



OPEN

Evolution algorithm with adaptive genetic operator and dynamic scoring mechanism for large-scale sparse many-objective optimization

Xia Wang^{1,2✉}, Wei Zhao^{1,2}, Jia-Ning Tang^{1,2✉}, Zhong-Bin Dai³ & Ya-Ning Feng^{1,2}

Large-scale sparse multi-objective optimization problems are prevalent in numerous real-world scenarios, such as neural network training, sparse regression, pattern mining and critical node detection, where Pareto optimal solutions exhibit sparse characteristics. Ordinary large-scale multi-objective optimization algorithms implement undifferentiated update operations on all decision variables, which reduces search efficiency, so the Pareto solutions obtained by the algorithms fail to meet the sparsity requirements. SparseEA is capable of generating sparse solutions and calculating scores for each decision variable, which serves as a basis for crossover and mutation in subsequent evolutionary process. However, the scores remain unchanged in iterative process, which restricts the sparse optimization ability of the algorithm. To solve the problem, this paper proposes an evolution algorithm with the adaptive genetic operator and dynamic scoring mechanism for large-scale sparse many-objective optimization (SparseEA-AGDS). Within the evolutionary algorithm for large-scale Sparse (SparseEA) framework, the proposed adaptive genetic operator and dynamic scoring mechanism adaptively adjust the probability of cross-mutation operations based on the fluctuating non-dominated layer levels of individuals, concurrently updating the scores of decision variables to encourage superior individuals to gain additional genetic opportunities. Moreover, to augment the algorithm's capability to handle many-objective problems, a reference point-based environmental selection strategy is incorporated. Comparative experimental results demonstrate that the SparseEA-AGDS algorithm outperforms five other algorithms in terms of convergence and diversity on the SMOP benchmark problem set with many-objective and also yields superior sparse Pareto optimal solutions.

Keywords Large-scale sparse evolutionary algorithms, Adaptive genetics, Dynamic scoring, Sparsity of Pareto solutions, Many-objective

A multitude of Multi-Objective Optimization Problems (MOPs)¹ exist in both real-world and industrial production. When decision variable numbers are hundreds or thousands, such problems are classified as Large-Scale Multi-Objective Optimization Problems (LSMOPs)². Instances include the optimization of millions of weights in deep neural networks³, unsupervised learning for data clustering⁴, network node issues in community detection⁵, vehicle routing problems⁶, and so on. More and more practical applications of LSMOPs need to be solved. Within these applications, a subset of LSMOPs is characterized by sparse Pareto optimal solutions, i.e., most decision variables of the optimal solutions are zero⁷. For instance, in community network node detection, nodes within the same community must be densely interconnected, whereas nodes across different communities should be sparsely linked⁵. In neural network training, a significant number of weights are set to zero to minimize both training error and model complexity⁸. Similarly, portfolio optimization problem is to maximize return and minimize risks, only a small fraction of all possible investments is selected⁹. These Large-Scale Sparse Multi-Objective Optimization Problems (LSSMOPs)¹⁰ are widespread in practical applications, and their study holds substantial practical significance.

¹School of Electrical and Information Technology , Yunnan Minzu University, Kunming 650504, China. ²Yunnan Key Laboratory of Unmanned Autonomous System , Yunnan Minzu University, Kunming 650504, China. ³Nanjing Branch of China Telecom Co., Ltd, Nanjing 210000, China. ✉email: wangxiacsu@163.com; tjin1216@163.com

During the past decades, a main focus of evolutionary algorithms has been to solve LSMOPs^{11–13}. A number of multi-objective evolutionary algorithms for solving LSMOPs have been developed^{14–17}, which can be roughly divided into three categories. The first category uses divide-and-conquer strategies¹⁸ to divide decision variables into several groups, and then optimizes each group of decision variables alternately. The second category adopts transformation strategies¹⁹ to transform the original problems into low-dimensional sub-problems, hence reducing the dimensionality of the decision variable. The last category employs novel search strategies²⁰ to effectively generate offspring solutions. In spite of the effectiveness of these strategies on LSMOPs, they cannot provide good optimization performance for LSMOPs, as they do not consider the sparse characteristic of Pareto optimal solutions^{21–23}. In LSSMOPs, only a few key decision variables determine the quality of the solution and the values of remaining decision variables should be zero in Pareto optimal solutions⁷. Ordinary large-scale multi-objective optimization algorithms implement undifferentiated update operations on all decision variables, which not only wastes computing resources and reduces search efficiency, but also makes the Pareto solutions obtained by the algorithms fail to meet the sparsity requirements, so it is difficult to obtain high-quality solutions.

Recently, some algorithms are especially designed for LSSMOPs. These algorithms can be broadly categorized into two classes¹¹. The first class consists of methods based on prior knowledge. Tian et al.⁷ introduced SparseEA in 2019, which employs a new initialization strategy and genetic operator to ensure solution's sparsity base on prior knowledge of decision variables. In 2021, Tan et al.²⁴ proposed MDRSAEA, which obtains prior knowledge of decision variables based on non-dominated sorting of feature selection and utilizes this knowledge to perform multi-stage dimension reduction on the search space. Ding et al.¹¹ proposed MSKEA in 2022, which introduces three different types of sparsity knowledge to estimate the sparse distribution of Pareto optimal solutions, and ultimately maintains a balance between exploration and exploitation. In 2022, Wang et al.²⁵ proposed ST-CCPSO, which uses prior knowledge of decision variables, and employs cumulative gradient values as the criterion for setting decision variables to zero. It ultimately achieves a balance between exploration and exploitation through a competitive particle swarm optimizer. Shao et al.²⁶ proposed NUCEA in 2023, which introduces a non-uniform clustering to divide binary variables into groups of different sizes based on prior knowledge of bi-level encoding of decision variables, so the sparsity of the algorithm could be controlled by optimizing binary variables. Another class of approaches is to mine the sparse distribution of Pareto optimal solutions during the evolution. Tian et al.²⁷ proposed MOEA/PSL in 2020, and two unsupervised neural networks are used to learn the sparse distribution and the compact representation of the decision variables. Additionally, Tian et al.²⁸ introduced PM-MOEA in 2020, which suggests an evolutionary pattern mining approach to mine the sparse distribution of Pareto optimal solutions, and thus, the search space is considerably reduced. In 2021, Wang et al.²⁹ presented the S-ECSO based on the CSO³⁰, where a strongly convex operator is proposed to directly generate sparse solutions by utilizing the sparse distribution of Pareto optimal solutions, enhancing algorithm's convergence and sparsity. In 2023, Wu et al.³¹ proposed SGECE, where a sparsity-guided elitism co-evolutionary framework is used to determine the distribution of sparse solutions and guide the generation of offspring.

So far, only a limited number of algorithms have been developed to solve LSSMOPs, and most of the algorithms in the first class of methods adopt the bi-level encoding strategy proposed in SparseEA as their base strategy, such as MDRSAEA, MSKE and NUCE. SparseEA adopts decision variable vector and binary mask vector to represent an individual, where the mask vector is used to control the sparsity of the individual, and the score is calculated for each decision variable. In iterative process of SparseEA, fixed crossover probability and mutation probability are used to perform crossover operation and mutation operation on the decision variable vector, and the crossover operation and mutation operation of mask vector are carried out by using the decision variable score during initialization. It can be seen that crossover probability and mutation probability will affect the update efficiency of the decision variable vector, and the decision variable score will affect the update quality of the mask vector, thus affecting the sparsity of the solution. However, in the process of evolution, individuals in the population will have differences between good and bad, and limited computing resources should be allocated to more promising individuals for crossover and variation. Therefore, the crossover probability, variation probability and decision variable score should change with the advantages and disadvantages of individuals in order to screen out more promising individuals. Otherwise, the invariable crossover probability, mutation probability and decision variable score will seriously restrict the improvement of algorithm performance.

To deal with above issue, this paper proposes an evolution algorithm with adaptive genetic operator and dynamic scoring mechanism for large-scale sparse many-objective optimization (SparseEA-AGDS) by building upon the SparseEA algorithm framework. The principal contributions are as follows:

(1) The adaptive genetic operator is proposed, where the probabilities of crossover and mutation for individuals are updated based on the changes of the non-dominated layer during each iteration. This update grants superior individuals increased chances for genetic operations, thereby aiding in the enhancement of the algorithm's convergence and diversity.

(2) Establish a dynamic scoring mechanism. In each iteration, based on the changes of individual's non-dominated layers, the decision variable score is recalculated using a weighted accumulation method. This method increases the chances of crossover and mutation of the superior decision variables, and improves sparsity of Pareto solutions.

It is worth mentioning that the SparseEA-AGDS algorithm has no additional parameter settings, so that the user of the algorithm does not face the difficulty of tuning parameters.

Related works

Definitions in LSSMOPs

Without loss of generality, considering minimization optimization as an example³², MOPs is defined as

$$\min F(\mathbf{X}_i) = \{f_1(\mathbf{X}_i), f_2(\mathbf{X}_i), \dots, f_M(\mathbf{X}_i)\}$$

$$\text{s.t. } \mathbf{X}_i = (x_1, x_2, \dots, x_D) \in \Omega \quad (1)$$

where Ω is the search space and $\{F(\mathbf{X}_i) | \mathbf{X}_i \in \Omega\}$ denotes the attainable objective space. M denotes the number of objective functions, and D denotes the number of the decision variable. In general, MOPs with $D \geq 100$ is called LSMOPs. In LSMOPs, if most decision variables of the optimal solutions are zero, then the problem is considered LSSMOPs.

SparseEA algorithm

The SparseEA algorithm is a multi-objective evolutionary algorithm specifically designed for LSSMOPs⁷, and the core of the SparseEA consists of the population initialization strategy and the genetic operator strategy. The method principle of SparseEA are as follows.

(1) The population initialization strategy is composed of three parts, which are sparse representation of decision variables, calculation of decision variable score and generation of initial population.

Sparse representation of the decision variables

Let \mathbf{X}_i denote the i -th individual, which is defined as

$$\mathbf{X}_i = (x_1, \dots, x_d, \dots, x_D) = (dec_1 \times mask_1, \dots, dec_d \times mask_d, \dots, dec_D \times mask_D) \quad (2)$$

where dec_d denotes the real variable and $mask_d$ denotes the binary variable of the d -th dimension of the decision variable. During the evolution, dec_d can record the best decision variable found so far, while $mask_d$ can record the decision variable that should be set to zero, thereby controlling the sparsity of the solutions. If the decision variables are binary numbers, dec_d are always set to 1.

Decision variable score

The score of decision variable is determined by the non-dominated rank of decision variable and reflects the probability that the decision variable should be set to zero. A smaller non-dominated rank corresponds to a lower score, which means the lower probability that this decision variable should be set to zero. To calculate the non-dominated rank of decision variable, matrix \mathbf{G} of size $D \times D$ is constructed, where the first D denotes the number of individuals and the second D denotes the decision variable dimension of individuals. The \mathbf{G} is obtained by the dot product of matrix **dec** and matrix **mask**:

$$G = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_D \end{bmatrix} = \mathbf{dec} \times \mathbf{mask} = \begin{bmatrix} dec_{1,1} \times mask_{1,1}, & dec_{1,2} \times mask_{1,2}, & \dots, & dec_{1,D} \times mask_{1,D} \\ dec_{2,1} \times mask_{2,1}, & dec_{2,2} \times mask_{2,2}, & \dots, & dec_{2,D} \times mask_{2,D} \\ \vdots, & \vdots, & \ddots, & \vdots, \\ dec_{D,1} \times mask_{D,1}, & dec_{D,2} \times mask_{D,2}, & \dots, & dec_{D,D} \times mask_{D,D} \end{bmatrix} \quad (3)$$

Where **dec** is a D -dimensional random matrix of real variable, and **mask** is a D -dimensional identity matrix, the actual computation of \mathbf{G} is given by (4).

$$G = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_D \end{bmatrix} = \begin{bmatrix} dec_{1,1}, & 0, & \dots, & 0 \\ 0, & dec_{2,2}, & \dots, & 0 \\ \vdots, & \vdots, & \ddots, & \vdots, \\ 0, & 0, & \dots, & dec_{D,D} \end{bmatrix} \quad (4)$$

As can be seen from (4), in each individual \mathbf{X}_i in \mathbf{G} at this time, only one dimension of the decision variable is non-zero, while the other dimensions are zero. So the non-dominated layer of individual \mathbf{X}_i can be used as the non-dominated rank of the decision variable $dec_{i,i}$ that is non-zero. Repeat the above operation, and accumulate the non-domination rank of each decision variable. The cumulative result is used as the score of each dimensional decision variable, denoted by S_i , and put into the score matrix $\mathbf{S} = [S_1, \dots, S_i, \dots, S_D]$.

Generation of initial population

In order to generate an initial population of size N , firstly, a random real variable matrix **dec** and a binary variable matrix **mask** of size $N \times D$ are generated, with all values of **mask** being 0. Then, each individual performs binary tournament selection mechanism³³ on decision variables according to the score matrix **S** to identify a quantity of $c \times D$ positions, where c denotes a random number between [0,1]. At these positions, the values in the **mask** are set to 1. Finally, initial population P is obtained by **dec** × **mask**.

(2) In the genetic operator strategy, parent populations Q generate offspring populations O through crossover and mutation operation of **dec** and **mask**.

Crossover and mutation operations on Dec

The position index matrix **K** of the parent population is first generated, with the size of $N \times D$. N row vectors in **K** correspond to N individuals in the population. Subsequently, a random matrix **d** with elements ranging from 0 to 1 is generated to select individuals for crossover and mutation operations, with the size of $N \times 1$. Finally, fixed crossover probability P_{c0} and mutation probability P_{m0} are separately set, and each element in **d** is compared with the P_{c0} and P_{m0} . If $d(i)$ is less than P_{c0} or P_{m0} , then crossover or mutation operation is performed on the i -th individual of the population, and the elements in row i of **K** are either 1 or 0, representing whether decision variable corresponding to that position should perform crossover and mutation operation or not.

Crossover and mutation operations on mask

In the parent population Q , parent individuals who perform crossover and mutation operations are denoted as Q_{k1} and Q_{k2} , and the offspring individuals are denoted as O_{k1} and O_{k2} , $k_1, k_2 \in [1, N]$ and $k_1 \neq k_2$. During the crossover operation, positions with the same location in Q_{k1} and Q_{k2} but different logical values are picked out, and two positions are randomly selected from them, which correspond to two decision variable dimensions, respectively. Based on the initialization decision variable scores of these two decision variables dimensions, the binary tournament selection strategy³³ is used to select a dimension for crossover operation. During the mutation operation, the initialization decision variable scores of the parent individuals and the binary tournament selection strategy³³ are again used to select a decision variable dimension to carry out logical value inversion.

Shortcomings of the SparseEA algorithm

Although the SparseEA algorithm has shown excellent experimental results in LSSMOPs, there are still three aspects that can be further optimized:

(1) In each iteration, the crossover probability and mutation probability of **dec** remain constant. Whether an individual performs crossover or mutation operation is primarily determined by the random matrix **d**, which has nothing to do with whether the individual is excellent. Thus, the potential development of outstanding individuals has been affected.

(2) In each round of crossover and mutation operation of **mask**, the positions for crossover or mutation are selected always based on the initialized decision variable score. However, as the updating of population, the relative merits of decision variables may change, and the score of decision variable should be updated dynamically.

(3) SparseEA employs crowding distance-based environmental selection strategy. When dealing with many-objective optimization problems, the number of non-dominated individuals grows exponentially, which means the selection pressure of Pareto dominance relations drops sharply. This situation makes it difficult to distinguish between the superior and inferior individuals, ultimately leading to a decline in algorithm performance.

SparseEA-AGDS algorithm

To address several improvable aspects of the SparseEA algorithm, this paper proposes the SparseEA-AGDS algorithm. Adaptive genetic operators strategy and dynamic scoring mechanism are proposed, and reference point-based environmental selection strategy is introduced. The above three strategies improve the ability of the SparseEA-AGDS algorithm to handle large-scale sparse many-objective problems.

Adaptive genetic operator

The adaptive genetic operator strategy is proposed to address the issues during the crossover and mutation process of **dec** in the SparseEA. It dynamically adjusts the crossover probability and mutation probability of individual according to individual's superiority or inferiority in each round of iteration. This strategy provides more chances for the excellent individuals and improves the convergence and diversity of the algorithm. The following are the specific implementation steps of the adaptive genetic operator strategy, and Algorithm 1 presents the pseudo-code of the strategy.

Calculation of the parent individual's non-dominant layer

The fast non-dominated sorting approach³⁴ was used to calculate the non-dominated layers of individuals in the parent population, and the non-dominated layer of \mathbf{X}_i is defined as r_i . The better \mathbf{X}_i corresponds to a smaller r_i .

Calculation of the probability that the parent individual to be selected

According to the r_i , the selected probability for parent individual is calculated by Eq. (5), where $P_{s,i}$ denotes the selected probability of \mathbf{X}_i , and $maxr$ denotes the largest non-dominated layer. It can be seen that the better \mathbf{X}_i is, the smaller r_i is, and the higher $P_{s,i}$ is.

$$P_{s,i} = \frac{maxr - r_i + 1}{maxr} \quad (5)$$

Calculation of adaptive crossover probability and adaptive mutation probability

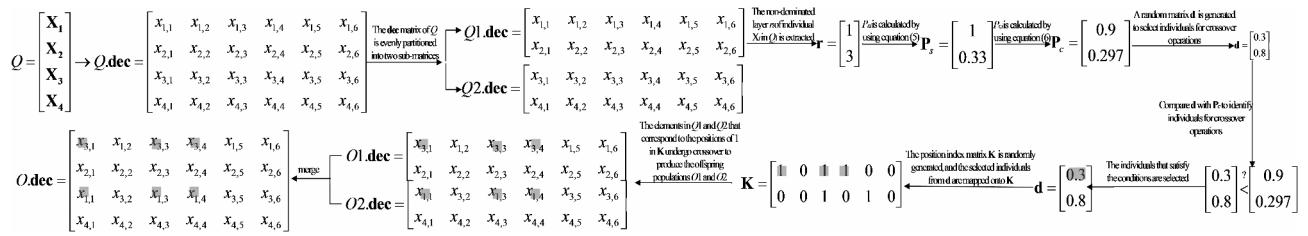
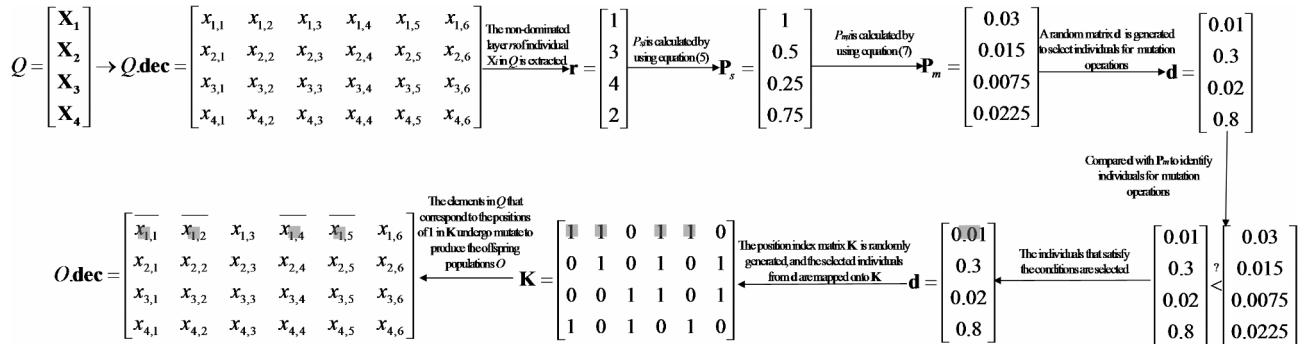
Adaptive crossover probability $P_{c,i}$ and adaptive mutation probability $P_{m,i}$ of \mathbf{X}_i are calculated by Eqs. (6) and (7) respectively. The better \mathbf{X}_i corresponds to the higher $P_{c,i}$ and $P_{m,i}$.

$$P_{c,i} = P_{C0} \times P_{s,i} \quad (6)$$

$$P_{m,i} = P_{m0} \times P_{s,i} \quad (7)$$

Adaptive inheritance to produce offspring

In adaptive inheritance process of **dec**, the generation and use of logical matrix **K** and matrix **d** are the same as that in SparseEA algorithm. The difference is that instead of using fixed probabilities P_{c0} and P_{m0} , adaptive probabilities $P_{c,i}$ and $P_{m,i}$ are used. So, each individual's probability of undergoing crossover and mutation operations will be changed according to their respective levels of superiority or inferiority.

