



Surrogate-assisted evolutionary algorithm with decomposition-based local learning for high-dimensional multi-objective optimization

Jiangtao Shen, Peng Wang*, Huachao Dong, Wenxin Wang, Jinglu Li

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, 710072, China



ARTICLE INFO

Keywords:

Surrogate-assisted evolutionary algorithm
Expensive high-dimensional multi-objective optimization
Decomposition-based approach
Multi-objective shape design

ABSTRACT

When the evolutionary algorithm is applied to handle high-dimensional expensive multi-objective optimization problems (MOPs), population evolution is crucial since it controls exploration and exploitation and decides if promising candidate solutions could be generated. However, little attention has been paid to this issue, evolution operators based on Genetic Algorithm (GA) and Differential Evolution (DE) are still the two most common approaches, whose convergence on high-dimensional MOPs with a limited number of fitness evaluations remains challenging. In this paper, we propose a decomposition-based local learning strategy to accelerate convergence in the high-dimensional search space of MOPs. Specifically, an individual is updated by learning from one of the best solutions of its corresponding local area based on the multi-objective decomposition approach. Accordingly, a surrogate-assisted evolutionary algorithm is proposed for better solving expensive high-dimensional MOPs. Experimental studies on MOPs with up to 100 decision variables and with 300 fitness evaluations demonstrate the effectiveness of the proposed method. Furthermore, we use the proposed method to solve a 2-objective shape design problem of the blended wing-body underwater glider (BWBUG) with 39 decision variables, and an impressive solution set is obtained.

1. Introduction

Numerous real-world optimization problems involve expensive multiple objectives, which are termed expensive multiobjective optimization problems (EMOPs) (Sonoda & Nakata, 2022). Since it is computationally expensive for fitness evaluations (FEs) of EMOPs, only a small number of FEs are allowed during the optimization process. For example, in the case of multiobjective airfoil design (Pan et al., 2018), the lift and drag of a solution require to solve Navier-Stokes equations by the technique of Computational Fluid Dynamics (CFD) (Arias-Montano, Coello, & Mezura-Montes, 2012) simulations, which is very time-consuming. Multiobjective evolutionary algorithms (MOEAs) are effective for solving multiobjective optimization problems (MOPs) with a large number of allowable FEs (Chugh, Jin, Miettinen, Hakkanen, & Sindhy, 2018), which are not suitable for EMOPs. Surrogate-assisted evolutionary algorithms (SAEAs) show effectiveness in reducing the number of FEs when handling EMOPs. Generally, SAEAs filter out promising solutions to be evaluated by estimating the quality of candidate solutions using cheap surrogate models (Wang, Wang, Song, Wang, & Wang, 2019). And the surrogate models are updated gradually by adding the newly evaluated solutions iteratively. In this paper, SAEAs

only represent multi-objective SAEAs (do not include single-objective SAEAs) if not stated differently.

To handle high-dimensional or large-scale MOPs (LMOPs) with cheap fitness evaluations, numerous methods have been proposed. This kind of approach is mainly based on the evolutionary search strategy since a large number of fitness evaluations are available. In the accelerated large-scale MOEA (ALMOEA) framework (Liu, Li, Lin, Tian, & Tan, 2022), FNN is applied to guide the search direction and reproduce new solutions efficiently. The discriminative reconstruction neural networks-based EA (DRNEA) (Liu, Lin, Feng, Wong, & Tan, 2022) uses the evolutionary multitasking framework to solve multiple LMOPs simultaneously. In the variable importance-based DE (VIDE) (Liu, Lin, Tian, & Tan, 2021), a DE algorithm that favors searching the variables with higher importance is proposed. In the large-scale competitive swarm optimizer (LMCSO) (Tian, Zheng, Zhang, & Jin, 2019), competitive swarm optimizer is applied as the evolution operator for solving LMOPs. The comprehensive competitive swarm optimizer (CCSO) (Liu, Lin, Li, & Tan, 2021) enhances the search capability of loser particles and provides diverse search directions for solving LMOPs, which performs better than LMCSO.

* Corresponding author.

E-mail addresses: shen_jiang_tao@163.com (J. Shen), wangpeng305@nwpu.edu.cn (P. Wang), hdong@nwpu.edu.cn (H. Dong), wxwang@mail.nwpu.edu.cn (W. Wang), lijinglu@mail.nwpu.edu.cn (J. Li).

Over the past two decades, many SAEAs have been proposed for solving EMOPs and expensive many-objective optimization problems (EMaOPs), such as the Pareto-based Efficient Global Optimization (EGO) (ParEGO) (Knowles, 2006), the *S*-metric based EGO (SMS-EGO) (Ponweiser, Wagner, Biermann, & Vincze, 2008), the classification-based pre-selection MOEA (CPS-MOEA) (Zhang, Zhou, & Zhang, 2015), the EGO-assisted decomposition-based MOEA (MOEA/D-EGO) (Zhang, Liu, Tsang, & Virginas, 2009), the Kriging-assisted reference vector-guided EA (RVEA) (Cheng, Jin, Olhofer, & Sendhoff, 2016) (K-RVEA) (Chugh et al., 2018), the classification-based surrogate-assisted EA (CSEA) (Pan et al., 2018), the hybrid surrogate-assisted MOEA (HSMEA) (Habib, Singh, Chugh, Ray, & Miettinen, 2019), and the Kriging-assisted two-archive EA (KTA2) (Song, Wang, He, & Jin, 2021), etc. However, most of the aforementioned methods are proposed to solve EMOPs with less than 30 decision variables. For EMOPs with more than 30 decision variables, termed high-dimensional EMOPs, these strategies are not working well due to two main reasons as follows. First, building accurate surrogate models for high-dimensional problems with a limited number of samples is a great challenge (Sonoda & Nakata, 2022; Wang et al., 2019). Second, the population in high-dimensional space is updated inefficiently, which leads to slow convergence and unsatisfying solutions within a small number of FEs. Actually, it is not uncommon that EMOPs also have high-dimensional decision variables. Let us consider the previous case of multiobjective airfoil design. Since the performance of an airfoil is very sensitive to the shape itself, a high-precision airfoil must be modeled with a great number of decision variables (Arias-Montano et al., 2012), which results in a high-dimensional EMOP. Therefore, it is of great significance to enhance the scalability of SAEAs to better solve high-dimensional EMOPs.

For high-dimensional EMOPs, an intuitive idea is to reduce the number of decision variables of the original problem by dimension reduction techniques such as the Principal Component Analysis (PCA) method (Moore, 1981), so that the model accuracy could be improved and the transformed problem is easier to handle. In the surrogate-assisted RVEA-based PCA (SA-RVEA-PCA) (Zhao et al., 2020), the Gaussian processes (GPs) model (Zhang et al., 2009) based on PCA is constructed for each objective function and RVEA is applied as an multiobjective optimizer. SA-RVEA-PCA demonstrates its effectiveness on EMOPs with up to 160 decision variables and 2 objectives. In the surrogate-assisted particle swarm optimization algorithm based on the adaptive dropout mechanism (ADSAPSO) (Lin, He, & Cheng, 2021), the statistical differences between different solution sets in the decision space are utilized to guide the selection of partial decision variables. Correspondingly, an infill criterion is proposed to optimize the selected decision variables based on surrogate models. ADSAPSO shows good performance on benchmark problems with up to 200 decision variables and 3 objectives.

Furthermore, some efforts to improve the accuracy of surrogate models have been made, including the application of surrogate ensemble or classifiers, and the management of these surrogates. The heterogeneous ensemble-based MOEA (He-EMOEA) (Guo, Jin, Ding, & Chai, 2019) uses the heterogeneous ensemble consisting of a least square support vector machine and two radial basis function networks to enhance the reliability of ensembles for uncertainty estimation. In addition, two sets of transformed variables that vary from the original decision variables are applied to promote the diversity of the ensemble. HeE-MOEA performs well on EMOPs with up to 80 decision variables and 3 objectives. The RBFs-assisted (radial basis functions-assisted) RVEA (R-RVEA) (Shen, Wang, & Wang, 2020) applies two kinds of RBFs to provide estimated errors and select a more suitable RBF for each objective function to improve the approximation accuracy. R-RVEA performs better than K-RVEA on EMOPs with 30 decision variables. In the efficient dropout neural network-assisted MOEA with reference point adaptation (EDN-ARMOEA) (Guo et al., 2022), an efficient neural network is proposed as a computationally scalable alternative

of the GPs model for better solving high-dimensional EMOPs. EDN-ARMOEA achieves good performance and computational efficiency on EMOPs with up to 100 decision variables and 20 objectives. In the relation-based evolutionary multiobjective optimization algorithm (REMO) (Hao, Zhou, Qian, & Zhang, 2022), a three-class feedforward neural network (FNN) (Svozil, Kvasnicka, & Pospichal, 1997) is trained to learn the dominance relationship of solution pairs based on the two subpopulations partitioned by the penalty-based boundary intersection (PBI) (Zhang & Li, 2007) method. REMO outperforms K-RVEA and KTA2 on benchmark MOPs with up to 50 decision variables and 10 objectives. Recently, a multiple classifiers-assisted evolutionary algorithm based on decomposition (MCEA/D) (Sonoda & Nakata, 2022) is proposed to solve high-dimensional EMOPs. MCEA/D constructs multiple local Support Vector Machine (SVM) classifiers, which show robustness on high-dimensional classification even with few training samples (Vapnik, 1999). MCEA/D shows better performance and higher efficiency on benchmark EMOPs with up to 150 decision variables and 11 objectives when compared with EDN-ARMOEA.

On the one hand, although large-scale MOEAs show good performance on MOPs with high-dimensional decision space, they consume a large number of FEs, which is prohibitive when time-consuming fitness evaluations are involved. On the other hand, to solve high-dimensional EMOPS, improving surrogate models for better accuracy and efficiency is common. Despite the significance of efficient searching on high-dimensional decision space, rare works have been done on search strategy. In this paper, different from the existing studies, we try to promote the performance of SAEAs on high-dimensional EMOPs from the perspective of population evolution. Fig. 1 presents the IGD (iterative inverse generation distance) (Coello & Cortés, 2005) values of EDN-ARMOEA and its variant EDN-ARMOEA* on 10D-3M-DTLZ1-4 (denotes DTLZ1-4 with 10 decision variables and 3 objectives) and 50D-3M-DTLZ1-4 (Deb, Thiele, Laumanns, & Zitzler, 2005), respectively. Differential Evolution (DE) (Das & Suganthan, 2011) is employed in EDN-ARMOEA* instead of Genetic Algorithm (GA) (Deb & Agrawal, 1995) in EDN-ARMOEA, the maximum number of FEs is 300 and other parameter settings are the same as that in Section 4. From the figures, three observations can be made. First, DE performs better than GA on the two multimodal MOPs (DTLZ1 and DTLZ3). Second, DE shows similar performance with GA on 10D-DTLZ2 and 10D-DTLZ4 but performs much better than GA on 50D-DTLZ2 and 50D-DTLZ4. Lastly, with the growth of problems in dimensionality, the performance gap between GA and DE is clearly expanding. The above instance demonstrates that evolution of the population is one of the crucial points for solving high-dimensional EMOPs since it controls exploration degree and determines if promising candidate solutions could be generated. However, scarce attention has been paid to this issue, evolution operators based on GA and DE (Li, Wang, Dong, & Shen, 2022) are still the most common approaches in this area, whose convergence on high-dimensional MOPs with a limited number of FEs remains challenging.

Following the above idea, this paper proposes a decomposition-based local learning strategy (DBLL) to accelerate the convergence of population in high-dimensional search space, where the individual is updated by learning from one of the best solutions of its corresponding local area based on the multiobjective decomposition approach. Correspondingly, an SAEA based on DBLL, termed SAEA-DBLL, is proposed in this paper for better solving high-dimensional EMOPs. In the proposed method, radial basis functions (RBFs) (Forrester & Keane, 2009) and RVEA (Cheng et al., 2016) are employed as the surrogate models and multiobjective optimizer, respectively. The main contributions of this paper are as follows.

- (1) To better explore and exploit the high-dimensional decision space, a novel evolution strategy based on decomposition of the objective space is proposed. Specifically, an individual is updated by learning from the best solution of its corresponding local area defined by the angles between solutions and reference vectors.

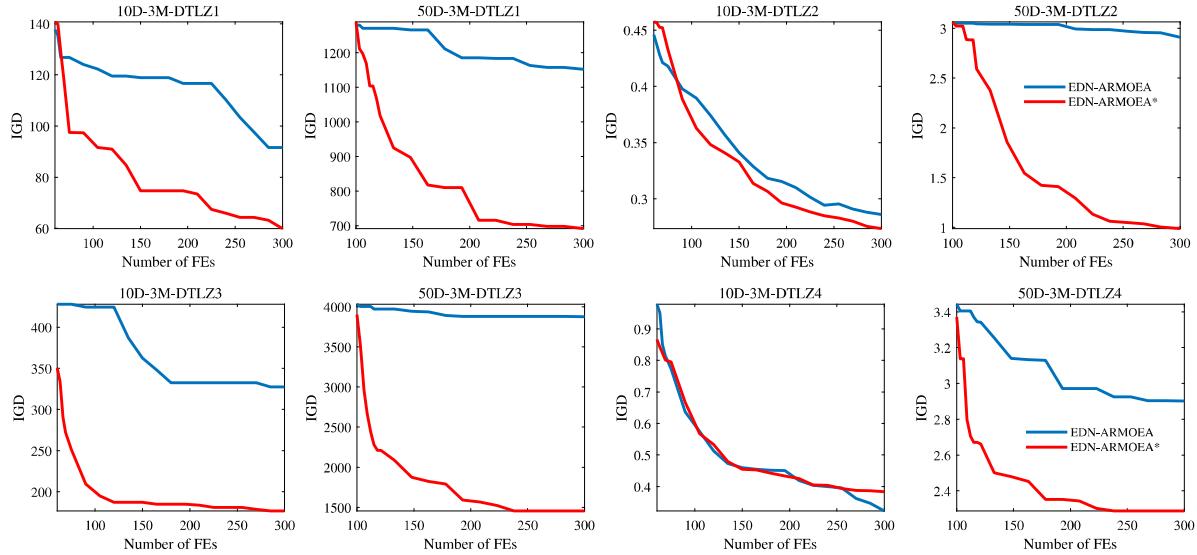


Fig. 1. Iterative IGD values of EDN-ARMOEA and EDN-ARMOEA* on 10D-3M-DTLZ1-4 and 50D-3M-DTLZ1-4, respectively.

Different from the existing methods based on dimension reduction or surrogate management, we improve the performance of SAEAs on high-dimensional EMOPs from the perspective of population evolution.

