



An improved estimation of distribution algorithm for multi-objective optimization problems with mixed-variable

Wenxiang Wang¹ · Kangshun Li^{1,2} · Hassan Jalil¹ · Hui Wang³

Received: 21 January 2022 / Accepted: 30 July 2022 / Published online: 28 August 2022

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Multi-objective evolutionary algorithms face many challenges in optimizing mixed-variable multi-objective problems, such as quantization error, low search efficiency of discontinuous discrete variables, and difficulty in coding non-integer discrete variables. To overcome these challenges, this paper proposes a mixed-variable multi-objective evolutionary algorithm based on estimation of distribution algorithm (MVMO-EDA). Compared with traditional multi-objective evolutionary algorithms, MVMO-EDA has the following improvements: (1) instead of crossover and mutation, statistics and sampling are used to generate offspring, which can avoid the quantization error caused by crossover and mutation operations; (2) using index coding for discrete variables to improve the search efficiency; and (3) a scalable histogram probability distribution model and two crowding distance-based diversity maintenance strategies are used to improve the global optimization ability. The performance of the proposed MVMO-EDA is evaluated on the modified ZDT and DTLZ benchmark sets with mixed-variable, and the results show that MVMO-EDA has a competitive performance both in convergence and diversity.

Keywords Multi-objective optimization · Mixed-variable · Evolutionary algorithm · Estimation of distribution algorithm · Scalable histogram

1 Introduction

The optimization problems with both continuous and discrete decision variables are called mixed-variable optimization problems. Mixed-variable multi-objective optimization problems (MMOPs) widely exist in daily life.

For example, the spring design problem [1]: in which the number of turns and wire diameter of the spring are discrete variables, while length and diameter of the spring are continuous variables, and the spring's weight and shear stress are two optimization goals. The cellular network optimization problem is another issue: in which the downtilt angle of the antenna is often taken as an integer to facilitate the implementation process, the transmit power of the antenna is continuous, and there are also multiple optimization objectives (coverage, capacity, power, etc.). In fact, considering factors such as material selection, operation accuracy, and industry specifications, the optimization schemes of many continuous optimization problems must be discretized during their actual implementation, but the optimization accuracy is likely to be lost in the discretization. Therefore, the study of MMOPs has important application significance.

There are two categories optimization methods for MMOPs: the first category is the traditional deterministic optimization methods, and the second category is the heuristic algorithms. The traditional methods first adopt

¹ College of Mathematics and Informatics, South China Agricultural University, Guangzhou 51006, China

² School of Computer Science, Guangdong University of Science and Technology, Dongguan 523000, China

³ School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen 518172, China

some integrated methods (such as Weighted Sum Method [2], Goal Programming [3], ε -Constraint Method [4], and Lexicographic Ordering [5]) to transform the multi-objective optimization problem into a single-objective optimization problem and then use mature mathematic methods to solve it. The mature optimization theories such as gradient descent method, Newton method, and convex optimization can be used for continuous variable optimization. While the linear programming methods such as Simplex Method [6], Cutting Plane Method [7], Branch and Bound [8], and Implicit Enumeration Method [9] can be used for discrete variable optimization. The traditional methods generally have specific requirements for the optimization models, such as being differentiable, continuous, and satisfying Slater conditions. Besides, these methods cannot solve black box problems. Therefore, the traditional methods have many limitations in MMOPs.

In the past two decades, heuristic algorithms have been widely used in multi-objective optimization problems. And the multi-objective evolutionary algorithm has become a hot spot in the field of multi-objective optimization. The state-of-the-art multi-objective evolutionary algorithms include vector evaluated genetic algorithm (VEGA) [10], non-dominated sorting genetic algorithm II (NSGA-II) [11], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [12], strength Pareto evolutionary algorithm (SPEA) [13], SPEA2 [14], etc. The research of multi-objective evolutionary algorithm has many achievements in the fields of many-objective optimization [15], high-dimensional multi-objective optimization [16], and dynamic multi-objective optimization [17], etc. However, the research on mixed-variable multi-objective evolutionary algorithm has not received enough attention. For instance, there exists only a handful of studies in multi-objective particle swarm optimization (MOPSO) where mixed-discrete variables are considered [18], and many algorithms face the influence of quantization error or low efficiency in MMOPs [19].

The estimation of distribution algorithm (EDA) [20] is an evolutionary algorithm based on the principles of statistics and sampling, which can avoid the quantization error caused by crossover and mutation operations and make it good for dealing with mixed variables. Therefore, this paper proposes a mixed-variable multi-objective evolutionary algorithm based on EDA (MVMO-EDA). The main contributions of this paper are as follows:

1. An extended estimation of distribution algorithm for multi-objective problems with mixed variables is proposed. We use the multi-objective optimization framework based on decomposition to extend the EDA into a multi-objective evolutionary algorithm and improve its elements such as probability distribution model,

offspring generation method, coding method, and selection strategy.

2. A scalable histogram probability model that is suitable for describing the probability distribution of mixed-variable is proposed for EDA. This model uses statistical histograms to replace the Gaussian distribution, Cauchy distribution, and other mathematical models, that are more suitable for the representation of discrete variable's probability distribution, and also suitable for continuous variables. Instead of count individual numbers, the statistical histograms record local minimum fitness to avoid the algorithm falling into local optimization.

3. We improve the coding method and selection strategy of the EDA. The index coding method for discrete variables can not only simplify the coding of non-integer discrete variables but also improve the search efficiency of non-uniform discrete variables. Besides, a selection strategy integrating the scalar fitness and crowding distance is used to improve the diversity of solution set.

The rest of this paper is organized as follows: Sect. 2 reviews the related work and proposes the motivation; Sect. 3 contains details of the proposed algorithm; Sect. 4 introduces the experiment study; and finally, Sect. 5 concludes this paper.

2 Related work

This paper focuses on the mixed-variable multi-objective optimization methods based on evolutionary algorithm. The current research status and basic algorithms related to this research are introduced as given below.

2.1 Previous research on mixed-variable multi-objective optimization problems

Scholars have done the following researches for the discrete variable optimization of MMOPs:

Firstly, MMOPs are treated as continuous variable optimization problems, and then, the optimization results are quantified to meet the requirements of discretization. These algorithms include Pareto GA [21], HSIA [22], D-ITGO [23], etc. These algorithms inherit the mature continuous multi-objective evolutionary algorithm and have the advantage of minor revises for original algorithms, but they are prone to large quantization errors in complex multimodal functions. Take the minimum optimization of the Gramacy & Lee (2012) function, for example, as is shown in Fig. 1, if it is regarded as a continuous optimization problem, then the optimization will search for the continuous optimal point A and then find

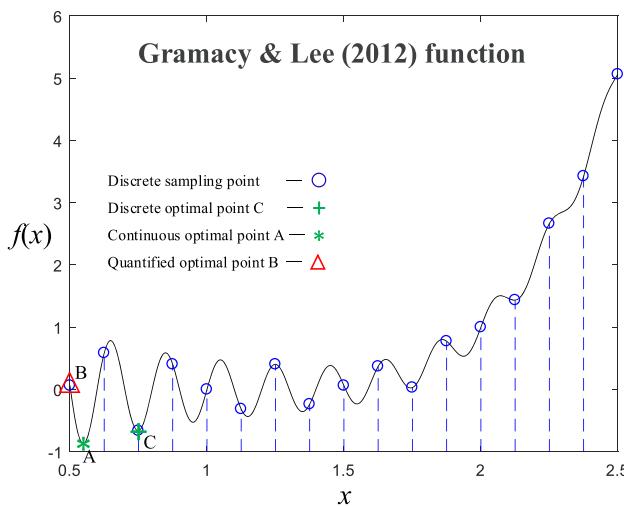


Fig. 1 Quantization error diagram

point A's nearest quantized point B, but the point C is the true discrete optimal point. Obviously, it brings a larger quantization error in the quantization process.

Secondly, the quantization operation is embedded into the coding, crossover, mutation, and other operations of evolutionary algorithms to obtain the quantified offspring. Among them, the binary coding method of the genetic algorithm [24] is very suitable for coding discrete variables. Such algorithms included DPSO [25], MV-PSP [26], MDPSO [27], FC-MOPSO [28], CES-SEO [29], etc. These algorithms reduce the quantization error to a certain extent, but adding quantization operation to the crossover and mutation operation will change the search direction of the algorithm, thereby reducing the algorithm's convergence speed.

Thirdly, the estimation of distribution algorithm is used to optimize discrete variables. For the algorithm is based on the principle of statistics and sampling, there is no quantization error when dealing with discrete variables. Besides, it also has the advantage of fast convergence. In 2019, Wang et al. proposed a single-objective mixed-variable optimization algorithm named EDA_{Mvn} [30], which achieved good results in the mixed-variable newsvendor problem.

The EDA-based method is easy to deal with discrete variables, which makes it more advantageous in optimizing MMOPs. However, due to some inherent defects of EDA (local optimization, poor in diversity, etc.), there is no report on mixed-variable multi-objective evolutionary algorithm based on EDA. The research of this paper is to take advantage of EDA's easy handling of discrete variables, then overcome its inherent defects, and finally propose an efficient mixed-variable multi-objective evolutionary algorithm.

2.2 Estimation of distribution algorithm

The idea of EDA was first proposed by H. Mühlenbein and G. Paaß [31] in 1996, and the formal EDA was proposed by Larranaga P, Lozano J A in 2002 [17]. The steps of typical EDA are as follows:

- Step 1: Initialize a population P ;
- Step 2: Select n elite individuals from P ;
- Step 3: Estimate a probability distribution model $f(\vec{x})$ from the selected elite individuals;
- Step 4: Generate the offspring population P' by sampling from $f(\vec{x})$;
- Step 5: Select N elite individuals from P and P' to replace P ;
- Step 6: Judge whether the termination conditions are satisfied, if satisfied then output the optimization results, otherwise skip to Step 2.

The probability distribution model of the population is the core of EDA. According to the correlation of the model's variables, EDAs can be divided into univariate EDA, tree-based EDA and multivariate EDA [32]. Univariate EDAs, such as PBIL [33], UMDA [34], and cGA [35], do not consider the correlation of variables, and they treat the variables of the model as independent variables. This makes the univariate EDA model simple, easy to implement, and running fast. But for high-dimensional problems, the algorithm has a low convergence speed. While multivariate EDA fully considers the correlation among all decision variables and has a complex probability model that can optimize high-dimensional problems. However, it also has problems such as the difficulty in establishing a probability model and increased computational complexity. Typical algorithms of multivariate EDA include ECGA [36], FDA [37], BOA [38], etc. The performance of tree-based EDA is between univariate EDA and multivariate EDA, and the representative algorithms include the MIMIC [39], COMIT [40] and BMDA [41]. Since the probability distribution model of EDAs relies on the statistics of elite individuals' numbers, once the algorithm falls into a local optimum, then it is difficult to escape from there, which makes EDAs easy to premature convergence.

Therefore, scholars have proposed multi-probability model [42] and Bayesian classification method [43] to overcome the premature convergence problem of EDAs and improve the high-dimensional optimization ability of EDAs through subspace model [44] and dimensionality reduction technology [45]. In addition, EDA is also extended to optimize continuous multi-objective problems [46, 47]. However, from the literature review, so far no reports of EDA-based mixed-variable multi-objective evolutionary algorithms have been found.