**Fig. 1.** dec crossover process of adaptive genetic operator strategy.**Fig. 2.** dec mutation process of adaptive genetic operator strategy.

From the above, it can be indicated that the more superior the \mathbf{X}_i , the higher the probability of undergoing crossover and mutation operations. Take $N=4$, $D=6$ as an example, Figs. 1 and 2 show the adaptive genetic operator strategy of **dec**, where the value of P_{C0} and P_{m0} are randomly set to 0.9 and 0.03, respectively.

Input: $Q.\text{dec}$ (dec of parent population), Q (Parent population)
Output: $O.\text{dec}$ (dec of offspring population)

1. $D \leftarrow$ Decision variable dimension;
2. $N \leftarrow$ Initialize population size;
3. $r = (r_1, \dots, r_i, \dots, r_N) \leftarrow$ non-dominated-sort(Q) //The fast non-dominated sorting approach is used to calculate the non-dominated layers of individuals in the Q .
4. $maxr \leftarrow max(r)$ //The largest non-dominated layer is identified.
5. $d \leftarrow rand(N, 1)$ //A random matrix d is generated to select individuals for crossover and mutation operations.
6. $K \leftarrow logical(rand(N, D))$ //The position index matrix K is randomly generated to select decision variables of individuals for crossover and mutation operations.
7. **for** $i=1:N$
 8. $P_{s,i} \leftarrow (maxr - r_i + 1) / maxr$ // $P_{s,i}$ is calculated by equation (5).
 9. $P_{c,i} \leftarrow P_{C0} \times P_i$ // $P_{c,i}$ and $P_{m,i}$ are calculated by equation (6) and (7) respectively.
 10. $P_{m,i} \leftarrow P_{m0} \times P_i$
 11. **if** $d(i) < P_{c,i}$ or $P_{m,i}$
 12. index \leftarrow find ($K(i, :)$)
 13. $O.\text{dec}(i, :) \leftarrow Q.\text{dec}(i, \text{index})$ undergoes crossover or mutation operation //If the condition is met, the i -th individual is selected for crossover or mutation operation.
 14. **else** $O.\text{dec}(i, :) \leftarrow Q.\text{dec}(i, :)$
 15. **end**
 16. **end**
17. **return** $O.\text{dec}$ //The dec of offspring population O with size of $N \times D$ is generated.

Algorithm 1. Pseudo-code for adaptive genetic operator strategy.

Dynamic scoring mechanism

The dynamic scoring mechanism is proposed to address the issues during crossover and mutation process of **mask** in SpaeseEA. This mechanism updates the decision variable score based on the changes of individual's non-dominated layers during the iteration process, so as to increase the chances of superior decision variables to perform crossover and mutation operation. The score update of the decision variable is based on two aspects,

one is the non-dominated layers of the individual, and the other is which decision variables of this individual are selected. The followings are the specific implementation steps of dynamic scoring mechanism, and Algorithm 2 presents the pseudo-code of it.

Calculation of the non-dominated layer score

The non-dominated layer score S_i_r of individual \mathbf{X}_i is calculated by Eq. (8), where r_i is the non-dominated layer of \mathbf{X}_i . The non-dominated layer scores of N individuals form a $N \times 1$ matrix $\mathbf{S}_r = [S_1_r, S_2_r, \dots, S_N_r]^T$. In calculating the non-dominated layer of individuals, we use fast non-dominated sorting method of literature³⁴. If an individual \mathbf{X}_i is not dominated by any other individuals in the population, it is in the first non-dominated layer, corresponding to $r_i = 1$. If an individual \mathbf{X}_i is not dominated by any other individuals in the population other than the individuals in the first non-dominated layer, it is in the second non-dominated layer, corresponding to $r_i = 2$, and so on. The better \mathbf{X}_i is, the smaller r_i is, and the higher S_i_r is.

$$S_i_r = \max r - r_i + 1 \quad (8)$$

Calculation of the weighted score for decision variable

The transpose matrix of \mathbf{S}_r is multiplied with **mask** to obtain a weighted score matrix **sumS** with size of $1 \times D$. The calculation process is shown in Eq. (9), where $\text{sum}S_d = \sum_{i=1}^N S_i_r \times \text{mask}_{i,d}$ denotes the weighted score of the d -th dimension decision variable. It can be seen from Eq. (9) that when $\text{mask}_{i,d}$ is 0, the result of multiplying $\text{mask}_{i,d}$ and S_i_r is 0, and only when the value of $\text{mask}_{i,d}$ is 1, the non-dominated layer score S_i_r could affect $\text{sum}S_d$. So, the weighted score is not only related to the S_i_r , but also related to whether the decision variable in **mask** is selected or not.

$$\begin{aligned} \text{Sum}S &= S_{r}^T \times \text{mask} \\ &= [S_1_r \ \dots \ S_i_r \ \dots \ S_N_r] \times \begin{bmatrix} \text{mask}_{1,1} & \dots & \text{mask}_{1,d} & \dots & \text{mask}_{1,D} \\ \vdots & \ddots & & & \vdots \\ \text{mask}_{i,1} & & \text{mask}_{i,d} & & \text{mask}_{i,D} \\ \vdots & & & \ddots & \vdots \\ \text{mask}_{N,1} & \dots & \text{mask}_{N,d} & \dots & \text{mask}_{N,D} \end{bmatrix} \quad (9) \\ &= \left[\sum_{i=1}^N S_i_r \times \text{mask}_{i,1} \ \dots \ \sum_{i=1}^N S_i_r \times \text{mask}_{i,d} \ \dots \ \sum_{i=1}^N S_i_r \times \text{mask}_{i,D} \right] \end{aligned}$$

Update of the decision variable score

The score of the d -th dimension decision variable, denoted as S_d , is calculated according to Eq. (10), where $\text{max}S$ is the largest value in **sumS**. It can be observed that a higher $\text{sum}S_d$ corresponds to a lower S_d .

$$S_d = \text{max}S - \text{sum}S_d + 1 \quad (10)$$

Inheritance to produce offspring

During genetic operation, unlike SparseEA algorithm, which uses fixed initialization decision variable score, SparseEA-AGDS algorithm adopts updated decision variable score S_d . It can be observed that the better \mathbf{X}_i is, the smaller the ordinal number of non-dominated layer r_i is, the higher the non-dominated layer score S_i_r is, the higher the weighted score $\text{sum}S_d$ is, and the smaller the decision variable score S_d is. The binary

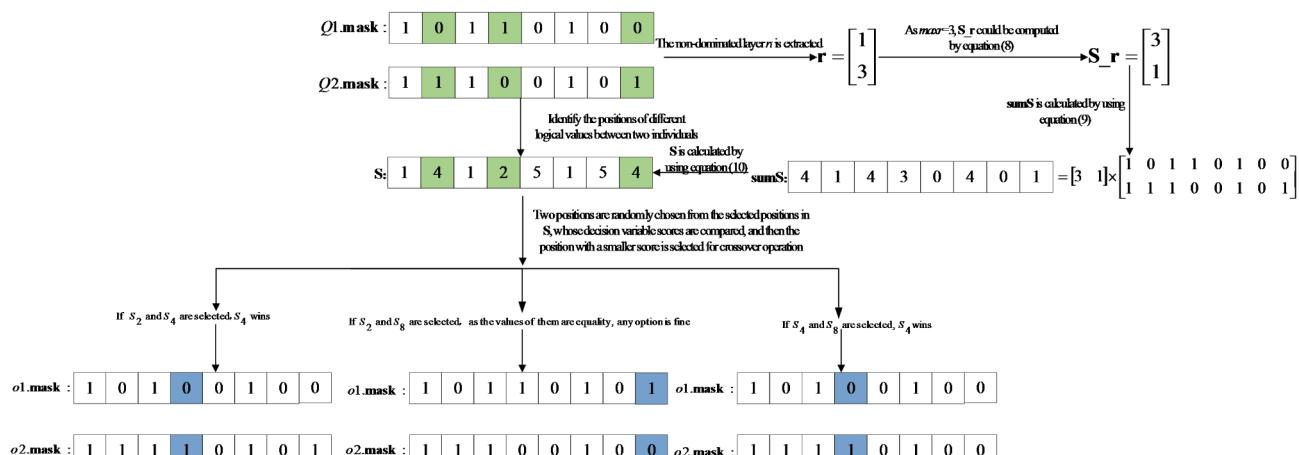


Fig. 3. Crossover process of **mask** in dynamic scoring mechanism.

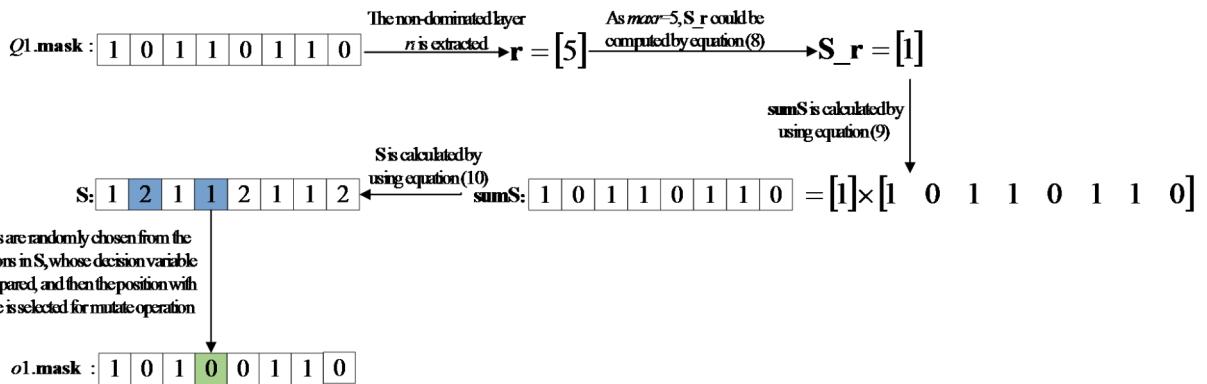


Fig. 4. Mutation process of **mask** in dynamic scoring mechanism.

tournament selection strategy³³ is used to select a dimension with a smaller updated decision variable score S_d for crossover and mutation operation. Take $D=8$ as an example, Figs. 3 and 4 show the crossover and mutation process of **mask** in dynamic scoring mechanisms, where the value of r_i is randomly set.

Input: $Q.\text{mask}$ (mask of parent populations) , Q (Parent population)
Output: $O.\text{mask}$ (mask of offspring population)

1. $D \leftarrow$ Decision variable dimension
2. $N \leftarrow$ Parent population size
3. **mask** \leftarrow The binary variable matrix of Q
4. $r = (r_1, \dots, r_b, \dots, r_N) \leftarrow$ non-dominated-sort(Q)
5. $maxr \leftarrow max(r)$
6. **for** $i=1:N$
7. $S_{i,r} \leftarrow maxr - r_i + 1$
8. **end**
9. $S_r \leftarrow [S_{1,r}, \dots, S_{i,r}, \dots, S_{N,r}]$
10. $\text{sumS} \leftarrow S_r \times \text{mask}$
11. **for** $d=1:D$
12. $S_d \leftarrow maxS - sumS_d + 1$
13. **end**
14. $S \leftarrow [S_1, \dots, S_d, \dots, S_D]$
15. **for** $i=1:N$
16. $O.\text{mask}(i,:) \leftarrow Q.\text{mask}(i,:)$ undergoes crossover or mutation operation
17. **end**
18. **return** $O.\text{mask};$

//The fast non-dominated sorting approach is used to calculate the non-dominated layers of individuals in the Q .
//The largest non-dominated layer is identified.
//The non-dominated layer score $S_{i,r}$ is calculated by equation (8).
//The weighted score for decision variable **sumS** is calculated by equation (9).
//The decision variable score S_d is calculated according to equation (10).
//The **mask** of the i -th individual undergoes crossover or mutation operation to generate the **mask** of offspring individual according to **S** and binary tournament selection strategy.
The **mask** of offspring population O with size of $N \times D$ is generated.

Algorithm 2. Pseudo-code for dynamic scoring mechanism.

Reference point-based environmental selection strategy

The non-dominated individuals grow exponentially as the number of objective functions increases, and the selection pressure of the Pareto dominance relationship drops sharply. In response, the reference point-based environmental selection strategy³⁵ is introduced into SparseEA-AGDS algorithm to address the issue of high-dimensional convergence difficulties. The following are the specific implementation steps of reference point-based environmental selection strategy.

Generation of reference points and normalization of population

The reference point matrix Z within the hyper-plane is firstly constructed based on the scale of the initial population N and the number of objective functions M . Subsequently, the population individuals and reference points undergo normalization.

Associate individuals with reference lines

Reference line of each reference point is calculated based on Z . Then, the distance between individual and its nearest reference line is calculated, thus each individual is associated with its nearest reference line.

Generation of parent population

Individuals in the non-criticality non-dominated layer are placed into population Q , firstly. Then, in the criticality non-dominated layer r_c , in order of the number of individuals associated with each reference line, from least to most, the reference line is selected and its individuals are placed into population Q , until Q is full of N individuals.

SparseEA-AGDS algorithm

As previously mentioned, the SparseEA-AGDS algorithm is proposed within the SparseEA algorithm framework and consists of three strategies, which are adaptive genetic operator, dynamic scoring mechanism and reference point-based environmental selection strategy. Algorithm 3 provides the pseudo-code for the SparseEA-AGDS.

Input: Population size N , decision variable dimension D Output: P_L (Last population) // Calculate the initial scores of variables 1. $D \leftarrow$ The decision variable dimension of individuals 2. $\text{dec} \leftarrow D \times D$ random matrix 3. $\text{mask} \leftarrow D \times D$ identity matrix 4. $G \leftarrow \text{dec} \times \text{mask}$ 5. $S = [S_1, \dots, S_b, \dots, S_D] \leftarrow$ Non-dominated-sort (G) // Generate the initial population 6. $\text{dec} \leftarrow N \times D$ matrix 7. $\text{mask} \leftarrow N \times D$ zero matrix 8. $pos \leftarrow rand(c \times D)$ 10. $\text{mask} \leftarrow \text{mask}_{pos} = 1$ 11. $P \leftarrow \text{dec} \times \text{mask}$ // Optimization 12. while termination criterion not fulfilled do 13. $O.\text{dec} \leftarrow$ Adaptive genetic operator ($P, P.\text{dec}$) 14. $O.\text{mask} \leftarrow$ Dynamic scoring mechanism ($P, P.\text{mask}$) 15. $O \leftarrow O.\text{dec} \times O.\text{mask}$ 16. $P \leftarrow$ Reference point-based environmental selection strategy [P, O] 17. end 18. return P_L	// The non-dominated layer of individuals in G is used to get the score of decision variable. // A quantity of $c \times D$ positions are identified, c denotes a random number between $[0, 1]$. // At these positions, the values in the mask are set to 1. // Initial population P is generated. //Algorithm 1 is performed to gain $O.\text{dec}$. //Algorithm 2 is performed to gain $O.\text{mask}$. // Offspring population O is gained. // Merging O with P , and the reference point-based environmental selection strategy is used to produce parent population of the next generation. // The population from the last iteration.
---	--

Algorithm 3. Pseudo-code for the proposed SparseEA-AGDS.

Computation complexity analysis*Time complexity analysis*

Assuming the population size is N , the dimension of individual decision variables is D , the number of objectives is M , and the number of iterations is γ_{iter} .

Consider the complexity of one iteration of the entire SparseEA algorithm, the basic operations and their worst-case complexities are as follows.

- (1) initializing decision variable scores is $O(D^2)$;
- (2) fast non-dominated sorting is $O(N^2M)$ [34];
- (3) population initialization is $O(ND)$;
- (4) crossover operation of dec is $O(ND)$;
- (5) mutation operation of dec is $O(ND)$;
- (6) crossover operation of mask is $O(ND)$;
- (7) mutation operation of mask is $O(ND)$;
- (8) crowding distance-based environmental selection strategy is $O(N^2M)$ ³⁴.

The population undergoing γ_{iter} iterations, and the overall complexity of the SparseEA algorithm is $\max\{O(D^2), O(KN^2M), O(KND)\}$.

Consider the complexity of one iteration of the entire SparseEA-AGDS algorithm, the basic operations and their worst-case complexities are as follows.

- (1) initializing decision variable scores is $O(D^2)$;
- (2) fast non-dominated sorting is $O(N^2M)$;

Problem	N	M	D	maxFE
SMOP1	100	2–10	100–1000	$100 \times D$
SMOP2	100	2–10	100–1000	$100 \times D$
SMOP3	100	2–10	100–1000	$100 \times D$
SMOP4	100	2–10	100–1000	$100 \times D$
SMOP5	100	2–10	100–1000	$100 \times D$
SMOP6	100	2–10	100–1000	$100 \times D$
SMOP7	100	2–10	100–1000	$100 \times D$
SMOP8	100	2–10	100–1000	$100 \times D$

Table 1. Parameter settings of the benchmark problems.