- (2) Based on the proposed evolution strategy, a surrogate-assisted evolutionary algorithm called SAEA-DBLL is presented to handle high-dimensional MOPs within a few hundred times of real FEs. SAEA-DBLL is compared with state-of-the-art SAEAs on MOPs with up to 100 decision variables and 10 objectives. Experimental results demonstrate the effectiveness of the proposed algorithm and its evolution strategy.
- (3) The proposed SAEA-DBLL is applied to handle a shape design problem of the blended-wing-body underwater glider (BWBBUG) (Li, Wang, & Gong, 2022), which is an EMOP that involves computationally expensive CFD simulations with 39 decision variables and 2 objectives. Results show that the proposed method can provide high-quality trade-off solutions with higher convergence speed.

The remainder of this paper is organized as follows. Section 2 provides the background of the study, including a definition of MOPs and introductions to RVEA and RBFs. The proposed SAEA-DBLL is detailed in Section 3. Section 4 presents the experimental studies on benchmark problems. And SAEA-DBLL is applied to the shape optimization of BWBUG in Section 5. Finally, conclusions and future works are given in Section 6.

2. Background

In this section, we first give a definition of MOP. RVEA, as one of the representatives of decomposition-based methods, is applied in this study as the multiobjective optimizer. The main components of RVEA is summarized in the second part. Finally, RBF is briefly introduced, which is the most popular surrogate model for high-dimensional expensive optimization (Wang et al., 2019).

2.1. Multi-objective optimization problem

Generally, an unconstraint MOP can be briefly stated as

$$\begin{aligned} \min \mathbf{f}(\mathbf{x}) = & (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{s.t. } \mathbf{x} \in & X \end{aligned} \quad (1)$$

where M denotes the number of objectives, $\mathbf{x} = (x_1, x_2, \dots, x_D)$ is the decision vector with D decision variables, and $X \subseteq \mathbb{R}^D$ is the decision space. Different from single-objective optimization, the optimum of an MOP is a set of trade-off solutions since the conflicting nature of objectives. It is termed Pareto optimal set (PS) in the decision space corresponding to Pareto optimal front (PF) in the objective space (Shen, Dong, Wang, Li, & Wang, 2023). For multiobjective optimization, Pareto dominance (Deb, Pratap, Agarwal, & Meyarivan, 2002) is the most common approach to distinguish the quality of candidate solutions. A solution \mathbf{x} is said to dominate a solution \mathbf{y} , denoted as $\mathbf{x} \prec \mathbf{y}$, if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for every $i \in \{1, 2, \dots, M\}$, and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ at least one index $j \in \{1, 2, \dots, M\}$.

2.2. Reference vector guided evolutionary algorithm

Algorithm 1 RVEA

Require: t_{max} , maximum number of generations; NV , number of reference vectors; $V_0 = \{v_{0,1}, v_{0,2}, \dots, v_{0,NV}\}$, a set of reference vector;

Ensure: $P_{t_{max}}$, nondominated solutions from the final population;

- 1: $P_0 \leftarrow$ Initialize the population with NV individual randomly;
- 2: $t \leftarrow 1$;
- 3: **while** $t \leq t_{max}$ **do**
- 4: $Q_t \leftarrow$ Generate offspring (P_t);
- 5: $P_t \leftarrow Q_t \cup P_t$;
- 6: $P_{t+1} \leftarrow$ Environmental selection based on reference vectors (t, P_t, V_t);
- 7: $V_{t+1} \leftarrow$ Adaptation of reference vectors (t, P_{t+1}, V_t);
- 8: $t \leftarrow t + 1$;
- 9: **end while**

The pseudocode of RVEA is presented in **Algorithm 1**, whose framework is similar to MOEAs that adopt the elitism strategy, such as the elitist nondominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002). RVEA distinguishes itself from NSGA-II in its environmental selection, which is based on a set of uniformly distributed reference vectors that considers both convergence and diversity (Cheng et al., 2016). The environmental selection can be divided into the following four steps.

(1) *Translation of objective value*: Since the initial point of the reference vectors is always the coordinate origin, objective values of the current population P_t should be translated by

$$\mathbf{f}'_{t,i} = \mathbf{f}_{t,i} - \mathbf{z}_t^{min} \quad (2)$$

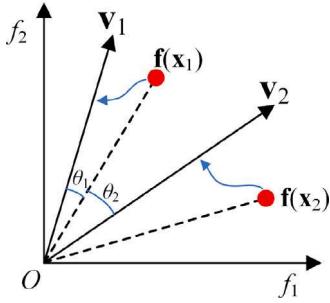


Fig. 2. An Illustration of assigning candidate solutions to their closest reference vectors according to angles.

where $i \in \{1, 2, \dots, |P_t|\}$, $\mathbf{f}_{t,i}$ denotes the objective vector of the i th individual at the t th generation, $\mathbf{f}'_{t,i}$ is the translated objective vector, $\mathbf{z}_t^{\min} = (z_{t,1}^{\min}, z_{t,2}^{\min}, \dots, z_{t,M}^{\min})$ consists of the M minimum objective values of P_t . Through the translation operation, all the translated objective values are inside the first quadrant, which is the same as the reference vectors.

(2) *Assignment of individuals to reference vectors*: In RVEA, a set of uniformly distributed reference vectors is generated using the approach introduced in Cornell (1843), which is applied to assist the optimization process. After the translation operation, the population is partitioned into N subpopulations by assigning each individual to its closest reference vector. Fig. 2 illustrates this process by a 2-objective instance with two reference vectors v_1, v_2 and two individuals x_1, x_2 . In this example, x_1 is assigned to v_1 since the angle θ_1 between $\mathbf{f}'(x_1)$ and v_1 is less than the angle θ_2 between $\mathbf{f}'(x_1)$ and v_2 . Similarly, x_2 is assigned to v_2 .

(3) *Calculation of angle-penalized distance (APD)*: After assigning individuals to reference vectors, the population is partitioned into multiple subpopulations corresponding to the reference vectors. Significantly, the number of subpopulations is no more than NV since there may be vectors that no individual is assigned to, termed no-active vectors or empty vectors. The APD considers both convergence and diversity by the following strategy. First, the convergence of \mathbf{x} can be represented by the distance from $\mathbf{f}'(\mathbf{x})$ to the ideal point. Second, the diversity of \mathbf{x} can be represented by the acute angle between $\mathbf{f}'(\mathbf{x})$ and the reference vector that \mathbf{x} is assigned. APD is formulated as

$$d_i = (1 + P(\theta_i)) \cdot \|\mathbf{f}'(\mathbf{x}_i)\| \quad (3)$$

where $\|\mathbf{f}'(\mathbf{x}_i)\|$ is distance from the translated objective values of the i th individual to the ideal point, θ_i represents the acute angle between the i th individual and its corresponding reference vector. $P(\theta_i)$ is the penalty function that can be formulated as

$$P(\theta_i) = M \cdot \left(\frac{1}{t_{\max}} \right)^{\alpha} \cdot \left(\frac{\theta_i}{\gamma_v} \right) \quad (4)$$

where M is the number of objectives, t_{\max} represents the maximal number of generations, γ_v is the smallest angle between the corresponding reference vector of the i th individual and the other reference vectors. α is a parameter to change the rate of $P(\theta_i)$, which can be applied to stress convergence in the early stage and diversity in the later stage during the entire search process. After calculating APD of all individuals, one individual of its corresponding subpopulation that has the minimum APD value is selected for the next generation.

(4) *Adaptation of reference vectors*: In order to handle MOPs with different scales of objectives, reference vectors are adapted according to the ranges of objective values of the current population. Reference vectors of the next generation $\mathbf{v}_{t+1,i}$ can be updated by

$$\mathbf{v}_{t+1,i} = \frac{\mathbf{v}_{0,i} \circ (\mathbf{z}_t^{\max} - \mathbf{z}_t^{\min})}{\|\mathbf{v}_{0,i} \circ (\mathbf{z}_t^{\max} - \mathbf{z}_t^{\min})\|} \quad (5)$$

where $i \in \{1, 2, \dots, NV\}$, \circ denotes the Hadamard product (Cheng et al., 2016), $\mathbf{v}_{t+1,i}$ is the i th reference vector of the $(t+1)$ th generation, $\mathbf{v}_{0,i}$ is the i th initial reference vector, \mathbf{z}_t^{\max} and \mathbf{z}_t^{\min} is the maximum and minimum objective values of the t th generation, respectively. Moreover, many efforts have been made to adapt the reference vectors (Jain & Deb, 2013; Liu, Lin, Tan, Gong, & Coello, 2020; Liu, Lin, Wong, et al., 2020) for better performance on MOPs with various PFs.

2.3. Radial basis function

In this paper, RBF (Forrester & Keane, 2009) is employed as the surrogate model, which is one of the most popular surrogate models with high efficiency on high-dimensional problems (Li, Zhang, Lin, & Gao, 2021). For an objective function $f(\mathbf{x})$, given NT training points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{NT}]^T$ in the decision space, and their responses $\mathbf{F} = [f_1, f_2, \dots, f_{NT}]^T$, the $f(\mathbf{x})$ can be approximated by RBF as

$$\hat{f}(\mathbf{x}) = \mathbf{r}^T(\mathbf{x}) \mathbf{R}^{-1} \mathbf{F} \quad (6)$$

where

$$\mathbf{R} = \begin{bmatrix} R(\|\mathbf{x}_1 - \mathbf{x}_1\|) & R(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & R(\|\mathbf{x}_1 - \mathbf{x}_{NT}\|) \\ R(\|\mathbf{x}_2 - \mathbf{x}_1\|) & R(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & R(\|\mathbf{x}_2 - \mathbf{x}_{NT}\|) \\ \vdots & \vdots & \ddots & \vdots \\ R(\|\mathbf{x}_{NT} - \mathbf{x}_1\|) & R(\|\mathbf{x}_{NT} - \mathbf{x}_2\|) & \dots & R(\|\mathbf{x}_{NT} - \mathbf{x}_{NT}\|) \end{bmatrix} \quad (7)$$

$$\mathbf{r}(\mathbf{x}) = [R(\|\mathbf{x}_1 - \mathbf{x}\|), R(\|\mathbf{x}_2 - \mathbf{x}\|), \dots, R(\|\mathbf{x}_{NT} - \mathbf{x}\|)]^T \quad (8)$$

The multiquadric basis function, formulated as $R(d) = \sqrt{d^2 + \sigma^2}$, is applied in this paper, where σ is a predefined parameter that is set to 1 in this study. Definitely, other basis functions could be considered too.

3. The proposed method

In this section, we first detail the proposed strategy of decomposition-based local learning. Then, the framework of the proposed SAEA-DBLL is presented.

3.1. The decomposition-based local learning strategy

Different from single-objective optimization, it is not trivial to calculate a fitness value to accurately represent the quality of an individual for multiobjective optimization. However, the decomposition-based approach alleviates this situation by partitioning the population into multiple subpopulations and assigning a more accurate fitness value for an individual in its corresponding subpopulation or local area. The evolution operator like GA or DE can be easily applied to generate offspring for multiobjective optimization since fitness values of individuals are needless, but they perform inefficiently on high-dimensional MOPs with a small number of FEs. Therefore, to accelerate the convergence speed of SAEA on EMOPs, we propose a decomposition-based local learning strategy, termed DBLL, to update the individuals in the decision space. The pseudocode of DBLL is presented in Algorithm 2.

Unlike most of the evolution operators deriving from single-objective optimization, the proposed DBLL is tailored for multiobjective optimization, which uses NV reference vectors to guide the update of population P with NP individuals. Here, $P = \{X|F\}$, $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{NP}\}$ represents the NP solutions and $F = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{NP}\}$ denotes NP corresponding objective vectors. $Vel = \{vel_1, vel_2, \dots, vel_{NP}\}$, where $vel_i = (vel_{i,1}, vel_{i,2}, \dots, vel_{i,D})$ is a vector with D dimensions that represents the velocity of the i th solution, $i = 1, 2, \dots, NP$. The main procedures of DBLL are as follows.

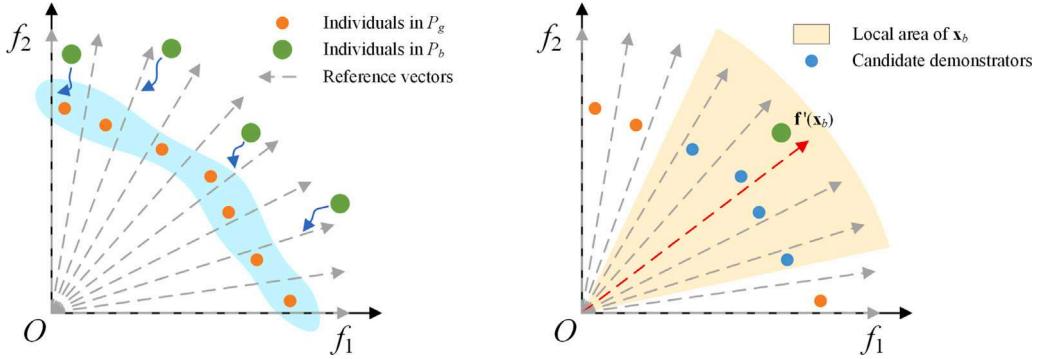


Fig. 3. An Illustration of decomposition-based local learning.