2.3 Decomposition-based multi-objective optimization framework

There are many decomposition methods in the field of multi-objective optimization. The MOEA/D framework was proposed by Zhang et al. in 2007 [12]; due to its excellent performance, it has achieved many achievements in many-objective optimization [48], high-dimensional optimization [49] and dynamic optimization [50]. The process of MOEA/D framework is as follows:

Step 1: Initialization

Step 1.1: Set $Pop = \emptyset$.

Step 1.2: Initialize neighborhood B .

Step 1.3: Initialize the population Pop .

Step 1.4: Calculate the reference point Z .

Step 2: Update

for $i = 1 : N$ *do*

Step 2.1: Reproduction. Generate a new solution y by using genetic operators.

Step 2.2: Improvement. Apply a problem-specific repair/improvement heuristic on y to produce y' .

Step 2.3: Update of reference point Z .

Step 2.4: Update of neighborhood.

Step 2.5: Update of Pop .

end

Step 3: Stopping Criteria. If stopping criteria is satisfied, then stop and output Pop . Otherwise, go to Step 2.

Although the MOEA/D algorithm has excellent performance in continuous multi-objective optimization problems, it was designed without considering the optimization of discrete variables, so it cannot be directly used in MMOPs.

3 The proposed algorithm

Since most of the multi-objective evolutionary algorithms have crossover and mutation operations, they will inevitably generate some offspring that exceeding the range of discrete space. Then, the quantization operations will be used at this time, but from the case in Fig. 1, it can be seen that the quantization operation may bring a large error in the multimodal functions. The EDA is an evolutionary algorithm without crossover operations, which can avoid the quantization error caused by crossover and mutation operations. Therefore, this paper proposes a mixed-variable multi-objective evolutionary algorithm based on EDA (MVMO-EDA) for the MMOPs. This section introduces the workflow and basic principles of the proposed MVMO-EDA.

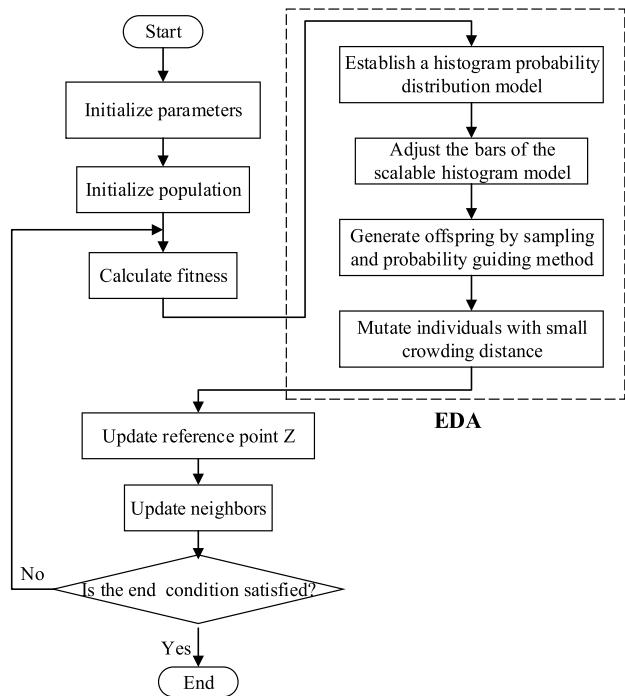


Fig. 2 Flow chart of the MVMO-EDA

3.1 Workflow of MVMO-EDA

The overall framework of the proposed MVMO-EDA is shown in Fig. 2. We use the MOEA/D framework to extend the EDA to a multi-objective evolutionary algorithm and propose a scalable histogram probability distribution model and two crowding distance-based diversity maintenance strategies to improve its global optimization ability. Besides, an index coding approach for discrete variables is used to improve the search efficiency. Relevant detail improvements will be described in the following subsections.

3.2 Coding approach for mixed-variable

In this paper, two different coding approaches are used for continuous and discrete variables of MMOPs. Continuous variables are coded by real numbers, while discrete variables are coded by integer indexes. The traditional real number codes or binary codes have low coding efficiency for non-uniform or non-integer discrete variables. To this end, this article adopts an index coding approach for discrete variables. In this method, the discrete variables are mapped to continuous integer indexes, as shown in Fig. 3, and then, the integer indexes are used for optimization. This coding approach not only improves the coding efficiency of non-uniform and non-integer variables but also compresses the search space of discrete variables. So the

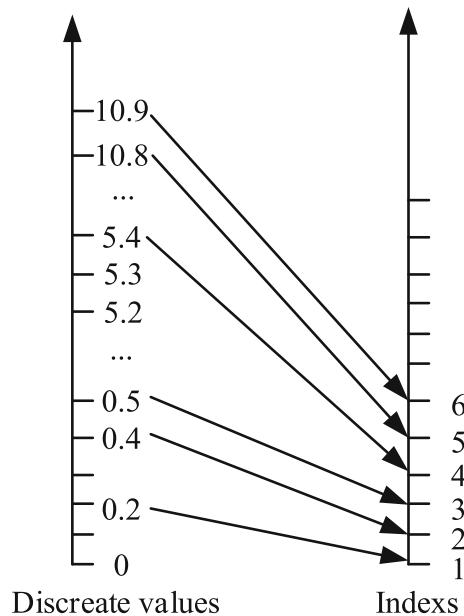


Fig. 3 Index coding for discrete variables

index coding approach can improve the search efficiency of the proposed MVMO-EDA.

3.3 Scalable histogram probability distribution model

The probability distribution model of the population is the core of the EDA. This paper proposes a scalable histogram probability distribution model to improve the global optimization capability of the basic EDA. In this model, the local optimal fitness, instead of the number of elite individuals, is used to calculate the probability distribution of the offspring, so that the algorithm can quickly jump out of the local optimal solution. And the histogram bars are adjusted by expanding and shrinking regularly to balance the global search ability and convergence performance of MVMO-EDA, as shown in Fig. 4. For a better understanding of the scalable histogram probability distribution model, the related concepts are introduced as follows.

Local optimal solution: for single-objective optimization problems, the local optimal solution is individual with the smallest fitness (this paper default to minimum optimization) in a certain region; and for multi-objective optimization problems, this is the solution with the smallest scalar fitness [51] in a certain region.

Sensitive area: if the local optimal solution of a region is smaller than that of its left and right adjacent regions (if any), then the region is a sensitive area of the scalable histogram, which is also a candidate area of the global optimal solution. All expanding and shrinking operations are performed in the sensitive areas.

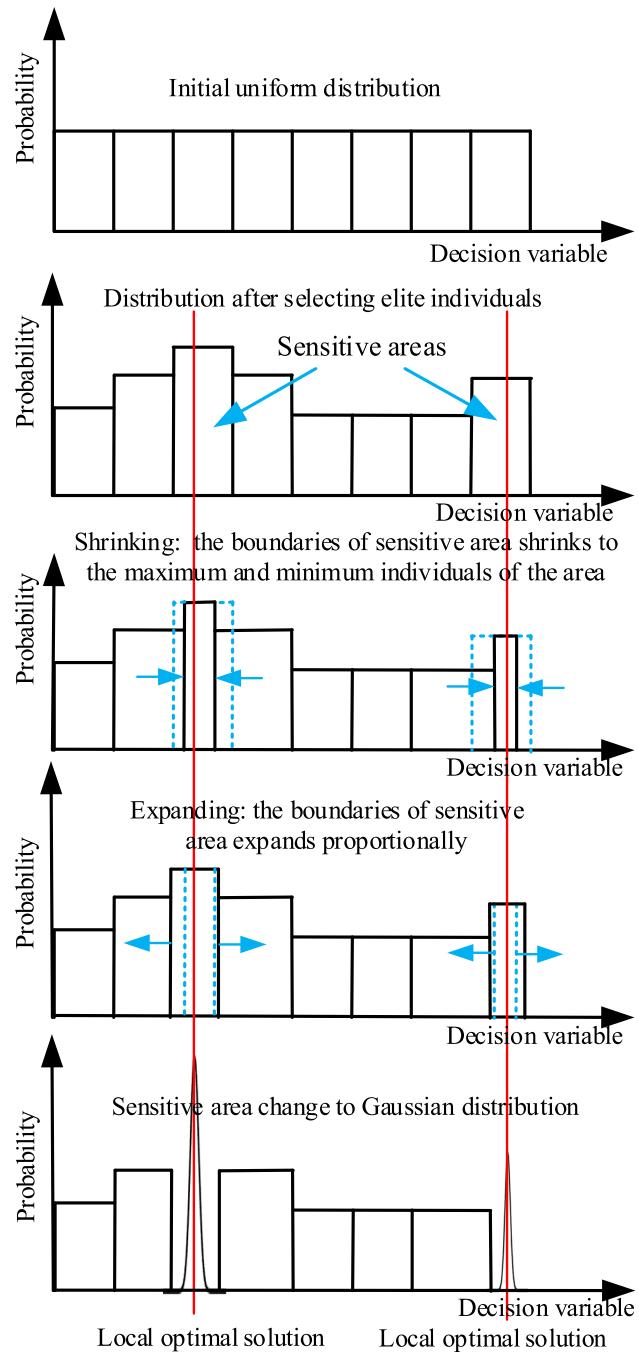


Fig. 4 The scalable histogram probability distribution model of MVMO-EDA

Sensitivity: the scalable histogram divides the search space into sensitive areas and non-sensitive areas according to the above definition and allocates the population probabilities of these two parts according to the sensitivity, which is defined as follows:

$$S_r = \frac{N_s}{N} \quad (1)$$

where N_s is the number of individuals in the sensitive area, and N is the number of the whole population. The range of sensitivity is $(0, 1)$. The larger S_r , the faster the convergence speed is, on the contrary, the smaller S_r , the stronger the global search capability is.

In the scalable histogram model, we set the probability distribution of the non-sensitive area to be a uniform distribution, which is good for global exploration, and set the sensitive area to be Gaussian distribution, which is good for in-depth search. The construction steps of the model are as follows:

Step 1 Probability allocation.

First, the primary probability allocation is carried out for the sensitive area and non-sensitive area of the model according to the sensitivity. The probability of all sensitive areas is N_s , and that of all non-sensitive areas is $1 - N_s$. Then, according to the local optimal solutions of all bars, the secondary probability allocation is carried out for each bar in the sensitive area and the non-sensitive area, respectively.

For sensitive areas, the probability of bars is sorted by local optimal solution's scalar fitness from small to large, and then, the probability of each bar is linearly allocated from large to small:

$$P_i = \frac{S_r}{K_s} - [(i - 0.5) - \frac{K_s}{2}] \cdot \sigma_1 \quad (2)$$

Among them, P_i is the probability of the i -th bar in sensitive areas after sorting, K_s is the number of sensitive areas in the histogram, and σ_1 is a probability difference among bars of all sensitive areas.

For non-sensitive areas, the probability of bars is also sorted by local optimal solution's scalar fitness from small to large and then linearly allocated from large to small:

$$P'_i = \frac{1 - S_r}{K - K_s} - [(i - K_s - 0.5) - (K - K_s)/2] \cdot \sigma_2 \quad (3)$$

where P'_i is the probability of the i -th bar after sorting, K is the number of all bars in the histogram, and σ_2 is the probability difference among bars in all non-sensitive areas.

