



A coevolutionary estimation of distribution algorithm based on dynamic differential grouping for mixed-variable optimization problems

Shijia Huang, Zhe Wang, Yang Ge, Feng Wang^{*}

School of Computer Science, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Keywords:

Mixed-variable optimization
Dynamic differential grouping
Estimation of distribution algorithm
Coevolutionary algorithms

ABSTRACT

The mixed variable optimization problems (MVOPs), which involves both continuous and discrete decision variables, are difficult to be solved due to the complex search space. Recently, many EA-based algorithms have been designed to address MVOPs. However, due to the mixed variables with different evolutionary operators and complex search space, it is difficult to handle the mixed variables effectively and the search efficiency cannot be guaranteed. How to solve MVOPs efficiently has been a challengeable issue. In this paper, we propose a mixed-variable optimization algorithm called coevolutionary estimation of distribution algorithm (CoEDA_{mv}). First, a dynamic differential grouping (DDG) method is employed to improve the search efficiency of CoEDA_{mv}, in which both the interaction of variables and search performance on the current search region are considered simultaneously. Second, two probabilistic models, i.e. fitness rank based continuous histogram (FRCH) and fitness rank based discrete histogram (FRDH), are proposed to handle continuous and discrete variables respectively, which can benefit from elite individuals obtained during the fitness ranking strategy and enhance the convergence performance with the elite neighborhood-based updating probability strategy. Compared with eight state-of-the-art algorithms, the experimental results on 28 artificial MVOPs show that CoEDA_{mv} is effective and efficient.

1. Introduction

Many real-world problems in scientific research and engineering technology can be formulated as mixed-variable optimization problems (MVOPs), e.g. financial optimization problems and resource scheduling problems, which involve a combination of discrete and continuous variables. Due to the mixed variables, the search space of the MVOPs becomes more complex, so that obtaining the optimal solutions has been difficult. In the literature, many efficient algorithms have been proposed to tackle discrete optimization problems (DOPs) and continuous optimization problems (CnOPs). However, the search performance of algorithms designed for CnOPs or DOPs usually performs not well when dealing with MVOPs. As the requirement of real applications, it is required to develop more efficient algorithms to tackle MVOPs.

The existing methods for solving MVOPs mainly can be classified into two classes. The first one includes some traditional mathematical programming methods, such as integer programming (Fetanat & Khorasaninejad, 2015), dynamic programming (Shuai et al., 2018), and branch-and-bound (Tanaka & Takii, 2015). The main idea of these traditional methods is to build a precise mathematical model and find the precise global optimal solution in a reasonable amount of time. These traditional methods can be used to solve simple optimization problems,

such as small-scale mixed variables, linear or convex problems. However, as the complexity of problem increases, it is impractical to provide the exact description of the problems. The efficiency and accuracy of those algorithms may quickly decrease (Brajević et al., 2022; Tejani et al., 2019). The second one is evolutionary algorithms, which are independent of the mathematical properties of optimization problems and employ random search strategies to promote the performance on complex MVOPs.

Evolutionary algorithms (EAs) are regarded as population-based optimization methods, which can obtain multiple candidate solutions in a single run (Aye et al., 2023; Kumar et al., 2021). EAs are well known due to their potential global search ability and easy-to-parallelized nature. The majority of EAs are tailored for solving CnOPs or DOPs, such as genetic algorithms (GA) (Sastry et al., 2005), estimation of distribution algorithm (EDA) (Larrañaga & Lozano, 2001), particle swarm optimization (PSO) (Van den Bergh & Engelbrecht, 2004) and ant colony optimization (ACO). There are many meta-heuristic algorithms in the literature. However, due to the complex search space of mixed variable optimization problems, most of these algorithms are difficult to handle the mixed variables effectively and the search efficiency cannot be guaranteed. Compared with other meta-heuristic algorithms,

^{*} Corresponding author.

E-mail addresses: huangsj@whu.edu.cn (S. Huang), z_wang@whu.edu.cn (Z. Wang), gy8911@whu.edu.cn (Y. Ge), fengwang@whu.edu.cn (F. Wang).

some variants of EAs have been proposed for mixed-variable optimization, such as ACO_{mv} (Liao et al., 2013), DE_{mv} (Lin et al., 2018), and PSO_{mv} (Wang et al., 2021). Since these EAs-based approaches are gradient-free and do not rely on the problem structure, these algorithms have found extensive application in handling diverse optimization problems from different domains (Li et al., 2023; Tejani et al., 2017; Wang & Wang, 2020).

In recent years, many researchers focus on the design of algorithms for handling MVOPs and numerous EA-based methods have been proposed. In summary, these EA-based methods for MVOPs are mainly divided into three categories in the literature. The first one lies in the discretization of continuous variables (Gallard, 1999), so that the reproduction operators designed to solve DOPs can be utilized to handle MVOPs. The second one transfer the discrete variables into the continuous variables via relaxation strategies (Chen et al., 2015; Gao & Hailu, 2010). However, it is significant to select the specific relaxation strategies, which leads to the unstable performance of such algorithms. The third one is named as hybrid algorithms (Lin et al., 2018; Wang et al., 2020), where two different evolutionary operators are used to handle discrete and continuous decision variables separately. Since these hybrid algorithms require no time for variable discretization or relaxation, and can deal with the mixed variables simultaneously, it has attracted more and more researchers' attentions (Kumar et al., 2021). As mentioned above, there are many different kinds of reproduction operators in these hybrid algorithms. However, poor performance among the reproduction operator or inconsistent performance among multiple reproduction operators may result in premature convergence and reduce the efficiency of algorithm (Singh et al., 2022). In addition, these algorithms with hybrid reproduction operators fail to use the population's historical information and would decrease the convergence speed (Kumar et al., 2022). How to increase the search efficiency and convergence ability of the current algorithms is a challenging issue.

The cooperative coevolutionary algorithm (CC) (Yang et al., 2008) decomposes the complex optimization problem into several subproblems by grouping decision variables and optimizes the subproblems separately, which can simplify MVOPs and accelerate the convergence speed (Wang et al., 2022). However, the search performance of CC strongly depends on the decomposition strategy. In contrast to other EAs, Estimation of distribution algorithm (EDA) will build the probabilistic models to acquire knowledge from the global information of the landscape (Wang et al., 2020). How to select and learn the customized probabilistic models for MVOPs is a challenging issue. While the efficiency of traditional probabilistic models, such as Gaussian models, Bayesian networks, is often influenced by control parameters and prior assumptions about the distribution of the data (Liang et al., 2018; Shahriari et al., 2015), the histogram probabilistic model is a non-parametric model that makes no explicit assumptions about the distribution of the data. This makes it more flexible and adaptable to various shapes and types of data distributions. In this paper, in order to increase the global search capabilities and convergence speed, a new variant of EAs named $CoEDA_{mv}$ is proposed, which is a learning-guided evolutionary algorithm that uses global information in the solution space and historical information in the evolution process to enhance the performance. To improve search efficiency, a dynamic differential strategy is employed at first. And two probabilistic models are proposed to handle discrete and continuous variables respectively. In detail, the major contributions of this paper are outlined as follows.

- A dynamic differential grouping method (DDG) is designed to simplify the complex search space. DDG takes both variable interactions and search performance on the current search region into account, which can divide the interacting variables into the same subgroup while grouping non-interacting or stagnant variables separately. Meanwhile, DDG dynamically modifies the grouping results during algorithm iterations to maintain better grouping accuracy. Thus, it can improve the search efficiency of $CoEDA_{mv}$.

- Two histogram probabilistic models are built to model the distribution features of mixed variables. Specifically, $CoEDA_{mv}$ utilizes a fitness rank based discrete histogram model (FRDH) for handling discrete decision variables, and a fitness rank based continuous histogram model (FRCH) for handling continuous decision variables. In order to enhance search ability, a fitness ranking strategy and an elite neighborhood-based updating probability strategy are used to learn the implicit information in populations, i.e. locating elite individuals and increasing the effect of elite neighborhoods on offspring generation. Due to the flexibility and cooperation of hybrid reproduction operators with different probabilistic models, $CoEDA_{mv}$ can enhance the convergence accuracy.
- To evaluate the effectiveness of $CoEDA_{mv}$, eight state-of-the-art EAs have been selected for comparison experiments on 28 artificial MVOPs. In addition, six combinations of the ablation experiments have been designed to testify the proposed DDG and two histogram probabilistic models. The experimental results demonstrate that the proposed $CoEDA_{mv}$ achieves competitive performance on MVOPs.

The rest of this article is organized as follows. Section 2 briefly introduces some related works about the EA-based algorithms for MVOPs and cooperative coevolutionary algorithms. Section 3 describes the details of the proposed $CoEDA_{mv}$. Then, Section 4 exhibit the experimental studies and results compared with 8 state-of-the-art EAs on 28 artificial MVOPs. Section 5 concludes the paper.

2. Related work

In this section, we give a brief introduction on the existing EAs for MVOPs. Then, we review the cooperative coevolutionary algorithms which utilize the variable interaction-based grouping strategies.

2.1. Evolutionary algorithms for MVOPs

In the last few years, many variants of EAs have been proposed to handle the mixed variable optimization problems. As the analysis in Section 1, The existing EA-based methods for MVOPs can be roughly classified to the following three categories.

The first one is to discretize the continuous variables, then the MVOPs can be solved as discrete optimization problems (DOPs). Binary-coded GA (Gallard, 1999) is a typical method for DOPs, which achieves the discretization by encoding the available value of each continuous variable into a binary string. Nevertheless, the length of the binary strings is limited in binary-coded GA, which affects the effectiveness and efficiency of the algorithm.

The second one transfers discrete variables into continuous variables through relaxation, but considering only the available values of the discrete variables during the objective function evaluation. It is worth noting that the discrete variables can be further divided into categorical variables or ordinal variables, depending on whether or not the available values of the variables have an order of magnitude. Ordinal variables are mapped to the continuous search space based on the magnitude of their values, and the search strategy for CnOPs is utilized to obtain the real values (Chen et al., 2015; Gao & Hailu, 2010; Nonut et al., 2022). However, since the available values for categorical variables usually represent different labels or symbols, it is difficult to apply the existing reproduction operators for CnOPs to solve MVOPs. Therefore, these algorithms are advantageous to deal with continuous and ordinal decision variables for MVOPs.

As the most commonly used algorithm of MVOPs, the third one combine different reproduction operators to handle different types of variables, which can take advantage of different operators and do not require additional calculations for variable type transformation. In order to enhance the convergence efficiency, these algorithms often integrate a local search strategy into the operators, such as ACO_{mv} (Liao

et al., 2013), DE_{mv} (Lin et al., 2018) and PSO_{mv} (Wang et al., 2021), EDA_{mun} (Wang et al., 2020), EDA_{mv} (Molina Pérez et al., 2022). And numerous algorithms have been applied in real-world MVOPs (Brajević et al., 2022; Jiang et al., 2022; Kim et al., 2020; Wang et al., 2023). Nevertheless, these algorithms rarely consider the historical information and the elite solutions within the population, which decreases the quality of offspring generation.

2.2. Cooperative coevolutionary algorithms

As the first cooperative coevolutionary (CC) algorithm, CCGA (Potter & De Jong, 1994) was proposed based on the idea of divide-and-conquer, which can decompose a D-dimensional problems into D one-dimensional subproblems. In each coevolutionary cycle, each subproblem corresponds to a subpopulation, and each subpopulation can be evolved by CCGA using an evolutionary optimizer in a round-robin fashion. The major key in utilizing the CC is choosing variable grouping strategy (He et al., 2021; Omidvar et al., 2015).