Algorithm 2 DBLL

Require: $P = \{X|F\}$, current population with NP individuals, $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{NP}\}$, $F = \{f_1, f_2, \dots, f_{NP}\}$; $Vel = \{vel_1, vel_2, \dots, vel_{NP}\}$, velocity of individuals; $V = \{v_1, v_2, \dots, v_{NV}\}$, NV reference vectors; t , number of current generation; T , number of neighbors of each reference vectors;

Ensure: X_{new} , updated individuals; Vel_{new} updated velocity;

- 1: $\mathbf{B}(i) = (i_1, i_2, \dots, i_T) \leftarrow$ Define indices of the T closest reference vectors to v_i , $i = 1, 2, \dots, NV$;
- 2: $[P_g, \mathbf{I}_g, Vel_g] \leftarrow$ Environmental selection based on APD (t, P, V);
- 3: $P_b \leftarrow P \setminus P_g$;
- 4: $X_{new} \leftarrow X_g$;
- 5: $Vel_{new} \leftarrow Vel_g$;
- 6: $F'_b \leftarrow$ Translation operation (F_b, F);
- 7: $[j_1, j_2, \dots, j_{|P_b|}] \leftarrow$ Assign individuals of P_b to V (F'_b, V);
- 8: **for** $k = 1 : |P_b|$ **do**
- 9: $\mathbf{I}_k \leftarrow \mathbf{B}(j_k) \cap \mathbf{I}_g$;
- 10: $s \leftarrow$ Randomly select a component from \mathbf{I}_k ;
- 11: $[\mathbf{x}_g, vel_g] \leftarrow$ The individual corresponding to v_s from P_g and its velocity;
- 12: $[\mathbf{x}_b, vel_b] \leftarrow$ The k th individual of P_b and its velocity;
- 13: $vel'_b \leftarrow r_1 \circ vel_b + r_2 \circ (vel_g - vel_b)$;
- 14: $\mathbf{x}'_b \leftarrow \mathbf{x}_b + vel'_b + r_3 \circ (vel'_b - vel_b)$;
- 15: $X_{new} \leftarrow X_{new} \cup \mathbf{x}'_b$;
- 16: $Vel_{new} \leftarrow Vel_{new} \cup vel'_b$;
- 17: **end for**
- 18: $X_{new} \leftarrow$ Mutation operation(X_{new});

First, indices of the T closest reference vectors of each reference vector are defined. $\mathbf{B}(i) = (i_1, i_2, \dots, i_T)$ means the T closest reference vectors of v_i are $\{v_{i_1}, v_{i_2}, \dots, v_{i_T}\}$. Then the population is divided into good population P_g and bad population P_b through the APD-based environmental selection, where P_g consists of individuals with the minimum APD values among their corresponding subpopulations, and P_b are the remaining individuals. Assume $NP > NV$, the number of individuals in P_b is always more than 0, i.e., $P_b \neq \emptyset$. Meanwhile, the indices of reference vectors corresponding to P_g can be also obtained, denoted $\mathbf{I}_g = (I_{g_1}, I_{g_2}, \dots, I_{g_{|P_g|}})$, where $v_{I_{g_i}}$ is the reference vector that the i th individual of P_g is assigned to. After conducting translation operation for P_b , individuals of P_b are assigned to the reference vector V again. The indices are denoted as $[j_1, j_2, \dots, j_{|P_b|}]$, and v_{j_k} is the reference vector corresponding to the k th individual in P_b . In the objective space, for the k th individual \mathbf{x}_b in P_b , its local area is defined by the neighboring reference vectors of v_{j_k} . An individual \mathbf{x}_g of P_g will be randomly selected from the local area of \mathbf{x}_b . Then, in the decision space, for \mathbf{x}_b in P_b , it learns the behavior of \mathbf{x}_g in P_g in the following manner.

$$vel'_b = r_1 \circ vel_b + r_2 \circ (vel_g - vel_b) \quad (9)$$

$$\mathbf{x}'_b = \mathbf{x}_b + vel'_b + r_3 \circ (vel'_b - vel_b) \quad (10)$$

where vel_b and vel_g are the velocities of \mathbf{x}_b and \mathbf{x}_g , respectively, \mathbf{x}'_b and vel'_b represent the updated position of \mathbf{x}_b and its updated velocity, r_1, r_2 , and r_3 are randomly generated vectors within $[0, 1]^D$. The inspiration of Eq. (9) and Eq. (10) comes from the competitive swarm optimizer (CSO) (Cheng & Jin, 2014; Tian et al., 2019) and the centripetal accelerated particle swarm optimization (CAPSO) (Beheshti & Shamsuddin, 2014), where $(vel'_b - vel_b)$ represents the acceleration of \mathbf{x}_b , which can improve the search efficiency during the optimization process. After individuals in P_b are updated by the learning strategy, they are combined into P_g to obtain a new population X_{new} . To improve the diversity of the population, mutation operation is applied to all the individuals of X_{new} . The widely used polynomial mutation (PM) (Deb & Agrawal, 1995) is adopted in this paper.

Fig. 3 illustrates the process of DBLL through a case with 2 objectives. In Fig. 3(left), the population is divided into P_g and P_b by the environmental selection based on APD. In Fig. 3(right), an \mathbf{x}_b in P_b will randomly select a candidate demonstrator \mathbf{x}_g to learn from its local area in the objective space, where T is set to 2. Then, the learning process is implemented in the decision space according to Eq. (9) and Eq. (10).

3.2. The proposed SAEA-DBLL

The proposed SAEA uses DBLL to update the population, whose framework is presented in Algorithm 3, which can be divided into the following steps.

(1) *Initialization*: An initial population P with NI individuals is generated using the Latin hypercube sampling method. Then individuals in P are copied to an empty archive A , which is applied to store individuals evaluated by real objective functions. A set of uniformly distributed reference vectors V_0 with NV individuals is also generated. Reference vectors in V_0 are copied to V and V_e , respectively.

(2) *Construction of RBF models*: M RBF models corresponding to M objective functions are constructed using individuals of the archive A . As optimization progresses, the accuracy of RBF models improves with the increase of the number of individuals in A .

(3) *DBLL-based evolutionary optimization*: RVEA with DBLL is applied to optimize the MOP approximated by RBF models. In this process, population P is updated by DBLL, and the objective values of individuals are estimated by RBF models.

(4) *Update of reference vectors*: After evolutionary optimization with w_{max} generations, two sets of reference vectors, i.e., V and V_e , are updated according to P .

(5) *Selection of new samples*: μ individuals (X_{new}) are selected from the final population P , which will be evaluated by real objective functions and added to the archive A .

SAEA-DBLL repeats steps (2)-(5) and terminates until the maximum number of FEs (FE_{max}) is reached. Significantly, before evolutionary optimization, velocity of the i th individual in P should be initialized as $v_i = (0, 0, \dots, 0)$, $i = 1, 2, \dots, |P|$, which is a D -dimensional zero vector. In addition, we use $\eta = (\frac{FE}{FE_{max}})^{\alpha}$ to replace $(\frac{t}{t_{max}})^{\alpha}$ of Eq. (4) for

calculating APD values. In the remaining part of this section, the update of reference vectors and selection of new samples will be elaborated, respectively.

Algorithm 3 SAEA-DBLL

Require: FE_{max} , maximum number of FEs; μ , number of infill solutions; w_{max} , maximum number of generations for evolutionary optimization; NI , size of initial population; NP , size of population; NV , number of reference vectors; T , number of neighboring reference vectors; K , parameter to control number of demonstrators; α , parameter to control rate of change of APD;

Ensure: A_{non} , nondominated solutions of archive A ;

- 1: $P \leftarrow$ Initialize population with NI individuals using Latin hypercube sampling method;
- 2: $A \leftarrow P$; $FE \leftarrow NI$; $w \leftarrow 1$;
- 3: $V_0 \leftarrow$ Generate NV uniformly distributed reference vectors;
- 4: $V \leftarrow V_0$; $V_e \leftarrow V_0$;
- 5: **while** $FE \leq FE_{max}$ **do**
- 6: $\mathfrak{R} \leftarrow$ Train an RBF model for each objective function by A ;
- 7: $P \leftarrow A$;
- 8: $Vel \leftarrow$ Initialize velocity of individual in P as zero vector;
- 9: $\eta \leftarrow (\frac{FE}{FE_{max}})^\alpha$;
- 10: **while** $w \leq w_{max}$ **do**
- 11: $Q \leftarrow$ Generate offspring by DBLL and evaluate them by RBFs ($P, \mathfrak{R}, Vel, V_e, \eta, T$);
- 12: $P' \leftarrow Q \cup P$;
- 13: $P \leftarrow$ Environmental selection based on APD (η, P', V);
- 14: $w \leftarrow w + 1$;
- 15: **end while**
- 16: $[V, V_e] \leftarrow$ Update V and V_e by Algorithm 4 (V_0, P, K);
- 17: $X_{new} \leftarrow$ Select new samples by Algorithm 5 (V, P, μ, η);
- 18: $P_{new} \leftarrow$ Evaluate X_{new} with real objective functions;
- 19: $A \leftarrow A \cup P_{new}$;
- 20: **end while**
- 21: $A_{non} \leftarrow$ Nondominated-sorting on individuals of A ;

Algorithm 4 Update of V and V_e

Require: $V_0 = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{NV}\}$, a set of uniformly distributed reference vectors; P , final population with NP individuals; K , parameter to control number of demonstrators, i.e., $|P_g|$;

Ensure: V , updated reference vectors for environmental selection and selection of new samples; V_e , updated reference vectors for DBLL;

- 1: $V \leftarrow$ Translate V_0 using Eq. 5;
- 2: $V_a \leftarrow$ Find active reference vectors through assigning individuals in P to V ;
- 3: $NV_e \leftarrow \lceil |P|/K \rceil$;
- 4: **if** $|V_a| > NV_e$ **then**
- 5: $[C_1, C_2, \dots, C_{NV_e}] \leftarrow$ Divide V_a into NV_e clusters by the k -means clustering method;
- 6: $V_e \leftarrow \emptyset$;
- 7: **for** $i = 1 : NV_e$ **do**
- 8: $V_i \leftarrow$ Find the reference vector that is closest to the center of C_i ;
- 9: $V_e \leftarrow V_e \cup V_i$;
- 10: **end for**
- 11: **else**
- 12: $V_e \leftarrow V_a$;
- 13: **end if**

3.2.1. Update of reference vectors

In the proposed method, V is adopted for environmental selection of the evolutionary optimization and selection of new samples, whose size is very close to $|P|$. Each reference vector in V_e can select the best individual assigned to itself according to the APD metric, and all of

the best individuals constitute P_g ($|P_g| \leq |V_e|$ since there may exist duplicate best individuals corresponding to reference vectors in V_e). When V_e is applied to the proposed DBLL strategy, whose size should be smaller than $|P|$ since P_b should contain adequate individuals for learning. If $|V_e| \geq |P|$, $|P_g| = |P|$ and $|P_b| = 0$ might occur. It is obvious that V_e influences the effect of the learning strategy dramatically. In this paper, we update V and V_e according to the final population P adaptively, and the process is presented in **Algorithm 4**. First, V is obtained by translating the uniformly distributed reference vectors in V_0 according to Eq. (5). Then, the active reference vectors, termed V_a , are found by assigning individuals in P to V . We set the number of reference vectors of V_e , termed NV_e , as $\lceil |P|/K \rceil$, where K is the parameter to control NV_e . If $|V_a| > NV_e$, we divide V_a into NV_e clusters by the k -means clustering method (Likas, Vlassis, & Verbeek, 2003), and V_e can be obtained by selecting the reference vector that is closest to its corresponding cluster center in each cluster. Otherwise, $V_e = V_a$. Fig. 4 illustrates the strategy of updating V_e by a 2-objective case, where $NV_e = 4$ and $|V_a| = 7$, thus 4 reference vectors are selected from 4 clusters to make up V_e .

3.2.2. Selection of new samples

Selecting new individuals to be evaluated using real objective functions is crucial to the performance of SAEA, where both convergence and diversity should be considered. The pseudocode of selecting new samples is presented in **Algorithm 5**, which can be divided into the following steps. First, the set of active reference vectors V_a in the translated reference vectors V is found. Then APD values of each individual in P are calculated. Afterward, we partition the reference vectors in V_a into μ clusters through the k -means clustering method. If the number of active reference vectors is smaller than the predefined μ , i.e., $|V_a| < \mu$, we set $\mu = |V_a|$. Note that each cluster of reference vectors corresponds to a cluster of individuals. Finally, μ individuals with the minimum APD values of their clusters are selected, termed X_{new} , for re-evaluation. Fig. 5 illustrates the sampling strategy by a 2-objective problem with 7 reference vectors and individuals, where 4 solutions with the minimum APD values are selected from 4 clusters of solutions.

Algorithm 5 Selection of new samples

Require: $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{NV}\}$, NV reference vectors; P , final population with NP individuals; μ , number of infill solutions; η , parameter used to calculate APD;

Ensure: X_{new} , new individuals for re-evaluation;

- 1: $V_a \leftarrow$ Find active reference vectors through assigning individuals in P to V ;
- 2: $[APD_1, APD_2, \dots, APD_{NP}] \leftarrow$ Calculate APD values of individuals in P (V_a, P, η);
- 3: $\mu \leftarrow \min\{|V_a|, \mu\}$;
- 4: $[C_1, C_2, \dots, C_\mu] \leftarrow$ Divide V_a into μ clusters by the k -means clustering method;
- 5: $X_{new} \leftarrow \emptyset$;
- 6: **for** $i = 1 : \mu$ **do**
- 7: $X_i \leftarrow$ Find the individual that has the minimum APD value among the individuals that assign to the reference vectors of C_i ;
- 8: $X_{new} \leftarrow X_{new} \cup X_i$;
- 9: **end for**

4. Experimental studies

In this section, the effectiveness of the proposed method is investigated by a series of numerical experiments. First, the experimental details are introduced, including the performance metric, benchmark problems, and parameter settings. Second, the effects of two extra parameters in SAEA-DBLL, i.e., T and K , are analyzed, respectively. Then, the components of the proposed DBLL is studied, and the effectiveness of the proposed DBLL is demonstrated as a whole, respectively.

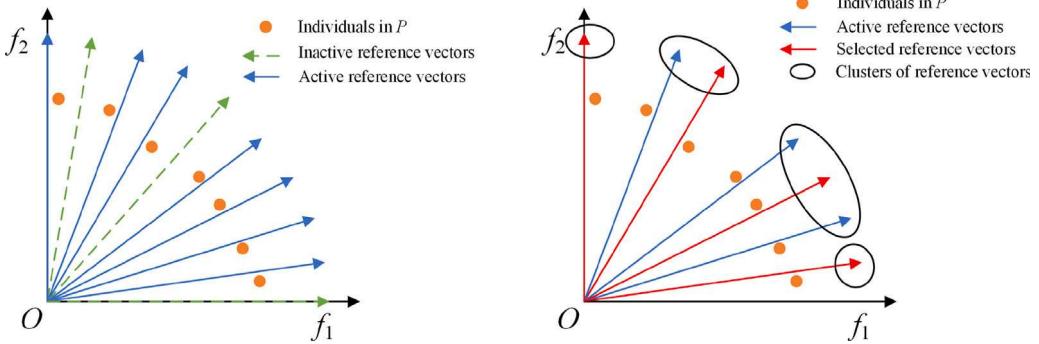


Fig. 4. An illustration of the strategy to update reference vectors.

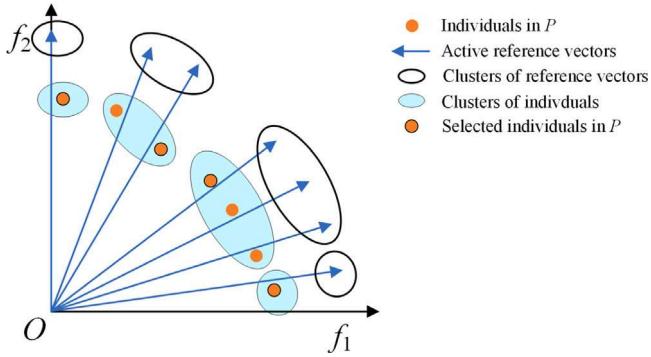


Fig. 5. An illustration of selecting individuals for re-evaluation.

Finally, we compare the proposed SAEA-DBLL with five state-of-the-art SAEAs, i.e., CPS-MOEA (Zhang et al., 2015), K-RVEA (Chugh et al., 2018), R-RVEA (Shen et al., 2020), END-ARMOEA (Guo et al., 2022), and MCEA/D (Sonoda & Nakata, 2022), to validate its superiority on expensive high-dimensional MOPs. All experiments in this section are implemented on PlatEMO (Tian, Cheng, Zhang, & Jin, 2017) with 2.9 GHz CPU and 16 GB RAM.