Step 2 Boundaries adjustment.

In order to achieve the convergence of MVMO-EDA, the boundaries of sensitive areas need to be shrunk continuously in each iteration. However, due to the limited number of samples in each generation, it is easy to miss the global optimal solution and fall into the local optimal solution if the sensitive area shrinks too fast. Therefore, after each shrinking operation, the boundaries of sensitive areas should be expanded in a certain proportion to improve the global search ability of the algorithm. The shrinking and expanding operations of the scalable histogram are as follows:

Shrinking: MVMO-EDA is an evolutionary algorithm based on statistics and sampling principles. Due to the limited number of samples, the population tends to gather within the boundary (generally not overlapping with the boundary), and then, the two boundaries in a sensitive area can be shrunk as follows:

$$\begin{cases} bl_i = \min(X_i) \\ bu_i = \max(X_i) \end{cases} \quad (4)$$

where bl_i and bu_i are the lower and upper boundaries of the i -th bar, respectively.

Expanding: In the above-mentioned shrinking process, the shrinking distance of the upper boundary and the lower boundary is often different. Usually, many elite individuals are distributed near the boundary with a small shrink distance, and the global optimal solution is more likely to be near the boundary. Therefore, the boundary with smaller shrink distance needs to be expanded to a larger distance. On the contrary, there are often few elite individuals near the boundary with a large shrink distance, and the possibility that the global optimal solution is there is low. So the boundary with a large shrink distance should have a smaller expand distance. Then, the boundaries of sensitive area are expanded according to the following formula:

$$\begin{cases} bl_{i,k+1} = bl_{i,k} - \alpha \cdot \Delta l_2 \\ bu_{i,k+1} = bu_{i,k} + \alpha \cdot \Delta l_1 \end{cases} \quad (5)$$

Among them, $bl_{i,k}$ and $bu_{i,k}$ are the lower and upper boundaries of the i -th bar in the k -th iteration, respectively; α is the expanding factor which range is $(0, 1)$, the larger the value, the better global search ability is, while the smaller the value, the better convergence is. According to the experimental experience, the recommended value of α is 0.8; Δl_1 and Δl_2 are the latest shrink distances of lower and upper boundaries, respectively, they are calculated by the following formula:

$$\begin{cases} \Delta l_1 = \min(X_i) - bl_{i,k} \\ \Delta l_2 = bu_{i,k} - \max(X_i) \end{cases} \quad (6)$$

Step 3 Sampling.

It is necessary to get the probability density function of the scalable histogram model before sampling. The probability distribution of the bars in sensitive areas is Gaussian distribution, and then, the probability density function is:

$$f_i(x) = \frac{P_i}{\sigma \sqrt{2\pi}} e^{-\frac{(x-x_{i,best})^2}{2\sigma^2}}, x \in [bl_i, bu_i] \quad (7)$$

where $\sigma = 0.2$, and $x_{i,best}$ is the local optimal solution of the i -th bar after sorting.

The probability in the non-sensitive areas of bars is the uniform distribution, and then, the probability density function is:

$$f_i(x) = \frac{P'_i}{bu_i - bl_i}, x \in [bl_i, bu_i] \quad (8)$$

where bl_i and bu_i are the lower and upper boundaries of the i -th bar, respectively.

The construction of the scalable histogram probability distribution model can be summarized as Algorithm 1.

Algorithm 1: Construction of scalable histogram probability distribution model

Input: a population $Pop = [X_1, X_2, \dots, X_N]^T$, a scalar fitness $Y = [Y_1, Y_2, \dots, Y_N]^T$, a boundary matrix of histogram L , X 's dimension D , the number of bars in the histogram K .

Output: an updated boundaries L , a group probability of histogram P .

$L_{Best} \leftarrow$ Search the local best individuals of all bars.

$Flag_S \leftarrow$ Find the sensitive areas by compare local optimal solutions of nearby bars.

for $j = 1; j \leq D; j \leftarrow j + 1$ **do**

for $k = 1; k \leq K; k \leftarrow k + 1$ **do**

if $Flag_S_{j,k} = 1$ **then** // Sensitive area

calculate $P_{j,k}$ by Formula (2) // probability allocation

update $L_{k,j}$ and $L_{k+1,j}$ by Formula (4) // shrinking

update $L_{k,j}$ and $L_{k+1,j}$ by Formula (5) // expanding

else // Non-sensitive area

calculate $P_{j,k}$ by Formula (3) // probability allocation

end

end

end

3.4 Offspring generation method guided by scalable histogram probabilistic distribution model

The univariate EDA algorithm does not consider the correlation among decision variables and has the advantages of easy to implement and low-computational complexity. But it also has the defect in high-dimensional optimization. To overcome the defect, the research proposed an offspring generation method guided by the scalable histogram probabilistic distribution model. The proposed offspring generation method can transfer the parent variables' correlation information to the offspring sampled by the probability model, thereby improving the high-dimensional optimization capability of the univariate EDA.

Suppose there is a parent population $X = [X_1, X_2, \dots, X_N]^T$ with N individuals, where $X_i = [X_{i,1}, X_{i,2}, \dots, X_{i,D}]$ represents the i -th individual of X , and D is the dimension of decision variables. The process of generating offspring by the probability guiding method is

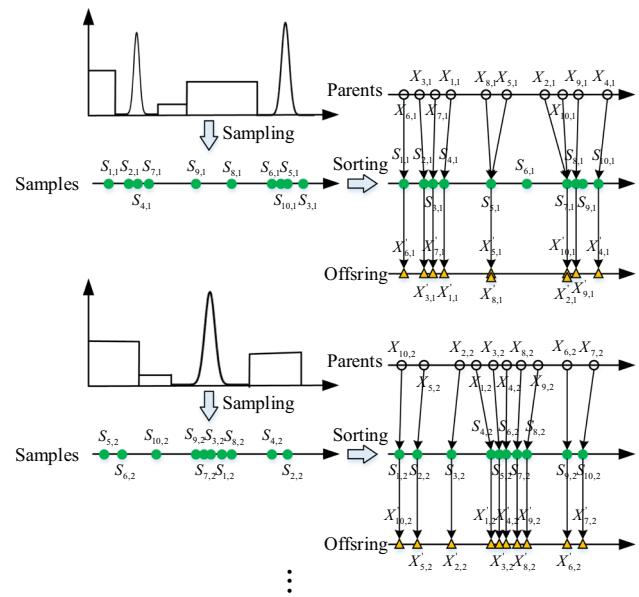


Fig. 5 Basic offspring generation method guided by the scalable histogram probabilistic distribution model

shown in Fig. 5. The method first sample in each dimension according to the probabilistic distribution of the scalable histogram and obtain a sample population $S = [S_1, S_2, \dots, S_D]$ with the same size as X , where $S_i = [S_{1,i}, S_{2,i}, \dots, S_{N,i}]^T$. After that, the data of each dimension of S are sorted independently from small to large. And then, copy X to create an offspring population X' . For any element $X'_{i,j}$ in X' , find a point $S_{x,j}$ with the smallest Euclidean distance from it in the j -th dimension samples, and use $S_{x,j}$ to replace X' . When all the variables are replaced, the final offspring population X' is obtained.

Although the probability guiding method can make the EDA algorithm quickly converge in the high-dimensional optimization problems, it also has the problems of insufficient population diversity. Then, we propose an improved probability guiding method based on the idea of neighbor learning, which replace “find a nearest point from samples” to “find a point near its two neighbors from samples.” The improved probability guiding method uses a fuzzy search strategy to improve the population diversity, as is shown in Algorithm 2.

3.5 Diversity maintenance strategies

Since the EDA algorithm has no crossover and mutation operations, its population is easy to lack of diversity. Thus, two diversity maintenance strategies based on crowding distance are introduced to the MVMO-EDA.

Crowding distance is an index that can reflect the diversity of a population; it is represented by the Euclidean

distance of the current individual and its neighbors [11]. Therefore, we first propose a mutation strategy based on crowding distance in which the individuals with a small crowding distance are selected to random mutate. The pseudocode of the mutation strategy based on crowding distance is shown in Algorithm 3.

Algorithm 2: Improved offspring generation method guided by the scalable histogram probabilistic distribution model

Input: a scalable histogram probability distribution model M , a neighborhood matrix $B = [B_1, B_2, \dots, B_N]^T$ of parent population Pop .

Output: a new population Pop' .

$S \leftarrow [S_1, S_2, \dots, S_D]$ //generate D -dimensional samples with the same size as Pop by sampling from M .

for $j = 1; j \leq D; j \leftarrow j + 1$ **do**

for $i = 1; i \leq N; i \leftarrow i + 1$ **do**

$[k_1, k_2] = randperm(H, 2)$

$n_1 = B_i(k_1), n_2 = B_i(k_2)$

$\Delta d = |n_1 - n_2| * 0.5 * (rand - 0.5)$

If $rand < 0.5$ **then**

$n = round(n_1 + \Delta d)$

else

$n = round(n_2 + \Delta d)$

end

$n = check_boundary(n)$

$X_{i,j} = S_{n,j}$

end

$X'_j = [X_{1,j}, X_{2,j}, \dots, X_{N,j}]^T$

end

$Pop' = [X'_1, X'_2, \dots, X'_D]$

Algorithm 3: Mutation strategy based on crowding distance

Input: a population $X = [X_1, X_2, \dots, X_N]^T$, the mutation rate γ , the mutation probability λ .

Output: an updated population X .

$fit \leftarrow$ calculate the fitness of X

$Cd \leftarrow$ calculate the crowding distance by fit

$X = sort(X, Cd)$ % sort the population by crowding distance from large to small

for $i = 1; i \leq N\gamma; i \leftarrow i + 1$ **do**

for $j = 1; j \leq D; j \leftarrow j + 1$ **do**

if $rand < \lambda / D$ **then**

$X_{i,j} = X_{\min} + rand * (X_{\max} - X_{\min})$

end

end

end

3.6 Pseudocode of MVMO-EDA

In order to get a full understand of the MVMO-EDA, we summarize the pseudocode of the MVMO-EDA, as shown in Algorithm 5.

Algorithm 4: Selection strategy based on crowding distance and scalar fitness

Input: a population $X = [X_1, X_2, \dots, X_N]^T$, a neighborhood $B = [B_1, B_2, \dots, B_N]^T$, a new population $X' = [X'_1, X'_2, \dots, X'_N]^T$, the fitness of population fit , the fitness of new population fit_new , the scalar fitness of population $scalarfit$, the scalar fitness of new population $scalarfit_new$, the crowding distance of population Cd , the crowding distance of new population Cd_new .

Output: an updated population X .

for $i = 1; i \leq N; i \leftarrow i + 1$ **do**

for $k = 1; k \leq H; k \leftarrow k + 1$ **do** // H is the dimension of B_i

$Flag = 0$

$n = B_{i,k}$

if $X'_i \succ X_n$ **then** % if X'_i dominate X_n

$Flag = 1$

else if $scalarfit_new(i) < 0.2 * scalarfit(n)$ **then**

$Flag = 1$

else if $scalarfit_new(i) < 0.8 * scalarfit(n)$ **&&**

$Cd_new(i) > 1.2 * Cd(n)$ **then**

$Flag = 1$

else if $scalarfit_new(i) < 1.2 * scalarfit(n)$ **&&**

$Cd_new(i) > 2 * Cd(n)$ **then**

$Flag = 1$

end

if $Flag == 1$ **then**

$X_i = X'_i$

$fit(i) = fit_new(n)$

end

end

end

Secondly, a selection strategy which combines scalar fitness and crowding distance is proposed. The pseudocode of the selection strategy is shown in Algorithm 4.

