1(8.51e−3)[2]
	200	3.8762e−1(5.85e−2)[3]=	3.3831e−1(6.33e−3)[1]+	4.4260e−1(2.06e−2)[4]−	5.8245e−1(2.91e−2)[6]−	5.3621e−1(1.68e−2)[5]−	3.6378e−1(7.17e−3)[2]
WFG9	50	3.4246e−1(2.94e−2)[4]−	3.5133e−1(1.48e−2)[5]−	3.1067e−1(4.41e−3)[2]−	4.7677e−1(2.90e−2)[6]−	3.4084e−1(2.31e−2)[3]−	2.6981e−1(8.56e−3)[1]
	80	3.1714e−1(2.77e−2)[3]−	3.5287e−1(1.29e−2)[4]−	3.0686e−1(7.99e−3)[2]−	5.9833e−1(4.48e−2)[6]−	3.5690e−1(3.37e−2)[5]−	2.6511e−1(1.21e−2)[1]
	100	3.1762e−1(2.76e−2)[3]−	3.5747e−1(1.45e−2)[4]−	3.0648e−1(9.93e−3)[2]−	6.7060e−1(4.79e−2)[6]−	3.6216e−1(2.93e−2)[5]−	2.6323e−1(1.32e−2)[1]
	200	2.8671e−1(2.53e−2)[2]−	3.6211e−1(1.56e−2)[5]−	3.0536e−1(9.46e−3)[3]−	7.5066e−1(7.17e−2)[6]−	3.6055e−1(2.60e−2)[4]−	2.5729e−1(1.58e−2)[1]
Ranking		4.2500	2.6944	2.6667	5.7778	3.9167	1.6944
+/−/=		4/31/1	6/29/1	8/28/0	0/36/0	6/28/2	

by IM-MOEA and MOEA/D-CMA with 6 and 4 best results, respectively. Considering the shapes of the WFG1 test instance, RAEA may perform unsatisfactorily on WFG1 due to the irregularity of PF.

In summary, we may conclude that RAEA performs better than the compared algorithms on the five test suites in terms of both IGD and HV indicator values. This suggests that RAEA is very competitive with these state-of-the-art MOEAs.

5. Conclusions

In this work, we propose a regularity augmented evolutionary algorithm, RAEA, for continuous multiobjective optimization. In RAEA, at each generation, we aim to accelerate the convergence of population by incrementally projecting these dominated solutions into the latent space built by the SVD method on these nondominated solutions. New trial solutions are generated by a newly developed dual-space search strategy (called DSS) that a DE operator is adopted to search in both the latent and decision space for offspring generation.

We have conducted a series of algorithm analyses, parameter settings and ablation studies, to examine the proposed algorithm. The research indicates that the proposed acceleration mechanism and DSS strategy can improve the performance of RAEA in continuous multiobjective optimization. Comparison experiments with five well-known MOEAs, including AMEA, IM-MOEA, MOEA/D-CMA, RM-MEDA and SMEA, were conducted on 42 instances with complex PSs and PFs to verify the performance of RAEA. Experimental results suggested that RAEA significantly outperforms the compared algorithms in convergence and diversity measured by IGD and HV indicators.

In the future, we intend to extend the proposed RAEA or its basic idea to deal with different kinds of MOPs:

1. Embedding the proposed dual-space search strategy in some representative MOEAs, i.e., 2REA [62], I_{SDE+} [63], MOEA/D-AWA [64] and CA-MOEA [65], to deal with MOPs with many-objective or irregular PFs;
2. Extending RAEA to attack large-scale MOPs. The scalability of RAEA on the WFG test suite indicates that RAEA has potential in dealing with large-scale MOPs by combining some large-scale optimization techniques [66].
3. As aforementioned, the degenerated MOPs may not meet the regularity property [30]. Moreover, the constrained MOPs with equality constraints may reduce the dimension of the manifold of Pareto optimal solutions [31]. Therefore, the effect of the proposed RAEA needs to be further investigated in these cases.

CRediT authorship contribution statement

Shuai Wang: Writing – original draft, Methodology, Software. **Bingdong Li:** Methodology, Software, Visualization. **Aimin Zhou:** Writing – review & editing, Methodology, Software.

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 Scientific and Technological Innovation 2030 Major Projects under Grant No. 2018AAA0100902, the Science and Technology Commission of Shanghai Municipality, China under Grant No. 19511120601, and the Fundamental Research Funds for the Central Universities, China.