4.1. Experimental details

4.1.1. Performance metric and benchmark problems

In this paper, the inverted generational distance (IGD) (Coello & Cortés, 2005) is applied to measure the performance of the algorithms, which considers both convergence and diversity and is one of the most widely used performance metrics for multiobjective optimization. For a solution set S , its IGD value can be formulated as

$$\text{IGD}(S, R) = \frac{1}{|R|} \sum_{i=1}^{|R|} \min_{1 \leq j \leq |S|} \text{dis}(\mathbf{f}(s_j), \mathbf{f}(r_i)) \quad (11)$$

where R is a set of evenly distributed reference points on the true Pareto front (PF), $\text{dis}(\mathbf{f}(s_j), \mathbf{f}(r_i))$ denotes the Euclidean distance between two individuals $s_j \in S$ and $r_i \in R$ in the objective space. In this paper, $|R|$ is set to 10000 for all instances. In general, a smaller IGD value means better performance.

The experiments are implemented on DTLZ1-DTLZ7 (Deb et al., 2005) and WFG1-WFG4 (Huband, Hingston, Barone, & While, 2006) with extensible number of decision variables and of objectives. For each problem, the numbers of decision variables and objectives are set to $D = \{30, 50, 100\}$ and $M = \{3, 5, 10\}$, respectively.

4.1.2. Parameter settings

(1) Number of individuals in the initialization phase N_I is set to $D + 50$, where D denotes the number of decision variables;

(2) For each instance, the maximum number of fitness evaluations FE_{max} is set to 300.

(3) For each instance, the number of independent runs is set to 30. And the Wilcoxon rank test is adopted at a significance level of 0.05. “+”, “-” and “=” indicate that the results of other algorithms are significantly better, significantly worse and statistically similar to the results of the proposed method.

(4) For simulated binary crossover (SBX) (Deb & Agrawal, 1995), the distribution index η_c is set to 20, and the crossover probability p_c is set to 1. For the DE operator (Das & Suganthan, 2011), the differential factor F is set as 0.5, and the crossover ratio CR is set as 1. For PM (Deb & Agrawal, 1995), the distribution index η_m is set to 20, and the mutation probability p_m is set to $1/D$.

(5) In K-RVEA, R-RVEA, MCEA/D, and SAEA-DBLL, the number of reference vectors $N_v = [45, 50, 55]$ corresponding to the number of objectives $M = [3, 5, 10]$.

(6) Specific parameter settings of each algorithm are listed in Table 1.

4.2. Sensitivity analysis of parameters

The proposed SAEA-DBLL introduces two new parameters, termed T and K . T is the number of neighbors of each reference vector, which controls the exploration degree of the proposed method. K is used to define $|V_e|$, which controls the size of P_g and P_b . To analyze the sensitivity of the two parameters, we run SAEA-DBLL with different settings of T and K on 100D-10M-DTLZ1-3 and WFG4.

We first test the performance of SAEA-DBLL with $T = 1, 2, 3, 5, 10$, and 15. The mean IGD values and ranks are listed in Table 2. It can be observed that SAEA-DBLL with $T = 3$ ranks first. Since T controls the number of choices of local learning for an individual, a larger T leads to greater global exploration, while a smaller T leads to greater local exploitation. It is of great importance to balance exploration and exploitation well during the optimization. Therefore, according to the experimental results, $T = 3$ is recommended in this paper.

Next, we test the performance of SAEA-DBLL with $K = 1, 2, 3, 5, 10$, and 15, and the results are presented in Table 3. From the table, SAEA-DBLL with $K = 5$ ranks first. Similar to T , a large K means a great number of learners but a small number of demonstrators, which reduces the diversity of evolution. While a small K leads to the opposite situation, where only a small number of individuals are updated by Eq. (9) and Eq. (10). Both too large and small values of K have a negative effect on the evolution of the population. According to the experimental results, we recommend $K = 5$.

4.3. Effectiveness of DBLL

To verify the effectiveness of the proposed DBLL and its components, we compare SAEA-DBLL with its five variants whose features are as follows. Variant 1 removes the term of acceleration, i.e., $(\text{vel}_b' - \text{vel}_b)$,

Table 1

Parameter settings of the compared algorithms.

Algorithm	Parameter settings
CPS-MOEA (Zhang et al., 2015)	(a) Population size is 10; (b) Number of offspring of each individual is 5; (c) DE operator;
K-RVEA (Chugh et al., 2018) and R-RVEA (Shen et al., 2020)	(a) Population size is equal to the number of reference vectors; (b) Parameter in RVEA $\alpha = 2$; (c) Number of generations before updating surrogate models w_{max} is 20; (d) Number of new samples in each generation is 5; (e) Parameter while updating the surrogate models $\delta = 0.05N$; (f) GA operator;
EDN-ARMOEA (Guo et al., 2022)	(a) Population size is 50; (b) Diversity threshold δ is 0.08; (c) Number of generations before updating surrogate models $iter = 20$; (d) Number of new samples in each generation is 5; (e) ANN parameters: $J = K = 40$, $wd = 10^{-5}$, $lr = 0.01$, $iter_{train} = 80000$, $iter_{test} = 100$, $iter_r = 8000$, $p_I = 0.8$, and $p_R = 0.5$; (f) GA operator;
MCEA/D (Sonoda & Nakata, 2022)	(a) Population size is equal to the number of reference vectors; (b) The probability of choosing parents locally $\delta = 0.9$; (c) Maximum repeat time of offspring generation $R_{max} = 10$; (d) Parameter of SVM model $\gamma = 1$ (e) DE operator;
SAEA-DBLL	(a) Population size is 50; (b) Number of generations before updating surrogate models w_{max} is 20; (c) Number of new samples in each generation is 5; (d) Parameter in RVEA $\alpha = 2$; (e) Number of neighboring reference vectors $T = 3$; (f) Parameter to control number of demonstrators $K = 5$;

Table 2Mean IGD values and ranks obtained by SAEA-DBLL with different settings of T on 100D-10M-DTLZ1-3 and WFG4 over 30 independent runs. The best result in each row is highlighted.

Problem	M	D	$T = 1$	$T = 2$	$T = 3$	$T = 5$	$T = 10$	$T = 15$
DTLZ1	10	100	9.1744e+2	9.0984e+2	8.3676e+2	8.5713e+2	8.5201e+2	8.8233e+2
DTLZ2	10	100	1.5824e+0	1.5855e+0	1.5551e+0	1.5992e+0	1.5870e+0	1.5842e+0
DTLZ3	10	100	2.2694e+3	2.2315e+3	2.2350e+3	2.2085e+3	2.2410e+3	2.2382e+3
WFG4	10	100	6.3068e+0	5.9288e+0	5.8611e+0	5.9159e+0	5.8740e+0	5.8677e+0
Average Rank			5.00	4.00	1.50	3.50	3.25	3.75

Table 3Mean IGD values and ranks obtained by SAEA-DBLL with different settings of K on 100D-10M-DTLZ1-3 and WFG4 over 30 independent runs. The best result in each row is highlighted.

Problem	M	D	$K = 1$	$K = 2$	$K = 3$	$K = 5$	$K = 10$	$K = 15$
DTLZ1	10	100	1.0528e+3	9.0593e+2	9.0284e+2	8.9988e+2	7.7065e+2	8.5117e+2
DTLZ2	10	100	1.3823e+0	1.4234e+0	1.5024e+0	1.5896e+0	1.6404e+0	1.6180e+0
DTLZ3	10	100	2.3097e+3	2.2795e+3	2.2794e+3	2.2289e+3	2.2167e+3	2.2377e+3
WFG4	10	100	6.2345e+0	5.9322e+0	5.9410e+0	5.9185e+0	6.0940e+0	6.3967e+0
Average Rank			4.50	3.25	3.75	2.50	3.00	4.00

in Eq. (10). Variant 2 only uses mutation operation for the update of population. Variants 3, 4, and 5 use GA, DE, and CSO operator to replace the strategy of DBLL, respectively. For the CSO operator, we adopt the shift-based density estimation (SDE) (Li, Yang, & Liu, 2013) strategy to calculate the fitness of each individual, which can be formulated as follows.

$$\text{Fitness}(\mathbf{x}) = \min_{\mathbf{y} \in P \setminus \{\mathbf{x}\}} \sqrt{\sum_{i=1}^M (\max\{0, f_i(\mathbf{y}) - f_i(\mathbf{x})\})^2} \quad (12)$$

where $f_i(\mathbf{x})$ denotes the i th objective value of \mathbf{x} , M is the number of objectives. SDE evaluates the quality of a solution in terms of both convergence and diversity, which is widely used in multiobjective optimization. The statistical results of IGD values obtained by the six algorithms on 100-D benchmark problems with 10 objectives (16 instances in total) are presented in Table 4. For better visualization, the convergence profile plots obtained by the six algorithms on 100D-10M-DTLZ1-4 are displayed in Fig. 6 (associated with the mean IGD values over 30 independent runs).

Variant 1 is employed to investigate the effects of the acceleration term in DBLL. Two observations can be made from these results. First, Variant 1 outperforms SAEA-DBLL mainly on unimodal problems (DTLZ2, DTLZ4, and DTLZ5), while SAEA-DBLL performs better on multimodal problems (DTLZ1, DTLZ3, DTLZ6, and WFG4), which indicates that the acceleration term promotes the exploration and can avoid the population falling into local optima. Second, SAEA-DBLL shows overall better performance than Variant 1, which demonstrates the effectiveness of the acceleration term in DBLL. Variant 2 is employed to study the influence of the mutation operator on the performance of DBLL. From the table and figures, SAEA-DBLL performs much better than Variant 2 on most instances, which demonstrates that the

excellent performance of SAEA-DBLL can mainly be attributed to the decomposition-based local learning rather than the mutation operation.

Variants 3, 4, and 5 are employed to investigate the effectiveness of the proposed DBLL. From the results, Variant 3 and Variant 4 show much worse performance than SAEA-DBLL, which indicates that the GA operator and DE operator are inefficient in searching high dimensional decision space. CSO operator performs well on large-scale optimization problems of dimensionality up to 5000. Since the application of the CSO operator, the performance of Variant 5 is obviously improved when compared with that of Variant 3 and Variant 4. However, the proposed SAEA-DBLL still outperforms Variant 5 under comprehensive consideration, which demonstrates the effectiveness of the proposed DBLL on high dimensional expensive MOPs.

4.4. Comparison over other algorithms

In this section, the proposed SAEA-DBLL is compared with CPS-MOEA, K-RVEA, R-RVEA, EDN-ARMOEA, and MCEA/D to demonstrate its superiority on handling high dimensional expensive MOPs. The comparison experiments are conducted on 30, 50, and 100-D DTLZ1-7 and WFG1-4 with 3, 5, and 10 objectives, respectively. Tables 5, 6, and 7 list the statistical IGD values of the compared algorithms on 30-D, 50-D, and 100-D benchmark problems, respectively. The best result of each instance is highlighted.

From the three tables, SAEA-DBLL shows the best performance on nearly all the instances of DTLZ1, 2, 3, 5, and 6, MCEA/D performs best on DTLZ4, and K-RVEA performs best on DTLZ7 and WFG1. For the remaining three problems, i.e., WFG2, 3, and 4, SAEA-DBLL performs best on most instances. For 99 instances in total, SAEA-DBLL shows

Table 4

Statistical results of IGD values obtained by SAEA-DBLL and its five variants on 100-D benchmark problems with 10 objectives over 30 independent runs. The best result in each row is highlighted.

Problem	<i>M</i>	<i>D</i>	Variant 1	Variant 2	Variant 3	Variant 4	Variant 5	SAEA-DBLL	
DTLZ1	10	100	1.6143e+3 (2.75e+2) –	1.9088e+3 (1.92e+2) –	1.9233e+3 (1.15e+2) –	1.9385e+3 (1.57e+2) –	1.0598e+3 (1.16e+2) –	8.5764e+2 (1.13e+2)	
DTLZ2	10	100	1.3752e+0 (1.23e-1) +	6.2400e+0 (3.38e-1) –	6.3640e+0 (2.25e-1) –	6.3886e+0 (2.73e-1) –	1.6200e+0 (2.64e-1) ≈	1.5862e+0 (2.00e-1)	
DTLZ3	10	100	2.8423e+3 (4.28e+2) –	7.4458e+3 (4.32e+2) –	7.6740e+3 (4.13e+2) –	7.7778e+3 (3.08e+2) –	2.3190e+3 (6.91e+1) –	2.2380e+3 (1.09e+2)	
DTLZ4	10	100	1.9853e+0 (2.27e-1) +	6.4289e+0 (3.09e-1) –	6.3489e+0 (2.05e-1) –	6.3606e+0 (2.89e-1) –	2.4945e+0 (4.28e-1) –	2.2356e+0 (2.92e-1)	
DTLZ5	10	100	6.7672e-1 (1.43e-1) +	5.8853e+0 (2.72e-1) –	5.8507e+0 (2.47e-1) –	5.8895e+0 (2.24e-1) –	9.2187e-1 (1.85e-1) ≈	9.0318e-1 (1.66e-1)	
DTLZ6	10	100	5.3287e+1 (5.31e+0) –	8.0479e+1 (5.00e-1) –	7.9244e+1 (8.00e-1) –	7.8956e+1 (8.54e-1) –	4.1008e+1 (2.80e+0) –	3.9214e+1 (3.46e+0)	
DTLZ7	10	100	2.5513e+1 (2.72e+0) –	3.1067e+1 (1.76e+0) –	2.6057e+1 (2.70e+0) –	2.6830e+1 (2.39e+0) –	2.2237e+1 (3.78e+0) –	1.4095e+1 (3.63e+0)	
WFG1	10	100	3.3558e+0 (5.72e-2) ≈	3.4045e+0 (3.98e-2) –	3.3410e+0 (3.94e-2) ≈	3.3507e+0 (4.02e-2) ≈	3.3956e+0 (3.89e-2) –	3.3645e+0 (5.17e-2)	
WFG2	10	101	3.2465e+0 (8.64e-1) –	4.3291e+0 (6.29e-1) –	2.9541e+0 (6.04e-1) +	3.0423e+0 (6.00e-1) –	4.0927e+0 (8.15e-1) –	3.3734e+0 (7.25e-1)	
WFG3	10	101	1.5656e+0 (6.66e-2) +	1.8632e+0 (3.09e-2) –	1.8644e+0 (3.21e-2) –	1.8568e+0 (2.75e-2) –	1.7648e+0 (6.15e-2) ≈	1.7364e+0 (7.12e-2)	
WFG4	10	100	8.3276e+0 (1.13e+0) –	9.6198e+0 (4.01e-1) –	8.6569e+0 (5.82e-1) –	8.4026e+0 (7.61e-1) –	5.9165e+0 (3.63e-1) ≈	5.8381e+0 (2.44e-1)	
+/ - / ≈		4/5/2		0/11/0		1/9/1		0/9/2	
+/ - / ≈		0/7/4		1/9/1		0/9/2		0/7/4	

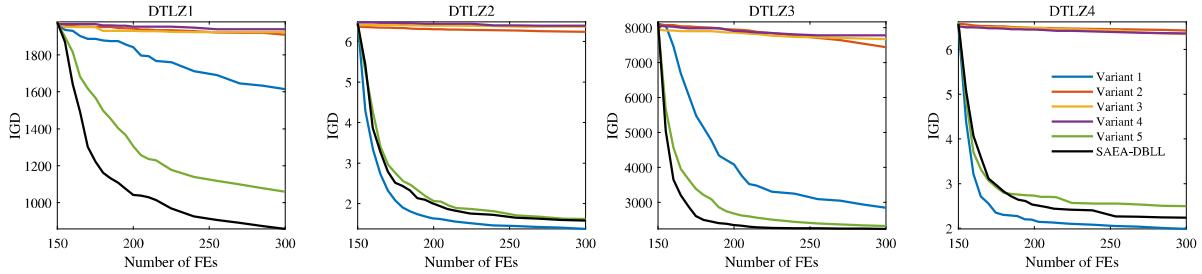


Fig. 6. Convergence trajectories of SAEA-DBLL and its five variants on 100D-10M-DTLZ1-4, which are associated with the mean IGD values of 30 independent runs.