Algorithm 5: MVMO-EDA

Input: an objective function $F(X)$, the boundaries of decision variables $[X_{\min}, X_{\max}]$, the dimension of decision variables D , the number of optimization objects M , the maximum number of iteration $Maxgen$.

Output: the optimized population X , the optimized objective values fit .

```

initialize algorithm parameters
 $I \leftarrow$  initialize index of discrete variables
 $X \leftarrow$  initialize population
 $W \leftarrow$  initialize weight vector
 $fit \leftarrow$  initialize fitness
for  $gen = 1; gen \leq Maxgen; gen \leftarrow gen + 1$  do
     $Y = scalarfit(fit, W)$ 
     $[L, P] \leftarrow$  construct the scalable histogram probability distribution model by using Algorithm 1
     $X' \leftarrow$  create a new population by using Algorithm 2
     $X' = mutarate(X', fit', M, \gamma, \lambda)$  //Algorithm 3
     $fit' = F(X')$ 
     $Y' = scalarfit(fit', W)$ 
     $Z \leftarrow$  update reference point  $Z$ 
     $Cd, Cd' \leftarrow$  update crowding distance
     $X = update\_neighbor(X, X')$  % Algorithm 4
     $fit = F(X)$ 
end
```

4 Experiment study

In this section, the performance of the proposed MVMO-EDA is evaluated on the modified ZDT and DTLZ benchmark sets with mixed-variable. Besides, other four mixed-variable multi-objective evolutionary algorithms are tested in the same condition for comparison. The experiments of mixed-variable multi-objective optimization are designed from three aspects: construction of benchmark function set, construction of evaluation index, and parameter settings.

4.1 Mixed-variable multi-objective optimization benchmark function set

Most of the benchmark functions for multi-objective optimization are set for continuous variables. Therefore, this research transforms the continuous multi-objective benchmark function set to be mixed-variable by quantization. We selected ten functions from the ZDT and DTLZ serious of benchmark functions and reformed to ten groups of mixed-variable benchmark functions, as shown in Table 1.

The number of discrete variables in the mixed-variable benchmark functions is calculated as follows:

$$D_1 = round(D \cdot \theta) \quad (9)$$

where D is the number of decision variables, and θ is the proportion of discrete variables. While the first to D_1 -th dimension decision variables of these functions is discretized. And then, a sampling vector Q is used to quantize these decision variables, and the sampling method is as follows:

$$X_d = X_{\min} + (X_{\max} - X_{\min}) \cdot *Q \quad (10)$$

Among them, X_d is the decision variable after quantization, and X_{\min} and X_{\max} are lower and upper boundaries of the decision variable, respectively. For a uniform quantization process, Q is marked as $Q_1 = [0.05, 0.1, 0.15, \dots, 1]$; while for the non-uniform quantization method, Q is marked as Q_2 , which calculated as follows:

$$Q_2(i) = \begin{cases} 0.01, & i = 1 \\ Q_2(1) + i(i-1) * 0.0025, & 2 \leq i \leq 20 \end{cases} \quad (11)$$

4.2 Evaluation indexes

Since the true Pareto fronts of these modified mixed-variable benchmark functions are difficult to obtain, the common evaluation indexes MS [52] and IGD [53] that rely on the true Pareto front cannot be calculated. Based on MS and IGD , this paper defines two new evaluation indexes:

Dominate Ratio (DR): suppose there are K mixed-variable multi-objective optimization algorithms to compare. We combine the optimization results of all algorithms into a solution set $U = \{U_1, U_2, \dots, U_K\}$ and then pick out the non-dominated solution set U' . After that, we find the intersection φ_i between U' and U_i , that is $\varphi_i = U_i \cap U'$, and the dominate ratio of the i -th algorithm is defined as:

$$DR_i = \frac{N_{\varphi_i}}{N_{U'}} \times 100\% \quad (12)$$

where N_{φ_i} and $N_{U'}$ are the element numbers of φ_i and U' , respectively.

DR reflects the Pareto dominance relationship among the solution sets of all comparative algorithms. The larger the value, the better performance of the evaluated algorithm is. It is a comprehensive evaluation index that not only reflects the convergence performance but also reflects the diversity characteristic of the evaluated algorithm. When the approximate Pareto fronts of two algorithms overlap with each other, then **DR** can well compare the performances of them, but when the approximate Pareto fronts of the two algorithms do not intersect, it cannot accurately evaluate the performance difference between the two algorithms.

Table 1 Benchmark function sets of MMOPs

	Original functions	Quantitative methods	Proportion of discrete variables		
			20%	50%	80%
ZDT1	Uniform	F_{1_11}	F_{1_12}	F_{1_13}	
	Non-uniform	F_{1_21}	F_{1_22}	F_{1_23}	
ZDT2	Uniform	F_{2_11}	F_{2_12}	F_{2_13}	
	Non-uniform	F_{2_21}	F_{2_22}	F_{2_23}	
ZDT3	Uniform	F_{3_11}	F_{3_12}	F_{3_13}	
	Non-uniform	F_{3_21}	F_{3_22}	F_{3_23}	
ZDT4	Uniform	F_{4_11}	F_{4_12}	F_{4_13}	
	Non-uniform	F_{4_21}	F_{4_22}	F_{4_23}	
ZDT6	Uniform	F_{5_11}	F_{5_12}	F_{5_13}	
	Non-uniform	F_{5_21}	F_{5_22}	F_{5_23}	
DTLZ1	Uniform	F_{6_11}	F_{6_12}	F_{6_13}	
	Non-uniform	F_{6_21}	F_{6_22}	F_{6_23}	
DTLZ2	Uniform	F_{7_11}	F_{7_12}	F_{7_13}	
	Non-uniform	F_{7_21}	F_{7_22}	F_{7_23}	
DTLZ3	Uniform	F_{8_11}	F_{8_12}	F_{8_13}	
	Non-uniform	F_{8_21}	F_{8_22}	F_{8_23}	
DTLZ5	Uniform	F_{9_11}	F_{9_12}	F_{9_13}	
	Non-uniform	F_{9_21}	F_{9_22}	F_{9_23}	
DTLZ6	Uniform	F_{10_11}	F_{10_12}	F_{10_13}	
	Non-uniform	F_{10_21}	F_{10_22}	F_{10_23}	

Table 2 Experiment grouping

Proportion of discrete variables	Uniform quantized	Non-uniform quantized
20%	Group 1: $F_{1_11}, F_{2_11}, \dots, F_{10_11}$	Group 2: $F_{1_21}, F_{2_21}, \dots, F_{10_21}$
50%	Group 3: $F_{1_12}, F_{2_12}, \dots, F_{10_12}$	Group 4: $F_{1_22}, F_{2_22}, \dots, F_{10_22}$
80%	Group 5: $F_{1_13}, F_{2_13}, \dots, F_{10_13}$	Group 6: $F_{1_23}, F_{2_23}, \dots, F_{10_23}$

Approximate Inverted Generational Distance (AIGD): If the above non-dominated solution set U' is substituted for the true Pareto front in the definition of IGD , then the calculated value is called approximate inverted generational distance. The calculation formula of $AIGD$ is as follows:

$$AIGD = \frac{\sum_{v \in PF} d(v, U'_i)}{|PF|} \quad (13)$$

where PF represents the true Pareto Front set, U'_i is the i -th individual of U' , v is the nearest individual to U'_i in PF , and $d(v, U'_i)$ is the Euclidean distance between v and U'_i .

$AIGD$ is also a comprehensive evaluation index, which can reflect both convergence and diversity performance of the algorithm. The smaller the $AIGD$, the better performance of the evaluated algorithm is. When the approximate Pareto fronts of two algorithms do not intersect, then $AIGD$ can well reflect the performance difference between the two algorithms, but when the approximate Pareto fronts

of two algorithms are overlapped with each other, then it is difficult to compare them by $AIGD$.

Therefore, DR and $AIGD$ are two complementary evaluation indexes, and their can well evaluate the comprehensive performance of the mixed-variable multi-objective evolutionary algorithms.

4.3 Experimental settings

The experimental settings are introduced from three aspects: experimental grouping, selection of comparative algorithms, and parameter settings.

Experimental grouping: According to the quantization method and the proportion of discrete variables, the experiments are classified into six groups, as shown in Table 2.

Comparative algorithms: As mentioned in Section 2, there are three categories of mixed-variable multi-objective evolutionary algorithms. The first category is to treat the MMOPs as continuous variable multi-objective

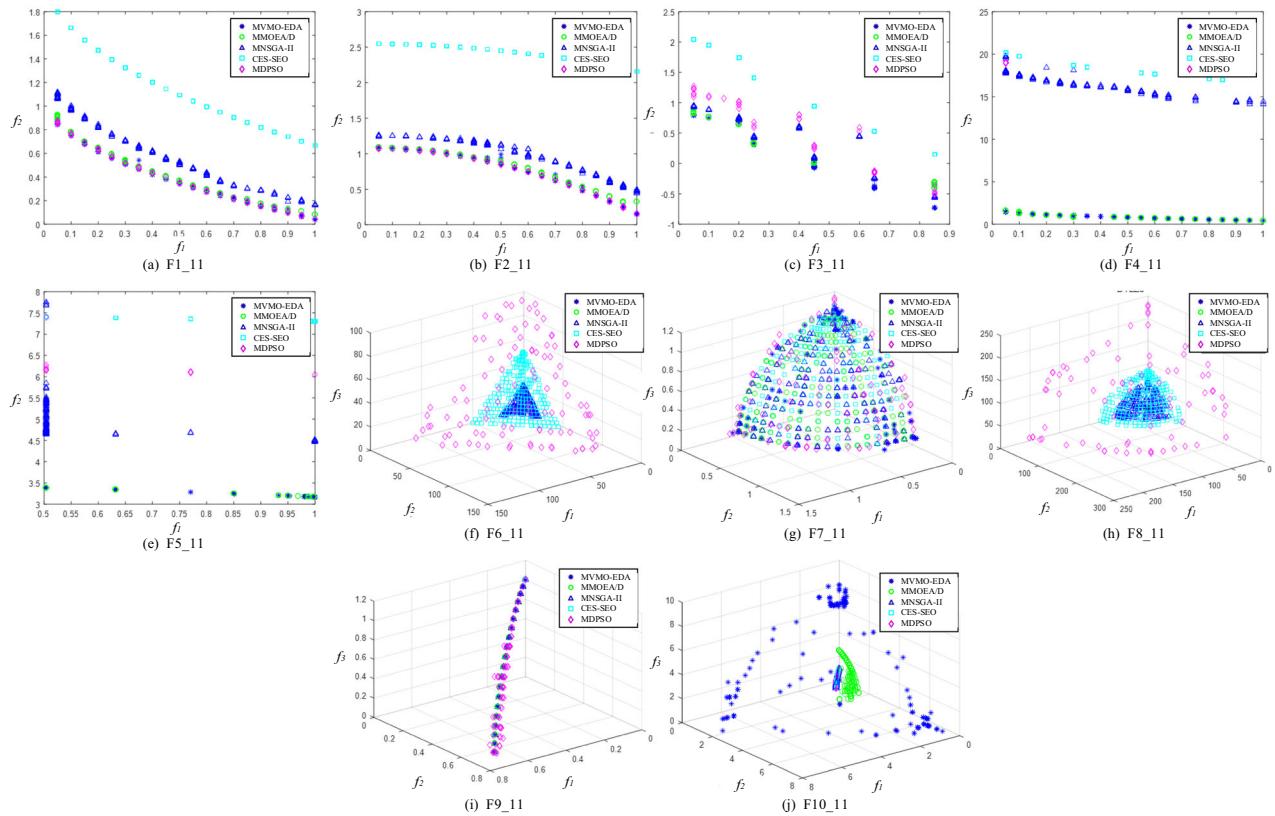


Fig. 6 The approximates Pareto fronts of mixed-variable multi-objective optimization (20% discrete variables, uniform quantized)