Over the past few years, various variable grouping methods have been developed to increase the search performance of CC. There are several variable grouping strategies, i.e. random grouping (Omidvar et al., 2010), linear grouping (Van Aelst et al., 2006), ordered grouping (Chen et al., 2010) and Bayesian network-based grouping (Shahriari et al., 2015). The subgroups should be constructed based on the interaction among decision variables, and the interactions between subgroups should be kept as small as possible. However, as the optimization problems with high-dimensional or mixed variables become more complex, the traditional grouping methods encounter challenges in efficiently classifying the interacting variables into the same subgroup. Moreover, Bayesian network-based grouping strategy acquires to learn the structure of the Bayesian network and its parameters (conditional probabilities), which is an NP-hard problem and makes it a poor choice for measuring interactions between variables.

Recently, many decomposition strategies based on the variable interaction analysis have been developed to enhance the convergence accuracy of the algorithms with different grouping methods. The differential grouping (DG) is the most popular grouping method which derives a set of finite difference equations for interaction detection from the definition of partially additive function. Compared with other grouping methods, DG is insensitive to computational roundoff errors due to its simpler computations. In the beginning, a differential grouping (DG) method is proposed in Omidvar et al. (2015), which can identify the interaction between a pair of variables, and show that the DG method is sensitive to the parameter ϵ used to determine whether two variables are inseparable or not. DG2 (Omidvar et al., 2017) is an enhanced variant of DG which adjusts the value of parameter ϵ adaptively. In order to reduce the fitness evaluates of DG, a recursive decomposition method (RDG) (Sun et al., 2017) studies the interaction between a pair of variable sets. Then, a variant of RDG (RDG2) (Sun et al., 2018) with adaptive parameter tuning is proposed to enhance the ability of detecting interaction between variable sets. To further save the fitness evaluates, Yang et al. demonstrate the redundant computation existing in the RDG and design an efficient recursive decomposition method (ERDG) (Yang, Zhou, Li & Yao, 2020). Recently, Chen et al. have proposed an efficient adaptive differential grouping (EADG) (Chen et al., 2023) based on the separability of the objective function, which uses an instance type identification procedure and adopts the corresponding grouping strategy. The selection of the appropriate decomposition strategy can significantly improve the search performance and grouping accuracy of CC.

As mentioned above, the cooperative coevolution used to simplify MVOPs helps to improve the search efficiency of existing algorithms. However, the static grouping strategy cannot deal with MVOPs efficiently but heavily rely on grouping accuracy. Furthermore, the existing algorithms tailored for MVOPs do not fully utilize the information of elite individuals in the population, resulting in insufficient convergence ability. Here, we propose a mixed variable optimization algorithm (CoEDA_{mv}) to enhance the convergence accuracy.

3. Proposed algorithm for MVOPs

In this section, a novel coevolutionary estimation of distribution algorithm (called CoEDA_{mv}) is proposed to deal with MVOPs. In CoEDA_{mv}, a dynamic differential grouping method (DDG) is proposed to simplify the search space and increase the convergence efficiency. To deal with mixed variables, two histogram probability models are constructed, in which a fitness ranking strategy and an elite neighborhood-based updating probability strategy are employed to update these probability models.

3.1. Dynamic differential grouping method

Cooperative coevolutionary (CC) algorithm decomposes a complex optimization problem by grouping decision variables into several subproblems, which can reduce the complexity of MVOPs. It is noted that the performance of the CC is closely related to the decomposition strategy of the variables. The existing grouping strategy based on the variable interaction is a competitive methodology. Grouping results are also obtained before evolution and then remain unchanged. However, when certain variables within the same subgroup tend to stagnate, considering only the interaction between variables often leads to remaining variables from the above subgroup to fall into a local optimum. Then, the static grouping strategy has the disadvantage of relying heavily on the grouping accuracy. To tackle this problem, inspired by Ma et al. (2016), CoEDA_{mv} proposes a dynamic differential grouping method (DDG) based on the interaction and search status of variables on the current search region. In the iteration of the algorithm, CoEDA_{mv} will update the grouping results dynamically. The dynamic differential grouping (DDG) derives a set of finite difference equations for interaction detection from the definition of partially additive function. Compared with the Bayesian network, DDG is insensitive to computational roundoff errors due to its simpler computations.

Give a optimization function $\min f(\mathbf{x}) = f(x_1, x_2, \dots, x_D)$, the interaction between two variables, x_i and x_j , is considered to be established when Eq. (1) is satisfied.

$$\begin{cases} f(\mathbf{x})|_{x_i=t_1, x_j=t_3} < f(\mathbf{x})|_{x_i=t_2, x_j=t_3} \\ f(\mathbf{x})|_{x_i=t_1, x_j=t_4} > f(\mathbf{x})|_{x_i=t_2, x_j=t_4} \end{cases} \quad (1)$$

where D represents the dimension of decision variables, t_1, t_2 are two values of the variable x_i and t_3, t_4 are two values of the variable x_j . Eq. (1) means that the fitness corresponding to the variable x_i taking the value of t_1 or t_2 varies with the variable x_j . To be specific, if a variable interacts with any other variable, these two variables should be assigned into the same subgroup; otherwise, it should be assigned to a new subgroup individually when there is no interaction. In an extreme cases, there could be a maximum of D subgroups, indicating complete separability of all variables; while if the variables are fully non-separable, only one subgroup will exist.

As mentioned before, some variables tend to stagnate in the later iterations of the algorithms, which causes others in the same subgroup to stop evolving. In other words, the subpopulation are typically trapped in local optima as the algorithms proceed. Therefore, if the stagnant variables can be identified and separated in advance, other interacting variables will be helped to jump out of the local optimum and speed up the convergence rate. Given a current optimal solution \mathbf{x}_{best} and a parent population \mathbf{P} , there are two criterion to determine whether the variable x_j is stagnant or not. The detailed formulation is shown as follows.

$$W_{s,j} = \begin{cases} 1, & \forall t, x_{t,j} = x_{best,j} \\ 1, & std_j < \epsilon \\ 0, & otherwise \end{cases} \quad (2)$$

where $t \in (1, 2, \dots, N)$, $x_{t,j}$ is the j th gene value of individual \mathbf{x}_t , $x_{best,j}$ is the j th gene value of individual \mathbf{x}_{best} , std_j is the standard deviation of the variable in dimension j for all individual, N is the population

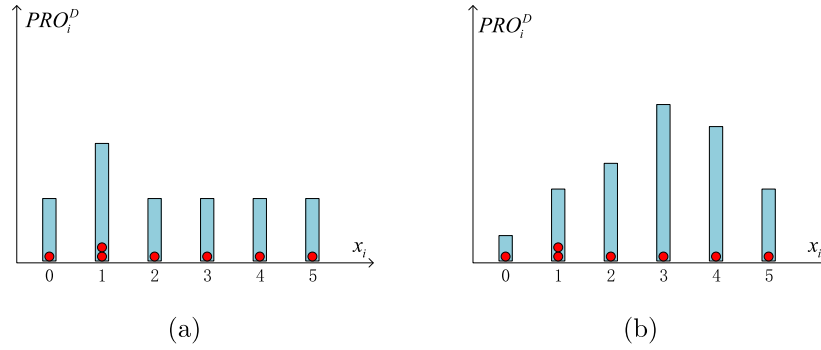


Fig. 1. Example of the fitness ranking strategy for the discrete variable x_i . (a) described the same effect of individuals for the probability model. (b) described the different effect of individuals for the probability model after fitness ranking.

size, ϵ is a threshold. If the j th gene value of each individual is equal to $x_{best,j}$ or to each other, the variable x_j is considered stagnant, and $W_{s,j} = 1$. Otherwise, $W_{s,j} = 0$. Noted that the judgment condition is too loose to isolate unstagnant variables, or it is difficult to find stagnant variables, so the top $N/2$ individuals after the fitness ranking are chosen to involve the judgment and the threshold ϵ is set to $1e-6$.

During the grouping process, DDG dynamically update grouping results by learning interaction and search status of variables in the current search area. Algorithm. 1 exhibits the detailed implementation of DDG. From line 2 to line 14, the interaction structure matrix and the stagnant vector are established. In line 15, a new set of grouping results C are obtained, and the variables judged to be stagnant of separable will be grouped separately, while other variables that are directly or indirectly interacting will be grouped in the same subgroup.

Algorithm 1: Dynamic differential grouping method(DDG)

Input: population size N , variable dimension D , the current best solution \mathbf{x}_{best} , the population P , Probabilistic models of variables Pr

Output: set of subgroups C

- 1 initialize the interaction structure matrix M_{in} , the stagnant vector W_s ;
- 2 **for** $i \leftarrow 1$ to D **do**
- 3 **for** $j \leftarrow i + 1$ to D **do**
- 4 $t_1, t_2 \leftarrow$ Sample from the probabilistic model of x_i ;
- 5 $t_3, t_4 \leftarrow$ Sample from the probabilistic model of x_j ;
- 6 **if** Eq. (1) is satisfied **then**
- 7 $M_{in}(i, j) \leftarrow 1$;
- 8 **else**
- 9 $M_{in}(i, j) \leftarrow 0$;
- 10 **end**
- 11 **end**
- 12 $P_{x_i} \leftarrow$ obtain the subpopulation of x_i from P ;
- 13 $W_{s,i} \leftarrow$ compute the search status of x_i using the Eq. (2), \mathbf{x}_{best} and P_{x_i} ;
- 14 **end**
- 15 $C \leftarrow$ separable or stagnant variables are grouped separately, and remaining variables are divided based on interaction.

3.2. Mixed-variable probabilistic models building of CoEDA_{mv}

The Bayesian networks and Gaussian models have been used to construct the probability models and have achieved promising results (Liang et al., 2018; Shahriari et al., 2015), but the performance of the constructed probabilistic models is affected by prior knowledge and control parameters. Based on the frequency distribution of

data, the histogram models are an efficient and non-parametric statistical method for analyzing the probability distribution. In addition, the histogram model is relatively simple to construct with high interpretability, which can reduce the difficulty of implementation. In CoEDA_{mv}, two histogram probability models are tailored to handle the discrete and continuous variables respectively. In this subsection, we introduce the fitness ranking strategy and these two models with the elite neighborhood-based updating probability strategy.

3.2.1. Fitness ranking strategy

In existing histogram probabilistic models (Molina Pérez et al., 2022; Wang et al., 2020; Zhou et al., 2015), each individual of the population has the same effect on the probabilistic model and offspring generation. They rarely make use of the historical information and decrease the convergence rate of the population.

CoEDA_{mv} designs the fitness ranking strategy (FR) to make a good use of the elite individuals. Specifically, the impact of each individual is not in the same for the histogram probabilistic models. On the contrary, the better the fitness of an individual, the higher its weight on updating the probabilistic model. In this paper, the maximum weight equals to population size N , the minimum weight equals to 1. For example, Fig. 1 described the effect of individuals for the discrete variable probability model Pr_i^D without and with the fitness ranking strategy. In Fig. 1(a), the number of the individuals located in the 1-th bin is highest, and the effect of the 1-th bin in Pr_i^D is largest. In Fig. 1(b), the value of the variable x_i in the top elite individual is 3, and the effect of the top elite individual in Pr_i^D is largest. After the fitness ranking strategy, the elite individuals have been located in the probability model, which can lead the population to learn from them and increase the search ability of CoEDA_{mv}.