Compared algorithm	Parameter settings
NSGA-III ³⁵	$p_{co} = 1$, $p_{mo} = \frac{1}{D}$, $\eta = 20$
CMOPSO ³⁸	$p_{co} = 1$, $p_{mo} = \frac{1}{D}$, $\eta = 20$
MOEA/D-DRA ³⁹	Neighborhood size $T = 0.1N$, Selected probability of the neighborhood $nr = 0.01N$
PM-MOEA ²⁸	Population size of evolutionary pattern mining method $np = 20$ The number of generations of evolutionary pattern mining method $tp = 10$, $p_{co} = 1$, $p_{mo} = \frac{1}{D}$, $\eta = 20$
MOEA/PSL ²⁷	$p_{co} = 1$, $p_{mo} = \frac{1}{D}$, $\eta = 20$
LMOCSO ³⁶	$p_{co} = 1$, $p_{mo} = \frac{1}{D}$, $\eta = 20$
SparseEA ⁷	$p_{co} = 1$, $p_{mo} = \frac{1}{D}$, $\eta = 20$

Table 2. Parameter setting of compared algorithms.

- (3) population initialization is $O(ND)$;
- (4) calculation of adaptive crossover probability and mutation probability is $O(N)$;
- (5) crossover operation of **dec** is $O(ND)$;
- (6) mutation operation of **dec** is $O(ND)$;
- (7) calculation of dynamic scores of decision variables is $O(ND)$;
- (8) crossover operation of **mask** is $O(ND)$;
- (9) mutation operation of **mask** is $O(ND)$;
- (10) reference point-based environmental selection strategy is $\max\{O(N^2M), O(N^2\log^{M-2}N)\}$ ³⁵.

The population undergoing γ_{iter} iterations, and the overall complexity of the SparseEA-AGDS algorithm is $\max\{O(D^2), O(KN^2\log^{M-2}N), O(KN^2M), O(KND)\}$.

Space complexity analysis

Assuming the population size is N , the dimension of individual decision variables is D , the number of objectives is M , and the number of iterations is γ_{iter} .

Consider the space complexity of the entire SparseEA and SparseEA-AGDS algorithm, the basic operations and their worst-case complexities are the same, as follows.

- (1) population space is $O(N(D + M))$;
- (2) space of decision variable scoring table is $O(D^2)$.

In summary, the overall space complexity of the algorithm is $\max\{O(D^2), O(N(D + M))\}$.

Experimental results and analysis

In this section, the experimental settings are firstly introduced. Subsequently, performance verification experiments of the adaptive genetic operator, dynamic scoring mechanism and reference point-based environmental selection strategy are conducted separately. Finally, the proposed SparseEA-AGDS algorithm has been tested on SMOP test suite, comparing against representative algorithms, i.e. NSGA-III³⁵, LMOCSO³⁶, MOEA-PSL²⁷, PM-MOEA²⁸, SparseEA⁷, NUCEA²⁶ and SGECF³¹.

Experimental settings

(1) Experimental environment: All experiments are carried out on a PC with AMD i5-8250U 1.6 GHz CPU, 64GB RAM, Windows 10. All the experiments are conducted on the evolutionary multi-objective optimization platform PlatEMO³⁷.

(2) Benchmark problems: The SMOP test suite that proposed by Tian et al.⁷ was utilized, and the parameters settings of the benchmark problems are as shown in Table 1, where maxFE denotes the maximum number of function evaluations.

(3) Parameter Settings of Compared algorithms: The parameters of all compared algorithms are consistent with the optimal parameter settings in the original articles of them, as shown in Table 2. For algorithms that use simulated binary crossover and polynomial variation methods, p_{c0} and p_{m0} denote fixed crossover probability and fixed mutation probability, and η denotes the distribution indices of crossover and mutation.

(4) Evaluation metrics: Since the true PF of all benchmark problems are known, the inverted generational distance (IGD)⁴⁰ is adopted to measure the quality of solution. IGD is calculated with 10,000 points that are uniformly sampled on the true Pareto front. For each algorithm on each problem, the mean value and standard deviation of solutions gained by 30 independent runs are as evaluation indicators, and the standard deviation is recorded in parentheses. The Wilcoxon rank-sum test⁴¹ was used to perform statistical analysis, with a significance level of 0.05. In all tables, “+”, “-” and “=” indicate that the result of compared algorithm is better than, worse than, or approximately equal to the SparseEA-AGDS, respectively. IGD is calculated by Eq. (11).

$$IGD(H, POP) = \frac{\sum_{i=1}^N pop_i \in POP dis(pop_i, H)}{N} \quad (11)$$

where H denotes the point set uniformly sampled on the real PF. POP denotes the Pareto solution set obtained by algorithm, and pop_i is a solution in the POP . The $dis(pop_i, H)$ denotes the minimum Euclidean distance between pop_i and each individual in H .

Ablation study

Effectiveness of the adaptive genetic operator

To verify the effect of adaptive genetic operator on the performance of algorithm, the strategy is applied to the SparseEA algorithm to form SparseEA-AG, which is accompanied with SparseEA. Under the same initial population conditions, comparison experiments are conducted on SMOP benchmark problems with $M=2, 3$ and $D=100$. The IGD values obtained from the experiments are presented in Table 3, where the best results are highlighted. The percentage of non-zero decision variables is illustrated in Fig. 5. The following conclusions can be drawn from Table 3; Fig. 5.

(1) In Table 3, according to the IGD results, SparseEA-AG achieves the best IGD values on 11 of 16 problems, and for the statistical results of the Wilcoxon rank-sum test, SparseEA-AG outperforms the SparseEA on 10 problems and is inferior to the SparseEA on 4 problems. It is apparent that adaptive genetic operator plays a positive role in enhancing the algorithm's convergence and diversity.

(2) From Fig. 5, the percentage of non-zero decision variables of SparseEA-AG surpass the SparseEA on SMOP2, SMOP6 and SMOP8 problems when $M=2$. And SparseEA-AG outperforms the SparseEA on 6 of 8 problems when $M=3$. It can be seen that the SparseEA-AG algorithm obtains Pareto optimal solution with better sparsity when dealing with many-objective problems.

In summary, adaptive genetic operator has a significant contribution on improving the convergence and diversity of algorithm, and sparsity of Pareto solutions for many-objective problems.

Problem	M	D	IGD	
			SparseEA	SparseEA-AG
SMOP1	2	100	1.0334e-2 (2.39e-3) -	9.7707e-3 (2.23e-3)
	3		7.8095e-2 (4.41e-3) -	7.6181e-2 (4.04e-3)
SMOP2	2	100	3.3230e-2 (7.43e-3) -	3.2528e-2 (8.33e-3)
	3		1.2126e-1 (9.21e-3) -	1.2031e-1 (8.74e-3)
SMOP3	2	100	1.7071e-2 (3.24e-3) -	1.6606e-2 (2.89e-3)
	3		8.2551e-2 (4.89e-3) +	8.3529e-2 (5.10e-3)
SMOP4	2	100	4.7931e-3 (2.38e-4) -	4.7150e-3 (2.59e-4)
	3		6.1633e-2 (9.71e-3) +	6.1654e-2 (1.10e-2)
SMOP5	2	100	6.2090e-3 (4.25e-4) -	6.1471e-3 (3.75e-4)
	3		6.1466e-2 (5.99e-3) -	6.1008e-2 (7.64e-3)
SMOP6	2	100	7.5651e-3 (5.33e-4) =	7.6828e-3 (7.49e-4)
	3		6.4626e-2 (7.90e-3) -	6.3115e-2 (5.87e-3)
SMOP7	2	100	4.1201e-2 (1.02e-2) =	4.0450e-2 (8.03e-3)
	3		1.3894e-1 (1.36e-2) +	1.3924e-1 (1.04e-2)
SMOP8	2	100	1.7364e-1 (2.54e-2) +	1.7860e-1 (2.88e-2)
	3		2.8406e-1 (2.64e-2) -	2.7044e-1 (3.60e-2)
$+/-=$			4/10/2	

Table 3. IGD results of two algorithms on SMOP benchmark problems.

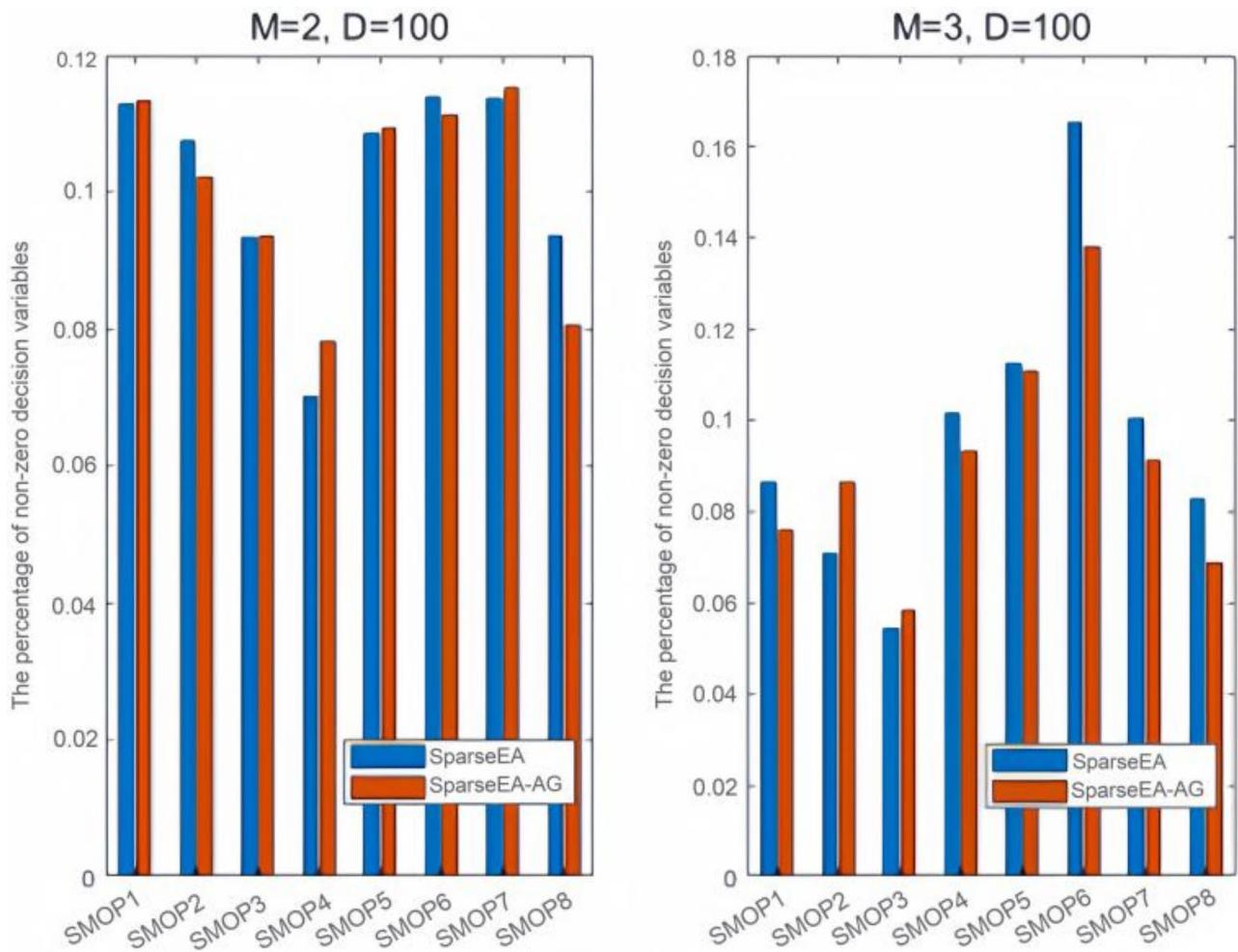


Fig. 5. Percentage of non-zero decision variables of two algorithms.

Effectiveness of dynamic scoring mechanism

To verify the effect of dynamic scoring mechanism on the performance of algorithm, the strategy is applied to the SparseEA algorithm to form SparseEA-DS, which is compared with SparseEA. Under the same initial population conditions, comparison experiments are conducted on SMOP benchmark problems with $M=2, 3$ and $D=100, 500, 1000$. The IGD values obtained from the experiments are presented in Table 4. The percentage of non-zero decision variables is illustrated in Fig. 6 and statistics of wins on this percentage is presented in Table 5. The following conclusions can be drawn from Table 4; Fig. 6; Table 5.

(1) In Table 4, according to the IGD results, SparseEA-DS achieves the best IGD values on 26 of 48 problems, and especially surpasses the SparseEA algorithm on SMOP4, SMOP5 and SMOP6 problems which are similar to the feature selection problem. For the statistical results of Wilcoxon rank-sum test, SparseEA-DS outperforms the SparseEA on 8 problems, which are all situated on SMOP4, SMOP5 and SMOP6. And the performance of SparseEA-DS is similar to that of SparseEA on 31 problems. It can be seen that the SparseEA-DS algorithm can show better diversity and convergence in dealing with problems similar to feature selection, but has a limited effect on improving IGD.

(2) It can be seen from the results in Fig. 6; Table 5 that the SparseEA-DS wins more times on the percentage of non-zero decision variables with $M=3$ when compared to that with $M=2$ at any dimension D , and also wins more times than SparseEA. This indicates that the SparseEA-DS algorithm can obtain Pareto optimal solution with better sparsity in high-dimensional problems.

Based on the aforementioned discussions, dynamic scoring mechanism can show good convergence and diversity in dealing with problems similar to feature selection, and obtains better sparse Pareto optimal solutions when facing large-scale many-objective problems.

Effectiveness of reference point-based environmental selection strategy

To verify the effect of the reference point-based environmental selection strategy on the performance of the algorithm, the strategy is applied to the SparseEA algorithm to form SparseEA-RP, which is compared with SparseEA. Under the same initial population conditions, comparison experiments are conducted on SMOP benchmark problems with $M=3, 5, 10$ and $D=100, 500, 1000$. The IGD values obtained from the experiments

Problem	M	D	IGD	
			SparseEA	SparseEA-DS
SMOP1	2	100	1.0266e-2 (2.42e-3) =	1.0883e-2 (3.31e-3)
		500	1.7398e-2 (2.93e-3) +	1.9333e-2 (2.28e-3)
		1000	2.5627e-2 (2.18e-3) +	2.6991e-2 (2.10e-3)
	3	100	7.7707e-2 (5.37e-3) =	7.8973e-2 (4.80e-3)
		500	7.8534e-2 (5.79e-3) +	8.0424e-2 (3.66e-3)
		1000	7.6016e-2 (4.37e-3) +	7.9579e-2 (5.14e-3)
SMOP2	2	100	3.1714e-2 (6.15e-3) =	3.4867e-2 (1.02e-2)
		500	4.9593e-2 (7.60e-3) =	4.8876e-2 (6.24e-3)
		1000	6.8045e-2 (6.39e-3) =	6.7336e-2 (6.48e-3)
	3	100	1.2001e-1 (9.76e-3) +	1.2755e-1 (6.46e-3)
		500	1.2467e-1 (6.11e-3) +	1.3353e-1 (6.37e-3)
		1000	1.2511e-1 (5.26e-3) +	1.2920e-1 (5.47e-3)
SMOP3	2	100	1.8208e-2 (3.97e-3) =	1.7467e-2 (3.60e-3)
		500	2.2287e-2 (3.07e-3) =	2.3909e-2 (3.75e-3)
		1000	2.8944e-2 (2.93e-3) =	2.9455e-2 (2.60e-3)
	3	100	8.2019e-2 (4.16e-3) =	8.4436e-2 (5.38e-3)
		500	8.5740e-2 (4.07e-3) =	8.7851e-2 (4.31e-3)
		1000	8.8299e-2 (3.64e-3) =	8.8225e-2 (2.98e-3)
SMOP4	2	100	4.8005e-3 (2.73e-4) =	4.7649e-3 (2.91e-4)
		500	4.7739e-3 (2.81e-4) =	4.7174e-3 (2.08e-4)
		1000	4.8575e-3 (3.54e-4) =	4.7384e-3 (2.32e-4)
	3	100	6.1667e-2 (7.84e-3) =	5.8816e-2 (8.07e-3)
		500	6.4770e-2 (7.34e-3) =	6.1828e-2 (6.55e-3)
		1000	6.7284e-2 (9.55e-3) =	6.2110e-2 (7.30e-3)
SMOP5	2	100	6.0811e-3 (3.26e-4) -	5.7951e-3 (4.38e-4)
		500	6.0378e-3 (3.94e-4) -	5.5896e-3 (2.55e-4)
		1000	5.8650e-3 (2.64e-4) -	5.4542e-3 (3.00e-4)
	3	100	6.4078e-2 (5.68e-3) -	5.6592e-2 (6.74e-3)
		500	6.6193e-2 (7.30e-3) =	5.5646e-2 (7.10e-3)
		1000	6.7092e-2 (1.13e-2) =	6.2529e-2 (7.89e-3)
SMOP6	2	100	7.6830e-3 (6.76e-4) -	6.8689e-3 (5.28e-4)
		500	7.1403e-3 (3.87e-4) -	6.2900e-3 (2.62e-4)
		1000	7.1827e-3 (5.33e-4) -	6.1859e-3 (2.65e-4)
	3	100	6.1868e-2 (5.62e-3) =	6.1656e-2 (5.87e-3)
		500	7.1621e-2 (8.86e-3) -	6.6965e-2 (8.29e-3)
		1000	6.9927e-2 (1.12e-2) =	6.6056e-2 (9.39e-3)
SMOP7	2	100	3.9925e-2 (7.63e-3) =	3.9197e-2 (1.00e-2)
		500	5.9428e-2 (9.17e-3) =	6.3652e-2 (9.39e-3)
		1000	8.5709e-2 (6.84e-3) =	8.7108e-2 (8.01e-3)
	3	100	1.3487e-1 (1.56e-2) =	1.4011e-1 (1.58e-2)
		500	1.6780e-1 (1.27e-2) +	1.7441e-1 (1.09e-2)
		1000	1.7090e-1 (9.83e-3) +	1.8053e-1 (7.02e-3)
SMOP8	2	100	1.6663e-1 (2.65e-2) =	1.7208e-1 (3.01e-2)
		500	2.1109e-1 (1.72e-2) =	2.0606e-1 (1.91e-2)
		1000	2.4511e-1 (1.59e-2) =	2.4225e-1 (1.33e-2)
	3	100	2.7629e-1 (3.01e-2) =	2.7222e-1 (2.78e-2)
		500	3.3060e-1 (2.54e-2) =	3.3462e-1 (2.00e-2)
		1000	3.6855e-1 (1.61e-2) =	3.7053e-1 (1.38e-2)
+/-/=			9/8/31	

Table 4. IGD results of two algorithms on SMOP benchmark problems.

are presented in Table 6 and statistics of wins is presented in Table 7. The percentage of non-zero decision variables is illustrated in Fig. 7. The following conclusions can be drawn from Tables 6 and 7; Fig. 7.