optimization problems and then quantify the optimization results. However, the algorithms by using this approach in Sect. 2 are not suite for multi-objective optimization. Then, we modified NSGA-II and MOEA/D to mixed-variable NSGA-II (MNSGA-II) and mixed-variable MOEA/D (MMOEAD), respectively, through this approach. The second category is to embed the quantization operation into the coding, crossover, mutation and other operations of evolutionary algorithms. For this approach, we select MDPSO [27] and CES-SEO [29] as the comparative algorithms. The third category is by using EDAs, but there is no report on EDA-based mixed-variable multi-objective evolutionary algorithm except for the proposed MVMO-EDA. So the four algorithms MMOEA/D, MNSGA-II, MDPSO and CES-SEO are selected as the comparative algorithms of the proposed MVMO-EDA.

Parameter settings: In order to ensure the fairness of experiments, common parameter settings of the five algorithms need to be the same: the population size is set to 220; for the ZDT-based benchmark functions, the iteration number is set to 100, the objective number is set to 2, and the dimension of the decision variable is set to 30; while for the DTLZ-based benchmark functions, the three above-

mentioned parameters are set to be 1000, 3, 12, respectively.

4.4 Experimental results and analysis

Performances of the proposed MVMO-EDA are evaluated in the six groups of modified benchmark functions. In order to comprehensively evaluate the performance of MVMO-EDA, this subsection analyzes the experimental results from three aspects: approximate Pareto front, evaluation index, and operating efficiency.

4.4.1 Approximate pareto front analysis

The approximate Pareto fronts of the five algorithms in six groups of experiments are shown in Figs. 6, 7, 8, 9, 10, and 11. It can be seen from the six figures that the approximate Pareto fronts of the mixed-variable functions are also discrete. In all six figures, the approximate Pareto fronts of MVMO-EDA have advantages in these functions: F_{1_mn} , F_{2_mn} , F_{3_mn} , F_{4_mn} , F_{5_mn} , F_{6_mn} , and F_{8_mn} , where, $m = \{1, 2\}$, $n = \{1, 2, 3\}$. While the approximate Pareto fronts in F_{7_mn} and F_{9_mn} are difficult to compare, and

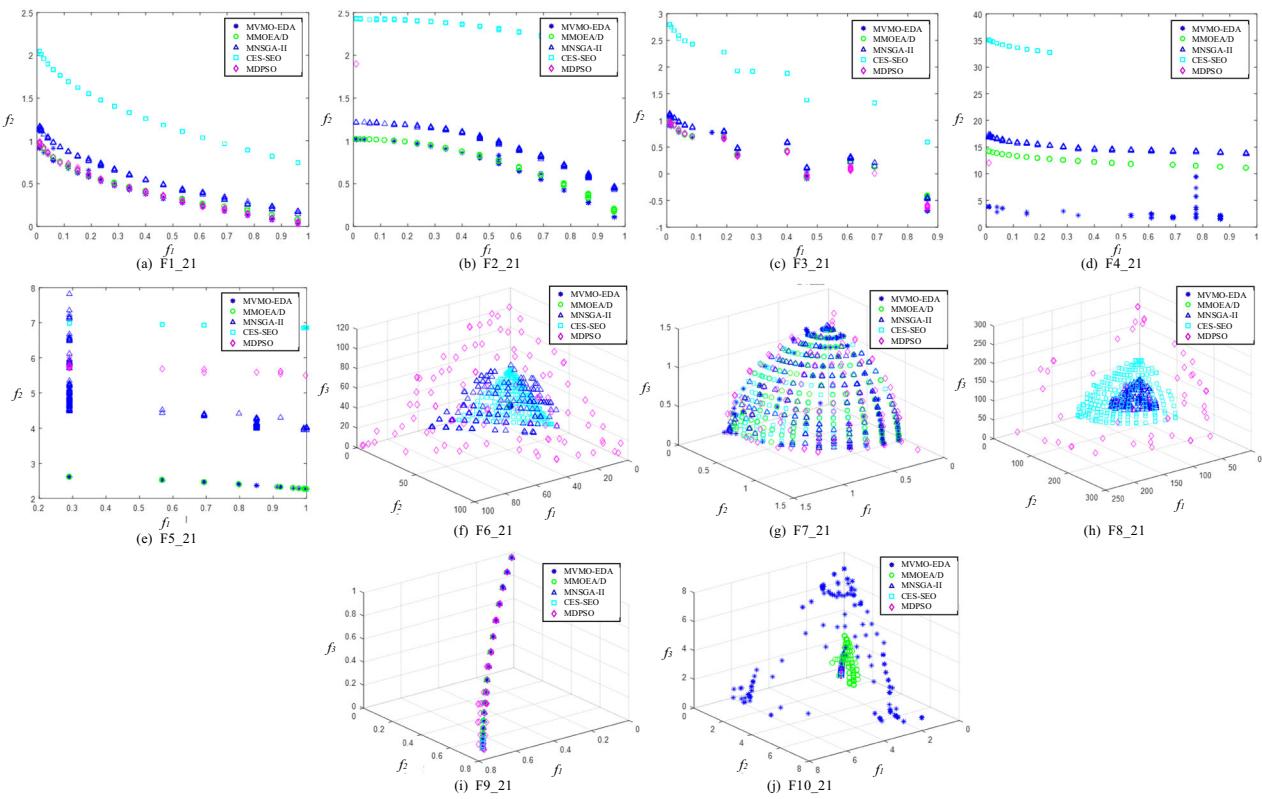


Fig. 7 The approximates Pareto fronts of mixed-variable multi-objective optimization (20% discrete variables, non-uniform quantized)

further evaluation index analysis is required. Only in F_{10_mn} , the approximate Pareto fronts of MVMO-EDA are inferior to that of the other four algorithms. Statistically, the approximate Pareto front of the MVMO-EDA is dominant in 70% of benchmark functions, required more accurate index evaluation in 20% of the functions, and is worse than that of the other four algorithms in only 10% of the functions.

Through further analysis, it is found that the reason for the poor convergence of the MVMO-EDA in F_{10_mn} series of functions is that the prototype function of F_{10_mn} is DTLZ6. And the true Pareto front of DTLZ6 is extremely narrow in the decision space. While MVMO-EDA is based on the principle of statistics and sampling, it is difficult to get the point on the true Pareto front of DTLZ6-based functions through sampling. So the convergence of MVMO-EDA on the F_{10_mn} series of functions is poor.

Comparing Figs. 6, 8, 10 with Figs. 7, 9, 11, it can be found that in most cases, the approximate Pareto fronts of the uniform quantized functions are smaller than that of corresponding non-uniform quantized functions when using the same algorithm. That is to say, the quantization method of MMOPs has an impact on the performance of mixed-variable multi-objective evolutionary algorithms,

and the difficulty of optimizing the non-uniform quantized functions is generally greater than that of optimizing the uniform quantized functions.

Comparing Figs. 7, 8 and 10, it is found that in most cases, the approximate Pareto fronts in the first five ZDT-based functions decrease as the proportion of discrete variables increases when using the same algorithm, but the approximate Pareto fronts of the last five DTLZ-based functions get the opposite rule. Comparing Figs. 7, 9, and 11, it is found that there is no obvious regularity in the approximate Pareto fronts of the first five ZDT-based functions, but the approximate Pareto fronts of the last five DTLZ-based functions increase as the proportion of discrete variables increases when using the same algorithm. It shows that the optimizing difficulty of MMOPs with the proportion of discrete variables has no obvious law.

4.4.2 Evaluation index analysis

In order to further analyze the performance of the five mixed-variable multi-objective evolutionary algorithms, the DR and AIGD indexes of these five algorithms are calculated for an in-depth analysis. Each group of experiments was repeated 20 times independently, and then, the

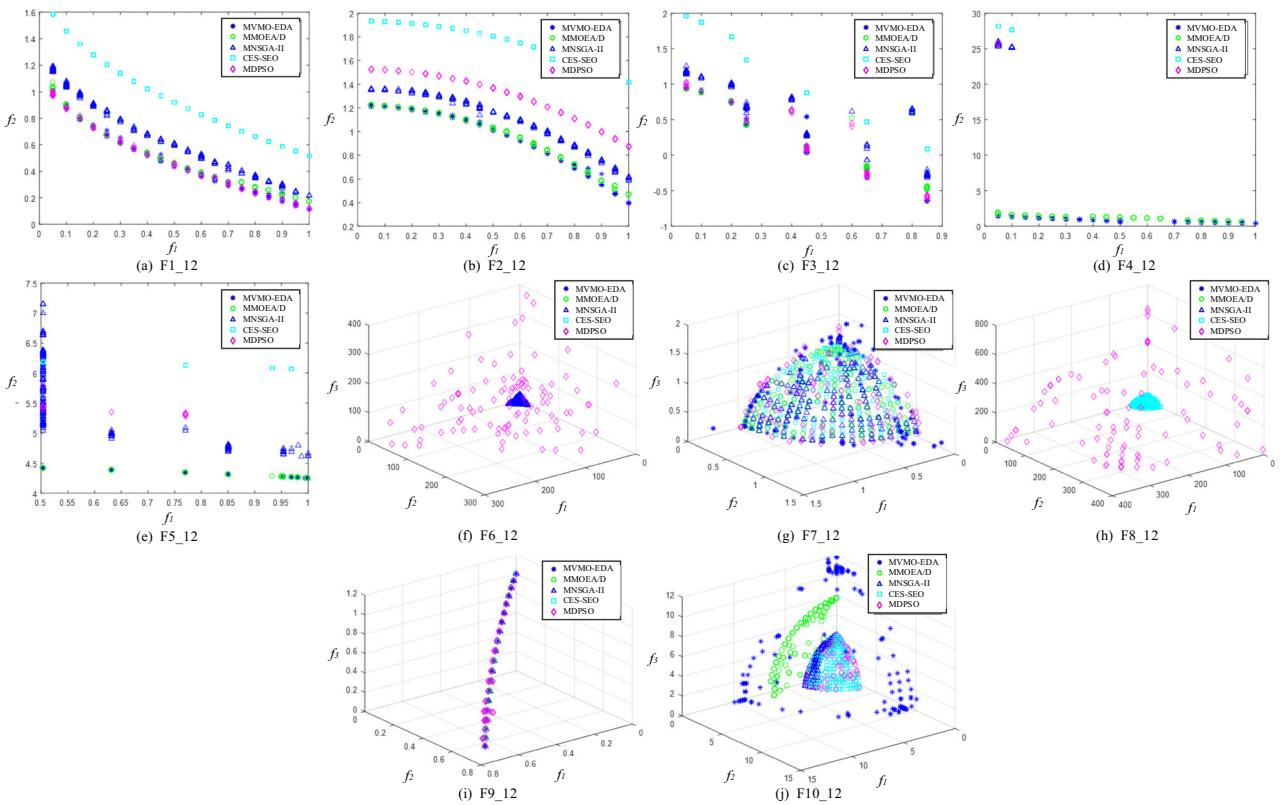


Fig. 8 The approximates Pareto fronts of mixed-variable multi-objective optimization (50% discrete variables, uniform quantized)

mean values of *DR* and *AIGD* are calculated for the five algorithms, as shown in Figs. 12 and 13, respectively.