3.2.2. Fitness rank based discrete histogram (FRDH) model

The FRDH model is proposed for handling discrete decision variables. By using the FRDH model, a histogram probability model Pr_i^D for a discrete variable x_i is established, in which the probability for each bin v of x_i , $v \in \{L_i, L_i + 1, \dots, U_i - 1, U_i\}$ is recorded as $Pr_{i,v}^D$, where $[L_i, U_i]$ is the boundary of variable x_i . At first, each bin of the discrete variable x_i should be considered with the equal probability. The initial probability is set as follows.

$$Pr_{i,v}^D(0) = \frac{1}{U_i - L_i + 1} \quad (3)$$

After that, the histogram model Pr_i^D will be updated as following steps. In the generation t_{gen} , the probability of the bin v for variable x_i is set as $Pr_{i,v}^D(t_{gen})$. Firstly, the sum of weight for individuals whose x_i value equals to v is calculated and defined as A_v . Secondly, sum all cases of A_k , where k can be any bin of x_i , and define it as A_{all} . Thirdly, the histogram model $Pr_{i,v}^D$ will be updated as follows.

$$Pr_{i,v}^D(t_{gen} + 1) = (1 - \gamma) \cdot Pr_{i,v}^D(t_{gen}) + \gamma \cdot \frac{A_v}{A_{all}} \quad (4)$$

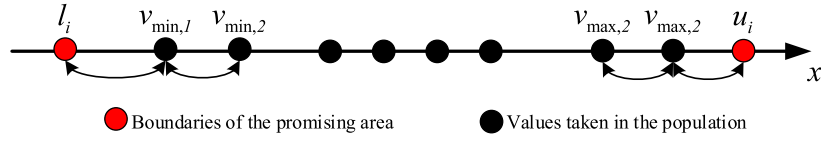


Fig. 2. Illustration of the boundary calculation of the existing model for the variable x_i .

Algorithm 2: FRDH model

Input: the fitness rank population P , population size N , the current generation t_{gen} , the maximum generation t_{max} , bounds $[L_i, U_i]$ of variable x_i , the current probability model $Pr_i^D(t_{gen})$

Output: the updated probability model $Pr_i^D(t_{gen} + 1)$

- 1 initialize the all-zero weight vector $A = [A_{L_i}, A_{L_i+1}, \dots, A_{U_i-1}, A_{U_i}]$ the maximum weight of any individual $weight_{max} = N$;
- 2 $\gamma \leftarrow$ update the learning rate by Eq. (6);
- 3 $R_i^D \leftarrow \max(\text{round}(U_i - L_i)/100, 1)$;
- 4 **for** $j \leftarrow 1$ to N **do**
- 5 $v \leftarrow$ the bin of the variable x_i in the j -th individual;
- 6 $A_v \leftarrow A_v + weight_{max} - j$;
- 7 **for** $k \leftarrow 1$ to R_i^D **do**
- 8 $A_{v+k} \leftarrow A_{v+k} + (weight_{max} - j) \cdot 0.05/R_i^D$;
- 9 $A_{v-k} \leftarrow A_{v-k} + (weight_{max} - j) \cdot 0.05/R_i^D$;
- 10 **end**
- 11 **end**
- 12 **for** $v \leftarrow L_i$ to U_i **do**
- 13 $Pr_{i,v}^D(t_{gen} + 1) \leftarrow$ update the probability by Eq. (4) and (5);
- 14 **end**
- 15 **return** $Pr_i^D(t_{gen} + 1)$;

$$A_{all} = \sum_{k=L_i}^{U_i} A_k \quad (5)$$

where the parameter γ represents the current population's learning rate and is defined as follows.

$$\gamma = 0.5 + (1 - \exp(\frac{-t_{max}}{t_{max} - t_{gen} + 1}))/2 \quad (6)$$

where t_{max} is the maximum number of generation, and t_{gen} is the current generation. As shown in Eq. (6), the minimum value of learning rate γ is set as 0.5, which avoids inefficiently updating the probabilistic model at the beginning. Then, γ will exhibit exponential growth to increase the search efficiency. As the current generation t_{gen} increases, FRDH will increasingly employ information from elite individuals to update Pr_i^D .

In addition, the FRDH model will be updated with the elite neighborhood-based updating probability strategy to help generate good quality offspring, in which the weight for both the current bin and its neighbor bin will be increased. In other words, if the variable x_i takes the value as v , the probability $Pr_{i,v}^D$ will be updated by Eqs. (4) to (6). Simultaneously, the updating probability of its neighbor bins, i.e. $x_i = (v - R_i^D, \dots, v - 1, v + 1, v + R_i^D)$, will be effected. Then, R_i^D represents the scope of the neighborhood. The pseudocode for updating the FRDH model is summarized in Algorithm. 2. The weight vector A of a discrete variable x_i is recalculated from line 5 to line 11. Each weight A_v consists of two parts, i.e. the weight sum for individuals whose x_i value equals to v , and the weight sum for individuals whose x_i value neighbor around v . Note that the better the individual's fitness, the greater the weight for the corresponding bin. From line 12 to line 15, the probability for each available discrete value of x_i is updated.

3.2.3. Fitness rank based continuous histogram (FRCH) model

Inspired by Wang et al. (2020), the FRCH model is proposed for handling continuous variables. In contrast to AWH model (Wang et al., 2020), the FRCH model will be updated by taking advantage of the good quality individuals and the elite neighborhood-based updating probability strategy, while the global search area $[L_i, U_i]$ and promising search area $[l_i, u_i]$ of continuous variable x_i can be adjusted adaptively.

In the AWH model (Wang et al., 2020), the promising search area $[l_i, u_i]$ is located from the global search area $[L_i, U_i]$ of the continuous variable x_i . All individuals are contained in the promising area, in which the W bins have the same width. Fig. 2 introduces how to determine the promising area for the variable x_i in the current population, in which $v_{min,1}$ and $v_{min,2}$ are the first and second smallest values, and $v_{max,1}$ and $v_{max,2}$ are the first and second largest values of variable x_i respectively. The new boundaries $[l_i, u_i]$ are evaluated based on $v_{min,1}$, $v_{min,2}$, $v_{max,1}$ and $v_{max,2}$. However, it is not enough to consider the maximum and minimum available values of x_i . Searching around the best values v_{best} of the variable x_i in the x_{best} , is more meaningful and beneficial in sampling for better solutions. As shown in Fig. 3, the FRCH model additionally references x_{best} to calculate the boundaries of the promising area. Here, we calculate $[l_i, u_i]$ as Eq. (7).

$$\begin{cases} c_1 = v_{max,1} - v_{best} \\ c_2 = v_{best} - v_{min,1} \\ c_3 = v_{max,1} - v_{min,1} \\ u_i = \min\{U_i, v_{max,1} + (v_{max,1} - v_{max,2}) \cdot c_2/c_3\} \\ l_i = \max\{L_i, v_{min,1} + (v_{min,2} - v_{min,1}) \cdot c_1/c_3\} \end{cases} \quad (7)$$

The search space $[L_i, U_i]$ of the variable x_i is divided into $(W_{column} + 2)$ bins. Specifically, the promising search area $[l_i, u_i]$ can be represented by W_{column} equal-width bins, in which all individuals are contained. In addition, there are no individuals on either side of the bins. The width of each bin W_{width} in the promising area is calculated by Eq. (8). Then, the probability model $Pr_{i,v}^C$ of the v th bin for the continuous variable x_i will be estimated by the following steps. Firstly, sum the weight of the individuals whose value of x_i is in the v th bin and set it as A_v . Secondly, sum all cases of A_v , where $v \in \{1, 2, \dots, W_{column} + 2\}$. Then the probability model $Pr_{i,v}^C$ updating rule is shown in Eq. (9).

$$W_{width} = \frac{1}{W_{column}}(u_i - l_i) \quad (8)$$

$$Pr_{i,v}^C = \frac{A_v}{\sum_{k=1}^{W_{column}+2} A_k} \quad (9)$$

Noted that the population tend to the convergence in the later iterations of the algorithm. The promising search area will become smaller, and the search areas at both sides with low sampling probabilities, i.e. $[L_i, l_i]$ and $[u_i, U_i]$, become larger. This situation makes it difficult for a global search strategy to be effective. Therefore, the FRCH model selects a portion of the global search area and updates the boundaries of the variable x_i . The details are given in Eq. (10).

$$\begin{cases} U_i = \min(U_i, u_i + \frac{U_i - l_i}{100}) \\ L_i = \max(L_i, l_i - \frac{U_i - L_i}{100}) \end{cases} \quad (10)$$

The pseudocode of the FRCH model is summarized in Algorithm. 3. From line 5 to line 9, the weight vector A of each bin is recalculated.

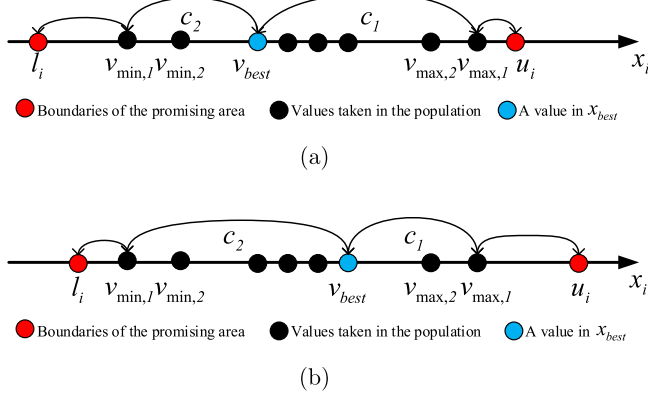


Fig. 3. Illustration of the boundary calculation of the FRCH model for the variable x_i . (a) and (b) described the effects of different v_{best} on u_i and l_i . When v_{best} is close to $v_{max,1}$ or $v_{min,1}$, the FRCH model adjusts the boundaries of the promising area to focus on the search area around v_{best} .

Algorithm 3: FRCH model

Input: the fitness rank population P , population size N , the global search boundaries $[L_i, U_i]$ of variable x_i , the number of bins W_{column}

Output: the updated probability model Pr_i^C , the updated global search boundaries $[L_i, U_i]$

- 1 initialize the all-zero weight vector $A = [A_1, A_2, \dots, A_{W_{column}+2}]$, the maximum weight of any individual $weight_{max} = N$;
- 2 $[l_i, u_i] \leftarrow$ calculate the boundaries of the promising area by Eq. (7);
- 3 $W_{width} \leftarrow$ calculate the bin width of the promising area by Eq. (8);
- 4 $[L'_i, U'_i] \leftarrow$ calculate the boundaries of the global area by Eq. (10);
- 5 **for** $j \leftarrow 1$ to N **do**
- 6 $k \leftarrow$ the bin of the variable x_i in the j -th individual;
- 7 $v \leftarrow \lfloor (k - l_i) / W_{width} \rfloor + 2$;
- 8 $A_v \leftarrow A_v + weight_{max} - j$;
- 9 **end**
- 10 **for** $v \leftarrow 1$ to $W_{column} + 2$ **do**
- 11 $Pr_{i,v}^C \leftarrow$ update the probability model by Eq. (9);
- 12 **end**
- 13 $[L_i, U_i] \leftarrow [L'_i, U'_i]$;

Parameter v in line 7 means the value k falling into the v th bin. Note that the better the individual's fitness, the greater the weight for the corresponding bin. From line 10 to line 12, the probability model Pr_i^C of the variable x_i is updated by Eq. (9).