(1) In Table 6, according to the IGD results, SparseEA-RP achieves the best IGD values on 54 of 72 problems, which is approximately 3/4 of the total problems. For the statistical results of Wilcoxon rank-sum test, SparseEA-

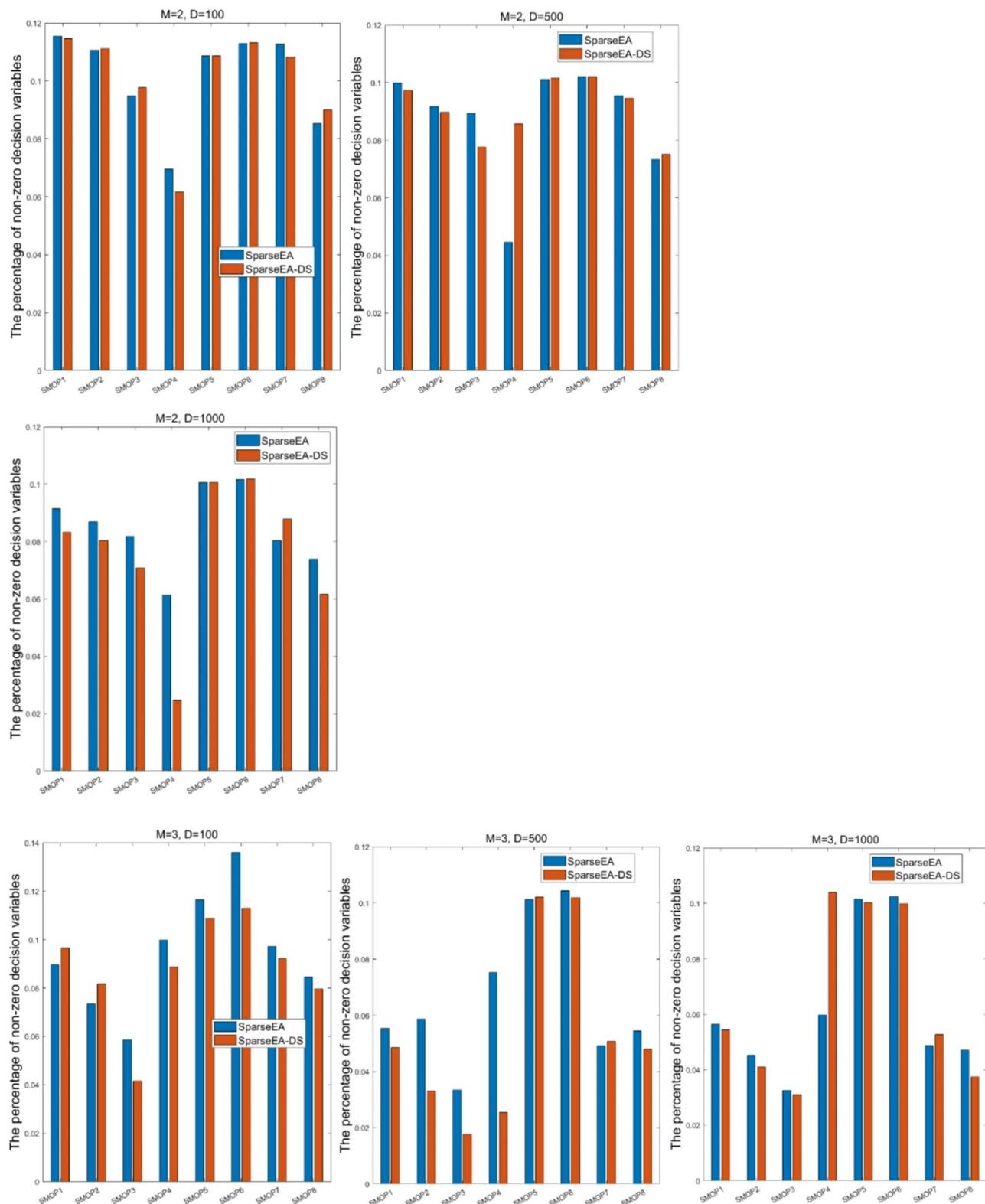


Fig. 6. Percentage of non-zero decision variables of two algorithms.

RP surpasses the SparseEA on 50 problems. It is apparent that the SparseEA-RP can indicate good convergence and diversity.

(2) In Table 7, when D increased from 100 to 1000 under $M=3,10$ conditions, the wins of SparseEA-RP are all significantly better than the SparseEA. However, as the dimension of decision variables increases under $M=5$ conditions, the performance decline of SparseEA-RP is pronounced. It can be seen that the performance

	D = 100		D = 500		D = 1000	
	M = 2	M = 3	M = 2	M = 3	M = 2	M = 3
SparseEA-DS	3	6	4	6	5	6
SparseEA	5	2	4	2	3	2

Table 5. Statistics of wins for two algorithms' on percentage of non-zero decision variables.

of the SparseEA-RP algorithm is easily influenced by the dimension of the decision variables, and an increase of dimension leads to a decrease of IGD values.

(3) From Fig. 7, the SparseEA-RP achieves better percentage of non-zero decision variables on 3 problems with M = 3. While it achieves better percentage of non-zero decision variables on all problems with M = 5, and achieves better percentage of non-zero decision variables on 7 problems with M = 10. It is apparent that the SparseEA-RP algorithm significantly enhances the sparsity of Pareto optimal solutions in large-scale many-objective sparse problems.

To conclude, although the IGD of SparseEA-RP is easily affected by the dimension of decision variables, the SparseEA-RP effectively resists the potential impact of increasing the dimension of the objective on the algorithm, making algorithm suitable for many-objective optimization problems. And the SparseEA-RP has better convergence and diversity when compared to the SparseEA on the statistical results of Wilcoxon rank-sum test. In addition, SparseEA-RP has a striking contribution to the sparsity of Pareto solutions when the dimension of M and D are large.

Computational efficiency of SparseEA-AGDS

To verify the efficiency of the proposed algorithm, SparseEA-AGDS is compared with seven representative algorithms, i.e., NSGA-III³⁵, LMOCSO³⁶, MOEA/PSL²⁷, PM-MOEA²⁸, SparseEA⁷, NUCEA²⁶ and SGECF³¹. The comparison experiments are conducted on SMOP benchmark problems with M = 3, 5, 8, 10, 15 and D = 100, 500, 1000, respectively. The IGD values obtained from the experiments are presented in Table 8, the histogram of IGD values is illustrated in Fig. 8 and statistics of wins on IGD values is presented in Table 9. The percentage of non-zero decision variables is illustrated in Fig. 9 and statistics of wins on this percentage is presented in Table 10. Since the eight benchmark problems, SMOP1–SMOP8, have three different frontiers, i.e., SMOP1–SMOP3 are linear, SMOP4–SMOP6 are concave, and SMOP7–SMOP8 are convex, three benchmark problems are randomly selected from three different frontiers to intuitively compare algorithms' Pareto front with the true Pareto front. Taking SMOP2, SMOP4, and SMOP8 as examples, the Pareto front of eight algorithms on them with M = 3 and D = 100, 1000 are illustrated in Fig. 10. Since neither NSGA-III and LMOCSO are specific algorithms for handling LSSMOPs, they do not participate in comparing the runtime of the algorithms. The statistics of wins on runtime for the MOEA / PSL, PM-MOEA, SparseEA, NUCEA, SGECF and SparseEA-AGDS algorithm are shown in Table 11. The following conclusions can be drawn from Table 8; Fig. 8; Table 9; Fig. 9; Table 10; Fig. 10; Table 11.

(1) In Table 8; Fig. 8, and Table 9, according to the statistical results of Wilcoxon rank-sum test, SparseEA-AGDS surpasses the LMOCSO on 83 of 120 problems, and surpasses the NUCEA on 78 problems, and surpasses the PM-MOEA on 73 problems, and surpasses the SparseEA, NSGA-III, SGECF and MOEA/PSL on 72, 71, 65, 55 problems, separately. For the histogram of IGD values and the statistics of wins on IGD values, SparseEA-AGDS achieves the best IGD values on 41 problems, and NSGA-III, LMOCSO, MOEA/PSL, PM-MOEA, SparseEA, NUCEA and SGECF achieves the best IGD values on 26, 2, 15, 6, 4, 3, 23 problems, separately. It is apparent that the SparseEA-AGDS can indicate good convergence and diversity.

(2) In Fig. 9; Table 10, under M = 3 conditions, the SparseEA-AGDS algorithm's wins on percentage of non-zero decision variables are less than the SparseEA. But all of the wins of SparseEA-AGDS surpass other seven algorithms when M = 5, 8, 10 and 15. It can be seen that the SparseEA-AGDS can obtain Pareto optimal solutions with better sparsity, especially when dealing with many-objective large-scale sparse optimization problem.

(3) In Fig. 10, with M = 3 and D = 100, the SparseEA-AGDS algorithm can better converge to the true Pareto front and has a more uniform distribution when compared to the other seven algorithms in SMOP2 and SMOP4 problems. And with M = 3 and D = 1000, SparseEA-AGDS shows the best convergence and uniformity in three problems when compared to the other seven algorithms. It can be seen that the SparseEA-AGDS algorithm can promote the convergence and uniformity of Pareto front.

(4) In Table 11, according to the statistical results of runtime Wilcoxon rank-sum test, although SparseEA-AGDS is inferior to SparseEA, it is far better than the other four comparison algorithms. The SparseEA-AGDS algorithm surpasses MOEA/PSL on 119 of 120 problems, surpasses PM-MOEA on all 120 problems, and surpasses the NUCEA and SGECF on 106, 101 problems, respectively. Due to the calculation of adaptive genetic operator and dynamic evaluation of decision variable scores, that the runtime of SparseEA-AGDS is longer than that of SparseEA. However, in the percentage of non-zero decision variables, SparseEA-AGDS outperformed SparseEA on 86 of 120 problems, and surpasses SparseEA on 72 problems in IGD values. The SparseEA-AGDS algorithm achieved superior performance of sparse optimization by the increased runtime.

To sum up, when dealing with many-objective large-scale sparse optimization problems, the SparseEA-AGDS algorithm can promote the convergence and uniformity of the Pareto front, and also can gain Pareto solutions with better sparsity.

Problem	M	D	IGD	
			SparseEA	SparseEA-RP
SMOP1	3	100	7.7707e-2 (5.37e-3) -	5.2253e-2 (3.19e-3)
		500	7.8289e-2 (6.05e-3) -	4.8940e-2 (2.32e-3)
		1000	7.5624e-2 (4.00e-3) -	6.7441e-2 (1.03e-1)
	5	100	3.8632e-1 (6.44e-2) -	2.7384e-1 (1.01e-1)
		500	3.9657e-1 (5.40e-2) -	3.2574e-1 (1.20e-1)
		1000	3.9066e-1 (3.05e-2) =	4.0516e-1 (1.91e-1)
	10	100	8.5286e-1 (7.96e-1) -	5.5898e-1 (1.37e-1)
		500	6.8913e-1 (1.03e-1) =	6.0132e-1 (1.85e-1)
		1000	5.4386e-1 (1.35e-1) =	5.5209e-1 (1.68e-1)
SMOP2	3	100	1.1860e-1 (8.66e-3) -	7.9832e-2 (7.65e-3)
		500	1.2554e-1 (8.58e-3) -	7.5642e-2 (5.26e-3)
		1000	1.2554e-1 (7.05e-3) -	9.8094e-2 (1.09e-1)
	5	100	4.7139e-1 (1.08e-1) -	2.5213e-1 (9.51e-2)
		500	5.5955e-1 (9.58e-2) -	4.1952e-1 (1.62e-1)
		1000	5.4279e-1 (8.01e-2) -	4.6058e-1 (2.08e-1)
	10	100	9.3025e-1 (1.44e-1) -	6.1064e-1 (1.02e-1)
		500	8.1507e-1 (1.12e-1) -	6.2880e-1 (1.82e-1)
		1000	7.1023e-1 (1.80e-1) =	6.3552e-1 (2.32e-1)
SMOP3	3	100	8.4194e-2 (4.19e-3) -	6.7528e-2 (7.74e-3)
		500	8.7454e-2 (4.12e-3) =	2.5446e-1 (2.83e-1)
		1000	8.8145e-2 (3.69e-3) =	3.9246e-1 (3.67e-1)
	5	100	2.9740e-1 (1.86e-1) +	3.7311e-1 (5.76e-2)
		500	5.2648e-1 (2.59e-1) =	6.0790e-1 (1.52e-1)
		1000	5.8548e-1 (2.53e-1) +	7.5485e-1 (1.69e-1)
	10	100	8.6561e-1 (1.10e-1) -	6.3676e-1 (5.49e-2)
		500	9.8027e-1 (5.29e-2) -	9.0792e-1 (7.45e-2)
		1000	1.0054e+0 (3.70e-2) -	9.5522e-1 (7.44e-2)
SMOP4	3	100	5.8589e-2 (9.12e-3) -	2.9098e-2 (9.56e-4)
		500	6.4131e-2 (9.15e-3) -	2.8595e-2 (1.60e-4)
		1000	6.6332e-2 (1.19e-2) -	2.8717e-2 (7.59e-5)
	5	100	2.4355e-1 (3.20e-2) +	4.2818e-1 (2.73e-1)
		500	2.1166e-1 (3.77e-2) +	7.3286e-1 (3.31e-1)
		1000	2.0532e-1 (3.67e-2) +	7.4893e-1 (2.96e-1)
	10	100	3.7333e-1 (5.76e-2) =	5.4843e-1 (3.68e-1)
		500	3.4036e-1 (1.31e-1) +	7.9620e-1 (3.12e-1)
		1000	3.2531e-1 (1.33e-1) +	9.0536e-1 (2.13e-1)
SMOP5	3	100	6.3326e-2 (6.50e-3) -	3.6159e-2 (1.29e-2)
		500	6.6230e-2 (8.45e-3) -	5.2579e-2 (1.29e-1)
		1000	6.8056e-2 (1.11e-2) -	2.8934e-2 (1.15e-4)
	5	100	3.0320e-1 (1.66e-1) =	2.9568e-1 (8.08e-2)
		500	3.9902e-1 (3.02e-1) +	8.2519e-1 (9.34e-2)
		1000	3.7844e-1 (3.11e-1) +	8.6336e-1 (5.44e-2)
	10	100	7.7116e-1 (8.84e-2) -	4.7043e-1 (7.87e-2)
		500	9.6017e-1 (5.14e-2) -	8.8774e-1 (7.87e-2)
		1000	9.2857e-1 (1.34e-1) -	9.1489e-1 (6.86e-2)
SMOP6	3	100	6.5653e-2 (6.33e-3) -	4.8343e-2 (5.40e-2)
		500	7.1262e-2 (1.28e-2) +	7.8879e-2 (1.88e-1)
		1000	6.6677e-2 (7.53e-3) -	5.6952e-2 (1.52e-1)
	5	100	4.6173e-1 (2.68e-1) =	3.5037e-1 (5.29e-2)
		500	4.8556e-1 (3.57e-1) +	8.2091e-1 (7.24e-2)
		1000	5.5047e-1 (3.73e-1) =	7.5012e-1 (2.40e-1)
	10	100	7.3595e-1 (1.10e-1) -	5.1089e-1 (8.45e-2)
		500	9.5320e-1 (3.75e-2) -	9.0152e-1 (5.57e-2)
		1000	9.5743e-1 (3.95e-2) -	8.8785e-1 (1.51e-1)

Continued

Problem	M	D	IGD		
			SparseEA	SparseEA-RP	
SMOP7	3	100	1.3533e-1 (1.55e-2) -	1.0965e-1 (1.40e-2)	
		500	1.6917e-1 (8.94e-3) -	1.0952e-1 (8.30e-3)	
		1000	1.7086e-1 (8.97e-3) -	1.1248e-1 (6.47e-3)	
	5	100	3.6966e-1 (1.24e-2) -	3.1113e-1 (8.90e-3)	
		500	3.7947e-1 (1.60e-2) -	2.9573e-1 (6.08e-3)	
		1000	3.7990e-1 (1.50e-2) -	2.8808e-1 (4.93e-3)	
	10	100	1.2215e+0 (1.12e-1) -	6.2046e-1 (9.01e-3)	
		500	1.2363e+0 (6.93e-2) -	6.0164e-1 (8.59e-3)	
		1000	1.2124e+0 (6.78e-2) -	5.9704e-1 (3.27e-2)	
+/-=	11/50/11				

Table 6. IGD results of two algorithms on SMOP benchmark problems.

	M = 3			M = 5			M = 10		
	D = 100	D = 500	D = 1000	D = 100	D = 500	D = 1000	D = 100	D = 500	D = 1000
SparseEA-RP	8	6	7	6	4	3	8	7	6
SparseEA	0	2	1	2	4	5	0	1	2

Table 7. Statistics of wins for two algorithms on the IGD values.

Conclusions

Many researchers utilize multi-objective evolutionary algorithms to solve real-world optimization problems⁴². However, the performance of multi-objective evolutionary algorithms is often not satisfactory when solving large-scale sparse many-objective optimization problems. Consequently, this paper proposes SparseEA-AGDS algorithm based on the SparseEA algorithm framework. The SparseEA-AGDS algorithm introduces adaptive genetic operators and dynamic scoring mechanism to dynamically updates the crossover probability, mutation probability, and decision variable scores based on the quality of individuals, thereby increasing the chances for superior individuals to undergo crossover and mutation operation. Additionally, SparseEA-AGDS introduces the reference point-based environmental selection strategy to improve the algorithm's ability to handle many-objective optimization problems.

The results of the ablation study indicate that, the SparseEA-AG has positively contributed in enhancing the diversity and convergence of the algorithm. The SparseEA-DS also can show good convergence and diversity in dealing with problems similar to feature selection, and obtain better sparse Pareto optimal solutions. The SparseEA-RP effectively resists the potential impact of increasing the dimension of the objective on the algorithm, making algorithm suitable for many-objective optimization problems. In summary, the SparseEA-AGDS algorithm can promote the convergence and diversity of the algorithm when dealing with many-objective large-scale sparse optimization problems, and also can obtain better sparse Pareto solutions.