Table 5

Statistical results of IGD values obtained by the six compared SAEAs on 30-D benchmark problems with 3, 5, and 10 objectives over 30 independent runs. The best result in each row is highlighted.

Problem	<i>M</i>	<i>D</i>	CPS-MOEA	K-RVEA	R-RVEA	EDN-ARMOEA	MCEA/D	SAEA-DBLL	
DTLZ1	3	30	5.0000e+2 (4.54e+1) –	5.6862e+2 (6.28e+1) –	2.4602e+2 (4.89e+1) –	3.1059e+2 (4.84e+1) –	3.2432e+2 (7.72e+1) –	2.2093e+2 (1.61e+1)	
	5	30	3.9959e+2 (3.46e+0) –	4.7140e+2 (4.14e+1) –	2.0626e+2 (5.48e+1) ≈	2.7950e+2 (3.64e+1) –	3.1967e+2 (5.55e+1) –	2.0416e+2 (2.12e+1)	
	10	30	2.6902e+2 (3.40e+1) –	2.2680e+2 (3.70e+1) –	1.2700e+2 (6.69e+1) +	2.0999e+2 (3.13e+1) –	2.2167e+2 (5.53e+1) –	1.7322e+2 (2.50e+1)	
DTLZ2	3	30	1.0940e+0 (1.63e-1) –	1.6245e+0 (8.24e-2) –	6.3134e-1 (9.84e-2) –	4.2437e-1 (1.82e-2) –	4.8514e-1 (4.38e-2) –	2.6295e-1 (3.03e-2)	
	5	30	1.3629e+0 (1.65e-1) –	1.6279e+0 (1.06e-1) –	1.0433e+0 (1.04e-1) –	7.0593e-1 (2.10e-2) –	7.1122e-1 (6.67e-2) –	6.0802e-1 (5.80e-2)	
	10	30	1.4045e+0 (9.81e-2) –	1.2735e+0 (1.19e-1) –	1.0498e+0 (5.53e-2) ≈	1.0139e+0 (2.70e-2) ≈	1.1669e+0 (6.48e-2) –	1.0225e+0 (6.18e-2)	
DTLZ3	3	30	1.4168e+3 (1.78e+2) –	1.6405e+3 (1.76e+2) –	7.1465e+2 (5.28e+1) –	8.5686e+2 (1.10e+2) –	6.9442e+2 (1.59e+2) ≈	6.8267e+2 (6.69e+0)	
	5	30	1.3237e+3 (1.41e+2) –	1.4703e+3 (1.78e+2) –	6.2159e+2 (5.48e+1) ≈	8.2460e+2 (7.75e+1) –	6.4715e+2 (1.43e+2) –	6.1980e+2 (3.32e+1)	
	10	30	1.0458e+3 (1.24e+2) –	9.7286e+2 (1.72e+2) –	2.9321e+2 (1.35e+2) +	5.3781e+2 (3.87e+1) –	5.9278e+2 (1.02e+2) –	4.9500e+2 (3.06e+1)	
DTLZ4	3	30	1.4957e+0 (1.37e-1) –	1.7637e+0 (1.42e-1) –	1.0976e+0 (9.82e-2) –	1.0337e+0 (2.20e-2) ≈	9.9028e-1 (2.99e-2) ≈	9.9475e-1 (1.03e-1)	
	5	30	1.6209e+0 (1.30e-1) –	1.7229e+0 (1.71e-1) –	1.3019e+0 (7.99e-2) –	1.2249e+0 (7.53e-2) –	1.1305e+0 (2.02e-2) +	1.1926e+0 (5.97e-2)	
	10	30	1.6321e+0 (8.54e-2) –	1.6174e+0 (1.32e-1) –	1.3794e+0 (8.20e-2) –	1.3605e+0 (7.85e-2) ≈	1.2806e+0 (1.84e-2) +	1.3267e+0 (6.29e-2)	
DTLZ5	3	30	1.0238e+0 (1.65e-1) –	1.5233e+0 (1.17e-1) –	5.3181e-1 (8.36e-2) –	2.5299e-1 (1.66e-2) –	2.8973e-1 (4.65e-2) –	1.7890e-1 (2.75e-2)	
	5	30	1.0870e+0 (2.10e-1) –	1.2816e+0 (1.43e-1) –	5.7540e-1 (1.16e-1) –	3.3658e-1 (2.37e-2) –	2.9871e-1 (5.64e-2) –	1.9457e-1 (3.99e-2)	
	10	30	9.1934e-1 (1.13e-1) –	8.2070e-1 (1.16e-1) –	3.7033e-1 (5.91e-2) –	3.0777e-1 (2.51e-2) –	5.2264e-1 (9.48e-2) –	2.0618e-1 (5.52e-2)	
DTLZ6	3	30	1.6871e+1 (1.07e+0) –	2.0055e+1 (9.97e-1) –	1.1194e+1 (1.97e+0) –	1.1277e+1 (1.46e+0) –	1.2393e+1 (1.50e+0) –	8.1735e+0 (1.82e+0)	
	5	30	1.6892e+1 (8.35e-1) –	1.9022e+1 (9.90e-1) –	9.7121e+0 (1.62e+0) –	9.9546e+0 (1.35e+0) –	1.1644e+1 (1.95e+0) –	7.7466e+0 (1.99e+0)	
	10	30	1.3406e+1 (8.23e-1) –	1.5974e+1 (7.29e-1) –	7.8275e+0 (1.10e+0) –	6.9771e+0 (8.13e-1) ≈	9.1343e+0 (1.46e+0) –	6.3941e+0 (1.39e+0)	
DTLZ7	3	30	8.0939e+0 (6.87e-1) –	1.8864e-1 (2.93e-2) +	4.8568e-1 (1.96e-1) –	3.2215e+0 (6.52e-1) –	7.7627e+0 (1.12e+0) –	7.3850e-1 (3.00e-1)	
	5	30	1.3082e+1 (1.28e+0) –	6.2996e-1 (7.97e-2) +	9.9104e-1 (2.16e-1) +	3.5222e+0 (7.32e-1) –	1.2177e+1 (2.28e+0) –	1.5465e+0 (3.57e-1)	
	10	30	2.4127e+1 (3.99e+0) –	2.3495e+0 (7.03e-1) ≈	2.0769e+0 (6.55e-1) ≈	6.0161e+0 (1.77e+0) –	2.7890e+1 (2.66e+0) –	3.5101e+0 (6.22e+0)	
WFG1	3	30	2.2857e+0 (5.39e-2) –	1.8440e+0 (8.96e-2) +	2.2663e+0 (5.77e-2) –	2.3091e+0 (4.51e-2) –	2.2539e+0 (6.72e-2) –	1.9586e+0 (1.28e-1)	
	5	30	2.6589e+0 (4.84e-2) –	2.3165e+0 (5.94e-2) ≈	2.6745e+0 (4.70e-2) –	2.7033e+0 (4.34e-2) –	2.6775e+0 (4.40e-2) –	2.3494e+0 (9.10e-2)	
	10	30	3.4467e+0 (3.15e-2) –	3.2768e+0 (4.89e-2) –	3.4307e+0 (3.13e-2) –	3.4517e+0 (3.62e-2) –	3.4480e+0 (3.58e-2) –	3.3684e+0 (6.29e-2)	
WFG2	3	30	8.1240e-1 (3.26e-2) –	8.4939e-1 (4.59e-2) –	8.6751e-1 (5.26e-2) –	8.1228e-1 (6.30e-2) –	6.9521e-1 (6.04e-2) ≈	6.9232e-1 (1.07e-1)	
	5	30	1.3014e+0 (8.55e-2) –	1.2278e+0 (7.28e-2) –	1.3743e+0 (1.71e-1) –	1.3184e+0 (6.51e-2) –	1.2580e+0 (2.46e-1) ≈	1.2110e+0 (1.69e-1)	
	10	31	3.0269e+0 (4.63e-1) +	1.9390e+0 (1.83e-1) +	2.9570e+0 (7.50e-1) +	2.8880e+0 (3.53e-1) +	3.9002e+0 (9.36e-1) ≈	3.6143e+0 (9.02e-1)	
WFG3	3	30	7.5935e-1 (3.01e-2) –	7.8782e-1 (1.59e-2) –	6.4326e-1 (5.62e-2) –	5.7549e-1 (1.62e-2) –	5.7172e-1 (3.27e-2) –	4.0935e-1 (3.54e-2)	
	5	30	1.0663e+0 (3.23e-2) –	1.0667e+0 (3.86e-2) –	9.1494e-1 (5.70e-2) –	1.0482e+0 (3.09e-2) –	8.3190e-1 (5.08e-2) –	6.4566e-1 (4.99e-2)	
	10	31	1.7013e+0 (6.55e-2) –	1.8274e+0 (6.34e-2) –	1.6658e+0 (9.82e-2) –	1.4704e+0 (6.28e-2) +	1.3965e+0 (8.57e-2) +	1.5613e+0 (1.05e-1)	
WFG4	3	30	5.9141e-1 (1.52e-2) –	5.8408e-1 (1.52e-2) –	5.9769e-1 (2.03e-2) –	6.1865e-1 (2.42e-2) –	5.5088e-1 (3.09e-2) –	5.0250e-1 (2.43e-2)	
	5	30	1.5163e+0 (6.08e-2) +	1.3521e+0 (5.37e-2) +	1.5598e+0 (3.80e-2) ≈	1.5530e+0 (4.42e-2) ≈	1.6493e+0 (8.99e-2) –	1.5947e-0 (1.05e-1)	
	10	30	6.5125e+0 (4.00e-1) –	6.4376e+0 (8.14e-1) –	5.1770e+0 (1.54e-1) +	5.3544e+0 (1.32e-1) +	7.2211e+0 (8.99e-1) –	5.6055e+0 (3.84e-1)	
+/ - / ≈		2/31/0		6/24/3		6/22/5		3/24/6	
+/ - / ≈		3/25/5		6/24/6		3/24/6		3/25/5	

the best performance on 55 instances, and CPS-MOEA, K-RVEA, R-RVEA, EDN-ARMOEA, and MCEA/D achieve the best performance on 0, 20, 10, 1, and 13 instances, respectively. Obviously, SAEA-DBLL performs best among the compared algorithms under comprehensive consideration.

The convergence trajectories of the six compared algorithms on 100-D DTLZ1-5 with 3, 5, and 10 objectives are plotted in Figs. 7, 8, and 9, respectively, which are associated with the median IGD values of 30 independent runs. From the figures, SAEA-DBLL achieves the best convergence on most instances of DTLZ1-3 and DTLZ5 with 100

Table 6

Statistical results of IGD values obtained by the six compared SAEAs on 50-D benchmark problems with 3, 5, and 10 objectives over 30 independent runs. The best result in each row is highlighted.