3.3. The framework of CoEDA_{mv}

Based on the methods introduced previously, the framework of CoEDA_{mv} algorithm can be presented in Algorithm. 4. From line 3 to line 5, CoEDA_{mv} updates the grouping results regularly. In line 6, CoEDA_{mv} obtains the subgroup C_{sub} and the corresponding subpopulation P_{sub} to undergo evolution by using the subgroup selection strategy proposed in CCFR2 (Yang, Zhou, Li, Guan et al., 2020). The reason is that this strategy has shown promising performance in terms of efficient allocation of fitness evaluations and reduction of computational costs. From line 7 to line 13, the specific histogram probability

Algorithm 4: Framework of CoEDA_{mv}

Input: the population size N , the variable dimension D , the boundary $[U, L]$, the maximum generation t_{max} , the dynamic grouping interval t_{group} , the number of bins W_{column}

Output: the optimal solution x_{best}

- 1 initialize the current generation t_{gen} , the population P , the current optimal solution x_{best} , the probability models Pr^D, Pr^C for the discrete variables and the continuous variables respectively;
- 2 **for** $t_{gen} \leftarrow 1$ to t_{max} **do**
- 3 **if** $\text{mod}(t_{gen}, t_{group}) = 1$ **then**
- 4 $C \leftarrow$ update grouping results by Algorithm. 1;
- 5 **end**
- 6 obtain the selected subgroup C_{sub} and the corresponding subpopulation P_{sub} ;
- 7 **for** $x_i \leftarrow C_{sub}(0)$ to $C_{sub}(\text{end})$ **do**
- 8 **if** x_i is a discrete variable **then**
- 9 $Pr_i^D \leftarrow$ update the FRDH model by Algorithm. 2;
- 10 **else**
- 11 $Pr_i^C \leftarrow$ update the FRCH model by Algorithm. 3;
- 12 **end**
- 13 **end**
- 14 $Q_{sub} \leftarrow$ generate N new individuals by sampling from the updated probability models;
- 15 $P_{sub} \leftarrow$ select the top N optimal individuals from $P_{sub} \cup Q_{sub}$;
- 16 update the optimal solution x_{best} and the contribution of C_{sub} ;
- 17 $t_{gen} \leftarrow t_{gen} + 1$;
- 18 **end**

models of the continuous and discrete decision variables grouped into the subgrouping C_{sub} are updated by Algorithm. 2 and 3 respectively. In addition, novel candidate solutions are generated in line 14 by sampling from the two histogram probability models. Finally, update the parent population P_{sub} and the current optimal solution x_{best} . To provide a clear overview of CoEDA_{mv}, its general flow diagram is distinctly depicted in Fig. 4.

As presented in Algorithm. 4, there are four main components in the framework of the proposed CoEDA_{mv}, including the dynamic differential grouping method in line 4, the FRDH model in line 9, the FRCH model in line 11 and the offspring generation in line 14. The computational complexity of CoEDA_{mv} can be obtained by analyzing these main components. At first, in order to generate a new set of subgroups, the dynamic differential grouping method processed in Algorithm. 1 will analysis the interaction and search status of decision variables. This requires the time complexity of $O(D^2)$, where D denotes the number of decision variables. Secondly, the FRDH model processed in Algorithm. 2 will update the probabilistic model of each discrete variable. This needs the time complexity of $O(N \times (U - L))$, where N , U and L stand for the number of individuals, the upper bound and the lower bound of each discrete variable, respectively. Thirdly, the FRCH model processed in Algorithm. 3 will update the probabilistic model of each continuous variable. This requires the time complexity of $O(N + W_{column})$, where W_{column} is the numbers of bins. To be specific, updating the probability models of mixed variables processed in line 7–13 in Algorithm. 4 requires the time complexity of $O((U - L) \times N \times D)$. Lastly, the offspring generation processed in line 14 in Algorithm. 4 needs the time complexity of $O(N \times D)$ by sampling from the updated probability models. Summarily, the overall time complexity of CoEDA_{mv} in one generation is $O(D^2 + N \times D)$.

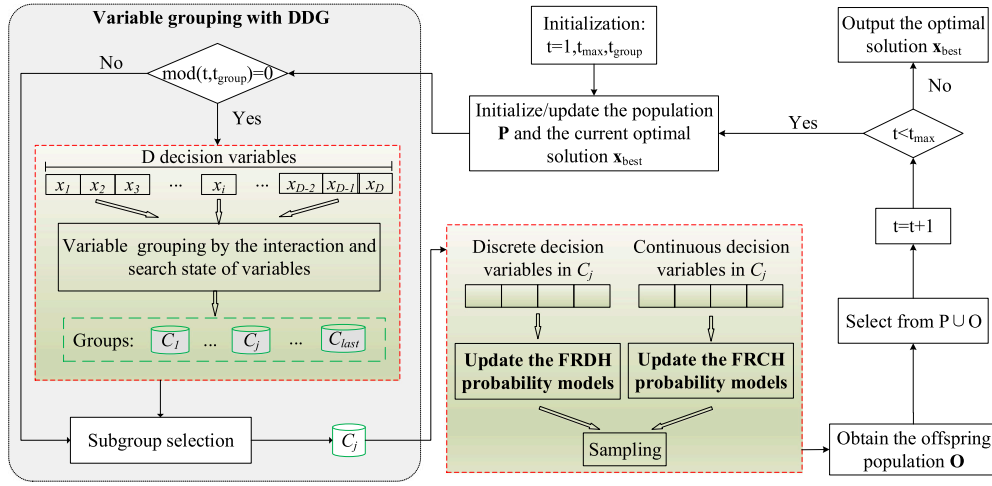


Fig. 4. General flow diagram of CoEDA_{mv}, including three main components: DDG, FRDH for discrete variables, and FRCH for continuous variables.

4. Experimental studies

In this section, the performance of CoEDA_{mv} is evaluated on 28 artificial MVOPs specifically for MVOPs. At first, we compare the CoEDA_{mv} with eight state-of-the-art algorithms to further validate its efficiency. Then, we design three ablation experiments to analysis the effectiveness of the proposed strategies in CoEDA_{mv}.

4.1. Experimental setup

In this paper, we use 28 artificial MVOPs which has been widely used as the test suite for MVOPs, and details of these functions can be found in Wang et al. (2021), which are developed from the CEC2013 (Liang et al., 2013). The decision variables in 28 artificial MVOPs comprise both continuous and discrete variables. The total number of mixed variables is set to D , which includes the Z -dimensional continuous variables and the V -dimensional discrete variables, i.e. $D = Z + V$. In 28 artificial MVOPs, there are five unimodal functions (f_1 to f_5). Then, there are 15 multimodal functions (f_6 to f_{20}). And the rest 8 functions (f_{21} to f_{28}) are composition functions including more complex features than the first 20 functions. Meanwhile, there are four functions (f_1, f_5, f_{11}, f_{14}) are separable and the rest of these functions are inseparable.

To verify the efficiency of the proposed CoEDA_{mv}, eight state-of-the-art algorithms are selected to compare with CoEDA_{mv}, i.e. EDA_{mvn} (Wang et al., 2020), EDA_{mv} (Molina Pérez et al., 2022), PSO_{mv} (Wang et al., 2021), AEDA (Shi et al., 2017), DE_{mv} (Lin et al., 2018), MSEFA (Peng et al., 2022), NOA (Abdel-Basset et al., 2023), SLPSO (Cheng & Jin, 2015). In this paper, we employed the first 5 algorithms to solve MVOPs directly, the last 2 algorithms are tailored for solving MVOPs by transferring discrete variables into continuous variables with relaxation, and the remaining algorithm is adopted for solving MVOPs by transferring continuous variables into discrete variables with discretization.

In this paper, we conduct 30 independent runs to ensure the generation of more reliable experimental results through statistical analysis. To provide fair comparisons, the maximum number of fitness function evaluations (maxFEs) is set as 3.6×10^5 for all involved algorithms. In 28 artificial MVOPs, there are a total of 50 decision variables, including 25 continuous variables and 25 discrete variables.

The parameters of the compared algorithms are described in Table 1, where N uniformly represents the number of individuals and t_{max} stands for the maximum number of iterations. Except for the proposed CoEDA_{mv}, it is guaranteed that the product of N and t_{max}

Table 1

The parameter settings for each algorithms.

Algorithms	Parameter settings
CoEDA _{mv}	$N = 300, t_{max} = 1000, t_{group} = 200, W_{column} = 10$
EDA _{mvn}	$N = 300, t_{max} = 1200, W_{column} = 10$
EDA _{mv}	$N = 50, t_{max} = 7000, W_{column} = 4, e_b = 2.3959, cp = 8$
PSO _{mv}	$N = 30, t_{max} = 12000, w = 0.4, c = 2.0, \alpha = 0.5$
DE _{mv}	$N = 30, t_{max} = 12000, F = 0.2, CR = 0.2$
AEDA	$N = 300, t_{max} = 1200, NP_{best} = 0.45, g_p = 0.5$
MSEFA	$N = 30, t_{max} = 12000, \alpha = 0.5, \gamma = 1, \beta_0 = 1, \beta_{min} = 0.2$
NOA	$N = 30, t_{max} = 12000, \alpha = 0.05, Pa = 0.2, Prb = 0.2$
SLPSO	$N = 30, t_{max} = 12000, w = 0.4, \alpha = 0.5, \beta = 0.01$

is equal to maxFEs. That is because that the variable grouping strategy of CoEDA_{mv} will consume some fitness evaluations. In addition, t_{group} is the dynamic grouping interval, W_{column} is the number of bins in the histogram probabilistic models. In EDA_{mv}, e_b is a small value to avoid premature convergence, and cp is a parameter to control the speed of constraint relaxation. In PSO_{mv} and SLPSO, w is the inertia weight, α is used to balance historical search information and current search information, c is the acceleration coefficients for learning toward the personal best position of a particle, and β is set as a small value to avoid premature convergence. In DE_{mv}, F represents the scaling factor during the mutation operation, and CR represents the crossover probability. In AEDA, the best NP_{best} individuals in the current population are used to estimate the probability models, and g_p is the probability of choosing the Gaussian distribution to sample new population. In MSEFA, α is the step length factor, β_0 and β_{min} are the initial and minimum attraction when the value of Euclidean distance between two individuals is 0. In NOA, α stands for the percent of attempts at avoiding local optima, Pa stands for the probability of exchanging between the cache of search stage and the recovery stage, and Prb stands for the percentage of exploration other regions within the search space.

4.2. Results comparison on benchmark functions

In order to validate the efficiency and performance of the proposed CoEDA_{mv}, eight start-of-the-art algorithms are selected for comparisons, and the numerical results on 28 artificial MVOPs are listed in Table 2. For each function, there are two statistical values that serve as indicators of comparison between different algorithms. The first one, MEAN, describes the average of the optimal solutions obtained by running the algorithm independently 30 times, and the smaller the indicator MEAN the better the algorithm's solution performance. The second one,

Table 2
Experimental results on 28 artificial MVOPs.