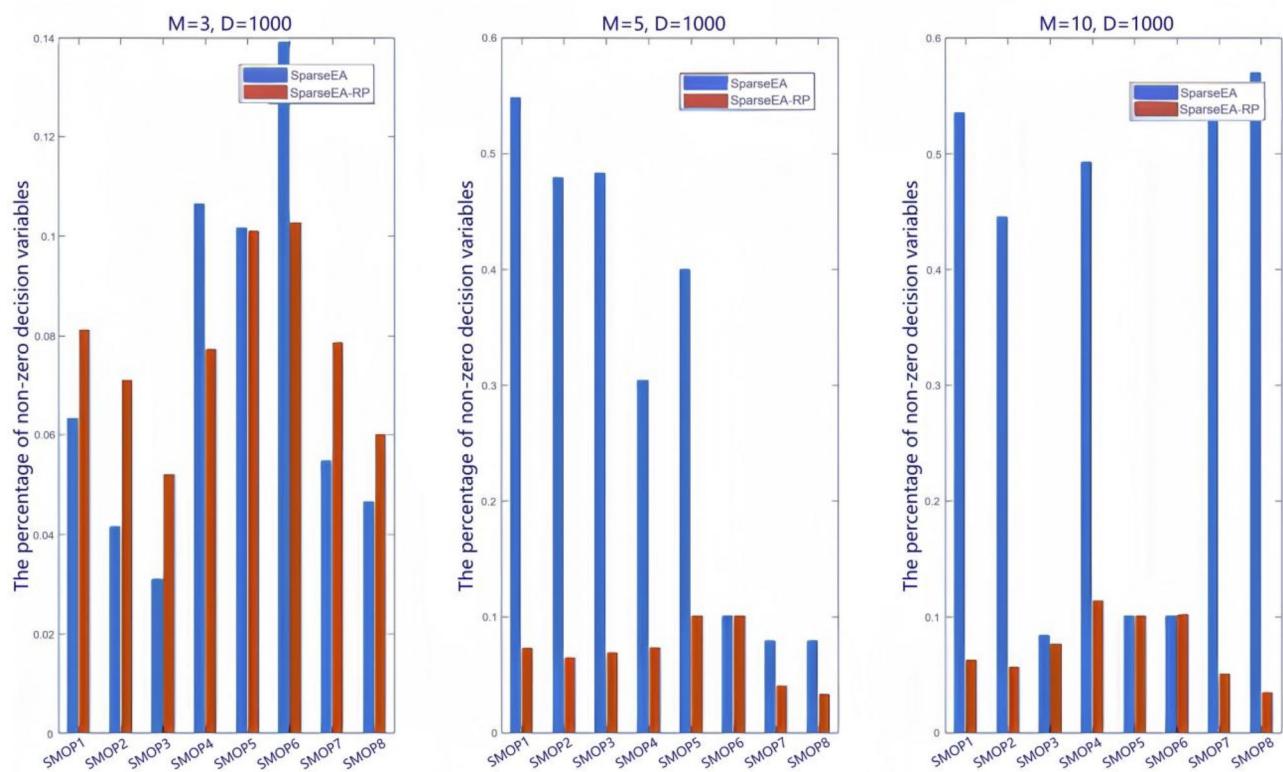


Fig. 7. Percentage of non-zero decision variables of two algorithms.

Problem	M	D	IGD							
			NSGAIII	LMOCSO	MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
SMOP1	3	100	2.8955e-1 (2.32e-3) -	5.0401e-1 (2.30e-2) -	7.7014e-2 (3.46e-3) -	7.5476e-2 (6.20e-3) -	8.1674e-2 (4.08e-3) -	6.0285e-2 (4.04e-3) -	5.0545e-2 (2.00e-3) -	4.8790e-2 (7.98e-4)
		500	2.0523e-1 (5.74e-3) -	5.3831e-1 (1.40e-2) -	6.6545e-2 (5.03e-3) -	6.9608e-2 (3.50e-3) -	7.5887e-2 (5.67e-3) -	5.2766e-2 (2.40e-3) -	4.8544e-2 (2.21e-3) +	5.1025e-2 (2.37e-3)
		1000	2.0092e-1 (9.17e-3) -	5.4945e-1 (4.50e-3) -	6.4408e-2 (5.83e-3) -	7.1199e-2 (3.02e-3) -	7.8788e-2 (2.75e-3) -	5.3607e-2 (3.01e-3) -	4.8080e-2 (1.56e-4) +	5.0527e-2 (1.99e-3)
	5	100	5.2945e-1 (8.38e-3) -	5.9616e-1 (3.79e-2) -	2.4494e-1 (2.20e-2) +	2.9559e-1 (5.57e-3) +	3.9837e-1 (7.34e-2) -	1.9468e-1 (3.20e-3) +	2.2973e-1 (6.56e-3) +	3.5249e-1 (9.34e-2)
		500	3.4203e-1 (8.67e-3) +	6.2500e-1 (5.17e-3) +	1.9212e-1 (5.76e-3) +	1.9654e-1 (9.37e-3) +	4.1472e-1 (4.04e-2) +	2.0302e-1 (1.21e-2) +	1.7415e-1 (5.43e-3) +	7.1334e-1 (1.64e-1)
		1000	3.1420e-1 (9.18e-3) +	6.3577e-1 (1.75e-2) +	1.7031e-1 (6.20e-3) +	1.8067e-1 (5.10e-3) +	3.8542e-1 (4.60e-2) +	2.1720e-1 (1.67e-2) +	1.6773e-1 (3.79e-3) +	7.4110e-1 (2.50e-1)
	8	100	5.4461e-1 (7.28e-2) -	9.7629e-1 (2.80e-1) -	6.5158e-1 (7.96e-2) -	8.6115e-1 (2.14e-1) -	7.7141e-1 (1.82e-1) -	1.2512e+0 (2.37e-1) -	1.0049e+0 (5.07e-2) -	5.1359e-1 (2.01e-1)
		500	5.0047e-1 (3.71e-3) +	9.7267e-1 (1.13e-1) -	4.6468e-1 (6.10e-2) +	7.7636e-1 (2.35e-1) +	6.4015e-1 (1.89e-2) +	1.7603e+0 (8.47e-1) -	9.3302e-1 (1.40e-1) -	9.0698e-1 (3.53e-2)
		1000	4.6248e-1 (1.18e-2) +	1.0249e+0 (1.84e-1) -	5.9467e-1 (3.71e-1) +	8.7173e-1 (3.00e-1) +	5.6395e-1 (7.06e-3) +	1.4634e+0 (6.19e-1) -	8.0172e-1 (6.66e-2) +	9.5909e-1 (1.76e-2)
	10	100	8.2538e-1 (5.01e-2) -	1.0437e+0 (6.91e-2) -	7.4313e-1 (1.01e-1) -	1.1771e+0 (1.84e-1) -	8.3853e-1 (4.68e-2) -	2.6445e+0 (3.56e-2) -	1.5171e+0 (7.81e-2) -	6.9138e-1 (4.62e-2)
		500	7.5340e-1 (2.01e-2) +	1.0322e+0 (3.44e-2) -	6.9348e-1 (2.69e-1) +	1.7701e+0 (1.95e-1) -	7.3625e-1 (3.94e-2) +	2.9816e+0 (8.23e-2) -	1.5760e+0 (5.14e-2) -	9.6069e-1 (1.42e-2)
		1000	7.6064e-1 (1.17e-2) +	1.0758e+0 (1.49e-1) -	6.5002e-1 (3.31e-1) +	1.6164e+0 (9.24e-2) -	5.3007e-1 (9.78e-2) +	2.7891e+0 (5.86e-1) -	1.5831e+0 (5.85e-2) -	9.7043e-1 (5.01e-2)
	15	100	5.0397e-1 (6.41e-2) +	1.1158e+0 (1.47e-1) -	7.4011e-1 (7.50e-2) +	1.2147e+0 (5.73e-3) -	8.9402e-1 (6.03e-2) =	2.9668e+0 (1.35e-1) -	1.4453e+0 (2.33e-1) -	9.3248e-1 (2.99e-2)
		500	6.5669e-1 (3.92e-2) +	1.1341e+0 (8.03e-2) -	6.0143e-1 (4.02e-2) +	1.6396e+0 (3.95e-2) -	8.6727e-1 (2.86e-2) +	3.3751e+0 (7.42e-1) -	2.5341e+0 (1.09e-1) -	1.0219e+0 (2.01e-2)
		1000	7.8198e-1 (1.82e-1) +	1.3924e+0 (4.78e-1) -	5.9170e-1 (3.47e-2) +	1.8436e+0 (8.41e-2) -	8.2606e-1 (7.96e-2) +	2.9373e+0 (8.57e-1) -	2.4916e+0 (2.78e-1) -	1.0109e+0 (5.19e-2)
SMOP2	3	100	9.3474e-1 (2.06e-2) -	1.5612e+0 (2.94e-2) -	1.0424e-1 (9.42e-3) -	1.1265e-1 (4.21e-3) -	1.2477e-1 (7.56e-3) -	8.0789e-2 (8.09e-3) +	7.2840e-2 (1.00e-3) +	8.5403e-2 (1.30e-2)
		500	7.8032e-1 (2.66e-2) -	1.5815e+0 (1.95e-2) -	1.0829e-1 (4.69e-3) -	1.0795e-1 (1.27e-2) -	1.2597e-1 (1.55e-3) -	9.2753e-2 (1.50e-2) -	7.2131e-2 (6.29e-3) +	7.3570e-2 (5.77e-3)
		1000	7.4165e-1 (8.45e-3) -	1.6282e+0 (1.74e-2) -	9.7653e-2 (6.90e-3) -	1.1894e-1 (1.54e-3) -	1.2809e-1 (2.59e-3) -	9.3610e-2 (2.78e-3) -	7.6138e-2 (3.19e-3) +	7.9901e-2 (6.08e-3)

Continued

Problem	M	D	IGD							
			NSGAIII	LMOCSO	MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
5	5	100	1.1709e+0 (3.83e-2) -	1.4188e+0 (1.80e-2) -	3.0318e-1 (9.39e-3) +	3.2348e-1 (3.94e-2) +	4.1965e-1 (7.37e-2) =	2.5599e-1 (8.07e-3) +	2.9853e-1 (2.21e-3) +	4.8383e-1 (1.01e-1)
		500	9.4643e-1 (6.52e-3) -	1.4637e+0 (1.17e-2) -	2.2182e-1 (7.46e-3) +	2.6772e-1 (6.19e-3) +	5.3937e-1 (1.64e-2) =	2.4290e-1 (3.77e-3) +	2.1081e-1 (5.67e-3) +	5.8003e-1 (2.32e-1)
		1000	8.8011e-1 (9.54e-3) -	1.4866e+0 (2.45e-2) -	2.1298e-1 (1.06e-2) +	2.4242e-1 (6.13e-3) +	4.5041e-1 (8.89e-2) -	2.5153e-1 (3.30e-2) +	2.0275e-1 (4.62e-3) +	4.2137e-1 (4.32e-1)
	8	100	1.1677e+0 (1.38e-2) -	1.4823e+0 (3.80e-1) -	8.6425e-1 (1.83e-1) -	8.2894e-1 (9.00e-2) -	8.3588e-1 (3.94e-2) -	1.1658e+0 (1.21e-1) -	1.2757e+0 (5.17e-2) -	6.4129e-1 (5.82e-2)
		500	1.1234e+0 (1.64e-2) -	1.6918e+0 (2.29e-1) -	5.4815e-1 (2.26e-1) +	1.0134e+0 (1.24e-1) -	7.1528e-1 (1.71e-1) +	1.4692e+0 (6.14e-1) -	9.2750e-1 (1.36e-1) -	8.9484e-1 (3.85e-2)
		1000	1.0466e+0 (5.44e-3) -	1.9662e+0 (4.61e-1) -	4.5906e-1 (6.04e-2) +	8.7763e-1 (3.33e-1) +	7.5152e-1 (6.02e-2) +	1.5318e+0 (5.18e-1) -	8.6424e-1 (7.24e-2) +	1.0313e+0 (5.27e-2)
10	10	100	1.6815e+0 (5.08e-2) -	2.1655e+0 (1.89e-1) -	8.0106e-1 (1.75e-1) -	1.2317e+0 (6.01e-2) -	9.0822e-1 (1.64e-1) -	1.3769e+0 (2.38e-1) -	1.4719e+0 (1.23e-1) -	6.8238e-1 (3.75e-2)
		500	1.4903e+0 (1.31e-2) -	2.1201e+0 (2.12e-1) -	8.4811e-1 (7.56e-2) =	1.5146e+0 (3.44e-1) -	7.3702e-1 (3.08e-1) +	2.2330e+0 (1.16e-1) -	1.6853e+0 (1.34e-1) -	8.5154e-1 (7.02e-2)
		1000	1.4191e+0 (3.19e-2) -	2.1237e+0 (2.04e-1) -	7.0776e-1 (1.16e-1) +	1.5555e+0 (3.81e-1) -	8.2622e-1 (1.74e-2) =	2.3314e+0 (1.89e-2) -	1.6144e+0 (1.19e-1) -	8.6336e-1 (3.19e-1)
	15	100	7.4375e-1 (8.86e-2) +	2.0423e+0 (3.63e-1) -	7.5919e-1 (4.18e-2) +	1.5129e+0 (3.17e-1) -	8.6014e-1 (8.29e-2) +	2.1975e+0 (1.04e+0) -	1.5601e+0 (7.19e-2) -	9.4098e-1 (1.74e-2)
		500	1.1111e+0 (2.75e-1) -	2.0748e+0 (1.08e-1) -	7.9997e-1 (1.05e-1) +	1.7859e+0 (1.46e-1) -	9.5407e-1 (2.23e-1) =	3.1151e+0 (1.78e-1) -	2.5827e+0 (1.35e-1) -	1.0772e+0 (3.17e-2)
		1000	9.6722e-1 (3.56e-2) +	2.0108e+0 (2.10e-1) -	1.0612e+0 (5.30e-3) =	2.3935e+0 (4.67e-1) -	9.0365e-1 (6.45e-2) +	2.8512e+0 (8.91e-1) -	2.6514e+0 (1.02e-1) -	1.1013e+0 (2.01e-2)
SMOP3	3	100	1.1086e+0 (1.92e-2) -	1.5187e+0 (5.88e-2) -	8.7509e-2 (3.34e-2) -	8.4081e-2 (5.48e-2) -	8.3078e-2 (3.21e-3) -	6.2954e-2 (1.74e-3) =	4.6654e-2 (3.36e-3) +	6.3735e-2 (1.42e-2)
		500	9.0318e-1 (3.19e-2) -	1.4644e+0 (5.65e-2) -	7.1957e-2 (5.23e-3) +	2.3321e-1 (2.55e-1) =	8.5052e-2 (1.50e-3) +	5.1174e-2 (2.37e-3) +	4.3444e-2 (7.78e-5) +	2.5777e-1 (3.50e-1)
		1000	9.0468e-1 (2.48e-2) -	1.4495e+0 (2.40e-2) -	7.4327e-2 (4.45e-3) +	6.8798e-2 (1.64e-2) +	8.7011e-2 (1.22e-3) +	4.9011e-2 (7.09e-4) +	4.5132e-2 (1.60e-3) +	5.3522e-1 (4.19e-1)
5	5	100	1.4879e+0 (3.60e-2) -	1.5918e+0 (8.87e-3) -	4.1508e-1 (2.34e-1) -	4.7935e-1 (3.45e-1) -	2.0935e-1 (1.52e-2) +	1.7586e-1 (9.15e-3) +	1.6551e-1 (2.08e-3) +	3.8364e-1 (3.12e-2)
		500	1.0907e+0 (5.08e-2) -	1.4291e+0 (6.42e-3) -	2.6702e-1 (3.01e-2) +	9.2270e-1 (2.19e-3) -	4.2661e-1 (3.18e-1) +	2.0831e-1 (1.37e-2) +	1.9511e-1 (5.27e-3) +	5.5926e-1 (1.11e-1)
		1000	1.0632e+0 (1.71e-2) -	1.3849e+0 (9.42e-3) -	1.0963e+0 (8.17e-1) -	5.8253e-1 (3.53e-1) +	5.9932e-1 (3.07e-1) +	2.0662e-1 (1.06e-2) +	1.9904e-1 (1.05e-2) +	7.2957e-1 (2.70e-1)
	8	100	1.5378e+0 (4.93e-2) -	1.6494e+0 (1.46e-1) -	9.7171e-1 (1.94e-1) -	8.2508e-1 (2.66e-1) -	7.4956e-1 (1.64e-1) -	7.3455e-1 (2.25e-1) -	1.4601e+0 (5.29e-1) -	6.4996e-1 (2.57e-2)
		500	1.3715e+0 (4.01e-2) -	1.5491e+0 (1.80e-2) -	5.0265e-1 (1.12e-1) +	1.0778e+0 (5.08e-2) -	9.3874e-1 (8.35e-2) -	9.3197e-1 (1.19e-1) -	1.9867e+0 (5.37e-1) -	8.3230e-1 (6.01e-2)
		1000	1.2994e+0 (1.18e-2) -	1.4934e+0 (6.61e-2) -	4.6124e-1 (3.11e-2) +	1.0467e+0 (4.30e-2) -	9.4367e-1 (1.20e-1) =	7.6795e-1 (1.86e-1) +	1.5600e+0 (1.71e-1) -	1.0021e+0 (1.07e-2)
	10	100	2.1354e+0 (1.80e-1) -	2.4917e+0 (2.11e-1) -	1.0950e+0 (5.07e-2) -	1.1320e+0 (8.46e-5) -	9.6308e-1 (4.42e-2) -	1.4996e+0 (4.54e-1) -	1.4803e+0 (7.11e-1) -	7.0115e-1 (3.85e-2)
		500	1.8520e+0 (4.48e-2) -	1.9750e+0 (2.34e-1) -	9.6664e-1 (2.66e-1) -	1.6357e+0 (6.92e-1) -	9.9283e-1 (5.39e-2) -	1.6078e+0 (7.70e-1) -	2.6902e+0 (7.83e-2) -	9.4705e-1 (1.73e-2)
		1000	1.7659e+0 (3.78e-3) -	2.1044e+0 (3.00e-1) -	7.3739e-1 (2.40e-1) +	1.3155e+0 (7.19e-1) -	9.4838e-1 (1.18e-1) =	1.9859e+0 (7.55e-1) -	2.6369e+0 (8.32e-2) -	9.6225e-1 (4.76e-2)
15	15	100	1.0091e+0 (4.20e-2) -	2.4798e+0 (2.35e-1) -	9.8269e-1 (1.43e-1) -	1.1603e+0 (4.82e-1) -	8.8543e-1 (6.17e-2) +	1.5626e+0 (3.94e-1) -	2.5537e+0 (3.38e-1) -	9.7432e-1 (5.93e-2)
		500	1.1735e+0 (1.59e-1) -	2.4426e+0 (1.14e-1) -	6.1581e-1 (3.74e-2) +	1.3009e+0 (2.74e-1) -	9.7030e-1 (9.39e-2) =	2.2751e+0 (9.65e-1) -	3.5225e+0 (9.62e-3) -	1.0521e+0 (8.55e-3)
		1000	1.2073e+0 (4.04e-2) -	2.3989e+0 (1.89e-1) -	9.8499e-1 (4.95e-2) +	2.1938e+0 (1.02e+0) -	9.8851e-1 (4.29e-2) =	3.1351e+0 (6.94e-2) -	3.2576e+0 (3.24e-1) -	1.0199e+0 (5.06e-2)
SMOP4	3	100	3.3577e-1 (2.44e-2) -	5.8343e-1 (7.29e-3) -	5.5384e-2 (3.23e-3) -	3.0278e-2 (3.35e-4) -	5.8250e-2 (1.06e-3) -	3.0470e-2 (5.36e-4) -	3.0241e-2 (1.85e-4) -	2.8565e-2 (2.28e-4)
	500		2.8642e-1 (6.73e-3) -	6.1738e-1 (6.49e-3) -	6.1733e-2 (1.12e-2) -	3.0346e-2 (4.16e-4) -	6.1630e-2 (8.38e-3) -	3.0111e-2 (6.23e-4) -	3.0570e-2 (4.56e-4) -	2.8643e-2 (1.09e-4)