Problem	<i>M</i>	<i>D</i>	CPS-MOEA	K-RVEA	R-RVEA	EDN-ARMOEA	MCEA/D	SAEA-DBLL
DTLZ1	3	50	5.8804e+2 (1.81e+2) –	1.1282e+3 (9.78e+1) –	4.4370e+2 (7.01e+1) –	6.6017e+2 (8.13e+1) –	6.8504e+2 (1.65e+2) –	3.7519e+2 (2.82e+1)
	5	50	6.6549e+2 (8.73e+1) –	9.5371e+2 (8.34e+1) –	4.0315e+2 (7.84e+1) ≈	5.9219e+2 (5.93e+1) –	6.6107e+2 (1.19e+2) –	3.7239e+2 (3.57e+1)
	10	50	5.5589e+2 (1.01e+2) –	5.5852e+2 (6.33e+1) –	2.6454e+2 (1.12e+2) +	4.5472e+2 (7.37e+1) –	5.1154e+2 (8.77e+1) –	3.7329e+2 (4.10e+1)
DTLZ2	3	50	1.0754e+0 (1.82e–1) –	2.9130e+0 (1.80e–1) –	9.8393e–1 (1.47e–1) –	6.3881e–1 (3.17e–2) –	6.9102e–1 (9.79e–2) –	4.4489e–1 (4.06e–2)
	5	50	1.3580e+0 (3.60e–1) –	2.9032e+0 (1.59e–1) –	1.4156e+0 (2.53e–1) –	9.4398e–1 (5.79e–2) –	8.6565e–1 (8.18e–2) –	7.8126e–1 (4.20e–2)
	10	50	1.6990e+0 (3.06e–1) –	2.4296e+0 (2.11e–1) –	1.2648e+0 (7.76e–2) –	1.2361e+0 (5.71e–2) –	1.4356e+0 (1.58e–1) –	1.1434e+0 (1.14e–1)
DTLZ3	3	50	1.2846e+3 (3.43e+2) –	3.3984e+3 (3.03e+2) –	1.2450e+3 (1.20e+2) –	1.8080e+3 (2.28e+2) –	1.5077e+3 (3.53e+2) –	1.1841e+3 (8.59e+0)
	5	50	1.6628e+3 (2.75e+2) –	3.2154e+3 (2.62e+2) –	1.1726e+3 (9.76e+1) –	1.7146e+3 (1.50e+2) –	1.5167e+3 (2.82e+2) –	1.1335e+3 (1.05e+1)
	10	50	1.5030e+3 (2.46e+2) –	2.6298e+3 (2.97e+2) –	6.0699e+2 (2.81e+2) +	1.1977e+3 (1.25e+2) –	1.3241e+3 (2.44e+2) –	1.0009e+3 (2.89e+1)
DTLZ4	3	50	1.5788e+0 (2.38e–1) –	3.1672e+0 (1.45e–1) –	1.3547e+0 (1.31e–1) –	1.1804e+0 (4.98e–2) ≈	1.0788e+0 (4.75e–2) +	1.1656e+0 (9.22e–2)
	5	50	1.6005e+0 (3.12e–1) ≈	3.1277e+0 (1.86e–1) –	1.5500e+0 (1.44e–1) ≈	1.4524e+0 (1.37e–1) ≈	1.2014e+0 (2.75e–2) +	1.4958e+0 (1.48e–1)
	10	50	1.8922e+0 (3.61e–1) –	2.7294e+0 (1.70e–1) –	1.6481e+0 (1.67e–1) –	1.7143e+0 (2.22e–1) –	1.3802e+0 (4.74e–2) +	1.5375e+0 (1.28e–1)
DTLZ5	3	50	9.5327e–1 (2.77e–1) –	2.8471e+0 (1.72e–1) –	9.0348e–1 (1.53e–1) –	4.9399e–1 (3.48e–2) –	4.7074e–1 (4.82e–2) –	3.2647e–1 (4.32e–2)
	5	50	1.0364e+0 (3.80e–1) –	2.7806e+0 (1.50e–1) –	9.4487e–1 (1.93e–1) –	5.8342e–1 (4.54e–2) –	4.4364e–1 (5.67e–2) –	3.8792e–1 (5.86e–2)
	10	50	9.3424e–1 (2.82e–1) –	2.0249e+0 (1.92e–1) –	6.5075e–1 (9.26e–2) –	5.3558e–1 (4.17e–2) –	9.2167e–1 (1.80e–1) –	3.8833e–1 (7.68e–2)
DTLZ6	3	50	2.3810e+1 (1.98e+0) –	3.8325e+1 (8.24e–1) –	2.0443e+1 (2.70e+0) –	2.4384e+1 (2.18e+0) –	2.4357e+1 (2.74e+0) –	1.7513e+1 (2.66e+0)
	5	50	2.5030e+1 (2.29e+0) –	3.7205e+1 (1.04e+0) –	1.8839e+1 (2.90e+0) –	2.2574e+1 (2.17e+0) –	2.2736e+1 (1.90e+0) –	1.6725e+1 (2.86e+0)
	10	50	2.3160e+1 (1.98e+0) –	3.4038e+1 (7.93e–1) –	1.6312e+1 (1.98e+0) –	1.7840e+1 (1.60e+0) –	1.8703e+1 (2.63e+0) –	1.4812e+1 (1.76e+0)
DTLZ7	3	50	9.1482e+0 (1.10e+0) –	6.4026e–1 (1.12e–1) +	1.7525e+0 (7.45e–1) –	5.3323e+0 (5.13e–1) –	9.1978e+0 (7.58e–1) –	1.1776e+0 (5.75e–1)
	5	50	1.5113e+1 (1.81e+0) –	1.2986e+0 (4.14e–1) +	3.1429e+0 (1.08e+0) –	7.3607e+0 (9.63e–1) –	1.5341e+1 (1.52e+0) –	2.1715e+0 (7.36e–1)
	10	50	3.1014e+1 (3.66e+0) –	7.5766e+0 (2.38e+0) –	6.1704e+0 (2.52e+0) –	1.5294e+1 (2.11e+0) –	3.2268e+1 (3.05e+0) –	5.5917e+0 (7.50e+0)
WFG1	3	50	2.2311e+0 (7.95e–2) –	1.8542e+0 (8.84e–2) –	2.2818e+0 (4.69e–2) –	2.2963e+0 (5.77e–2) –	2.2779e+0 (5.59e–2) –	2.0228e+0 (9.15e–2)
	5	50	2.6739e+0 (4.90e–2) –	2.3510e+0 (6.70e–2) ≈	2.6827e+0 (3.84e–2) –	2.6717e+0 (4.95e–2) –	2.6665e+0 (5.07e–2) –	2.3707e+0 (7.38e–2)
	10	50	3.4478e+0 (2.35e–2) –	3.3213e+0 (4.17e–2) +	3.4366e+0 (3.42e–2) –	3.4544e+0 (3.81e–2) –	3.4604e+0 (2.81e–2) –	3.3595e+0 (4.44e–2)
WFG2	3	50	7.7412e–1 (6.71e–2) –	8.8323e–1 (1.92e–2) –	8.8570e–1 (3.70e–2) –	8.4250e–1 (4.69e–2) –	7.1479e–1 (6.43e–2) ≈	7.0582e–1 (6.42e–2)
	5	50	1.4337e+0 (2.50e–1) –	1.3426e+0 (7.06e–2) –	1.4299e+0 (1.49e–1) –	1.3477e+0 (7.42e–2) –	1.3264e+0 (2.34e–1) –	1.1450e+0 (1.28e–1)
	10	51	3.4170e+0 (7.28e–1) ≈	2.1445e+0 (1.51e–1) +	2.7948e+0 (7.48e–1) +	2.9604e+0 (2.91e–1) +	4.0796e+0 (8.03e–1) –	3.6458e+0 (8.87e–1)
WFG3	3	50	6.6397e–1 (4.56e–2) –	8.0901e–1 (1.67e–2) –	6.9614e–1 (3.15e–2) –	6.2066e–1 (2.77e–2) –	5.9350e–1 (3.76e–2) –	4.4989e–1 (2.41e–2)
	5	50	9.5338e–1 (7.15e–2) –	1.1008e+0 (2.43e–1) –	9.6773e–1 (3.26e–2) –	1.0718e+0 (4.14e–2) –	8.2959e–1 (3.84e–2) –	6.9461e–1 (3.77e–2)
	10	51	1.5353e+0 (6.89e–2) +	1.8514e+0 (4.42e–2) –	1.6870e+0 (7.66e–2) ≈	1.6020e+0 (5.78e–2) +	1.4602e+0 (7.49e–2) +	1.6520e+0 (8.45e–2)
WFG4	3	50	5.6131e–1 (3.50e–2) –	5.8408e–1 (1.26e–2) –	6.1005e–1 (2.02e–2) –	6.2889e–1 (2.12e–2) –	5.5871e–1 (2.99e–2) –	5.2601e–1 (3.04e–2)
	5	50	1.6129e+0 (8.06e–2) ≈	1.3480e+0 (5.33e–2) +	1.5907e+0 (4.43e–2) ≈	1.5509e+0 (3.80e–2) +	1.6369e+0 (7.90e–2) ≈	1.6222e+0 (6.86e–2)
	10	50	6.1891e+0 (4.70e–1) –	7.4026e+0 (9.72e–1) –	5.2488e+0 (1.48e–1) +	5.3154e+0 (1.04e–1) +	7.0488e+0 (7.02e–1) –	5.6983e+0 (5.57e–1)
+/ – / ≈		1/29/3	6/26/1	4/25/4	4/27/2	4/27/2	4/27/2	

Table 7

Statistical results of IGD values obtained by the six compared SAEAs on 100-D benchmark problems with 3, 5, and 10 objectives over 30 independent runs. The best result in each row is highlighted.

Problem	<i>M</i>	<i>D</i>	CPS-MOEA	K-RVEA	R-RVEA	EDN-ARMOEA	MCEA/D	SAEA-DBLL
DTLZ1	3	100	1.4554e+3 (3.41e+2) –	2.6423e+3 (3.13e+2) –	1.1042e+3 (2.10e+2) –	1.5948e+3 (1.65e+2) –	1.8296e+3 (4.00e+2) –	8.3693e+2 (5.73e+1)
	5	100	1.6568e+3 (2.10e+2) –	2.1753e+3 (1.12e+2) –	1.0817e+3 (2.12e+2) –	1.4669e+3 (1.26e+2) –	1.7193e+3 (2.14e+2) –	8.7159e+2 (6.59e+1)
	10	100	1.3977e+3 (1.32e+2) –	1.5959e+3 (3.13e+2) –	6.7952e+2 (2.30e+2) +	1.1806e+3 (1.82e+2) –	1.3810e+3 (2.06e+2) –	8.5764e+2 (1.13e+2)
DTLZ2	3	100	2.0428e+0 (4.34e–1) –	6.4664e+0 (2.93e–1) –	1.9786e+0 (3.53e–1) –	1.1409e+0 (8.92e–2) –	1.0106e+0 (1.29e–1) –	7.7600e–1 (8.02e–2)
	5	100	2.4384e+0 (6.68e–1) –	6.4637e+0 (2.65e–1) –	2.4091e+0 (6.02e–1) –	1.5740e+0 (1.20e–1) –	1.1307e+0 (1.63e–1) +	1.2156e+0 (1.23e–1)
	10	100	3.0936e+0 (9.03e–1) –	6.0467e+0 (3.49e–1) –	1.8191e+0 (2.69e–1) –	1.8375e+0 (1.84e–1) –	2.3845e+0 (4.17e–1) –	1.5862e+0 (2.00e–1)
DTLZ3	3	100	3.2008e+3 (8.71e+2) –	8.3522e+3 (2.74e+2) –	3.0010e+3 (3.54e+2) –	4.6376e+3 (4.74e+2) –	4.2484e+3 (7.01e+2) –	2.4432e+3 (9.98e+0)
	5	100	4.0236e+3 (8.53e+2) –	8.0048e+3 (3.57e+2) –	2.7864e+3 (2.43e+2) –	4.4120e+3 (4.60e+2) –	4.3399e+3 (7.32e+2) –	2.3738e+3 (1.30e+2)
	10	100	3.9183e+3 (6.84e+2) –	7.4687e+3 (2.75e+2) –	1.8902e+3 (5.11e+2) ≈	3.3541e+3 (4.05e+2) –	4.0224e+3 (7.05e+2) –	2.2380e+3 (1.09e+2)
DTLZ4	3	100	2.3257e+0 (4.58e–1) –	6.7173e+0 (2.56e–1) –	2.2470e+0 (4.40e–1) –	1.6579e+0 (1.09e–1) ≈	1.3621e+0 (1.56e–1) +	1.6736e+0 (2.61e–1)
	5	100	2.5675e+0 (6.59e–1) –	6.6526e+0 (2.41e–1) –	2.5716e+0 (5.50e–1) –	2.4098e+0 (5.15e–1) –	1.3837e+0 (7.60e–2) +	2.1300e+0 (2.96e–1)
	10	100	3.0331e+0 (6.49e–1) –	6.1004e+0 (3.47e–1) –	2.0502e+0 (3.78e–1) ≈	2.7427e+0 (6.02e–1) –	1.8064e+0 (4.91e–1) +	2.2355e+0 (2.92e–1)
DTLZ5	3	100	1.8779e+0 (5.09e–1) –	6.4105e+0 (2.69e–1) –	1.9536e+0 (4.32e–1) –	9.8067e–1 (8.88e–2) –	9.6299e–1 (1.61e–1) –	6.7935e–1 (9.03e–2)
	5	100	2.2619e+0 (8.55e–1) –	6.2884e+0 (2.56e–1) –	2.1590e+0 (4.01e–1) –	1.1962e+0 (1.45e–1) –	7.9061e+0 (1.11e–1) ≈	7.8691e+0 (1.33e–1)
	10	100	2.2865e+0 (9.32e–1) –	5.6443e+0 (3.51e–1) –	1.3366e+0 (3.02e–1) –	1.1480e+0 (1.78e–1) –	1.9078e+0 (5.09e–1) –	9.0318e–1 (1.66e–1)
DTLZ6	3	100	5.3103e+1 (4.82e+0) –	8.2825e+1 (7.61e–1) –	4.7376e+1 (4.08e+0) –	5.9931e+1 (3.09e+0) –	5.7414e+1 (4.99e+0) –	4.2573e+1 (3.95e+0)
	5	100	5.3884e+1 (4.25e+0) –	8.2198e+1 (8.81e–1) –	4.6404e+1 (3.51e+0) –	5.5739e+1 (3.30e+0) –	5.7935e+1 (4.75e+0) –	4.1801e+1 (4.29e+0)
	10	100	5.5126e+1 (3.48e+0) –	7.8903e+0 (8.96e–1) –	3.9837e+1 (2.67e–1) ≈	4.6984e+1 (3.61e+0) –	4.6580e+1 (3.43e+0) –	3.9214e+1 (3.46e+0)
DTLZ7	3	100	1.0184e+1 (5.66e–1) –	3.3015e+0 (5.49e–1) ≈	5.3101e+0 (9.83e–1) –	7.7786e+0 (4.02e–1) –	1.0262e+1 (6.62e–1) –	4.2518e+0 (1.90e+0)
	5	100	1.7609e+1 (8.21e–1) –	5.5357e+0 (7.80e–1) +	9.3136e+0 (1.31e+0) ≈	1.2004e+1 (6.70e–1) –	1.7152e+1 (1.08e+0) –	9.1084e+0 (2.95e+0)
	10	100	3.5347e+1 (2.27e+0) –	1.8744e+1 (2.41e+0) –	1.8275e+1 (2.45e+0) –	2.5110e+1 (1.43e+0) –	3.6577e+1 (1.78e+0) –	1.4095e+1 (3.63e+0)
WFG1	3	100	2.2676e+0 (9.23e–2) –	1.9002e+0 (9.22e–2) +	2.2829e+0 (3.73e–2) –	2.3011e+0 (2.92e–2) –	2.2998e+0 (4.96e–2) –	2.0349e+0 (1.02e–1)
	5	100	2.6522e+0 (4.23e–2) –	2.4040e+0 (7.05e–2) ≈	2.6654e+0 (3.53e–2) –	2.6664e+0 (4.02e–2) –	2.3913e+0 (9.16e–2) –	
	10	100	3.4348e+0 (2.67e–2) –	3.3419e+0 (4.50e–2) ≈	3.4242e+0 (3.48e–2) –	3.4293e+0 (3.12e–2) –	3.4379e+0 (2.85e–2) –	3.3645e+0 (5.17e–2)
WFG2	3	100	7.8972e–1 (6.44e–2) –	8.9740e–1 (4.64e–2) –	9.0739e–1 (4.06e–2) –	8.6767e–1 (3.12e–2) –	7.6192e–1 (5.35e–2) –	7.2407e–1 (6.10e–2)
	5	100	1.3513e+0 (1.51e					

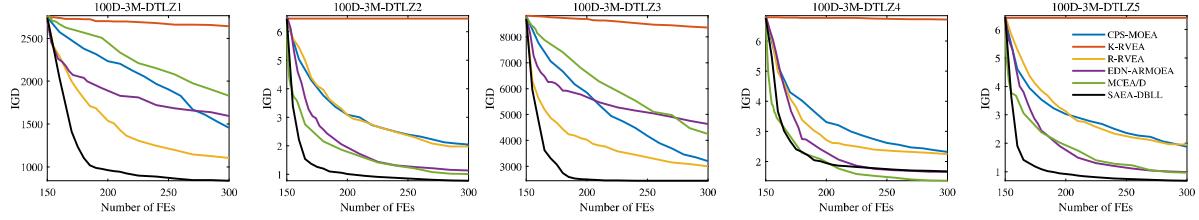


Fig. 7. Convergence trajectories of CPS-MOEA, K-RVEA, R-RVEA, EDN-ARMOEA, MCEA/D, and SAEA-DBLL on 100-D DTLZ1-5 with 3 objectives, which are associated with the mean IGD values of 30 independent runs.