		CoEDA _{mv}	EDA _{mv}	EDA _{mv}	PSO _{mv}	AEDA	DE _{mv}	MSEFA	NOA	SLPSO
F1	MEAN	0.00E+00	9.47E+00(+)	7.06E+01(+)	2.12E-13(+)	1.75E+01(+)	4.48E+01(+)	1.40E+03(+)	4.53E-02(+)	3.31E+01(+)
	STD	0.00E+00	3.87E+00	3.27E+01	3.11E-13	1.81E+02	3.87E+01	1.75E-04	2.04E-02	6.51E+01
F2	MEAN	5.86E+06	6.36E+06(=)	5.27E+07(+)	8.77E+06(+)	1.54E+08(+)	5.26E+07(+)	1.45E+06(-)	3.49E+06(-)	1.05E+07(+)
	STD	1.44E+06	1.72E+06	1.26E+07	2.30E+07	5.89E+08	4.32E+07	4.45E+05	1.11E+08	4.10E+06
F3	MEAN	2.23E+07	4.28E+09(+)	2.18E+10(+)	1.99E+07(+)	8.56E+09(+)	5.68E+08(+)	4.88E+10(+)	1.22E+09(+)	1.03E+09(+)
	STD	1.03E+08	1.45E+09	8.73E+09	8.28E+07	2.36E+10	1.02E+09	8.54E+10	8.38E+08	1.15E+09
F4	MEAN	2.64E+04	4.85E+04(+)	2.61E+04(=)	3.97E+03(-)	8.85E+04(+)	5.38E+04(+)	1.89E+03(-)	8.29E+02(-)	3.31E+04(+)
	STD	4.58E+03	5.34E+03	7.71E+03	2.14E+04	5.52E+04	3.36E+04	2.49E+02	2.53E+05	7.38E+03
F5	MEAN	9.53E-42	5.89E+00(+)	2.62E+01(+)	3.33E-02(+)	4.20E+02(+)	2.85E+01(+)	1.00E+03(+)	1.00E+02(+)	4.72E+01(+)
	STD	2.64E-41	1.71E+00	2.24E+01	9.83E-01	4.01E+03	1.90E+01	2.71E-04	2.50E-01	4.41E+01
F6	MEAN	5.42E+01	6.19E+01(+)	1.61E+02(+)	1.18E+02(+)	2.18E+02(+)	1.41E+02(+)	9.54E+02(+)	1.18E+02(+)	1.78E+02(+)
	STD	2.61E-01	3.51E+00	5.17E+01	1.99E+02	1.79E+02	1.93E+02	4.57E-01	2.57E+01	4.83E+01
F7	MEAN	2.96E+01	3.20E+01(=)	1.15E+02(+)	4.31E+01(+)	7.81E+01(+)	1.18E+02(+)	8.42E+02(+)	8.40E+01(+)	4.93E+01(+)
	STD	9.39E+00	6.90E+00	2.49E+01	6.67E+01	6.59E+01	6.25E+01	2.75E+00	5.38E+00	1.33E+01
F8	MEAN	2.11E+01	2.11E+01(=)	2.11E+01(=)	2.11E+01(-)	2.11E+01(=)	2.11E+01(=)	7.21E+02(+)	2.11E+01(=)	2.11E+01(=)
	STD	3.55E-02	4.11E-02	3.44E-02	2.49E-01	2.32E-01	1.86E-01	1.51E-01	6.64E-02	7.63E-02
F9	MEAN	1.52E+01	6.94E+01(+)	3.66E+01(+)	3.54E+01(+)	6.97E+01(+)	5.10E+01(+)	6.04E+02(+)	6.49E+01(+)	8.16E+01(+)
	STD	1.75E+01	1.38E+00	3.68E+00	3.50E+01	1.11E+01	1.38E+01	6.02E+00	4.23E+00	2.89E+00
F10	MEAN	8.62E+00	1.67E+01(+)	3.75E+02(+)	1.27E+01(+)	3.17E+01(+)	1.56E+02(+)	5.01E+02(+)	5.05E+01(+)	6.51E+01(+)
	STD	2.02E+01	5.73E+00	1.43E+02	2.53E+01	1.22E+02	1.53E+02	7.26E-02	1.77E+00	2.31E+01
F11	MEAN	1.03E-01	5.71E+00(+)	1.24E+02(+)	1.55E+01(+)	1.49E+02(+)	1.02E+02(+)	4.26E+02(+)	1.45E+02(+)	8.77E+01(+)
	STD	3.14E-01	2.27E+00	5.15E+01	2.44E+01	5.74E+01	4.30E+01	3.72E+00	2.20E+02	1.42E+01
F12	MEAN	3.31E+02	3.19E+02(-)	3.60E+02(+)	1.77E+02(-)	3.59E+02(+)	4.14E+02(+)	3.28E+02(=)	2.85E+02(-)	1.29E+02(-)
	STD	9.35E+00	1.01E+01	8.64E+01	2.36E+02	8.00E+01	1.77E+02	7.33E+01	4.00E+02	3.32E+01
F13	MEAN	3.27E+02	3.22E+02(-)	4.06E+02(+)	3.05E+02(-)	3.56E+02(+)	4.72E+02(+)	2.70E+02(-)	4.21E+02(+)	2.46E+02(-)
	STD	1.56E+01	1.23E+01	6.74E+01	3.28E+02	9.86E+01	1.49E+02	1.06E+02	4.39E+02	4.82E+01
F14	MEAN	6.82E-01	1.12E+03(+)	3.30E+03(+)	5.50E+02(+)	6.06E+03(+)	1.25E+03(+)	1.89E+03(+)	2.20E+03(+)	3.25E+03(+)
	STD	1.34E+00	7.85E+02	1.38E+03	1.35E+03	1.91E+03	7.29E+02	1.05E+03	3.52E+03	6.67E+02
F15	MEAN	1.33E+04	1.32E+04(=)	1.35E+04(+)	7.19E+03(-)	1.37E+04(+)	1.09E+04(-)	2.06E+03(=)	7.65E+03(-)	6.00E+03(-)
	STD	3.05E+02	3.96E+02	3.32E+02	8.40E+03	1.94E+03	3.02E+03	3.53E+02	8.10E+03	9.14E+02
F16	MEAN	3.52E+00	3.38E+00(-)	3.41E+00(=)	2.83E+00(-)	3.36E+00(-)	3.27E+00(-)	3.05E-01(-)	2.01E+00(-)	1.12E+00(-)
	STD	2.24E-01	3.06E-01	3.43E-01	1.70E+00	1.48E+00	1.74E+00	5.46E-01	2.03E-01	4.57E-01
F17	MEAN	3.18E+01	1.84E+02(+)	2.97E+02(+)	4.38E+01(+)	5.18E+02(+)	1.42E+02(+)	9.81E+02(+)	1.80E+02(+)	1.62E+02(+)
	STD	3.57E+01	2.78E+01	1.10E+02	2.31E+02	1.00E+02	5.87E+01	5.27E+02	1.23E+02	3.23E+01
F18	MEAN	3.55E+02	3.56E+02(=)	4.80E+02(+)	3.85E+02(+)	4.96E+02(+)	5.35E+02(+)	7.74E+02(+)	2.09E+02(-)	2.72E+02(-)
	STD	2.83E+01	2.54E+01	9.66E+01	3.79E+02	1.28E+02	1.93E+02	1.55E+02	3.20E+03	3.80E+01
F19	MEAN	2.95E+01	2.87E+01(-)	3.68E+01(=)	2.24E+01(-)	3.39E+01(+)	3.89E+01(+)	8.05E+00(-)	5.27E+01(+)	1.61E+01(-)
	STD	1.41E+00	1.01E+00	3.02E+01	3.25E+01	9.11E+00	1.45E+01	4.79E+00	7.23E+00	4.42E+00
F20	MEAN	2.15E+01	2.17E+01(+)	2.16E+01(=)	2.16E+01(=)	2.34E+01(+)	2.37E+01(+)	2.50E+01(+)	2.48E+01(+)	2.24E+01(+)
	STD	7.27E-01	3.68E-01	7.37E-01	8.76E+00	3.96E+00	2.17E+00	1.42E+00	1.03E+00	1.70E+00
F21	MEAN	2.00E+02	2.01E+02(+)	2.07E+02(+)	2.01E+02(+)	2.01E+02(+)	2.04E+02(+)	2.26E+02(+)	2.02E+02(+)	2.04E+02(+)
	STD	1.24E-04	2.68E-01	4.27E+00	3.77E+00	1.62E+01	3.86E+00	1.97E-02	5.19E-02	7.18E+00
F22	MEAN	1.01E+02	8.79E+02(+)	3.29E+03(+)	6.00E+02(+)	6.12E+03(+)	1.38E+03(+)	4.57E+03(+)	3.13E+03(+)	3.22E+03(+)
	STD	9.73E-01	7.32E+02	1.22E+03	1.19E+03	2.31E+03	7.45E+02	5.47E+02	5.16E+03	6.65E+02
F23	MEAN	1.34E+04	1.34E+04(=)	1.35E+04(=)	1.23E+04(-)	1.66E+04(+)	1.49E+04(+)	3.82E+03(-)	8.16E+03(-)	6.03E+03(-)
	STD	3.53E+02	3.47E+02	3.76E+02	4.31E+03	1.14E+03	1.44E+03	6.74E+02	8.81E+03	1.14E+03
F24	MEAN	1.21E+03	1.02E+03(-)	1.22E+03(=)	9.82E+02(-)	1.43E+03(+)	1.50E+03(+)	2.07E+02(-)	1.34E+03(+)	8.29E+02(-)
	STD	9.14E+01	7.81E+01	2.71E+02	5.84E+02	2.76E+02	2.62E+02	4.87E+02	9.30E+02	7.80E+01
F25	MEAN	4.57E+02	4.40E+02(-)	4.97E+02(+)	4.33E+02(-)	5.22E+02(+)	5.41E+02(+)	2.97E+02(-)	5.12E+02(+)	4.17E+02(-)
	STD	1.84E+01	1.58E+01	2.19E+01	9.35E+01	9.71E+01	6.00E+01	4.51E+01	1.46E+02	2.18E+01
F26	MEAN	7.18E+02	4.99E+02(-)	2.14E+03(+)	1.10E+03(+)	6.32E+02(=)	1.48E+03(+)	2.79E+02(-)	1.60E+03(+)	1.02E+03(+)
	STD	3.25E+02	5.91E+01	6.00E+02	1.02E+03	9.99E+02	7.20E+02	6.88E+02	1.23E+02	2.17E+02
F27	MEAN	4.72E+03	4.86E+03(+)	5.16E+03(+)	5.01E+03(+)	5.13E+03(+)	5.23E+03(+)	4.79E+03(+)	5.52E+03(+)	5.15E+03(+)
	STD	1.06E+02	7.82E+01	1.41E+02	5.97E+02	3.74E+02	3.29E+02	4.97E+02	9.81E+03	1.56E+02
F28	MEAN	1.03E+04	9.95E+03(=)	1.05E+04(=)	7.58E+03(-)	1.19E+04(+)	1.06E+04(=)	4.08E+02(-)	6.29E+03(-)	6.31E+03(-)
	STD	1.35E+03	1.60E+03	1.39E+03	5.41E+03	1.30E+03	1.54E+03	7.89E+02	2.15E+04	9.42E+02
+/-/-			14/7/7	20/8/0	16/1/11	25/2/1	24/2/2	16/2/10	19/1/8	17/1/10

STD, describes the standard deviation of the optimal solutions after 30 independent runs, and the smaller the indicator STD, the better the algorithm's stability.

In our experimental study, we employ the rank sum test to examine the performance of the proposed CoEDA_{mv} against other algorithms for each function. The performance statistics are listed in the last row of Table 2. When the performance of CoEDA_{mv} is significantly outstanding, similar, or significantly worse than that of the other comparison algorithms, a notation '+', '=', or '-' respectively will be marked behind the corresponding algorithm. Then, the dark color and the lighter color are emphasized to distinguish between optimal and sub-optimal results.