Continued

Problem	M	D	IGD							
			NSGAIII	LMOCSO	MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
		1000	2.6097e-1 (1.47e-3) -	6.0264e-1 (2.78e-2) -	9.4272e-2 (5.38e-2) -	3.0352e-2 (4.66e-4) -	5.7218e-2 (4.84e-3) -	3.0151e-2 (2.95e-4) -	2.9837e-2 (1.23e-4) =	2.8767e-2 (2.32e-5)
	5	100	3.0051e-1 (2.27e-2) +	3.3845e-1 (3.88e-3) +	1.2001e-1 (3.09e-3) +	1.0266e-1 (1.13e-3) +	3.0244e-1 (4.48e-2) +	1.2299e-1 (2.07e-2) +	8.7577e-2 (4.03e-3) +	5.4422e-1 (2.43e-1)
		500	2.7989e-1 (4.93e-3) +	3.6390e-1 (3.53e-3) +	1.2685e-1 (3.24e-3) +	8.7710e-2 (2.62e-3) +	2.1904e-1 (1.28e-2) +	1.1922e-1 (1.99e-2) +	8.4564e-2 (1.95e-3) +	7.3305e-1 (2.87e-1)
		1000	2.6599e-1 (9.77e-4) =	4.0535e-1 (3.36e-2) -	1.3137e-1 (8.63e-3) +	8.5188e-2 (1.06e-3) +	2.0448e-1 (4.74e-2) =	1.3336e-1 (1.33e-2) +	8.3651e-2 (4.49e-3) +	2.7794e-1 (3.70e-1)
	8	100	1.9686e-1 (3.80e-2) +	2.3183e-1 (1.89e-2) =	3.5748e-1 (1.50e-1) -	3.7789e-1 (1.44e-2) -	3.4696e-1 (9.67e-2) -	6.5865e-1 (1.35e-1) -	3.3186e-1 (1.96e-2) -	2.9522e-1 (1.45e-1)
		500	2.2201e-1 (5.50e-2) +	4.3988e-1 (1.17e-1) +	3.9686e-1 (3.56e-1) +	2.3659e-1 (4.09e-3) +	3.3302e-1 (1.94e-2) +	5.5772e-1 (6.22e-2) +	2.5886e-1 (5.98e-2) +	9.2698e-1 (2.51e-2)
		1000	2.9124e-1 (3.48e-2) +	4.1747e-1 (7.82e-2) +	4.3869e-1 (4.67e-1) +	1.9807e-1 (2.48e-2) +	2.7751e-1 (1.77e-2) +	4.4655e-1 (1.24e-1) +	1.9977e-1 (4.36e-2) +	9.7363e-1 (6.36e-3)
	10	100	2.6510e-1 (2.90e-2) +	2.9862e-1 (7.82e-2) +	2.6885e-1 (6.10e-2) +	4.2036e-1 (2.89e-2) +	4.0550e-1 (6.59e-2) +	7.8218e-1 (7.97e-2) -	4.1005e-1 (4.73e-2) +	6.2995e-1 (4.04e-1)
		500	2.6719e-1 (3.43e-2) +	4.1698e-1 (1.56e-1) +	2.3862e-1 (1.21e-2) +	4.5722e-1 (8.39e-2) +	3.3834e-1 (5.98e-2) +	4.5730e-1 (1.49e-1) +	3.1706e-1 (5.11e-2) +	9.7054e-1 (2.75e-2)
		1000	1.9676e-1 (3.82e-2) +	3.4697e-1 (8.90e-2) +	2.1653e-1 (4.70e-2) +	2.8920e-1 (7.04e-2) +	5.2409e-1 (4.00e-1) +	5.7398e-1 (1.05e-1) +	3.1984e-1 (1.27e-1) +	9.8478e-1 (2.87e-3)
	15	100	1.9585e-1 (2.28e-2) +	4.5008e-1 (3.08e-2) +	3.3066e-1 (1.08e-1) +	4.7170e-1 (1.12e-1) +	7.1594e-1 (3.87e-1) +	9.7845e-1 (1.94e-1) -	5.0726e-1 (9.52e-2) +	8.2297e-1 (1.77e-1)
		500	2.5860e-1 (7.46e-2) +	6.6223e-1 (2.01e-1) +	7.2607e-1 (4.72e-1) +	7.8981e-1 (1.70e-1) +	7.6737e-1 (4.01e-1) +	1.2333e+0 (1.91e-1) -	7.3897e-1 (9.01e-2) +	9.9868e-1 (3.79e-6)
		1000	3.4890e-1 (2.49e-1) +	6.9003e-1 (7.12e-2) +	7.4348e-1 (4.33e-1) +	6.0172e-1 (1.18e-1) +	5.2508e-1 (4.16e-1) +	1.1925e+0 (2.02e-1) -	7.9565e-1 (1.94e-1) +	9.7097e-1 (4.23e-2)
SMOP5	3	100	2.5725e-1 (2.15e-3) -	2.6270e-1 (2.57e-3) -	5.4701e-2 (1.02e-2) -	3.3174e-2 (7.10e-4) -	6.2754e-2 (1.61e-3) -	3.5011e-2 (9.06e-4) -	3.5052e-2 (7.13e-4) -	3.1033e-2 (2.29e-3)
		500	2.3649e-1 (1.63e-3) -	2.6708e-1 (4.54e-4) -	9.5695e-2 (4.72e-2) -	3.1220e-2 (5.92e-4) -	7.2042e-2 (1.20e-3) -	3.3675e-2 (8.78e-4) -	3.1830e-2 (5.39e-4) -	2.8691e-2 (1.20e-4)
		1000	2.3024e-1 (1.42e-3) -	2.6722e-1 (1.39e-3) -	1.0555e-1 (4.62e-2) -	3.1086e-2 (6.22e-4) -	6.2978e-2 (5.09e-3) -	3.1920e-2 (1.37e-3) -	3.1653e-2 (5.09e-4) -	2.8825e-2 (4.34e-6)
	5	100	2.0824e-1 (1.96e-3) =	1.5720e-1 (5.48e-3) +	1.3146e-1 (1.05e-2) +	1.2035e-1 (1.00e-2) +	2.7540e-1 (7.06e-2) -	1.3713e-1 (1.87e-2) +	1.3432e-1 (1.24e-2) +	2.4916e-1 (1.03e-1)
		500	1.8852e-1 (4.71e-3) +	1.8947e-1 (4.09e-2) +	1.3161e-1 (5.66e-3) +	8.9825e-2 (9.50e-4) +	4.6256e-1 (4.18e-1) +	1.7679e-1 (5.50e-2) +	9.0342e-2 (3.70e-3) -	8.3372e-1 (1.79e-2)
		1000	1.7874e-1 (1.76e-3) +	1.8133e-1 (2.27e-2) +	1.4952e-1 (5.84e-3) +	9.2640e-2 (3.03e-4) +	4.2915e-1 (4.33e-1) +	1.5922e-1 (3.01e-2) +	9.4563e-2 (4.14e-3) -	8.5206e-1 (7.06e-2)
	8	100	1.3827e-1 (1.04e-2) +	2.1226e-1 (8.10e-2) +	4.4158e-1 (2.52e-1) -	4.7355e-1 (8.79e-2) -	6.3064e-1 (2.39e-1) -	1.0144e+0 (4.73e-2) -	4.7149e-1 (9.20e-2) -	3.5571e-1 (9.38e-2)
		500	1.4832e-1 (2.68e-2) +	2.9182e-1 (1.88e-2) +	4.5918e-1 (2.67e-1) +	2.7282e-1 (7.50e-2) +	7.7419e-1 (3.37e-1) +	6.1539e-1 (5.71e-2) +	2.7330e-1 (4.11e-2) +	8.5642e-1 (1.22e-1)
		1000	1.7895e-1 (5.12e-2) +	3.9051e-1 (1.38e-1) +	2.4831e-1 (1.35e-1) +	1.7561e-1 (5.42e-3) +	5.3275e-1 (3.70e-1) +	1.0470e+0 (2.97e-1) -	2.5978e-1 (4.80e-2) +	8.9057e-1 (4.06e-2)
	10	100	1.7779e-1 (2.05e-2) +	2.3256e-1 (8.35e-2) +	3.0769e-1 (1.51e-2) +	4.9636e-1 (3.96e-2) -	6.5774e-1 (2.82e-1) -	1.1850e+0 (1.02e-1) -	5.7489e-1 (1.52e-1) -	4.5622e-1 (1.06e-1)
		500	1.9803e-1 (5.69e-2) +	3.6722e-1 (6.32e-2) +	2.1709e-1 (2.61e-2) +	6.3051e-1 (1.05e-1) +	9.7851e-1 (4.45e-4) -	1.5051e+0 (2.05e-1) -	4.4483e-1 (1.32e-1) +	9.3560e-1 (5.44e-2)
		1000	1.8956e-1 (8.90e-3) +	3.7458e-1 (8.63e-2) +	2.0235e-1 (8.78e-3) +	4.7728e-1 (6.51e-2) +	9.7752e-1 (1.10e-2) -	1.1535e+0 (3.89e-1) -	3.3170e-1 (3.46e-2) +	8.9946e-1 (1.07e-1)
	15	100	2.3930e-1 (1.51e-1) +	4.4530e-1 (9.70e-2) +	3.6632e-1 (2.07e-1) +	4.6564e-1 (5.10e-2) +	7.7592e-1 (4.86e-2) +	1.3490e+0 (3.30e-1) -	4.9681e-1 (1.62e-1) +	9.4358e-1 (4.77e-2)
		500	2.7730e-1 (9.43e-2) +	4.9181e-1 (1.56e-1) +	3.1035e-1 (3.05e-2) +	9.6766e-1 (1.54e-1) =	9.4913e-1 (7.62e-2) =	1.5107e+0 (7.05e-1) -	1.0692e+0 (2.94e-1) -	9.7214e-1 (1.66e-2)
		1000	2.0435e-1 (9.85e-2) +	5.8328e-1 (6.51e-2) +	2.6156e-1 (4.58e-2) +	9.1646e-1 (5.03e-1) =	8.9763e-1 (6.42e-2) =	2.1047e+0 (7.76e-1) -	8.1051e-1 (6.67e-2) +	9.5810e-1 (3.44e-2)