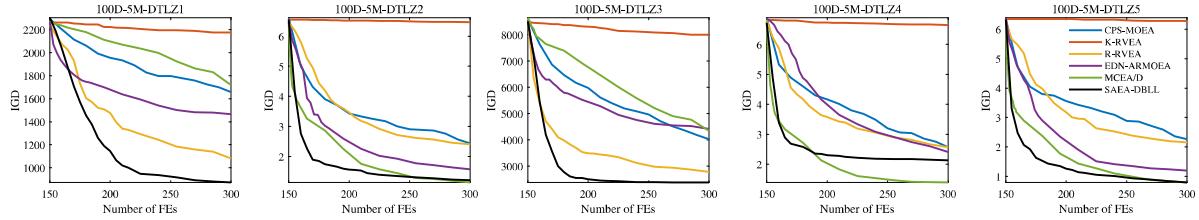


Fig. 8. Convergence trajectories of CPS-MOEA, K-RVEA, R-RVEA, EDN-ARMOEA, MCEA/D, and SAEA-DBLL on 100-D DTLZ1-5 with 5 objectives, which are associated with the mean IGD values of 30 independent runs.

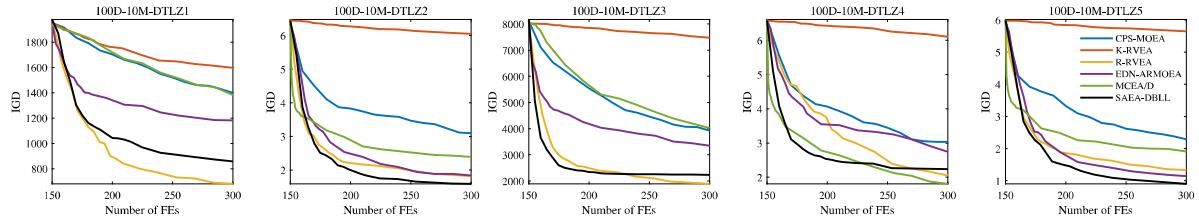


Fig. 9. Convergence trajectories of CPS-MOEA, K-RVEA, R-RVEA, EDN-ARMOEA, MCEA/D, and SAEA-DBLL on 100-D DTLZ1-5 with 10 objectives, which are associated with the mean IGD values of 30 independent runs.

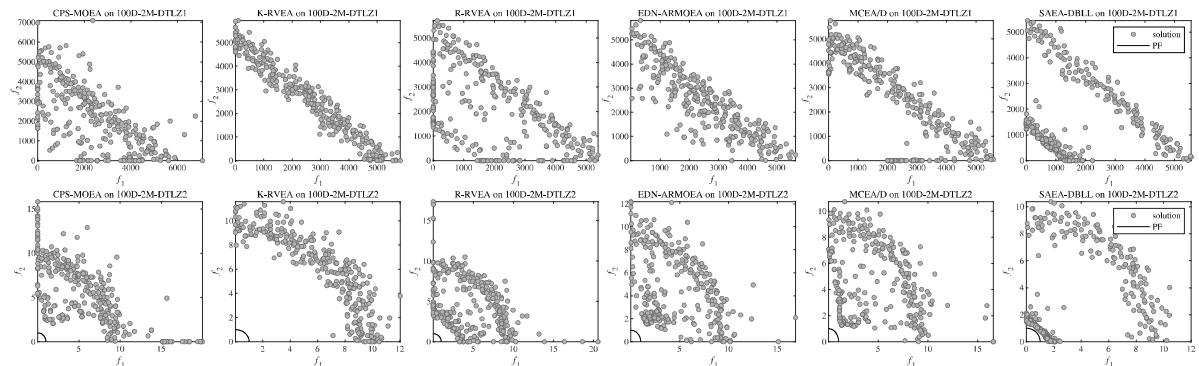


Fig. 10. Solutions obtained by CPS-MOEA, K-RVEA, R-RVEA, EDN-ARMOEA, MCEA/D, and SAEA-DBLL on 100-D 2-objective DTLZ1 and DTLZ2, which are associated with the mean IGD values of 30 independent runs.

decision variables. For DTLZ4, SAEA-DBLL and MCEA/D achieve the top two performance, and MCEA/D behaves best. From the five peer competitors, only R-RVEA and MCEA/D is comparable with the proposed SAEA-DBLL on the five problems. The search capacities of CPS-MOEA, K-RVEA, and END-ARMOEA sharply degenerate when tackling high-dimensional MOPs.

To further study the effectiveness of the proposed method, solutions obtained by the six compared algorithms on 100D-2M-DTLZ1 and 100D-2M-DTLZ2 are presented in Fig. 10. From the figure, K-RVEA performs badly in terms of convergence on the two instances, where its solutions are trapped on the nearby region of the initial samples. For EDN-ARMOEA and MCEA/D, their solutions evolve toward PF obviously but show low efficiency. CPS-MOEA shows good convergence but poor diversity on the two instances. R-RVEA shows poor diversity

and poor convergence on DTLZ1 and DTLZ2, respectively. In contrast, SAEA-DBLL shows high convergence efficiency on the two instances, where its solutions evolve toward PF strikingly and maintain good diversity.

In a word, the experimental results of this section demonstrate the superiority of the proposed method based on the decomposition-based local learning strategy when compared with state-of-the-art SAEAs on handling high-dimensional EMOPs.

4.5. Comparison of runtime

The computational time of the proposed method is studied in this subsection. In Table 8, the mean runtime and standard deviation of the six compared algorithms on all the instances with $D = 30, 50$, and 100

Table 8

Mean runtime obtained by the six compared algorithms, where the unit is [sec].

D	CPS-MOEA	K-RVEA	R-RVEA	EDN-ARMOEA	MCEA/D	SAEA-DBLL
30	4.93e-02 (7.91e-02)	6.21e+01 (2.59e+01)	3.72e+00 (5.44e-01)	8.88e+02 (3.80e+01)	3.18e+00 (3.28e-01)	8.47e+00 (1.87e+00)
50	6.37e-02 (1.11e-01)	7.86e+02 (3.63e+02)	4.53e+00 (3.40e-01)	6.92e+02 (3.28e+01)	4.52e+00 (1.04e+00)	1.18e+01 (3.04e+00)
100	4.51e-02 (5.88e-02)	7.03e+03 (3.73e+03)	6.19e+00 (8.75e-01)	1.15e+03 (5.14e+02)	3.72e+00 (5.76e-01)	1.28e+01 (3.23e+00)

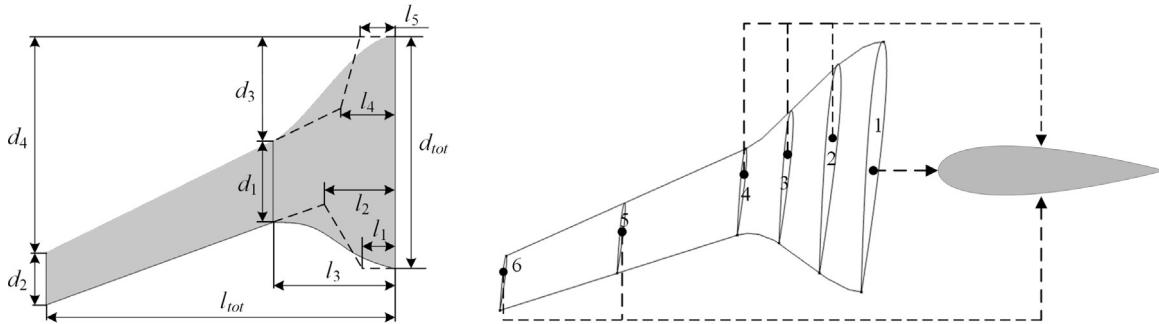


Fig. 11. Illustration of the shape design problem of BWBUG, where 39 decision variables including 9 plane variables and 30 section variables are considered.

Table 9

Lower and upper boundaries of the shape design problem of BWBUG.

	c_1^1	c_2^1	c_3^1	c_4^1	c_5^1	c_1^2	c_2^2	c_3^2	c_4^2	c_5^2
lower	-0.028675	-0.024699	-0.027838	-0.019125	-0.028710	-0.028675	-0.024699	-0.027838	-0.019125	-0.028710
upper	0.028675	0.024699	0.027838	0.019125	0.028710	0.028675	0.024699	0.027838	0.019125	0.028710
	c_1^3	c_2^3	c_3^3	c_4^3	c_5^3	c_1^4	c_2^4	c_3^4	c_4^4	c_5^4
lower	-0.028675	-0.024699	-0.027838	-0.019125	-0.028710	-0.028675	-0.024699	-0.027838	-0.019125	-0.028710
upper	0.028675	0.024699	0.027838	0.019125	0.028710	0.028675	0.024699	0.027838	0.019125	0.028710
	c_1^5	c_2^5	c_3^5	c_4^5	c_5^5	c_1^6	c_2^6	c_3^6	c_4^6	c_5^6
lower	-0.028675	-0.024699	-0.027838	-0.019125	-0.028710	-0.028675	-0.024699	-0.027838	-0.019125	-0.028710
upper	0.028675	0.024699	0.027838	0.019125	0.028710	0.028675	0.024699	0.027838	0.019125	0.028710
	d_1	d_2	d_3	d_4	l_1	l_2	l_3	l_4	l_5	
lower	300	150	400	860	100	300	450	300	100	
upper	400	300	500	1000	250	400	600	440	250	

are presented, which corresponds to the experimental results listed in Tables 5, 6, and 7, respectively. Obviously, CPS-MOEA takes much less time than other compared algorithms since it uses a computationally efficient KNN model and evaluates more individuals in each iteration. SAEA-DBLL performs slower than two classification-based SAEAs, i.e., CPS-MOEA and MCEA/D, since their computational time is less dependent on the increase of D . SAEA-DBLL consumes 12.8 s on 100-D problems on average, which is less than that of K-RVEA, EDN-ARMOEA at least one order of magnitude, and is similar to that of R-RVEA. The main reason is that SAEA-DBLL and R-RVEA use radial basis functions as surrogate models, which perform sufficiently faster than the other two approximation-based SAEAs, i.e., K-RVEA and EDN-ARMOEA. In brief, two conclusions can be made as follows. First, compared with the good performance of the proposed SAEA-DBLL, the computational time is acceptable. Second, the computational time of SAEA-DBLL is negligible when time-consuming simulations are involved.

5. Shape design of BWBUG

In this section, to investigate the effectiveness of the proposed method on real-world high-dimensional EMOPs, it is applied to optimize the shape of a BWBUG (Li, Wang, Dong, & Shen, 2022). MCEA/D is also applied as a peer competitor since it is a recently proposed algorithm for high-dimensional EMOPs.

5.1. Problem description

Fig. 11 illustrates the problem of shape optimization of BWBUG, where 9 plane variables ($d_1, d_2, d_3, d_4, l_1, l_2, l_3, l_4, l_5$) based on Bezier curves are used to control the plane shape of BWBUG, and 30 section variables ($c_1^i, c_2^i, c_3^i, c_4^i, c_5^i$, where $i = 1, 2, 3, 4, 5, 6$) based on CST method

are used to control the section shape of 6 hydrofoils. To promote the dynamic performance of the BWBUG, two objectives including maximizing the lift (L) and minimizing the drag (D) are considered. For convenience, the maximizing objective L is reformulated to be minimizing $-L$ to convert the above problem into a 2-objective minimization optimization problem. Specifically, the problem can be formulated as follows.

$$\begin{aligned} \min \mathbf{f}(\mathbf{x}) = & (-L, D) \\ \text{s.t. } \mathbf{x} \in & [\mathbf{x}_{lower}, \mathbf{x}_{upper}]^T \end{aligned} \quad (13)$$

where \mathbf{x}_{lower} and \mathbf{x}_{upper} are the lower boundary and upper boundary of the decision variables, respectively, which are given in Table 9. The baseline of the 6 sections are NACA0022, NACA0020, NACA0018, NACA0016, NACA0014, and NACA0012, respectively. The range of each section is from NACA[$\alpha - 2$] to NACA[$\alpha + 2$], where NACA[α] represents the baseline of its corresponding section. For example, the range of Section 1 is [NACA0020, NACA0024] since its baseline is NACA0022. The unit of the 9 plane variables is [mm]. A half of wing span (l_{tot}) and the chord length (d_{tot}) of the BWBUG are fixed to 1500 mm and 1000 mm, respectively.

Calculating the objective values ($-L, D$) of one solution requires the processes of parametric modeling, mesh generation, and CFD simulation, which takes half an hour in total. Therefore, only a few hundred times of real function evaluations can be afforded. Parameter settings of this section are the same as that in Section 4, where the number of initial samples N_I is set to 100, and FE_{max} is set to 300.

The size of the simulation area for CFD is 15 m \times 10 m \times 6 m, and the O-Block technique is applied on the surface of the BWBUG to improve the analysis accuracy. Fig. 12 shows more detailed information about the simulation area and mesh. The boundary conditions include the inlet velocity = 1.028 m/s, the outlet pressure = 0 Pa, and the

Table 10
 L , D , and $\frac{L}{D}$ of the baseline solution and the six representative solutions.

	Baseline solution	Solution 1	Solution 2	Solution 3	Solution 4	Solution 5	Solution 6
L (N)	69.43	80.75 ↑	78.65 ↑	74.42 ↑	68.66 ↓	65.31 ↓	60.37 ↓
D (N)	5.57	6.30 ↓	5.99 ↓	5.62 ↓	5.31 ↑	5.10 ↑	4.87 ↑
$\frac{L}{D}$	10.67	12.82 ↑	13.12 ↑	13.23 ↑	12.93 ↑	12.81 ↑	12.41 ↑

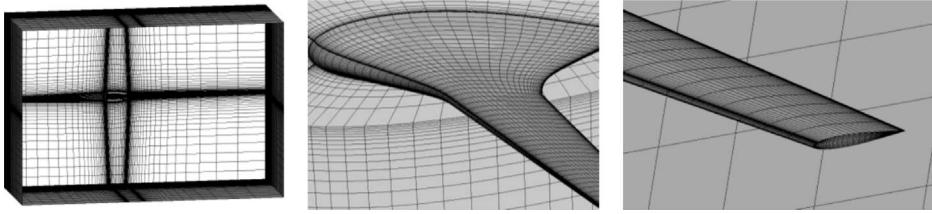


Fig. 12. Illustration of the simulation area and mesh of CFD.