References

[1] K. Miettinen, Nonlinear Multiobjective Optimization, Vol. 12, Springer Science & Business Media, 2012.
[2] K. Deb, Multi-objective optimisation using evolutionary algorithms: An introduction, in: Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing, Springer, 2011, pp. 3–34.
[3] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm Evol. Comput. 1 (1) (2011) 32–49.
[4] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63.
[5] R. Cheng, Y. Jin, K. Narukawa, B. Sendhoff, A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling, IEEE Trans. Evol. Comput. 19 (6) (2015) 838–856.
[6] J. Sun, H. Zhang, A. Zhou, Q. Zhang, K. Zhang, A new learning-based adaptive multi-objective evolutionary algorithm, Swarm Evol. Comput. 44 (2019) 304–319.
[7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[8] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, TIK-Rep. 103 (2001).
[9] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842.
[10] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European J. Oper. Res. 181 (3) (2007) 1653–1669.
[11] J.G. Falcón-Cardona, C.A.C. Coello, Indicator-based multi-objective evolutionary algorithms: A comprehensive survey, ACM Comput. Surv. 53 (2) (2020) 1–35.
[12] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[13] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2008) 284–302.
[14] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Trans. Evol. Comput. 21 (3) (2016) 440–462.
[15] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2010) 4–31.
[16] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution: An updated survey, Swarm Evol. Comput. 27 (2016) 1–30.
[17] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
[18] C.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, IEEE, 2002, pp. 1051–1056.
[19] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, vol. 2, Springer Science & Business Media, 2001.
[20] H. Zhang, A. Zhou, S. Song, Q. Zhang, X.-Z. Gao, J. Zhang, A self-organizing multiobjective evolutionary algorithm, IEEE Trans. Evol. Comput. 20 (5) (2016) 792–806.
[21] T. Liu, X. Li, L. Tan, S. Song, A novel adaptive greedy strategy based on Gaussian mixture clustering for multiobjective optimization, Swarm Evol. Comput. 61 (2021) 100815.
[22] C. Hillermeier, et al., Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach, Vol. 135, Springer Science & Business Media, 2001.
[23] Y. Jin, B. Sendhoff, Connectedness, regularity and the success of local search in evolutionary multi-objective optimization, in: The 2003 Congress on Evolutionary Computation, 2003, Vol. 3, CEC'03, IEEE, 2003, pp. 1910–1917.
[24] A. Zhou, Estimation of distribution algorithms for continuous multiobjective optimization (Ph.D. thesis), University of Essex, 2009.
[25] Y. Wang, J. Xiang, Z. Cai, A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator, Appl. Soft Comput. 12 (11) (2012) 3526–3538.
[26] H. Wang, Q. Zhang, L. Jiao, X. Yao, Regularity model for noisy multiobjective optimization, IEEE Trans. Cybern. 46 (9) (2015) 1997–2009.
[27] Y. Sun, G.G. Yen, Z. Yi, Improved regularity model-based EDA for many-objective optimization, IEEE Trans. Evol. Comput. 22 (5) (2018) 662–678.
[28] B. Dong, A. Zhou, G. Zhang, Sampling in latent space for a mulitiobjective estimation of distribution algorithm, in: 2016 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2016, pp. 3027–3034.
[29] R. Wang, N.-J. Dong, D.-W. Gong, Z.-B. Zhou, S. Cheng, G.-H. Wu, L. Wang, PCA-assisted reproduction for continuous multi-objective optimization with complicated Pareto optimal set, Swarm Evol. Comput. 60 (2021) 100795.
[30] L. Zhen, M. Li, R. Cheng, D. Peng, X. Yao, Multiobjective test problems with degenerate Pareto fronts, 2018, arXiv preprint arXiv:1806.02706.
[31] O. Cuate, L. Uribe, A. Lara, O. Schütze, A benchmark for equality constrained multi-objective optimization, Swarm Evol. Comput. 52 (2020) 100619.
[32] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[33] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[34] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.
[35] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[36] X. Li, H. Zhang, S. Song, A self-adaptive mating restriction strategy based on survival length for evolutionary multiobjective optimization, Swarm Evol. Comput. 43 (2018) 31–49.
[37] X. Li, H. Zhang, S. Song, MOEA/D with the online agglomerative clustering based self-adaptive mating restriction strategy, Neurocomputing 339 (2019) 77–93.