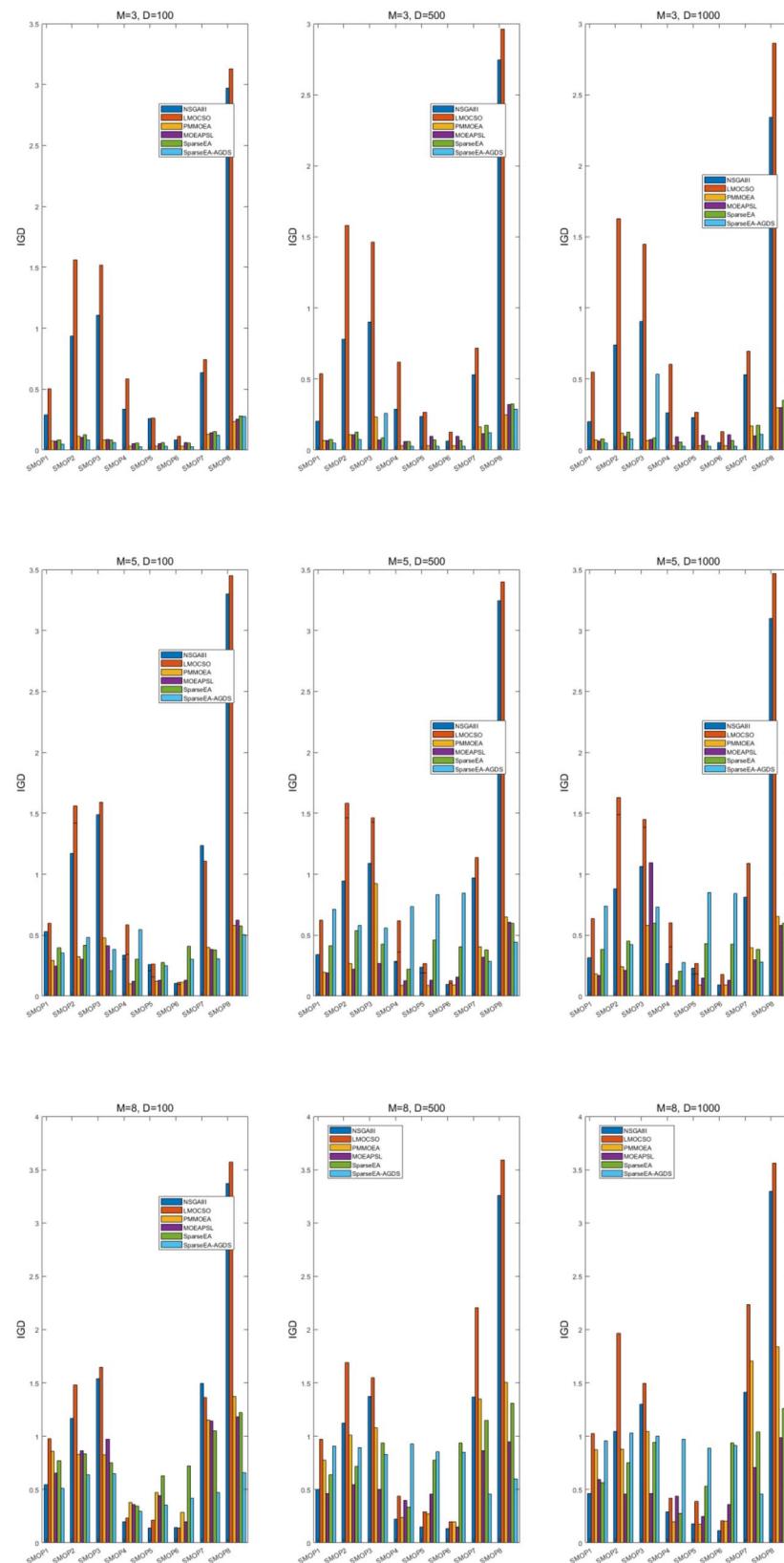
Continued

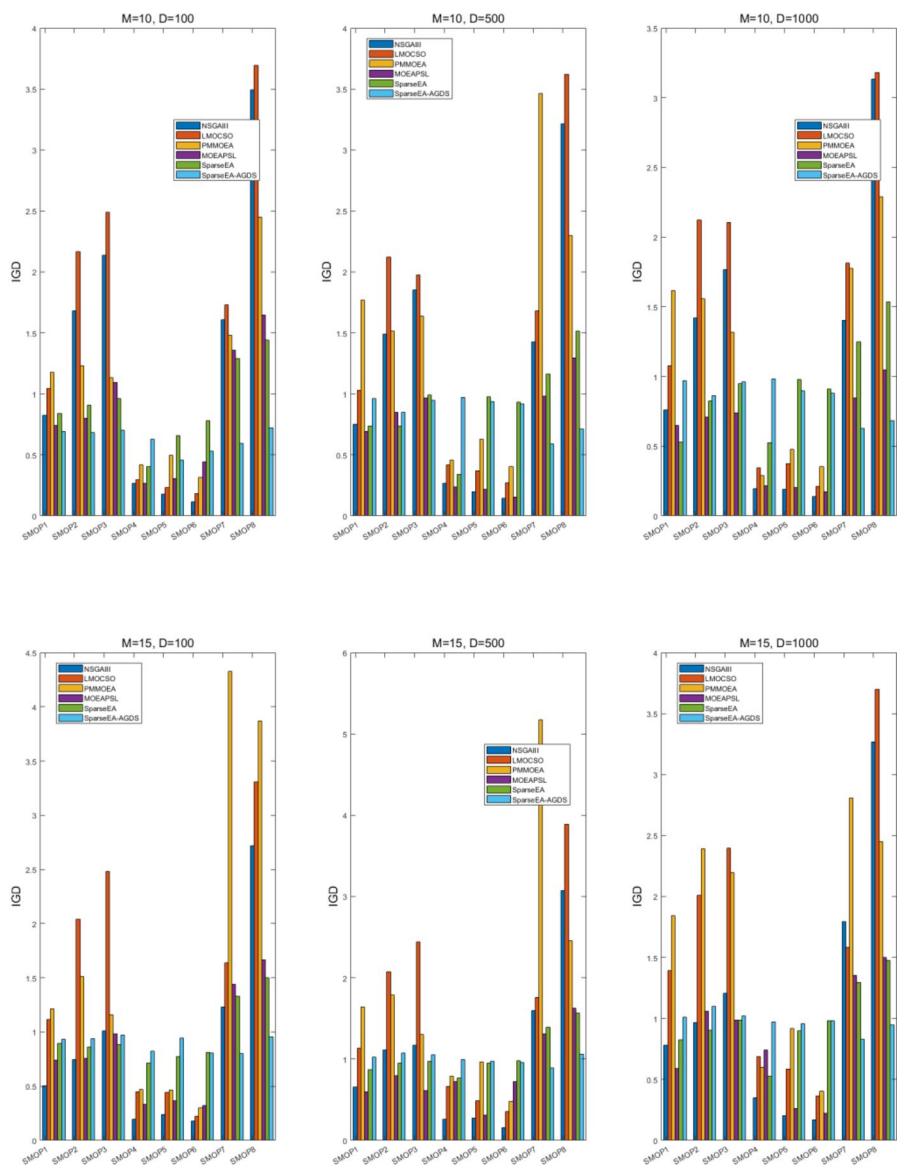
Problem	M	D	IGD							
			NSGAIII	LMOCSO	MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
SMOP6	3	100	8.2328e-2 (3.90e-3) -	1.1445e-1 (1.33e-3) -	6.0332e-2 (6.25e-4) -	3.3736e-2 (1.10e-2) -	5.8936e-2 (1.43e-3) -	3.5752e-2 (8.24e-4) -	3.5290e-2 (8.24e-4) -	2.9892e-2 (4.29e-4)
		500	6.3463e-2 (1.79e-3) -	1.2518e-1 (4.40e-3) -	9.9129e-2 (1.38e-2) -	3.1949e-2 (7.03e-4) -	6.8358e-2 (3.67e-3) -	3.3832e-2 (8.67e-4) -	3.2436e-2 (5.93e-4) -	2.9133e-2 (2.28e-4)
		1000	5.5057e-2 (2.06e-3) -	1.2866e-1 (9.31e-4) -	1.0698e-1 (4.90e-2) -	3.1309e-2 (7.17e-4) -	6.7985e-2 (4.88e-3) -	3.2531e-2 (9.03e-4) -	3.2037e-2 (1.33e-3) -	2.8986e-2 (3.30e-5)
	5	100	1.0638e-1 (5.14e-3) +	9.7428e-2 (1.37e-3) +	1.3024e-1 (1.58e-2) +	1.1313e-1 (7.27e-3) +	4.1035e-1 (2.96e-1) -	1.4608e-1 (5.17e-3) +	1.1563e-1 (3.27e-3) +	3.0325e-1 (5.94e-2)
		500	9.4897e-2 (7.48e-4) +	1.0327e-1 (2.05e-3) +	1.5631e-1 (2.42e-2) +	9.1618e-2 (2.30e-3) +	4.0623e-1 (4.20e-1) +	1.0678e-1 (5.50e-3) +	9.2562e-2 (2.17e-3) +	8.4562e-1 (5.97e-2)
		1000	9.1793e-2 (1.49e-3) +	1.7647e-1 (1.28e-1) +	1.3036e-1 (8.84e-3) +	9.0591e-2 (8.56e-4) +	4.2606e-1 (4.51e-1) +	1.2157e-1 (3.20e-2) +	9.0578e-2 (2.87e-3) +	8.4131e-1 (6.40e-2)
	8	100	1.4258e-1 (9.86e-3) +	1.4068e-1 (3.91e-2) +	2.0031e-1 (2.90e-2) +	2.8568e-1 (3.25e-3) +	7.2023e-1 (2.86e-2) -	3.3339e-1 (9.41e-2) +	2.4171e-1 (1.01e-2) +	4.1923e-1 (1.46e-1)
		500	1.3595e-1 (3.55e-2) +	1.9905e-1 (2.32e-2) +	1.5057e-1 (1.53e-2) +	1.9818e-1 (1.09e-2) +	9.3661e-1 (5.74e-2) -	2.5787e-1 (1.16e-1) +	2.1367e-1 (2.35e-2) +	8.4972e-1 (3.02e-2)
		1000	1.1740e-1 (3.44e-2) +	2.0880e-1 (1.70e-2) +	3.5873e-1 (3.34e-1) +	2.0520e-1 (2.07e-3) +	9.3996e-1 (3.68e-2) -	2.6963e-1 (4.13e-2) +	1.6247e-1 (2.14e-3) +	9.1392e-1 (5.72e-2)
	10	100	1.1339e-1 (7.80e-3) +	1.8383e-1 (1.71e-2) +	4.4401e-1 (3.88e-1) +	3.1768e-1 (4.33e-2) +	7.8287e-1 (1.53e-3) -	5.0534e-1 (1.03e-1) +	3.4941e-1 (4.62e-2) +	5.3195e-1 (7.24e-2)
		500	1.4356e-1 (1.33e-2) +	2.6930e-1 (1.85e-1) +	1.5477e-1 (1.21e-2) +	4.0156e-1 (3.44e-2) +	9.3260e-1 (5.18e-2) -	5.6563e-1 (1.56e-1) +	2.9787e-1 (4.67e-2) +	9.1866e-1 (4.74e-2)
		1000	1.4009e-1 (8.85e-3) +	2.1360e-1 (5.69e-2) +	1.7496e-1 (2.03e-2) +	3.5283e-1 (3.00e-2) +	9.0843e-1 (1.16e-1) -	2.6660e-1 (1.17e-1) +	2.3257e-1 (2.79e-2) +	8.8034e-1 (9.74e-2)
	15	100	1.8169e-1 (4.61e-2) +	2.2144e-1 (1.69e-2) +	3.2203e-1 (1.12e-1) +	3.0108e-1 (1.55e-2) +	8.1175e-1 (2.87e-2) -	5.0736e-1 (3.92e-2) +	2.2339e-1 (4.11e-2) +	8.0647e-1 (5.88e-2)
		500	1.5618e-1 (2.07e-2) +	3.5822e-1 (3.64e-2) +	7.2668e-1 (3.99e-1) +	4.8102e-1 (3.31e-2) +	9.8078e-1 (1.59e-2) -	6.7039e-1 (3.63e-1) +	5.7798e-1 (1.36e-1) +	9.5782e-1 (2.20e-2)
		1000	1.6912e-1 (2.30e-2) +	3.6444e-1 (6.68e-2) +	2.2300e-1 (8.66e-2) +	4.0448e-1 (6.58e-2) +	9.8287e-1 (1.27e-2) -	4.3198e-1 (7.16e-2) +	5.2755e-1 (1.34e-1) +	9.7932e-1 (8.57e-3)
SMOP7	3	100	6.3624e-1 (4.39e-2) -	7.4326e-1 (2.34e-2) -	1.4192e-1 (2.94e-2) -	1.3150e-1 (1.78e-2) -	1.5422e-1 (1.56e-2) -	8.2589e-2 (1.17e-2) +	8.9706e-2 (8.10e-3) +	1.2062e-1 (8.78e-3)
		500	5.3176e-1 (9.38e-3) -	7.1682e-1 (1.45e-2) -	1.1559e-1 (1.32e-2) +	1.6230e-1 (2.56e-2) -	1.7306e-1 (2.15e-2) -	1.2934e-1 (2.01e-2) -	1.1211e-1 (1.54e-2) +	1.2451e-1 (1.18e-2)
		1000	5.2936e-1 (4.85e-2) -	6.9705e-1 (2.67e-2) -	1.0093e-1 (3.93e-3) +	1.7094e-1 (4.36e-3) -	1.7353e-1 (4.61e-3) -	1.1627e-1 (1.38e-2) -	1.2428e-1 (9.95e-3) -	1.1230e-1 (4.10e-3)
	5	100	1.2355e+0 (5.09e-2) -	1.1062e+0 (3.96e-2) -	3.8319e-1 (2.51e-2) -	4.0220e-1 (1.65e-2) -	3.7977e-1 (3.72e-3) -	3.4607e-1 (6.61e-3) -	3.8583e-1 (2.04e-2) -	3.0797e-1 (2.26e-2)
		500	9.7044e-1 (2.23e-2) -	1.1371e+0 (2.99e-2) -	3.1833e-1 (1.47e-2) -	4.0525e-1 (2.00e-2) -	3.8093e-1 (1.37e-2) -	3.7022e-1 (1.56e-2) -	4.4098e-1 (2.28e-2) -	2.8276e-1 (1.31e-3)
		1000	8.1135e-1 (1.78e-2) -	1.0909e+0 (3.76e-2) -	2.9720e-1 (1.85e-2) -	3.9531e-1 (3.66e-3) -	3.8186e-1 (7.53e-3) -	3.7799e-1 (2.22e-3) -	5.3363e-1 (4.87e-2) -	2.8063e-1 (2.94e-3)
	8	100	1.4944e+0 (1.07e-1) -	1.3615e+0 (1.34e-2) -	1.1428e+0 (1.90e-1) -	1.1548e+0 (2.13e-1) -	1.0509e+0 (6.86e-2) -	2.7046e+0 (1.52e+0) -	1.3583e+0 (1.63e-1) -	4.7039e-1 (6.53e-3)
		500	1.3700e+0 (4.50e-2) -	2.2077e+0 (1.06e+0) -	8.6514e-1 (7.74e-2) -	1.3502e+0 (9.34e-2) -	1.1498e+0 (1.18e-1) -	2.7050e+0 (1.21e+0) -	2.5408e+0 (1.21e+0) -	4.6001e-1 (9.84e-3)
		1000	1.4121e+0 (1.20e-1) -	2.2330e+0 (1.22e+0) -	7.0595e-1 (1.99e-2) -	1.7086e+0 (3.56e-1) -	1.0421e+0 (4.69e-2) -	4.0067e+0 (5.11e-1) -	1.8004e+0 (9.85e-1) -	4.5892e-1 (1.27e-3)
	10	100	1.6070e+0 (1.50e-1) -	1.7316e+0 (1.70e-1) -	1.3597e+0 (5.71e-2) -	1.4799e+0 (1.28e-1) -	1.2888e+0 (8.04e-2) -	3.3083e+0 (1.30e+0) -	2.6202e+0 (1.09e+0) -	5.9325e-1 (7.15e-3)
		500	1.4277e+0 (1.52e-1) -	1.6831e+0 (4.80e-1) -	9.8274e-1 (1.22e-1) -	3.4654e+0 (1.67e+0) -	1.1610e+0 (5.98e-2) -	6.8028e+0 (4.59e+0) -	4.6778e+0 (5.67e-2) -	5.8800e-1 (6.54e-3)
		1000	1.4044e+0 (1.18e-1) -	1.8157e+0 (5.93e-1) -	8.4434e-1 (1.41e-2) -	1.7754e+0 (5.37e-1) -	1.2474e+0 (5.90e-3) -	4.7304e+0 (3.97e+0) -	4.1106e+0 (2.31e-1) -	6.2741e-1 (6.29e-2)
	15	100	1.2325e+0 (4.66e-2) -	1.6383e+0 (1.77e-1) -	1.4406e+0 (1.37e-2) -	4.3273e+0 (4.30e-1) -	1.3283e+0 (8.49e-2) -	7.9811e+0 (3.30e-1) =	6.0139e+0 (1.69e-1) -	7.9927e-1 (6.82e-2)
		500	1.5963e+0 (1.53e-1) -	1.7596e+0 (3.06e-1) -	1.3120e+0 (5.83e-2) -	5.1707e+0 (9.21e-2) -	1.3951e+0 (5.07e-2) -	7.5416e+0 (4.34e+0) -	5.4110e+0 (2.19e-1) -	8.9346e-1 (1.91e-1)
		1000	1.7921e+0 (1.56e-1) -	1.5816e+0 (2.84e-1) -	1.3520e+0 (1.14e-1) -	2.8081e+0 (1.75e+0) -	1.2959e+0 (1.75e+0) -	6.6434e+0 (6.11e-2) -	4.5661e+0 (3.74e+0) -	8.2867e-1 (3.50e-2)

Continued

Problem	M	D	IGD							
			NSGAIII	LMOCSO	MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
SMOP8	3	100	2.9715e+0 (1.26e-2) -	3.1284e+0 (7.03e-2) -	2.5314e-1 (3.63e-2) =	2.3370e-1 (4.53e-2) +	2.8004e-1 (2.99e-2) -	2.0006e-1 (1.21e-2) =	1.7513e-1 (2.44e-2) +	2.7538e-1 (6.42e-2)
		500	2.7437e+0 (9.90e-2) =	2.9620e+0 (5.05e-2) -	3.2020e-1 (3.19e-2) -	2.4908e-1 (2.55e-2) +	3.2652e-1 (7.22e-3) -	2.3745e-1 (2.60e-2) +	1.7016e-1 (2.05e-2) +	2.8996e-1 (1.98e-2)
		1000	2.3407e+0 (2.57e-1) =	2.8606e+0 (4.64e-2) -	2.9746e-1 (2.59e-2) -	2.9839e-1 (1.32e-2) -	3.5225e-1 (1.23e-2) -	2.2233e-1 (2.02e-2) +	1.7305e-1 (6.83e-3) +	2.9008e-1 (1.33e-2)
	5	100	3.3007e+0 (3.88e-2) -	3.4501e+0 (7.69e-2) -	6.2130e-1 (1.40e-2) -	5.8036e-1 (3.10e-2) -	5.7808e-1 (1.52e-2) -	5.6377e-1 (1.43e-2) -	6.0214e-1 (2.29e-2) -	5.0491e-1 (1.61e-2)
		500	3.2444e+0 (7.02e-2) -	3.3993e+0 (2.70e-2) -	6.0829e-1 (1.84e-2) -	6.5088e-1 (1.77e-2) -	5.9570e-1 (1.67e-2) -	5.6522e-1 (1.09e-2) -	7.0942e-1 (3.30e-2) -	4.4378e-1 (1.20e-2)
		1000	3.1016e+0 (1.13e-1) -	3.4677e+0 (4.68e-2) -	5.8189e-1 (1.98e-2) -	6.5486e-1 (1.44e-2) -	5.9920e-1 (3.57e-3) -	5.9303e-1 (2.00e-2) -	9.3126e-1 (1.70e-2) -	4.2774e-1 (2.73e-3)
	8	100	3.3710e+0 (6.02e-2) -	3.5732e+0 (3.82e-2) -	1.1805e+0 (1.42e-1) -	1.3710e+0 (2.14e-1) -	1.2231e+0 (1.13e-1) -	1.3216e+0 (5.87e-2) -	1.6130e+0 (2.67e-1) -	6.5923e-1 (3.72e-2)
		500	3.2582e+0 (1.24e-1) -	3.5889e+0 (4.29e-2) -	9.4682e-1 (4.18e-2) -	1.5034e+0 (7.65e-2) -	1.3111e+0 (4.54e-2) -	1.7361e+0 (1.57e-1) -	2.2009e+0 (1.21e+0) -	5.9798e-1 (4.96e-3)
		1000	3.2996e+0 (2.18e-2) -	3.5636e+0 (1.23e-2) -	9.8435e-1 (7.14e-2) -	1.8402e+0 (9.77e-2) -	1.2631e+0 (5.04e-3) -	2.6670e+0 (9.44e-1) -	2.9472e+0 (3.07e-1) -	5.8630e-1 (1.91e-2)
	10	100	3.4945e+0 (2.62e-2) -	3.6934e+0 (7.35e-3) -	1.6497e+0 (1.33e-2) -	2.4488e+0 (5.67e-1) -	1.4414e+0 (1.28e-1) -	1.8243e+0 (1.72e-1) -	1.6228e+0 (7.76e-2) -	7.2138e-1 (3.52e-2)
		500	3.2154e+0 (1.40e-1) -	3.6207e+0 (4.75e-2) -	1.2950e+0 (1.11e-1) -	2.2998e+0 (1.17e+0) -	1.5159e+0 (6.65e-2) -	3.1240e+0 (1.34e+0) -	3.7776e+0 (8.40e-2) -	7.1251e-1 (8.44e-3)
		1000	3.1325e+0 (5.90e-2) -	3.1823e+0 (4.08e-1) -	1.0470e+0 (8.66e-2) -	2.2914e+0 (7.79e-1) -	1.5346e+0 (8.33e-2) -	3.0969e+0 (1.29e+0) -	3.6317e+0 (5.34e-2) -	6.8152e-1 (7.59e-3)
	15	100	2.7161e+0 (6.88e-2) -	3.3058e+0 (2.74e-1) -	1.6646e+0 (4.39e-2) -	3.8692e+0 (7.17e-2) -	1.5025e+0 (3.15e-2) -	3.1156e+0 (1.24e+0) -	3.8585e+0 (2.37e-2) -	9.5325e-1 (8.55e-2)
		500	3.0739e+0 (2.64e-2) -	3.8881e+0 (6.63e-1) -	1.6288e+0 (7.80e-2) -	2.4576e+0 (1.36e+0) -	1.5656e+0 (4.78e-2) -	3.1983e+0 (1.34e+0) -	3.8689e+0 (6.01e-2) -	1.0641e+0 (2.23e-1)
		1000	3.2670e+0 (9.68e-2) -	3.6973e+0 (8.37e-1) -	1.4984e+0 (2.23e-1) -	2.4521e+0 (1.37e+0) -	1.4739e+0 (6.50e-2) -	3.1845e+0 (1.28e+0) -	3.7614e+0 (3.25e-2) -	9.4680e-1 (4.08e-2)
+/-/=	45/71/4	36/83/1	62/55/3	44/73/3	36/72/12	39/78/3	54/65/1			

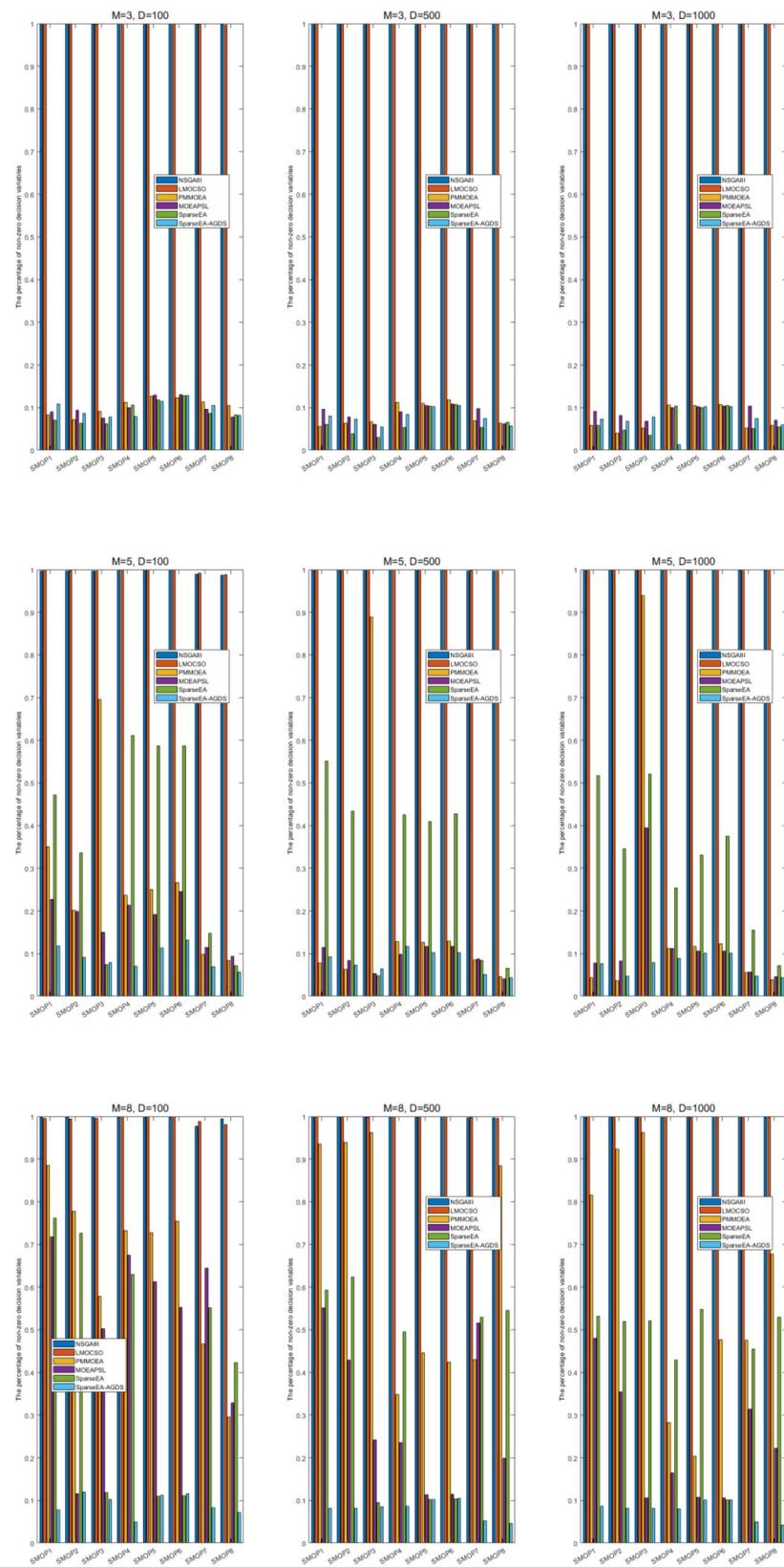
Table 8. IGD results of eight algorithms on SMOP benchmark problems.