It can be seen from Fig. 12 that the *DR* of the MVMO-EDA is significantly better than the other four comparative algorithms in all six groups of experiments. According to statistics, the MVMO-EDA has the largest *DR* index in 39 functions among the 60 test functions of the six groups of experiments, which accounts for 65%, while that proportion of MMOEA/D, MNSGA-II, CES-SEO, and MDPSO are 11.67%, 6.67%, 18.33%, and 0%, respectively. Additionally, in the three uniform quantized groups of experiments, the proportion of MVMO-EDA with the greatest *DR* is 60%, while in the three non-uniform quantized groups, the proportion raised to 70%.

This shows that the advantage of MVMO-EDA in the *DR* index is more significant in the non-uniform quantized functions. Finally, by comparing Fig. 12a–c and Fig. 12(d), (e), (f), respectively, it is found that with the increasing proportion of discrete variables, the *DR* indexes of most algorithms are decreasing, which indicate that the optimization difficulty is also increased.

It should be noted that since the *AIGD* of the five algorithms differs greatly, the ordinate value in Fig. 13 is the logarithm result of *AIGD* to 10 to get a better display. It

can be seen from Fig. 13 that the *AIGD* index of MVMO-EDA is also significantly better than that of the other four comparative algorithms in all six groups of experiments. According to statistics, among 60 test functions of the six groups of experiments, the MVMO-EDA has the smallest *AIGD* index in 36 functions, which accounts for 60%, while that proportion of MMOEA/D, MNSGA-II, CES-SEO, and MDPSO are 15%, 10%, 15%, and 8.33%, respectively. Additionally, the proportion of MVMO-EDA with the smallest *AIGD* is 63.33% in the three group's uniform quantized experiments, while the proportion reduced to 56.67% in the three non-uniform quantized groups, which shows that the advantage of the MVMO-EDA in the *AIGD* index is more significant in the uniform quantized functions. Finally, it is found that the *AIGD* in the experiments with 20% discrete variables (Fig. 13a and d) is overall smaller than that of experiments with 50% and 80% discrete variables.

Through the above evaluation index analysis, it is found that the MVMO-EDA is significantly better than the other four comparative algorithms in both *DR* and *AIGD* index of all the six groups of experiments. Meanwhile, it is also found that in most cases, the higher proportion of discrete variables, the worse *DR* and *AIGD* indexes, indicating that

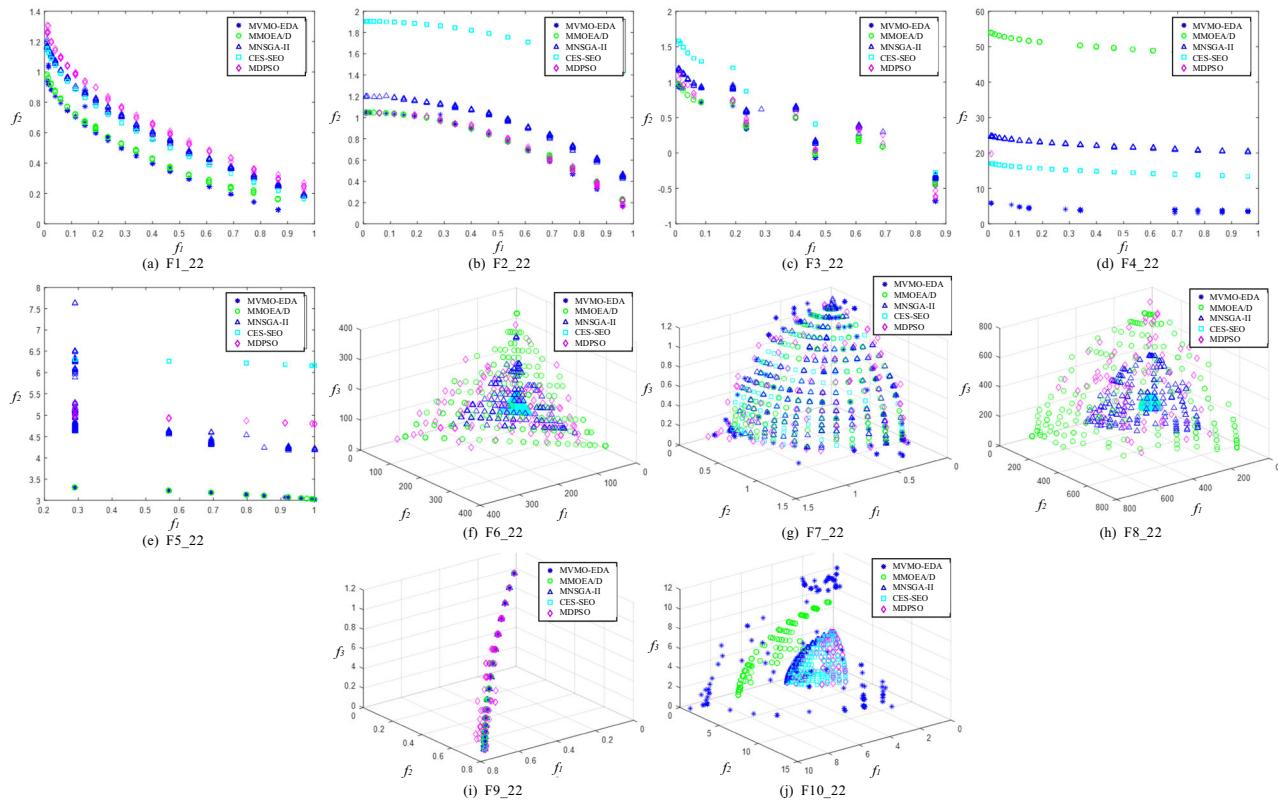


Fig. 9 The approximates Pareto fronts of mixed-variable multi-objective optimization (50% discrete variables, non-uniform quantized)

the difficulty of algorithm's optimization is rising with the proportion of discrete variables.

4.4.3 Operation efficiency analysis

We first analyze the computational complexity of the five algorithms. The main computation of MVMO-EDA is consumed in Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4. According to the flow of these four algorithms, the computational complexity of Algorithm 1 is $O(DK)$, the computational complexity of Algorithm 2 is $O(DN)$, the computational complexity of Algorithm 3 is $O(DN\gamma)$, and that of Algorithm 4 is $O(NH)$, and then, the computational complexity of MVMO-EDA is $O(DN\gamma)$. According to reference [12], the computational complexity of MOEA/D and NSGA-II is $O(DNT)$ and $O(DN^2)$, respectively, and the computational complexity of MMOEA/D is the same as that of MOEA/D, and the computational complexity of MNSGA-II is the same as that of NSGA-II. According to the analysis of the literature [29], the computational complexity of CES-SEO is the same as that of NSGA-II, which is also $O(DN^2)$. According to reference [27], we analysis that the computational complexity of MDPSO is $O(DN\gamma)$. It can be seen that the

computational complexity of the five algorithms is in the same order of magnitude.

Then, we analyze the computational efficiency of the five algorithms experimentally. All the experiments in this paper are carried out in the PC environment of Win10 64 bit operating system, Intel i7-6700 CPU (3.40 GHz) and 8 GB RAM. The running time of the five algorithms in a group of experiments (discrete variables accounted for 50%, uniform quantization) was recorded, as shown in Table 3. It can be seen from Table 3 that the running time of the five algorithms is at the same level of tens of seconds, and the shortest running time is 30.101 s for the MNSGA-II, followed by 34.089 s for the MVMO-EDA. In addition, the running time of CES-SEO and MMOEA/D is also at the same level, while MDPSO has the longest time. The running speed of MVMO-EDA is slightly faster than that of the MMOEA/D algorithm. The reason is that the offspring of MVMO-EDA is generated by the probabilistic guiding method which replaces the complicated SBX operation of MMOEA/D.

To sum up, through the experimental analysis in six groups of mixed-variable multi-objective optimization experiments, we get the following conclusions:

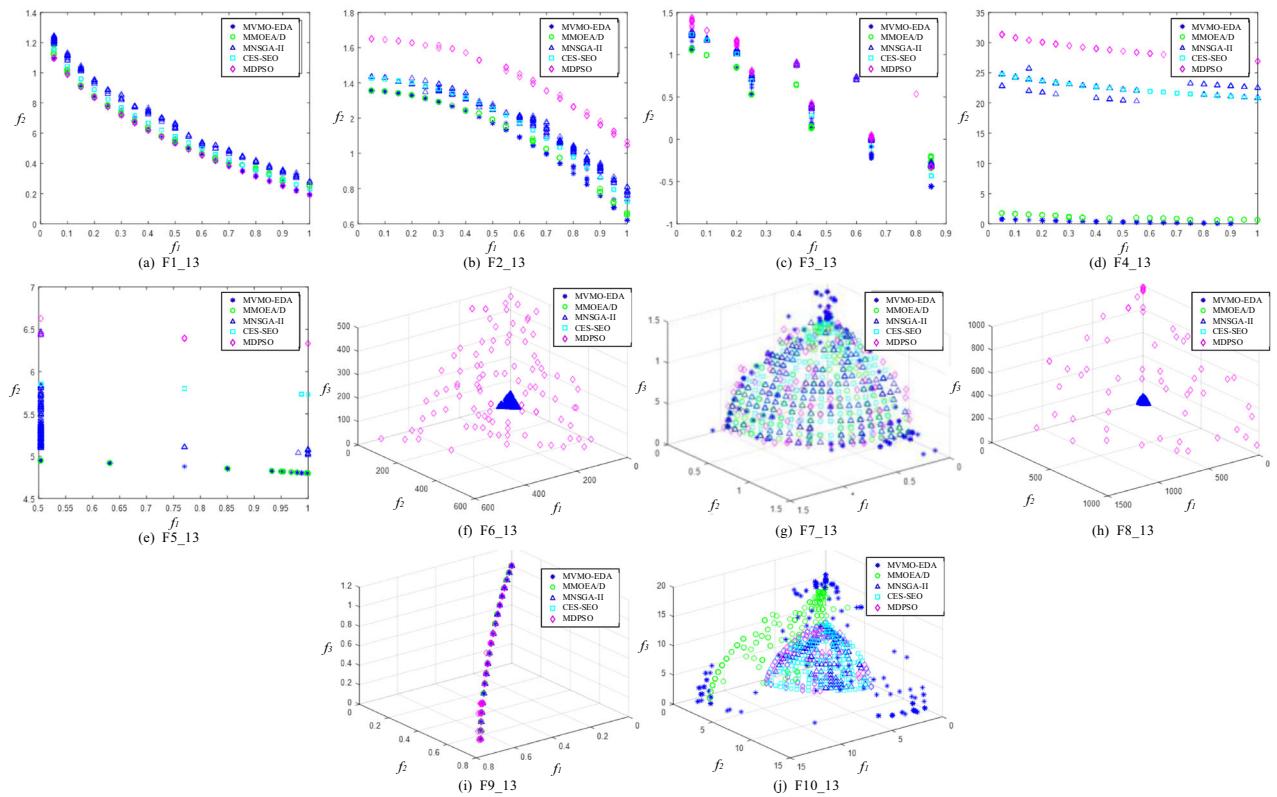


Fig. 10 The approximates Pareto fronts of mixed-variable multi-objective optimization (80% discrete variables, uniform quantized)