From Table 2, we can observe that CoEDA_{mv} obtains the optimal result on 14 out of 28 functions, including 2 unimodal functions(f_1 , f_5),

9 multi functions(f_6 , f_7 , f_8 , f_9 , f_{10} , f_{11} , f_{14} , f_{17} , f_{20}) and 3 composition functions(f_{21} , f_{22} , f_{27}). As for the other functions, CoEDA_{mv} obtains the sub-optimal result on 3 out of 15 functions, including 2 unimodal function(f_2 , f_3), 1 multi functions(f_8). EDA_{mv}, AEDA and DE_{mv} exhibit slightly inferior performance compared to the other algorithms under consideration, and these three algorithms demonstrate comparable efficiency with CoEDA_{mv} only on a few functions. The reason is that, in order to increase the ability of handling continuous and discrete variables separately, CoEDA_{mv} utilizes two different histogram models with both the fitness ranking strategy and the elite neighborhood-based updating probability strategy. When updating the probabilistic models, CoEDA_{mv} will assign different impact weights to different individuals based on the ranking of their fitness. Compared with recent

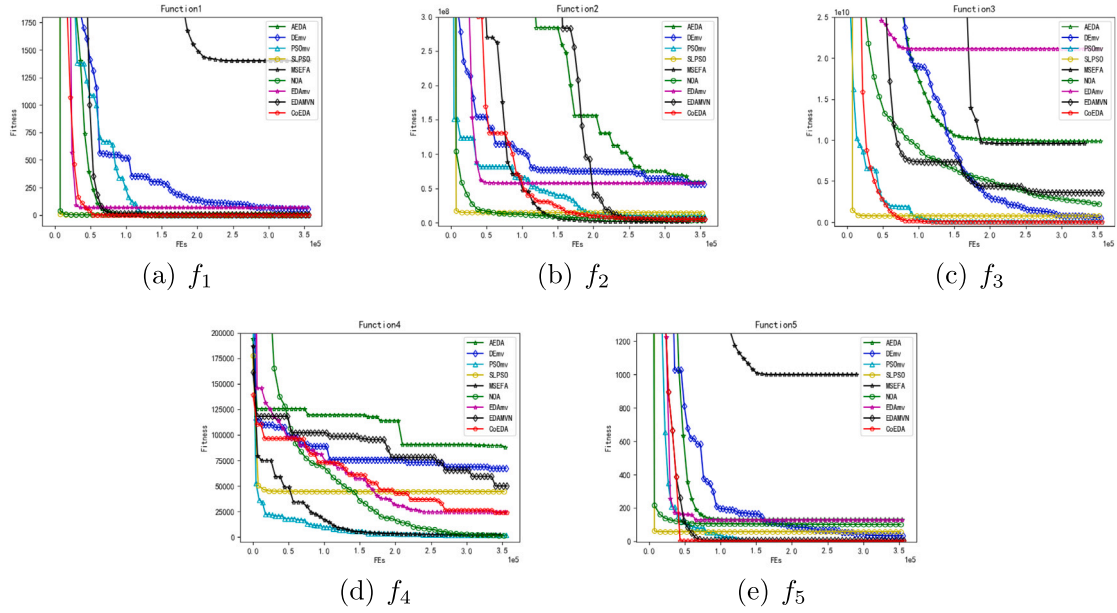


Fig. 5. Convergence process on 5 unimodal benchmarks.

metaheuristics, i.e. EDA_{mun} , PSO_{mu} , $MSEFA$ and NOA , $CoEDA_{mu}$ shows better performance on most functions, and comparable or even lower efficiency on a few functions. Considering the variable interaction of benchmark functions, $CoEDA_{mu}$ achieves the best results than the other algorithms on 4 separable functions (f_1, f_5, f_{11}, f_{14}), and the competitive results on the remaining benchmark functions. The reason is that the $CoEDA_{mu}$ takes the interaction and the convergence of variables on the current search region into consideration throughout its iterations. For non-separable problems, the fitness evaluations can be allocated more to the non-converged variables after considering the convergence. Meanwhile, for separable problems, the dynamic grouping strategy can be used to partition the problem during the algorithm's operation process to obtain better results. These statistical results indicate that $CoEDA_{mu}$ exhibits outstanding performance on the majority of functions.

Figs. 5–7 exhibit the convergence of all algorithms in 5 unimodal benchmarks, 15 multimodal benchmarks and 8 composition benchmarks, respectively. From the figure, it can be seen that $CoEDA_{mu}$ does not occupy an absolute advantage in the optimization speed of the algorithm, which is due to the fact that $CoEDA_{mu}$ groups the variables and optimizes only some variables at a time, resulting in a slower optimization speed of the function. However, $CoEDA_{mu}$ improves the histogram probability models and proposes two different probability models (FRCH and FRDH) based on the strategy of the ranking fitness, which improves the algorithm's convergence ability and global search ability. Therefore, the $CoEDA_{mu}$ is still a competitive and efficient evolution algorithm to deal with mixed-variable optimization problems.

4.3. Ablation study

In this subsection, we design the ablation experiments to analyze the effectiveness of the involved methods (DDG and improved histogram-based probabilistic models) in $CoEDA_{mu}$. In Specific, we have provided 6 combinations to verify the performance of each contribution. The first one is the original $CoEDA_{mu}$. Second, to verify the proposed DDG, we compare $CoEDA_{mu}$ with an algorithm using static grouping strategy (named as $CoEDA_{mu}$ -DDG), in which the decision variables are divided into multiple subgroups using a fixed decomposition method and the subgroups remain unchanged during the whole optimization process. Third, to verify the effectiveness of the histogram-based probabilistic

models (HPM), we compare $CoEDA_{mu}$ with an algorithm using Gaussian mixture module (GMM) instead of the HPM (named as $CoEDA_{mu}$ -HPM). The fourth one is an algorithm using static grouping strategy and GMM (named as $CoEDA_{mu}$ -DDG-HPM). Finally, to verify the two improved histogram probabilistic models designed for continuous variables and discrete variables, we compare the performance of $CoEDA_{mu}$ with an algorithm not using the FRDH probabilistic model but the LBH probabilistic model (Wang et al., 2020) for discrete variables (named as $CoEDA_{mu}$ -FRDH), and not using the FRCH probabilistic model but the AWH probabilistic model (Wang et al., 2020) the for continuous variables (named as $CoEDA_{mu}$ -FRCH), respectively.

The ablation experimental results on 28 artificial MVOPs are shown in Table 3. To assess the significant difference between two sets of results, a rank sum test is applied with a significance level of $\alpha = 0.05$. The optimal result is distinguished with a deep shade, whereas the second-best result is indicated using a lighter shade. The performance statistics can be found in the last line of Table 3.

Table 3 shows the results of ablation study. From the results of $CoEDA_{mu}$ and $CoEDA_{mu}$ -DDG, we can see that the DDG get better results on 8 functions and similar results on 20 functions. From the results of $CoEDA_{mu}$ -HPM and $CoEDA_{mu}$ -DDG-HPM, we notice that the DDG can improve the convergence accuracy on 5 functions, while obtain similar performance on 20 functions. It means that, compared to the static grouping method, the DDG is helpful to get better results. That is because that the DDG updates the groupings dynamically by the learning the interactions and search status among variables, whereas the results of the static grouping will not change during the evolution process.

From the results of $CoEDA_{mu}$ and $CoEDA_{mu}$ -HPM, it is obvious that HPM achieves the improvement of convergence accuracy on 25 functions. And from the results of $CoEDA_{mu}$ -DDG and $CoEDA_{mu}$ -DDG-HPM, the HPM is efficient to enhance the convergence accuracy on 24 functions. The reason can be concluded that the HPM does not require any prior information about the distribution of solutions, which can learn the global distribution of landscapes and lead to a better solution. Noted that the FRDH and the FRCH can obtain better solutions on 16 functions and 9 functions respectively, due to the guidance of elite individuals and adaptive probabilistic updating.

Finally, We can confirm that the involved methods are helpful and effective to improve the search ability of the $CoEDA_{mu}$, in particular,

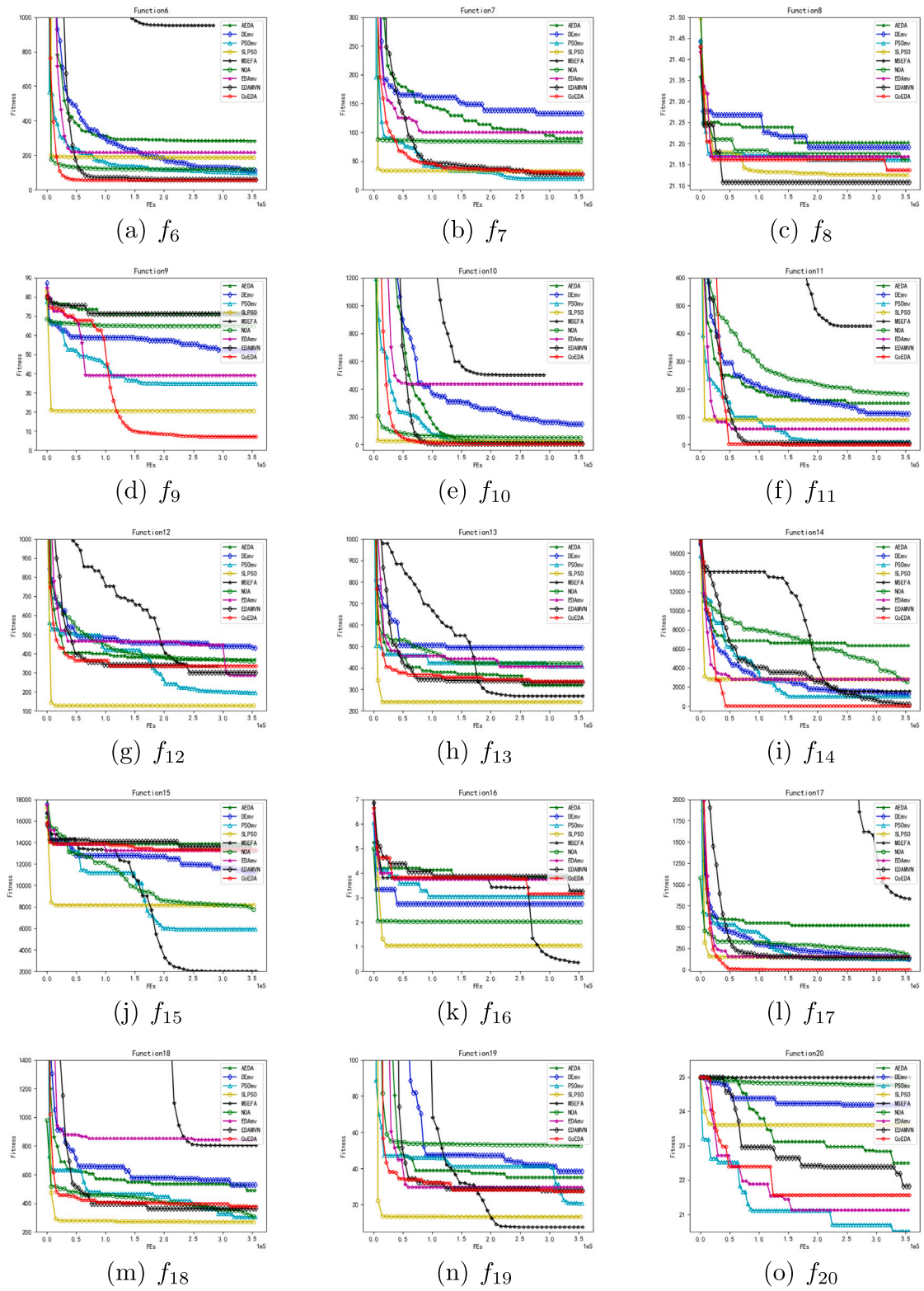


Fig. 6. Convergence process on 15 multimodal benchmarks.