[38] J. Sun, H. Zhang, A. Zhou, Q. Zhang, K. Zhang, Z. Tu, K. Ye, Learning from a stream of nonstationary and dependent data in multiobjective evolutionary optimization, IEEE Trans. Evol. Comput. 23 (4) (2018) 541–555.
[39] T. Liu, X. Li, L. Tan, S. Song, An incremental-learning model-based multiobjective estimation of distribution algorithm, Inform. Sci. 569 (2021) 430–449.
[40] L. Mo, G. Dai, J. Zhu, The RM-MEDA based on elitist strategy, in: International Symposium on Intelligence Computation and Applications, Springer, 2010, pp. 229–239.
[41] J.Y. Tey, R. Ramli, A.S. Abdullah, A new multi-objective optimization method for full-vehicle suspension systems, Proc. Inst. Mech. Eng. D 230 (11) (2016) 1443–1458.
[42] C. He, S. Huang, R. Cheng, K.C. Tan, Y. Jin, Evolutionary multiobjective optimization driven by generative adversarial networks (GANs), IEEE Trans. Cybern. 51 (6) (2020) 3129–3142.
[43] Z. Wang, H. Hong, K. Ye, G.-E. Zhang, M. Jiang, K.C. Tan, Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks, IEEE Trans. Neural Netw. Learn. Syst. (2021).
[44] C. Munteanu, V. Lazarescu, Improving mutation capabilities in a real-coded genetic algorithm, in: Workshops on Applications of Evolutionary Computation, Springer, 1999, pp. 138–149.
[45] Y.-l. Li, J. Zhang, W.-n. Chen, Differential evolution algorithm with PCA-based crossover, in: Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, 2012, pp. 1509–1510.
[46] P. Pošík, Using kernel principal component analysis in evolutionary algorithms as an efficient multi-parent crossover operator, None,
[47] W. Huang, T. Xu, K. Li, J. He, Multiobjective differential evolution enhanced with principle component analysis for constrained optimization, Swarm Evol. Comput. 50 (2019) 100571.
[48] G.H. Golub, C. Reinsch, Singular value decomposition and least squares solutions, in: Linear Algebra, Springer, 1971, pp. 134–151.
[49] V. Klema, A. Laub, The singular value decomposition: Its computation and some applications, IEEE Trans. Automat. Control 25 (2) (1980) 164–176.
[50] C.F. Van Loan, Generalizing the singular value decomposition, SIAM J. Numer. Anal. 13 (1) (1976) 76–83.
[51] J. Mandel, Use of the singular value decomposition in regression analysis, Amer. Statist. 36 (1) (1982) 15–24.
[52] M.E. Wall, A. Rechtsteiner, L.M. Rocha, Singular value decomposition and principal component analysis, in: A Practical Approach to Microarray Data Analysis, Springer, 2003, pp. 91–109.
[53] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407.
[54] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of Dimensionality Reduction in Recommender System-A Case Study, Tech. Rep., Minnesota Univ Minneapolis Dept of Computer Science, 2000.
[55] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Incremental singular value decomposition algorithms for highly scalable recommender systems, in: Fifth International Conference on Computer and Information Science, Vol. 1, Citeseer, 2002, pp. 27–28.
[56] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, SIAM Rev. 37 (4) (1995) 573–595.
[57] H. Li, Q. Zhang, J. Deng, Biased multiobjective optimization and decomposition algorithm, IEEE Trans. Cybern. 47 (1) (2016) 52–66.
[58] F. Gu, H.-L. Liu, K.C. Tan, A multiobjective evolutionary algorithm using dynamic weight design method, Int. J. Innovative Comput. Inf. Control 8 (5 (B)) (2012) 3677–3688.

[59] Y. Tian, R. Cheng, X. Zhang, M. Li, Y. Jin, Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier], IEEE Comput. Intell. Mag. 14 (3) (2019) 61–74.
[60] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2005, pp. 280–295.
[61] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.
[62] Z. Liang, K. Hu, X. Ma, Z. Zhu, A many-objective evolutionary algorithm based on a two-round selection strategy, IEEE Trans. Cybern. 51 (3) (2019) 1417–1429.
[63] T. Pamulapati, R. Mallipeddi, P.N. Suganthan, I_{SDE+}—An indicator for multi and many-objective optimization, IEEE Trans. Evol. Comput. 23 (2) (2018) 346–352.
[64] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264.
[65] Y. Hua, Y. Jin, K. Hao, A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts, IEEE Trans. Cybern. 49 (7) (2018) 2758–2770.
[66] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K.C. Tan, Y. Jin, Evolutionary large-scale multi-objective optimization: A survey, ACM Comput. Surv. 54 (8) (2021) 1–34.