1. The convergence performance of the proposed MVMO-EDA is significantly better than that of the other four comparative algorithms in all groups of experiments. This is because the MVMO-EDA algorithm adopts the working principle of statistics and sampling, which can avoid the quantization error caused by the crossover and quantization operations of the traditional evolutionary algorithm, thus achieving higher accuracy.
2. MVMO-EDA is an efficient MMOEA, ranking the second fastest in the average running time test. The reason is that the MVMO-EDA algorithm uses an index coding method for discrete variables, which improves the coding efficiency of discrete variables and speeds up the search process of discrete variables.
3. From the optimization results of the DTLZ series of functions with 30-dimensional decision variables, it can be seen that the MVMO-EDA algorithm has the ability to optimize high-dimensional problems. This shows that the proposed probabilistic guiding method can indeed improve the high-dimensional optimization capability of the EDAs.
4. The proportion of discrete variables and the quantization method have an impact on the optimization difficulty of MMOPs; in most cases, the difficulty

increases with the proportion of discrete variables, while the optimization difficulty of non-uniform quantized functions is generally greater than that of uniform quantized.

5 Conclusions and future work

This paper proposes a MVMO-EDA for the MMOPs to overcome the problems of quantification error and low efficiency that traditional multi-objective evolutionary algorithms faced. The MVMO-EDA inherits the high efficiency of the MOEA/D framework, gives full play to the advantages of the EDA without crossover operation, and has no quantization errors when optimizing the discrete variables. It also overcomes the shortcomings of the basic EDA such as poor global search ability and difficulty in optimizing high-dimensional problems. In all six groups of experiments, the MVMO-EDA is significantly better than the other four comparative algorithms in terms of *DR* and *AIGD* indexes. And its running speed is also ranked second among the five algorithms in the operation efficiency test experiments. This shows that the MVMO-EDA is a

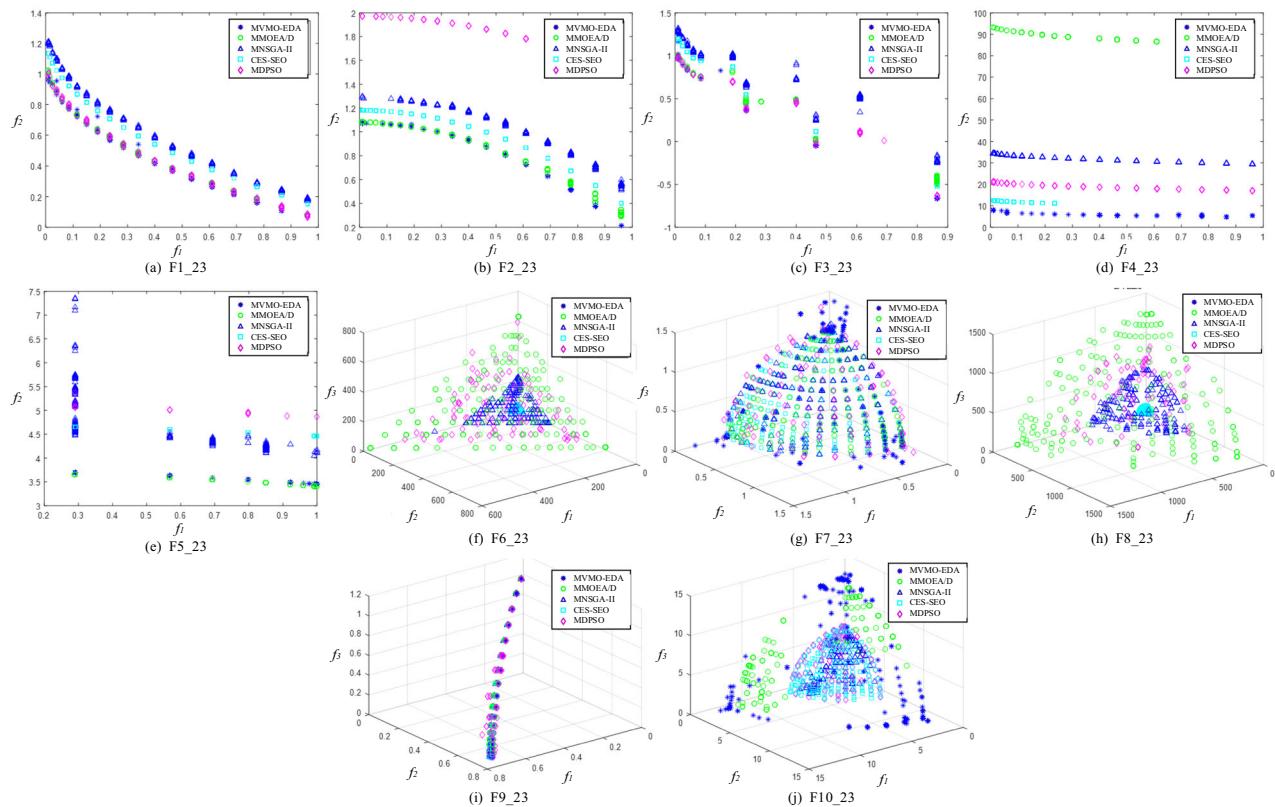


Fig. 11 The approximates Pareto fronts of mixed-variable multi-objective optimization (80% discrete variables, non-uniform quantized)