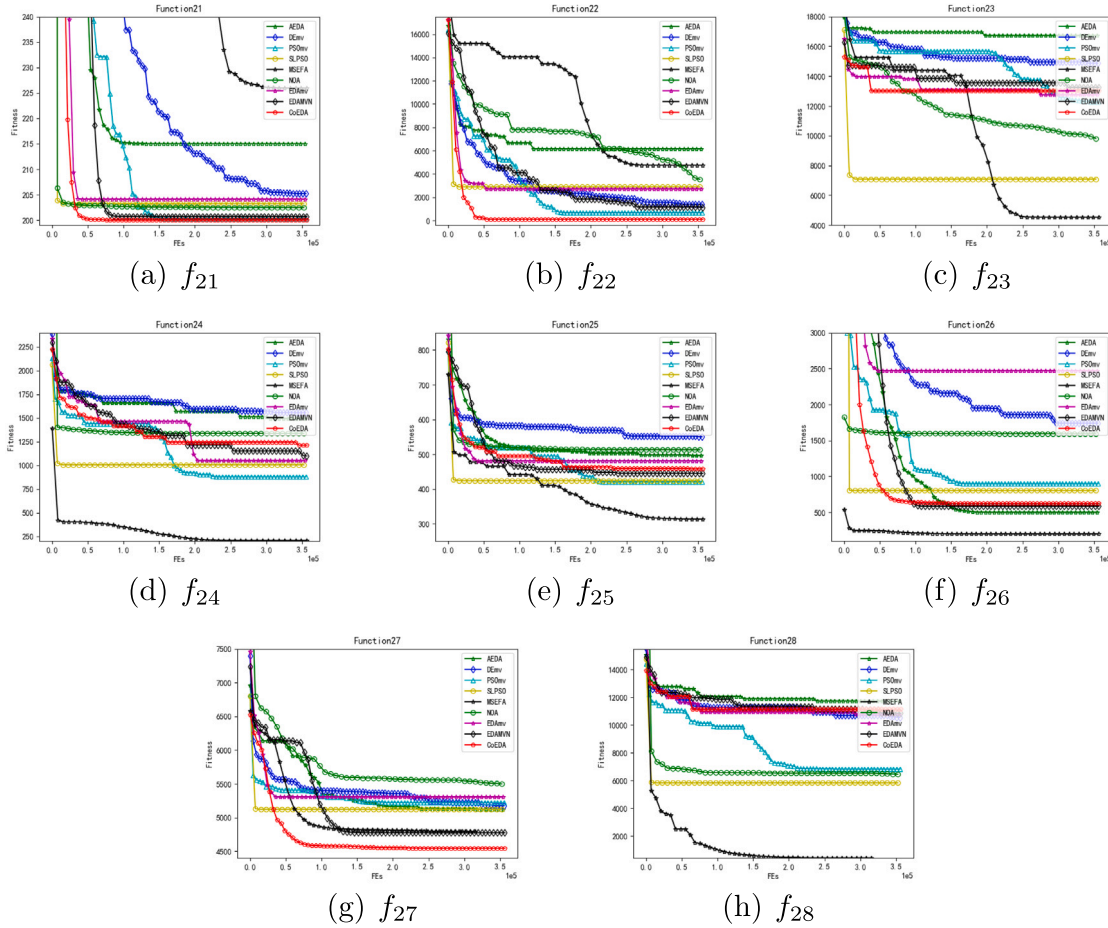


Fig. 7. Convergence process on 8 composition benchmarks.

handling the unimodal functions and multimodal functions from 28 artificial MVOPs.

5. Conclusion

In this paper, we propose a dynamic differential grouping-based co-evolutionary estimation of distribution algorithm (CoEDA_{mv}) to handle the complex mixed-variable optimization problems. First, the proposed dynamic differential grouping method (DDG) will dynamically update the groupings during the evolution process, which analyzes the variable interaction and the search performance in the current search region. DDG can guarantee that CoEDA_{mv} requires more fitness evaluations to the promising subproblem and improve the search efficiency by simplifying the complex MVOPs. Second, due to the histogram probabilistic model do not require any prior knowledge about distribution of solutions, CoEDA_{mv} proposes different histogram probabilistic models, i.e. FRCH and FRDH, to deal with continuous and discrete variables respectively. FRCH and FRDH will make a good use of the elite individuals obtained from the fitness ranking strategy, and help search the global optimum with the elite neighborhood-based updating probability strategy. The elite individual-guided probabilistic updating strategy adopted in FRDH and FRCH can well balance exploration and exploitation. Finally, the experimental results on 28 artificial MVOPs show the proposed CoEDA_{mv} can get better performance than some state-of-the-art algorithms for MVOPs.

We have not yet improved the sampling methods to match different landscapes of mixed variables. In future, we will devote to designing sampling methods which can take into account the difference in search

space of mixed variables. In addition, we will try to utilize CoEDA_{mv} to solve more complicated MVOPs with large scale decision variables or constraints, and investigate the applications on some real-world complex MVOPs.

CRediT authorship contribution statement

Shijia Huang: Methodology, Writing – original draft, Writing – review & editing, Software. **Zhe Wang:** Writing – original draft, Software, Data curation. **Yang Ge:** Writing – original draft, Writing – review & editing. **Feng Wang:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is supported by the National Nature Science Foundation of China [Grant Nos. 62173258, 61773296].

Table 3
The ablation experiments of CoEDA_{mv} on the 28 benchmarks.

		CoEDA _{mv}	CoEDA _{mv} -DDG	CoEDA _{mv} -HPM	CoEDA _{mv} -DDG-HPM	CoEDA _{mv} -FRDH	CoEDA _{mv} -FRCH
F1	MEAN	0.00E+00	0.00E+00(=)	8.75E+04(+)	8.77E+04(+)	1.92E+01(+)	2.10E-14(+)
	STD	0.00E+00	0.00E+00	8.04E+03	5.74E+03	8.12E+00	1.13E-13
F2	MEAN	5.86E+06	1.12E+08(+)	2.05E+09(+)	1.94E+09(+)	1.02E+07(+)	7.82E+06(=)
	STD	1.44E+06	1.91E+08	3.24E+08	4.16E+08	5.37E+06	3.82E+06
F3	MEAN	2.23E+07	3.94E+10(+)	1.35E+18(+)	2.91E+17(+)	1.29E+10(+)	1.79E+10(+)
	STD	1.03E+08	1.13E+11	2.69E+18	4.23E+17	1.59E+10	2.66E+10
F4	MEAN	2.64E+04	1.30E+05(+)	1.58E+05(+)	1.61E+05(+)	5.84E+04(+)	6.37E+04(+)
	STD	4.58E+03	8.04E+04	1.81E+04	1.52E+04	1.83E+04	2.45E+04
F5	MEAN	9.53E-42	3.88E-34(=)	2.64E+03(+)	5.92E+03(+)	7.02E+00(+)	1.53E-22(+)
	STD	2.64E-41	2.13E-33	4.98E+03	1.02E+04	3.11E+00	8.37E-22
F6	MEAN	5.42E+01	6.30E+01(+)	1.59E+04(+)	1.75E+04(+)	6.41E+01(+)	7.20E+01(+)
	STD	2.61E-01	1.34E+01	2.75E+03	2.34E+03	8.25E+00	2.04E+01
F7	MEAN	2.96E+01	3.24E+01(=)	4.09E+05(+)	3.77E+05(+)	3.12E+01(=)	2.86E+01(=)
	STD	9.39E+00	9.85E+00	3.92E+05	2.33E+05	6.96E+00	8.05E+00
F8	MEAN	2.11E+01	2.11E+01(=)	2.11E+01(-)	2.11E+01(-)	2.11E+01(=)	2.12E+01(=)
	STD	3.55E-02	4.86E-02	4.09E-02	3.33E-02	4.96E-02	4.72E-02
F9	MEAN	1.52E+01	3.71E+01(+)	7.97E+01(+)	8.05E+01(+)	6.50E+01(+)	5.28E+01(+)
	STD	1.75E+01	2.90E+01	1.63E+00	1.31E+00	1.34E+01	2.54E+01
F10	MEAN	8.62E+00	2.58E+01(+)	1.21E+04(+)	1.23E+04(+)	5.76E+01(+)	4.72E+01(+)
	STD	2.02E+01	3.13E+01	1.75E+03	1.29E+03	4.67E+01	4.55E+01
F11	MEAN	1.03E-01	3.55E-02(=)	1.35E+03(+)	1.37E+03(+)	1.07E+01(+)	6.64E-02(=)
	STD	3.14E-01	1.94E-01	8.98E+01	5.06E+01	2.29E+00	2.52E-01
F12	MEAN	3.31E+02	3.30E+02(=)	1.60E+03(+)	1.59E+03(+)	3.24E+02(-)	3.29E+02(=)
	STD	9.35E+00	1.04E+01	1.17E+02	1.12E+02	1.09E+01	1.22E+01
F13	MEAN	3.27E+02	3.27E+02(=)	1.59E+03(+)	1.55E+03(+)	3.25E+02(=)	3.24E+02(=)
	STD	1.56E+01	1.37E+01	9.01E+01	1.35E+02	1.26E+01	1.41E+01
F14	MEAN	6.82E-01	4.99E-01(=)	9.87E+03(+)	9.72E+03(+)	4.34E+01(+)	1.43E-01(=)
	STD	1.34E+00	9.94E-01	4.76E+02	6.07E+02	1.98E+01	3.20E-01
F15	MEAN	1.33E+04	1.33E+04(=)	1.55E+04(=)	1.59E+04(-)	1.32E+04(=)	1.32E+04(=)
	STD	3.05E+02	3.82E+02	4.99E+02	3.67E+02	5.04E+02	2.93E+02
F16	MEAN	3.52E+00	3.39E+00(=)	3.32E+00(-)	3.36E+00(-)	3.46E+00(=)	3.38E+00(=)
	STD	2.24E-01	3.26E-01	2.79E-01	3.41E-01	3.13E-01	3.45E-01
F17	MEAN	3.18E+01	9.02E+01(+)	3.73E+03(+)	3.71E+03(+)	1.14E+02(+)	1.11E+02(+)
	STD	3.57E+01	1.01E+02	4.15E+02	3.15E+02	8.63E+01	9.60E+01
F18	MEAN	3.55E+02	3.55E+02(=)	4.31E+03(+)	4.24E+03(+)	3.71E+02(+)	3.40E+02(-)
	STD	2.83E+01	2.92E+01	3.08E+02	2.41E+02	2.37E+01	2.71E+01
F19	MEAN	2.95E+01	3.19E+01(=)	2.65E+06(+)	3.15E+06(+)	3.05E+01(=)	2.90E+01(=)
	STD	1.41E+00	1.14E+01	8.91E+05	6.50E+05	7.62E+00	1.31E+00
F20	MEAN	2.15E+01	2.48E+01(+)	2.50E+01(+)	2.50E+01(+)	2.41E+01(+)	2.40E+01(+)
	STD	7.27E-01	6.88E-01	5.83E-09	2.51E-09	1.46E+00	1.55E+00
F21	MEAN	2.00E+02	2.00E+02(=)	4.85E+03(+)	4.93E+03(+)	2.05E+02(+)	2.09E+02(=)
	STD	1.24E-04	4.08E-02	3.00E+02	4.05E+02	1.35E+01	3.98E+01
F22	MEAN	1.01E+02	1.03E+02(=)	9.88E+03(+)	9.77E+03(+)	1.39E+02(+)	1.01E+02(=)
	STD	9.73E-01	1.13E+00	4.88E+02	4.98E+02	1.60E+01	9.54E-01
F23	MEAN	1.34E+04	1.33E+04(=)	1.56E+04(+)	1.57E+04(+)	1.35E+04(=)	1.33E+04(=)
	STD	3.53E+02	3.75E+02	4.93E+02	7.79E+02	3.33E+02	3.55E+02
F24	MEAN	1.21E+03	1.20E+03(=)	2.24E+03(+)	2.25E+03(+)	1.21E+03(=)	1.19E+03(=)
	STD	9.14E+01	6.08E+01	4.49E+01	3.29E+01	5.55E+01	6.08E+01
F25	MEAN	4.57E+02	4.51E+02(=)	7.99E+02(+)	7.92E+02(+)	4.52E+02(=)	4.56E+02(=)
	STD	1.84E+01	1.37E+01	2.15E+01	2.08E+01	1.71E+01	1.40E+01
F26	MEAN	7.18E+02	6.71E+02(=)	2.69E+04(+)	2.74E+04(+)	6.39E+02(=)	6.67E+02(=)
	STD	3.25E+02	2.83E+02	1.77E+03	2.70E+03	1.80E+02	2.10E+02
F27	MEAN	4.72E+03	4.71E+03(=)	7.13E+03(+)	7.17E+03(+)	4.83E+03(+)	4.74E+03(=)
	STD	1.06E+02	9.68E+01	1.22E+02	1.48E+02	8.09E+01	9.78E+01
F28	MEAN	1.03E+04	1.07E+04(=)	1.49E+04(+)	1.50E+04(+)	1.08E+04(=)	9.82E+03(=)
	STD	1.35E+03	1.22E+03	3.30E+02	3.84E+02	1.06E+03	1.59E+03
+/-/-			8/20/0	25/1/2	25/3/0	16/11/1	9/18/1