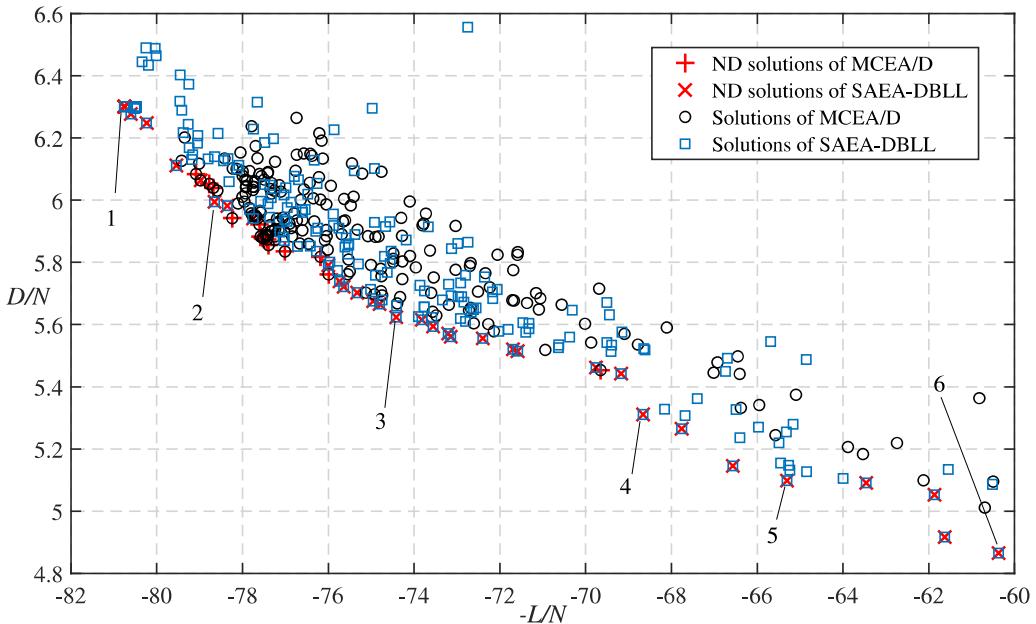


Fig. 13. Solutions obtained by MCEA/D and SAEA-DBLL on the shape design problem of BWBUG, where the nondominated (ND) solutions and six representative solutions of SAEA-DBLL are marked.

attack angle = 6° . To solve the three-dimensional incompressible steady viscous flow, Reynolds-Averaged-Navier-Stokes (RANS) equations with the $k - \omega$ turbulence model are adopted.

5.2. Optimization results

Fig. 13 presents the solutions obtained by MCEA/D and SAEA-DBLL on the shape design problem of BWBUG, where nondominated solutions are marked. It is obvious from the nondominated solutions that the lift is increased with the reduction of drag, which indicates the two objectives are in conflict. Compared with MCEA/D, the proposed SAEA-DBLL obtains a set of solutions with better convergence and diversity on the high-dimensional EMOP. Fig. 14 provides the iterative HV values of MCEA/D and SAEA-DBLL on the practical problem. It can be observed that the proposed method shows higher convergence speed during the whole optimization process, which validates the efficacy of the decomposition-based local learning strategy.

In addition, six representative solutions from the nondominated solution set of SAEA-DBLL are selected to display. The sections and pressure nephograms of the six solutions are presented in Fig. 15. According to the figure, with dwindling drag, the proportion of negative

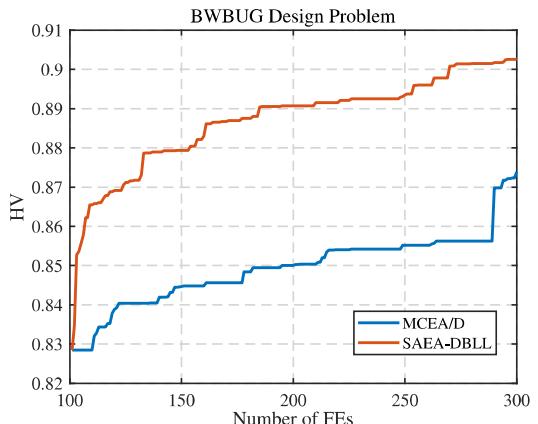


Fig. 14. Iterative HV values of MCEA/D and SAEA-DBLL on the BWBUG design problem.

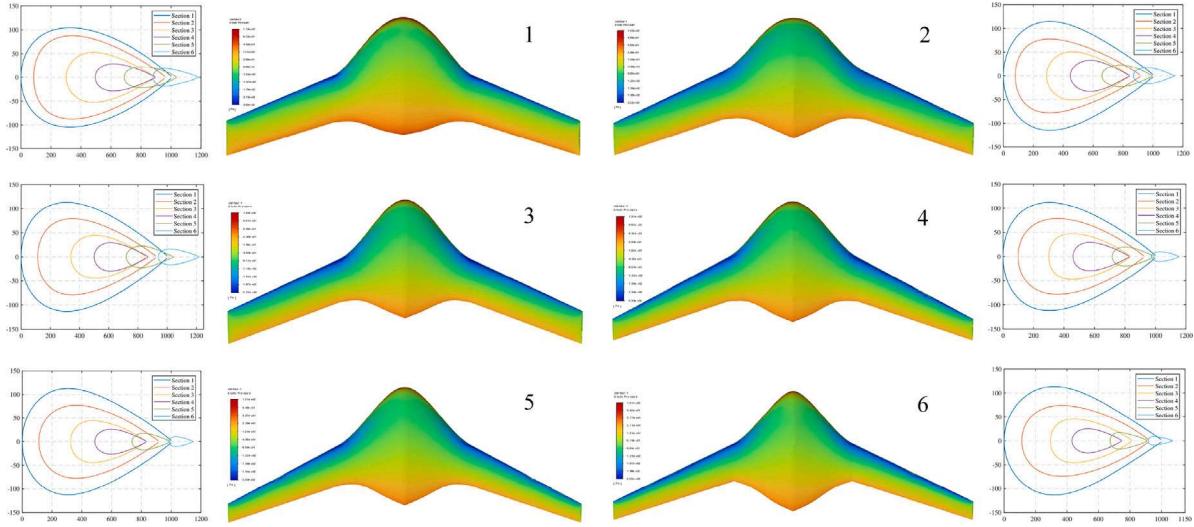


Fig. 15. Sections and pressure nephograms (the upper surfaces) of the six representative solutions, respectively.

pressure and sweep angle is increased, and the area of the plane shape and thickness of the wing is reduced. Values of L , D , and $\frac{L}{D}$ of the baseline solution (the solution in the center of the decision space) and the six representative solutions are listed in Table 10, where \uparrow and \downarrow denote the value of the representative solutions is better and worse than that of the baseline solution. Solution 1 shows the largest L , Solution 6 shows the smallest D , and the other four solutions have better comprehensive properties when both L and D are considered. $\frac{L}{D}$ of all the representative solutions are improved when compared with the baseline solution, and $\frac{L}{D}$ of solution 3 is 13.23, which is increased by 23.99% when compared with the baseline solution. In brief, the proposed SAEA-DBLL shows better performance than the recently proposed MCEA/D on the shape design problem of BWBUG, it obtains a set of high-quality solutions with different dynamic properties.

6. Conclusions

Different from the existing works that pay attention to the construction or management of surrogate models, like EDN-ARMOEA and MCEA/D, this paper promotes the performance of SAEAs on high-dimensional EMOPs from the perspective of population evolution. In this paper, a decomposition-based local learning strategy (DBLL) is proposed for the update of population in the high-dimensional search space of MOPs. Then a surrogate-assisted evolutionary algorithm based on DBLL is proposed for the promotion of solving high-dimensional EMOPs. In the proposed DBLL, the population is updated by learning from one of the best solutions of its corresponding local area based on the decompositional approach of multiobjective optimization. In SAEA-DBLL, radial basis functions are applied as the surrogate models to approximate the real objective functions, evolution search and selection of new samples are based on the reference vector guided method and calculation of APD values.

First, the effectiveness of the proposed DBLL is demonstrated by comparing it with several evolution operators including its two variants, GA operator, DE operator, and CSO operator. And the proposed SAEA-DBLL is compared with five state-of-the-art SAEAs on a set of widely-used benchmark MOPs with 30, 50, and 100 decision variables. The experimental results verify the superiority of the proposed method and the importance of evolution operator to solve EMOPs. Furthermore, SAEA-DBLL is applied to design the shape of a BWBUG and impressive solutions are obtained. In the future, EMOPs with multi-constraints (Li & Zhang, 2021) and with different time consumption for different objectives should be considered.

CRediT authorship contribution statement

Jiangtao Shen: Conceptualization, Data curation, Writing – original draft, Writing – review & editing. **Peng Wang:** Funding acquisition, Project administration, Supervision. **Huachao Dong:** Formal analysis, Funding acquisition. **Wenxin Wang:** Validation, Writing – review & editing. **Jinglu Li:** Validation, Writing – review & editing.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work is partially sponsored by the National Natural Science Foundation of China (Grant No. 52175251, 52205268), the National Basic Defense Scientific Research Program of China (Grant No. JCKY2021 206B005), and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University, China (Grant No. CX2022007).

References

- Arias-Montano, A., Coello, C. A. C., & Mezura-Montes, E. (2012). Multi-objective airfoil shape optimization using a multiple-surrogate approach. In *2012 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.
- Beheshti, Z., & Shamsuddin, S. M. H. (2014). CAPSO: centripetal accelerated particle swarm optimization. *Information Sciences*, 258, 54–79.
- Cheng, R., & Jin, Y. (2014). A competitive swarm optimizer for large scale optimization. *IEEE Transactions on Cybernetics*, 45(2), 191–204.
- Cheng, R., Jin, Y., Olhofer, M., & Sendhoff, B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5), 773–791.
- Chugh, T., Jin, Y., Miettinen, K., Hakkanen, J., & Sindhya, K. (2018). A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1), 129–142.

- Coello, C. A. C., & Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6, 163–190.
- Cornell, J. A. (1843). *Experiments with mixtures: Designs, models, and the analysis of mixture data*, vol. 255. Wiley-Interscience.
- Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1), 4–31.
- Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9(2), 115–148.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization: Theoretical advances and applications* (pp. 105–145). London: Springer London.
- Forrester, A. I., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1–3), 50–79.
- Guo, D., Jin, Y., Ding, J., & Chai, T. (2019). Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Transactions on Cybernetics*, 49(3), 1012–1025.
- Guo, D., Wang, X., Gao, K., Jin, Y., Ding, J., & Chai, T. (2022). Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4), 2084–2097.
- Habib, A., Singh, H. K., Chugh, T., Ray, T., & Miettinen, K. (2019). A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23(6), 1000–1014.
- Hao, H., Zhou, A., Qian, H., & Zhang, H. (2022). Expensive multiobjective optimization by relation learning and prediction. *IEEE Transactions on Evolutionary Computation*, 26(5), 1157–1170.
- Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5), 477–506.
- Jain, H., & Deb, K. (2013). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4), 602–622.
- Knowles, J. (2006). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1), 50–66.
- Li, J., Wang, P., Dong, H., & Shen, J. (2022). Multi/many-objective evolutionary algorithm assisted by radial basis function models for expensive optimization. *Applied Soft Computing*, 122, Article 108798.
- Li, G., Wang, Z., & Gong, M. (2022). Expensive optimization via surrogate-assisted and model-free evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(5), 2758–2769.
- Li, M., Yang, S., & Liu, X. (2013). Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3), 348–365.
- Li, G., & Zhang, Q. (2021). Multiple penalties and multiple local surrogates for expensive constrained optimization. *IEEE Transactions on Evolutionary Computation*, 25(4), 769–778.
- Li, G., Zhang, Q., Lin, Q., & Gao, W. (2021). A three-level radial basis function method for expensive optimization. *IEEE Transactions on Cybernetics*, 52(7), 5720–5731.
- Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, 36(2), 451–461.
- Lin, J., He, C., & Cheng, R. (2021). Adaptive dropout for high-dimensional expensive multiobjective optimization. *Complex & Intelligent Systems*, 8(1), 271–285.
- Liu, S., Li, J., Lin, Q., Tian, Y., & Tan, K. C. (2022). Learning to accelerate evolutionary search for large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 27(1), 67–81.
- Liu, S., Lin, Q., Feng, L., Wong, K. C., & Tan, K. C. (2022). Evolutionary multitasking for large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 27(4), 863–877.
- Liu, S., Lin, Q., Li, Q., & Tan, K. C. (2021). A comprehensive competitive swarm optimizer for large-scale multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(9), 5829–5842.
- Liu, S., Lin, Q., Tan, K. C., Gong, M., & Coello, C. A. C. (2020). A fuzzy decomposition-based multi/many-objective evolutionary algorithm. *IEEE Transactions on Cybernetics*, 52(5), 3495–3509.
- Liu, S., Lin, Q., Tian, Y., & Tan, K. C. (2021). A variable importance-based differential evolution for large-scale multiobjective optimization. *IEEE Transactions on Cybernetics*, 52(12), 13048–13062.
- Moore, B. (1981). Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1), 17–32.
- Pan, L., He, C., Tian, Y., Wang, H., Zhang, X., & Jin, Y. (2018). A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23(1), 74–88.
- Ponweiser, W., Wagner, T., Biermann, D., & Vincze, M. (2008). Multiobjective optimization on a limited budget of evaluations using model-assisted-metric selection. In *International conference on parallel problem solving from nature* (pp. 784–794).
- Shen, J., Dong, H., Wang, P., Li, J., & Wang, W. (2023). An inverse model-guided two-stage evolutionary algorithm for multi-objective optimization. *Expert Systems with Applications*, 225, Article 120198.
- Shen, J., Wang, P., & Wang, X. (2020). Managing radial basis functions for evolutionary many-objective optimization. In *2015 IEEE congress on evolutionary computation* (pp. 1–8). IEEE.
- Song, Z., Wang, H., He, C., & Jin, Y. (2021). A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 25(6), 1013–1027.
- Sonoda, T., & Nakata, M. (2022). Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multiobjective problems. *IEEE Transactions on Evolutionary Computation*, 26(6), 1581–1595.
- Svozil, D., Kvasnicka, V., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39(1), 43–62.
- Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4), 73–87.
- Tian, Y., Zheng, X., Zhang, X., & Jin, Y. (2019). Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. *IEEE Transactions on Cybernetics*, 50(8), 3696–3708.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988–999.
- Wang, X., Wang, G. G., Song, B., Wang, P., & Wang, Y. (2019). A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(5), 815–827.
- Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731.
- Zhang, Q., Liu, W., Tsang, E., & Virginas, B. (2009). Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3), 456–474.
- Zhang, J., Zhou, A., & Zhang, G. (2015). A classification and Pareto domination based multiobjective evolutionary algorithm. In *2015 IEEE congress on evolutionary computation* (pp. 2883–2890). IEEE.
- Zhao, M., Zhang, K., Chen, G., Zhao, X., Yao, C., Sun, H., et al. (2020). A surrogate-assisted multi-objective evolutionary algorithm with dimension-reduction for production optimization. *Journal of Petroleum Science and Engineering*, 192, Article 107192.