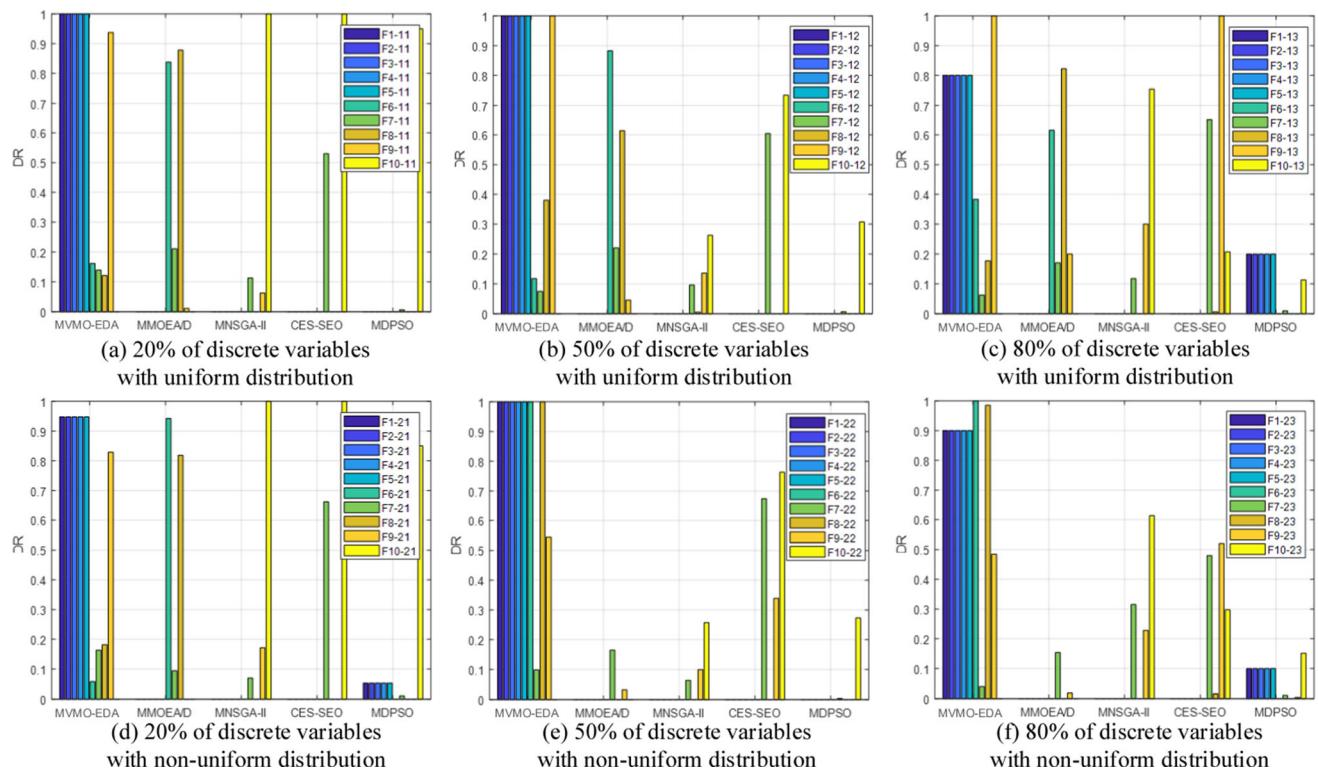


Fig. 12 Statistical chart of DR indexes

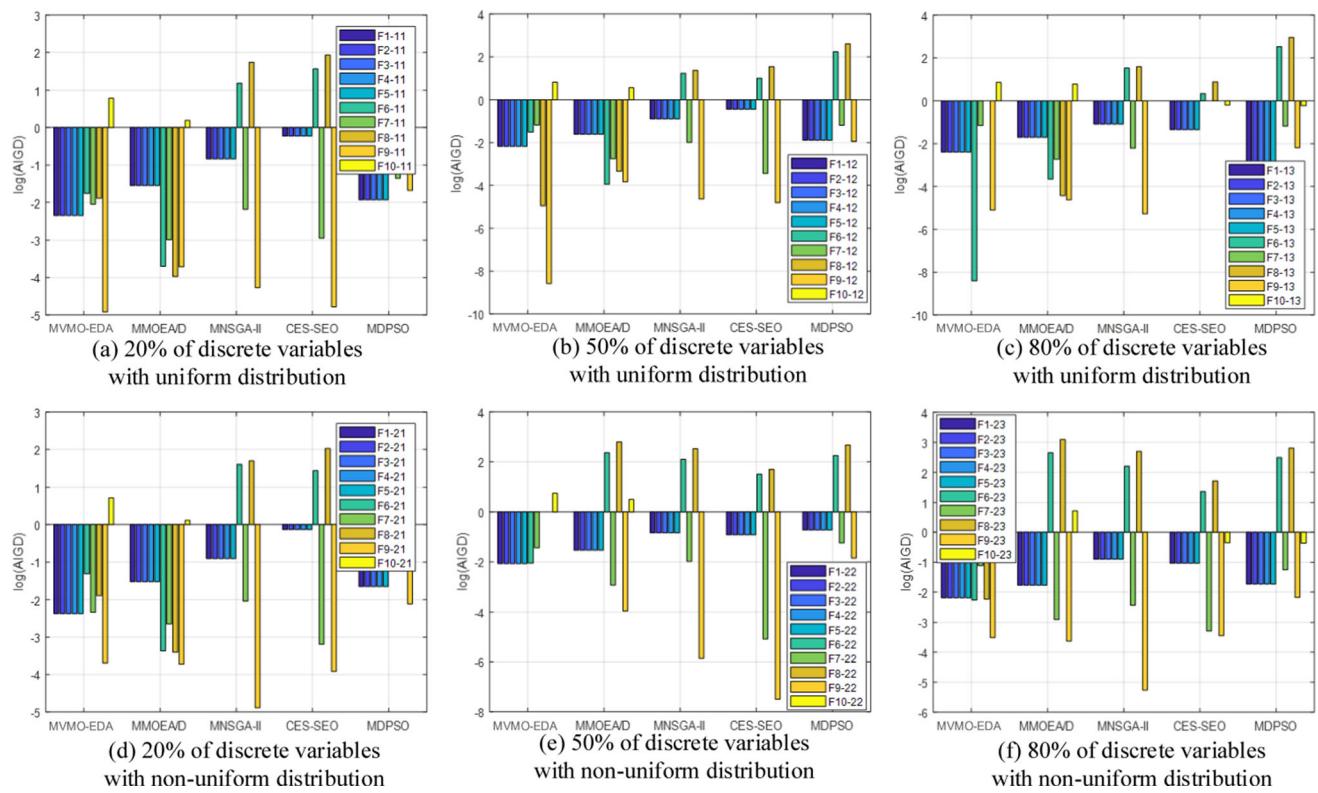


Fig. 13 Statistical chart of AIGD indexes (note: the figures show the logarithm result of AIGD for a better display)

Table 3 Running time statistics of five algorithms (discrete variables account for 50%, uniform quantified)

Function number	Running time/s				
	MVMO-EDA	MMOEAD	MNSGA-II	CES-SEO	MDPSO
F1_12	5.975	8.036	4.011	10.939	7.075
F2_12	5.959	8.329	7.241	13.839	4.971
F3_12	5.963	8.111	7.760	18.552	5.881
F4_12	5.387	7.997	12.587	18.079	4.861
F5_12	5.155	7.868	12.304	20.566	3.571
F6_12	61.735	78.136	73.389	81.902	176.27
F7_12	61.663	77.984	25.767	43.625	164.747
F8_12	61.723	78.029	72.897	82.286	86.455
F9_12	61.641	77.004	27.336	50.692	133.519
F10_12	65.693	78.411	57.718	52.745	374.918
Average running time	34.089	42.990	30.101	39.323	96.227

competitive mixed-variable multi-objective evolutionary algorithm.

However, there are still several tasks that need further research in the future: (1) the MVMO-EDA is not well performed in some special function which Pareto front is extremely narrow in decision space, such as DTLZ6; and (2) the parameter adaptive strategy of MVMO-EDA is also required in the future.

Funding This work was supported by the Key Field Special Project of Guangdong Provincial Department of Education with No.2021ZDZX1029. And it was also supported by Guangdong Basic and Applied Basic Research Foundation under No. Grant 2022A1515011447.

Declarations

Conflict of 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, “An Improved Estimation of Distribution Algorithm for Multi-objective Optimization Problems with Mixed-Variable.”

References

- Feyzioglu A, Kar AK (2017) Axiomatic design approach for nonlinear multiple objective optimizaton problem and robustness in spring design. *Cybern Inf Technol* 17(1):63–71
- Geoffrion AM (1968) Proper efficiency and the theory of vector maximization. *J Math Anal Appl* 22(3):618–630
- Charnes A, Cooper WW (1957) Management models and industrial applications of linear programming. *Manage Sci* 4(3):274–275
- Haimes YY, Lasdon LS, Wismer DA (1971) On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Trans Syst Man Cybern* 1(3):296–297
- Ben-Tal A (1980) Characterization of pareto and lexicographic optimal solutions. Springer 177:1–11
- Dantzig G, Fulkerson R, Johnson S (1954) Solution of a large-scale traveling salesman problem. *Oper Res* 2(4):393–410
- Gomory RE (1958) Outline of an algorithm for integer solutions to linear programs. *Bull Am Math Soc* 64(5):275–278
- Land AH, Doig AG (1960) An automatic method for solving discrete programming problems. Springer 28(3):497–520
- Balas E (1979) Disjunctive Programming. *Annals of Discrete Mathematics* 5(2): 3–51
- Schaffer JD (1985) Some experiments in machine learning using vector evaluated genetic algorithms. *Electr Eng* 3:31–37
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Int Conf Parallel Problem Solv from Nat* 1917:849–858
- Zhang Q (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans on Evolutionary Comput* 11(6):712–731
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. *Tech Rep Gloriastrasse* 5:1–21
- Mz A, Lei WA, WI B, Bo HA, DI A, Qw A (2021) Many-objective evolutionary algorithm with adaptive reference vector. *Inf Sci* 563:70–90
- Gu Q, Wang R, Xie H, Li X, Xiong N (2021) Modified non-dominated sorting genetic algorithm iii with fine final level selection. *Appl Intell* 51(5):1–34
- Jiang S, Yang S (2017) Evolutionary dynamic multiobjective optimization: benchmarks and algorithm comparisons. *IEEE Trans Cybern* 47(1):198–211
- Tong W, Chowdhury S, Messac A (2016) A multi-objective mixed-discrete particle swarm optimization with multi-domain diversity preservation. *Struct Multidiscip Optim* 53(3):1–18
- Shenlin Y, Wei L, Xiaochao Q et al (2019) Multi objective optimization of system effectiveness based on complex simulation. *Control Dec* 36(3):1–10
- Larranga P, Lozano JA (2001) Estimation of distribution algorithms: a new tool for evolutionary computation. *Kluwer Acad Publ* 2:344–360
- Xuejun Z, Deng P, Anling W et al (2000) Implementation of Pareto genetic algorithm for mixed variable multi-objective optimization design. *J Shanghai Jiaotong Univ* 34(3):411–414
- Chuang-Xin G, Jia-Sheng H, Bin Y, Yi-Jia C (2004) Swarm intelligence for mixed-variable design optimization. *Zhejiang Univ* 5(7):851–860
- Zhou J, Dong SB, Tang DY (2018) Task scheduling algorithm in cloud computing based on invasive tumor growth optimization. *Chin J Comput* 41(6):1140–1155
- Dhingra AK, Lee BH (2010) A genetic algorithm approach to single and multiobjective structural optimization with discrete-continuous variables International. *J for Numerical Methods Eng* 37(23):4059–4080
- Chen J, Chen G, Guo W (2009) A discrete PSO for multi-objective optimization in VLSI floorplanning. *Adv Comput Intell* 5821:400–410
- Khokhar ZO, Vahabzadeh H, Ziai A, Wang G, Menon C (2010) On the performance of the pareto set pursuing (PSP) method for mixed-variable multi-objective design optimization. *J Mech Des* 132(7):1–11
- Chowdhury S, Jie Z, Messac A (2012) Avoiding Premature convergence in a mixed-discrete particle swarm optimization (MDPSO) algorithm. *Struct Struct Dyn Mater Conf* 1678:1–20
- Mokarram V, Banan MR (2018) A new pso-based algorithm for multi-objective optimization with continuous and discrete design variables. *Struct Multidipl Optim* 57(2):509–533
- Yang SL et al (2021) Multi-objective optimization of system effectiveness based on complex and expensive simulation. *Control Dec* 36(3):589–598
- Wang F, Li Y, Zhou A, Tang K (2019) An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Trans Evol Comput* 99:1–15
- Muhlenbein H, Paass G (1996) From recombination of genes to the estimation of distributions I binary parameters. *Parallel Problem Solving From Nat -PPSN IV* 1141(19):178–187
- Hauschild M, Pelikan M (2011) An introduction and survey of estimation of distribution algorithms. *Swarm Evol Comput* 1(3):111–128
- Baluja S, Caruana R (1995) Removing the genetics from the standard genetic algorithm. *Machine Learning Proceedings 1995*. Elsevier, pp 38–46. <https://doi.org/10.1016/B978-1-55860-377-6.50014-1>
- Muhlenbein H (1997) The equation for response to selection and its use for prediction. *Evolut Comput* 5(3):303–346
- Harik G (1998) The compact genetic algorithm. In: *Proceedings of the IEEE Conference on Evolutionary Computation*, 3(4): 287–297
- Harik GR, Lobo FG, Sastry K (2006) Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ECGA). *Scalable Optimiz via Prob Model* 33:39–61
- Muhlenbein H, Mahnig T (1999) FDA—a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evol Comput* 7(4):353–376
- Pelikan M (1999) BOA: The Bayesian optimization algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp 525–532
- Bonet J, Jr C, Viola PA (1997) MIMIC: Finding Optima by Estimating Probability Densities. In: *Advances in Neural Information Processing Systems*, pp 424–430
- Baluja S, Davies S (1997) Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space. In: *Proceedings of the 14th International Conference on Machine Learning*, pp 30–38
- Pelikan M, Muehlenbein H, et al (1998) The bivariate marginal distribution algorithm. In: *advances in soft computing - engineering design and manufacturing*, pp 521–535

42. Zhong J, Zhang J, Fan Z (2010) MP-EDA: a robust estimation of distribution algorithm with multiple probabilistic models for global continuous optimization. *Adv Neural Inf Process Syst* 6457:85–94
43. MiquélezBengoetxea TE, Larrañaga P (2004) Evolutionary computations based on bayesian classifiers. *Int J Appl Math Comput* 14:101–115
44. Dong W, Chen T, Tino P et al (2013) Scaling up estimation of distribution algorithms for continuous optimization. *IEEE Trans Evol Comput* 17(6):797–822
45. Sanyang ML, Kabán A (2019) Large-scale estimation of distribution algorithms with adaptive heavy tailed random projection ensembles. *J Comput Sci Technol* 34(6):1241–1257
46. Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput* 12(1):41–63
47. Wang Y, Xiang J, Cai Z (2012) A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator. *Appl Soft Comput* 12(11):3526–3538
48. Yin ZA, Ggw A, Kl B, Wcy C, Mj D, Id A (2020) Enhancing moea/d with information feedback models for large-scale many-objective optimization. *Inf Sci* 522:1–16
49. Li QQ, Chu QX, Chang YL, Dong J (2019) Tri-objective compact log-periodic dipole array antenna design using MOEA/D-GPSO. *IEEE Trans Antennas Propag* 68(4):2714–2723
50. Wang F, Liao F, Li Y, Wang H (2021) A new prediction strategy for dynamic multi-objective optimization using Gaussian mixture model. *Inf Sci* 580:331–351
51. Peng X, Gao X (2009) A hybrid multi-objective optimal approach to multiple ucavs coordinated planning. *Int Conf Intell Human-Mach Syst Cybern* 2:23–28
52. Adra FS, Fleming JP (2011) Diversity management in evolutionary many-objective optimization. *IEEE Trans Evol Comput* 15(2):183–195
53. Czyżak P, Adrezej J (1998) Pareto simulated annealing: a metaheuristic technique for multiple-objective combinatorial optimization. *J Multi Criteria Dec Anal* 7(1):34–47

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.