References

- Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowledge-Based Systems*, 262, Article 110248.
- Aye, C. M., Wansaseub, K., Kumar, S., Tejani, G. G., Bureerat, S., Yildiz, A. R., & Pholdee, N. (2023). Airfoil shape optimisation using a multi-fidelity surrogate-assisted metaheuristic with a new multi-objective infill sampling technique. *CMES-Computer Modeling in Engineering & Sciences*, 137.
- Van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8, 225-239.
- Brajević, I., Stanimirović, P. S., Li, S., Cao, X., Khan, A. T., & Kazakovtsev, L. A. (2022). Hybrid sine cosine algorithm for solving engineering optimization problems. *Mathematics*, 10, 4555.
- Chen, A., Ren, Z., Guo, W., Liang, Y., & Feng, Z. (2023). An efficient adaptive differential grouping algorithm for large-scale black-box optimization. *IEEE Transactions on Evolutionary Computation*, 27, 475-489.
- Chen, W., Weise, T., Yang, Z., & Tang, K. (2010). Large-scale global optimization using cooperative coevolution with variable interaction learning. In *Parallel problem solving from nature, PPSN XI* (pp. 300-309).
- Chen, Y., Xie, W., & Zou, X. (2015). A binary differential evolution algorithm learning from explored solutions. *Neurocomputing*, 149, 1038-1047.
- Cheng, R., & Jin, Y. (2015). A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291, 43-60.
- Fetanat, A., & Khorasaninejad, E. (2015). Size optimization for hybrid photovoltaic-wind energy system using ant colony optimization for continuous domains based integer programming. *Applied Soft Computing*, 31, 196-209.
- Gallard, R. H. (1999). Genetic algorithms+ data structure= evolution programs, zbignew michalewicz. *Journal of Computer Science and Technology*, 1, 1.
- Gao, L., & Hailu, A. (2010). Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems. *International Journal of Computational Intelligence Systems*, 3, 832-842.

- He, Z., Peng, H., Chen, J., Deng, C., & Wu, Z. (2021). A spark-based differential evolution with grouping topology model for large-scale global optimization. *Cluster Computing*, 24, 515–535.
- Jiang, D., Lin, Y., Zhu, W., & He, Z. (2022). A parallel based evolutionary algorithm with primary-auxiliary knowledge. *Information Sciences*, 610, 1122–1142.
- Kim, T.-H., Cho, M., & Shin, S. (2020). Constrained mixed-variable design optimization based on particle swarm optimizer with a diversity classifier for cyclically neighboring subpopulations. *Mathematics*, 8, 2016.
- Kumar, S., Tejani, G. G., Pholdee, N., & Bureerat, S. (2021). Multi-objective passing vehicle search algorithm for structure optimization. *Expert Systems with Applications*, 169, Article 114511.
- Kumar, S., Tejani, G. G., Pholdee, N., & Bureerat, S. (2022). Performance enhancement of meta-heuristics through random mutation and simulated annealing-based selection for concurrent topology and sizing optimization of truss structures. *Soft Computing*, 26, 5661–5683.
- Larrañaga, P., & Lozano, J. A. (2001). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media.
- Li, S., Wang, F., He, Q., & Wang, X. (2023). Deep reinforcement learning for multi-objective combinatorial optimization: A case study on multi-objective traveling salesman problem. *Swarm and Evolutionary Computation*, 83, Article 101398.
- Liang, J. J., Qu, B., Suganthan, P. N., & Hernández-Díaz, A. G. (2013). *Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization: Technical Report, 201212*, (pp. 281–295). Zhengzhou University.
- Liang, Y., Ren, Z., Yao, X., Feng, Z., Chen, A., & Guo, W. (2018). Enhancing gaussian estimation of distribution algorithm by exploiting evolution direction with archive. *IEEE Transactions on Cybernetics*, 50, 140–152.
- Liao, T., Socha, K., de Oca, M. A. M., Stützle, T., & Dorigo, M. (2013). Ant colony optimization for mixed-variable optimization problems. *IEEE Transactions on Evolutionary Computation*, 18, 503–518.
- Lin, Y., Liu, Y., Chen, W.-N., & Zhang, J. (2018). A hybrid differential evolution algorithm for mixed-variable optimization problems. *Information Sciences*, 466, 170–188.
- Ma, X., Liu, F., Qi, Y., & Gong, M. (2016). A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation*, 20, 275–298.
- Molina Pérez, D., Alfredo Portilla-Flores, E., & Vega-Alvarado, E. (2022). An improved estimation of distribution algorithm for solving constrained mixed-integer nonlinear programming problems. In *IEEE congress on evolutionary computation* (pp. 01–08).
- Nonut, A., Kanokmedhakul, Y., Bureerat, S., Kumar, S., Tejani, G. G., Artrit, P., Yıldız, A. R., & Pholdee, N. (2022). A small fixed-wing uav system identification using metaheuristics. *Cogent Engineering*, 9, Article 2114196.
- Omidvar, M. N., Li, X., Mei, Y., & Yao, X. (2015). Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 18, 378–393.
- Omidvar, M. N., Li, X., Yang, Z., & Yao, X. (2010). Cooperative co-evolution for large scale optimization through more frequent random grouping. In *IEEE congress on evolutionary computation* (pp. 1–8).
- Omidvar, M. N., Yang, M., Mei, Y., Li, X., & Yao, X. (2017). Dg2: A faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Transactions on Evolutionary Computation*, 21, 929–942.
- Peng, H., Xiao, W., Han, Y., Jiang, A., Xu, Z., Li, M., & Wu, Z. (2022). Multi-strategy firefly algorithm with selective ensemble for complex engineering optimization problems. *Applied Soft Computing*, 120, Article 108634.
- Potter, M. A., & De Jong, K. A. (1994). A cooperative coevolutionary approach to function optimization. In *International conference on parallel problem solving from nature* (pp. 249–257).
- Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic algorithms. In *Search methodologies Introductory tutorials in optimization and decision support techniques* (pp. 97–125).
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2015). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104, 148–175.
- Shi, W., Chen, W.-N., Lin, Y., Gu, T., Kwong, S., & Zhang, J. (2017). An adaptive estimation of distribution algorithm for multipolicy insurance investment planning. *IEEE Transactions on Evolutionary Computation*, 23, 1–14.
- Shuai, H., Fang, J., Ai, X., Tang, Y., Wen, J., & He, H. (2018). Stochastic optimization of economic dispatch for microgrid based on approximate dynamic programming. *IEEE Transactions on Smart Grid*, 10, 2440–2452.
- Singh, P., Kottath, R., & Tejani, G. G. (2022). Ameliorated follow the leader: algorithm and application to truss design problem. In *Structures*, Vol. 42 (pp. 181–204). Elsevier.
- Sun, Y., Kirley, M., & Halgamuge, S. K. (2017). A recursive decomposition method for large scale continuous optimization. *IEEE Transactions on Evolutionary Computation*, 22, 647–661.
- Sun, Y., Omidvar, M. N., Kirley, M., & Li, X. (2018). Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition. In *Proceedings of the genetic and evolutionary computation conference* (pp. 889–896).
- Tanaka, S., & Takii, K. (2015). A faster branch-and-bound algorithm for the block relocation problem. *IEEE Transactions on Automation Science and Engineering*, 13, 181–190.
- Tejani, G. G., Pholdee, N., Bureerat, S., Prayogo, D., & Gandomi, A. H. (2019). Structural optimization using multi-objective modified adaptive symbiotic organisms search. *Expert Systems with Applications*, 125, 425–441.
- Tejani, G. G., Savsani, V. J., Patel, V. K., & Bureerat, S. (2017). Topology, shape, and size optimization of truss structures using modified teaching-learning based optimization. *Advances in Computational Design*, 2, 313–331.
- Van Aelst, S., Wang, X. S., Zamar, R. H., & Zhu, R. (2006). Linear grouping using orthogonal regression. *Computational Statistics & Data Analysis*, 50, 1287–1312.
- Wang, F., Huang, Z., & Wang, S. (2023). I ϵ + lgea a learning-guided evolutionary algorithm based on i ϵ + indicator for portfolio optimization. *Complex System Modeling and Simulation*, 3, 191–201.
- Wang, F., Li, Y., Zhou, A., & Tang, K. (2020). An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Transactions on Evolutionary Computation*, 24, 479–493.
- Wang, J.-J., & Wang, L. (2020). A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50, 1805–1819.
- Wang, F., Wang, X., & Sun, S. (2022). A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Information Sciences*, 602, 298–312.
- Wang, F., Zhang, H., & Zhou, A. (2021). A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm and Evolutionary Computation*, 60, Article 100808.
- Yang, Z., Tang, K., & Yao, X. (2008). Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178, 2985–2999.
- Yang, M., Zhou, A., Li, C., Guan, J., & Yan, X. (2020). Ccfr2: A more efficient cooperative co-evolutionary framework for large-scale global optimization. *Information Sciences*, 512, 64–79.
- Yang, M., Zhou, A., Li, C., & Yao, X. (2020). An efficient recursive differential grouping for large-scale continuous problems. *IEEE Transactions on Evolutionary Computation*, 25, 159–171.
- Zhou, A., Sun, J., & Zhang, Q. (2015). An estimation of distribution algorithm with cheap and expensive local search methods. *IEEE Transactions on Evolutionary Computation*, 19, 807–822.