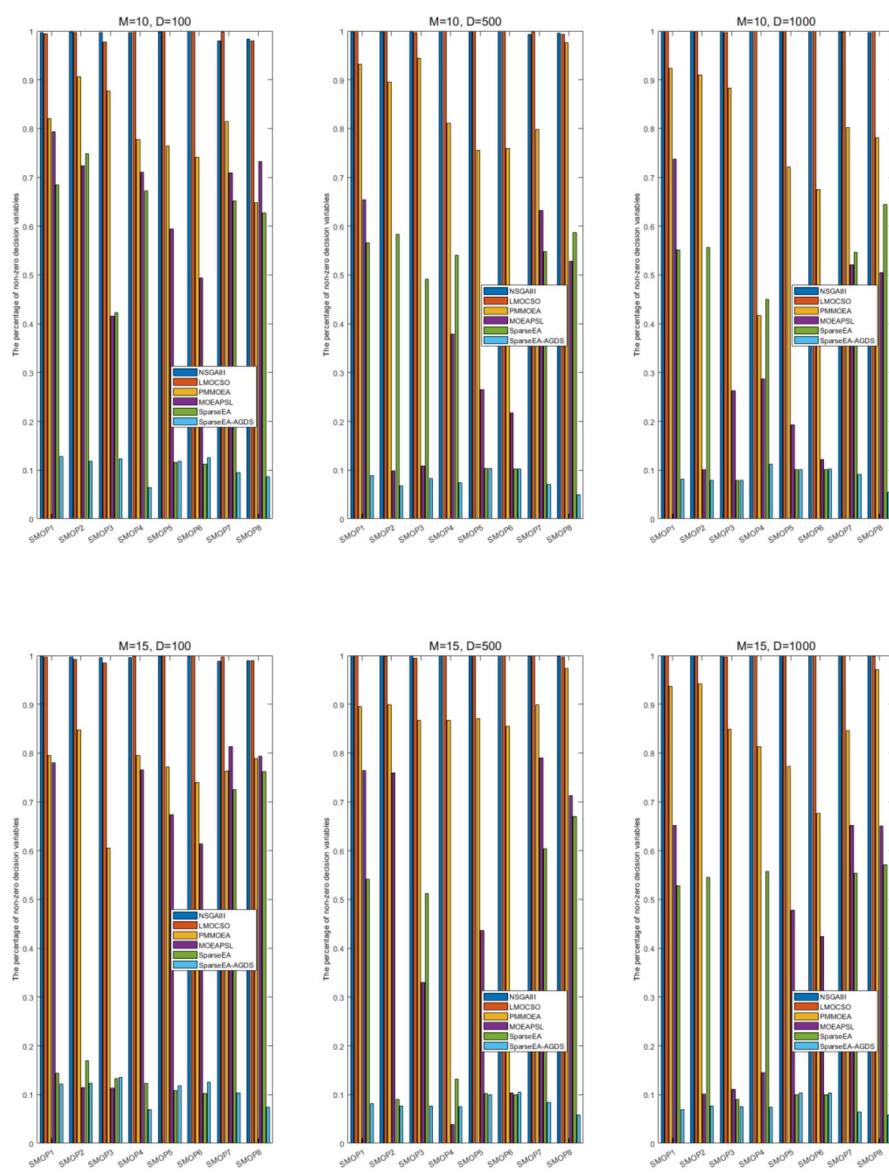
**Fig. 8.** Histogram of IGD values.

**Figure 8.** (continued)

	M=3			M=5			M=8			M=10			M=15			Total wins
	D=100	D=500	D=1000													
NSGAIII	0	0	0	0	0	0	2	3	3	3	2	3	5	3	3	26
LMOCSO	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	2
MOEA/PSL	0	0	1	0	0	0	0	3	2	0	2	2	0	3	2	15
PM-MOEA	0	0	1	2	1	0	0	2	0	0	0	0	0	0	0	6
SparseEA	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	4
NUCEA	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	3
SGECF	3	5	4	2	4	5	0	0	0	0	0	0	0	0	0	23
SparseEA-AGDS	4	3	3	2	2	5	2	5	3	2	2	2	2	2	2	41

Table 9. Statistics of wins for eight algorithms on the IGD values.

**Fig. 9.** Percentage of non-zero decision variables of eight algorithms.

**Figure 9.** (continued)

	M=3				M=5				M=8				M=10				M=15				Total wins
	D=100	D=500	D=1000	D=500	D=100	D=500	D=1000	D=500	D=100	D=500	D=1000	D=500	D=100	D=500	D=100	D=500	D=100	D=500	D=100	D=500	
NSGAIII	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LMOCSO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PM-MOEA	1	1	2	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
MOEA/PSL	1	0	0	0	2	0	1	0	0	0	0	0	0	0	0	2	1	0	0	0	7
SparseEA	4	3	3	0	0	0	2	1	0	2	1	0	2	2	3	2	1	2	2	25	
NUCEA	0	2	1	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
SGECF	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
SparseEA-AGDS	2	2	2	6	3	3	5	7	8	6	6	5	4	6	6	6	6	6	6	71	

Table 10. Statistics of wins for eight algorithms' on percentage of non-zero decision variables.

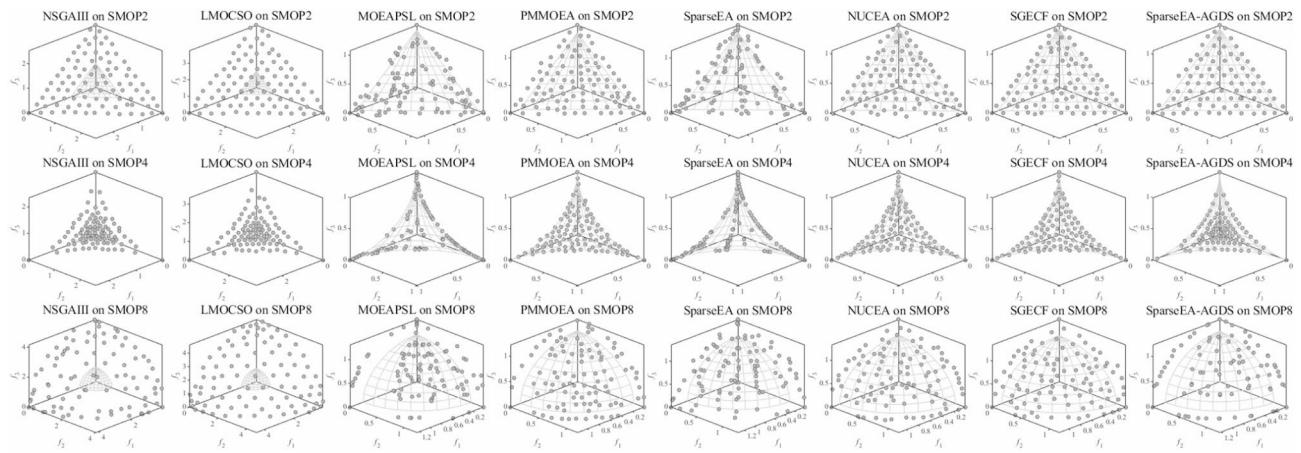
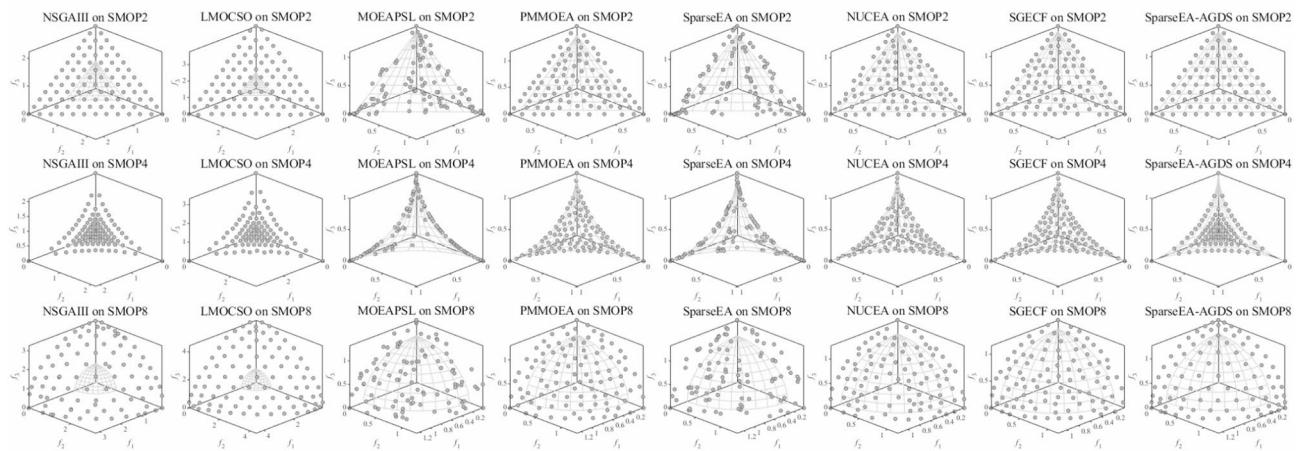
(a) $D = 100$ (b) $D = 1000$

Fig. 10. Pareto front of eight algorithms on the SMOP2, SMOP4, SMOP8 problems with $M=3$ and $D=100$ and 1000.

Problem	M	D	Runtime					
			MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
SMOP1	3	100	3.3688e+0 -	7.3755e+0 -	1.4384e + 0 +	6.2078e+0 -	6.1876e+0 -	1.8833e+0
		500	5.8559e+1 -	7.2847e+1 -	2.1572e + 1 +	5.9760e+1 -	5.1031e+1 -	2.3542e+1
		1000	2.5512e+2 -	2.3717e+2 -	7.8945e + 1 +	1.3720e+2 -	1.1143e+2 -	8.8713e+1
	5	100	5.9132e+0 -	1.1274e+1 -	1.3940e + 0 +	7.8570e+0 -	8.5474e+0 -	2.0881e+0
		500	1.5577e+2 -	1.3311e+2 -	2.5857e + 1 +	5.9356e+1 -	5.5854e+1 -	3.0000e+1
		1000	5.4211e+2 -	2.8699e+2 -	8.5415e + 1 +	1.5060e+2 -	1.4870e+2 -	9.2126e+1
	8	100	1.4117e+1 -	2.6944e+1 -	3.6477e + 0 +	3.9628e+0 -	3.9137e+0 +	5.2663e+0
		500	3.4943e+2 -	2.5976e+2 -	2.9217e+1 +	3.0634e+1 +	2.5674e + 1 +	3.5794e+1
		1000	7.1133e+2 -	1.4696e+3 -	9.0810e + 1 +	1.7326e+2 -	1.4246e+2 -	1.0346e+2
	10	100	6.5078e+0 -	1.3514e+1 -	1.3837e + 0 +	5.1450e+0 -	4.7567e+0 -	2.3697e+0
		500	1.7537e+2 -	3.9237e+2 -	2.3302e + 1 +	4.9401e+1 -	3.5276e+1 -	3.0789e+1
		1000	9.5179e+2 -	2.5237e+3 -	9.1032e+1 -	1.9244e+2 -	1.5137e+2 -	8.9836e+1
	15	100	9.0235e+0 -	1.6825e+1 -	1.8899e + 0 +	4.7633e+0 -	4.7276e+0 -	4.2947e+0
		500	3.4550e+2 -	3.4427e+2 -	2.6343e + 1 +	3.6031e+1 -	3.0729e+1 +	3.2224e+1
		1000	2.4801e+3 +	2.9877e+3 -	8.5631e + 1 +	1.9353e+2 +	1.4756e+2 +	3.0674e+4
SMOP2	3	100	3.8562e+0 -	7.8953e+0 -	1.3674e + 0 +	7.2883e+0 -	6.8829e+0 -	2.4121e+0
		500	6.4993e+1 -	8.8256e+1 -	2.0900e + 1 +	7.9975e+1 -	6.5089e+1 -	2.7043e+1
		1000	2.6074e+2 -	2.2668e+2 -	8.1699e + 1 +	1.5168e+2 -	1.1876e+2 -	9.2957e+1
	5	100	4.7650e+0 -	1.1026e+1 -	1.4282e + 0 +	1.1222e+1 -	1.0892e+1 -	2.2976e+0
		500	1.0711e+2 -	1.2100e+2 -	2.3142e + 1 +	5.2385e+1 -	5.0200e+1 -	3.0531e+1
		1000	7.0111e+2 -	4.1766e+2 -	8.4038e + 1 +	1.5464e+2 -	1.3128e+2 -	9.3823e+1
	8	100	1.0241e+1 -	1.6183e+1 -	2.0965e + 0 +	4.6700e+0 -	4.2257e+0 -	3.3876e+0
		500	1.8149e+2 -	2.5851e+2 -	2.1271e + 1 +	3.5508e+1 -	2.6347e+1 +	2.6910e+1
		1000	2.1923e+3 -	1.0517e+3 -	9.8749e + 1 +	1.8542e+2 -	1.5737e+2 -	9.9133e+1
	10	100	7.4978e+0 -	1.3980e+1 -	1.4931e + 0 +	6.3867e+0 -	5.8204e+0 -	2.7121e+0
		500	8.9115e+1 -	2.4433e+2 -	2.3041e + 1 +	5.1245e+1 -	4.1693e+1 -	3.0082e+1
		1000	3.7023e+2 -	2.0259e+3 -	8.9387e + 1 +	1.9691e+2 -	1.6780e+2 -	1.0148e+2
	15	100	4.1548e+0 -	1.4597e+1 -	1.4713e + 0 +	5.4649e+0 -	5.0339e+0 -	3.4648e+0
		500	4.0035e+2 -	2.9995e+2 -	2.2417e + 1 +	3.7174e+1 -	2.9892e+1 +	3.2587e+1
		1000	1.5126e+2 -	4.1830e+3 -	9.3157e+1 +	8.6250e+1 +	7.7770e + 1 +	1.1218e+2
SMOP3	3	100	3.9393e+0 -	8.1073e+0 -	1.3732e + 0 +	5.7951e+0 -	3.7191e+0 -	2.1820e+0
		500	4.9461e+1 -	1.9477e+2 -	1.9846e + 1 +	5.0527e+1 -	3.6050e+1 -	2.5576e+1
		1000	1.9807e+2 -	6.1770e+2 -	6.6971e+1 +	7.5226e+1 +	5.2253e + 1 +	8.0241e+1
	5	100	6.3109e+0 -	1.1721e+1 -	1.4915e + 0 +	7.8020e+0 -	4.5103e+0 -	2.4549e+0
		500	8.2915e+1 -	2.5761e+2 -	2.1621e + 1 +	5.3653e+1 -	4.3493e+1 -	2.6769e+1
		1000	6.4418e+2 -	7.2686e+2 -	7.7764e+1 +	8.0374e+1 +	6.0499e + 1 +	8.3092e+1
	8	100	1.2188e+1 -	2.1109e+1 -	2.8671e + 0 +	3.7648e+0 +	3.4998e+0 +	4.6594e+0
		500	2.6415e+2 -	3.9118e+2 -	2.3942e + 1 +	3.3465e+1 -	2.6533e+1 +	3.2205e+1
		1000	4.5869e+2 -	3.2984e+3 -	9.1892e+1 -	9.1342e+1 -	7.1317e + 1 +	9.0295e+1
	10	100	7.5955e+0 -	1.6845e+1 -	1.5677e + 0 +	5.8040e+0 -	6.1877e+0 -	2.7269e+0
		500	8.2677e+1 -	5.6245e+2 -	2.1804e + 1 +	5.9961e+1 -	5.4357e+1 -	2.8848e+1
		1000	3.7119e+2 -	2.3998e+3 -	6.6102e + 1 +	1.1268e+2 -	9.0839e+1 -	7.7493e+1
	15	100	6.5822e+0 -	1.7096e+1 -	1.8372e + 0 +	4.3898e+0 +	4.4925e+0 -	4.5004e+0
		500	3.3145e+2 -	5.7543e+2 -	2.4021e + 1 +	3.6510e+1 +	6.6673e+1 -	3.9586e+1
		1000	3.6566e+2 -	5.5026e+3 -	1.1081e+2 +	1.2458e+2 +	1.0119e + 2 +	1.3301e+2

Continued

Problem	M	D	Runtime					
			MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
SMOP4	3	100	5.1210e+0 -	1.2690e+1 -	1.7238e+0 +	1.0228e+1 -	1.1590e+1 -	2.4248e+0
		500	1.0779e+2 -	1.1609e+2 -	1.9261e+1 +	6.7786e+1 -	6.0774e+1 -	2.5705e+1
		1000	4.4804e+2 -	3.5900e+2 -	7.4917e+1 +	1.2604e+2 -	9.9255e+1 -	9.3382e+1
	5	100	7.6679e+0 -	1.4249e+1 -	1.6903e+0 +	7.4387e+0 -	7.5940e+0 -	2.7758e+0
		500	1.3428e+2 -	1.3255e+2 -	2.1784e+1 +	5.8874e+1 -	6.5064e+1 -	2.6010e+1
		1000	6.4155e+2 -	4.1866e+2 -	8.4563e+1 +	1.2650e+2 -	1.1811e+2 -	8.8734e+1
	8	100	9.2733e+0 -	1.8325e+1 -	1.9530e+0 +	8.5176e+0 -	8.3861e+0 -	3.6600e+0
		500	1.8372e+2 -	2.7939e+2 -	2.3603e+1 +	5.5209e+1 -	4.9129e+1 -	3.0102e+1
		1000	6.8879e+2 -	1.4102e+3 -	9.1758e+1 +	1.5075e+2 -	1.2113e+2 -	1.1928e+2
	10	100	1.1412e+1 -	1.8127e+1 -	1.8679e+0 +	8.6573e+0 -	7.0181e+0 -	3.3819e+0
		500	3.5834e+2 -	2.8286e+2 -	2.0828e+1 +	5.2613e+1 -	4.4523e+1 -	2.7095e+1
		1000	2.0600e+3 -	1.9597e+3 -	7.8432e+1 +	1.6984e+2 -	1.2154e+2 -	8.9517e+1
	15	100	1.3311e+1 -	3.1043e+1 -	3.1624e+0 +	1.0602e+1 -	8.7582e+0 -	6.7106e+0
		500	1.5208e+2 -	3.6278e+2 -	2.7792e+1 +	6.0633e+1 -	5.2865e+1 -	5.2119e+1
		1000	6.0922e+2 -	2.1845e+3 -	9.7410e+1 +	1.8604e+2 -	1.4016e+2 -	1.2971e+2
SMOP5	3	100	6.6653e+0 -	1.4253e+1 -	2.0700e+0 +	7.1668e+0 -	4.9100e+0 -	3.0681e+0
		500	1.0789e+2 -	1.1477e+2 -	2.0987e+1 +	6.1016e+1 -	4.8701e+1 -	2.7801e+1
		1000	4.5319e+2 -	3.1264e+2 -	8.2086e+1 +	1.8299e+2 -	1.3160e+2 -	1.0214e+2
	5	100	9.7164e+0 -	1.8099e+1 -	2.1778e+0 +	7.3145e+0 -	8.1021e+0 -	3.5651e+0
		500	1.6088e+2 -	1.3745e+2 -	2.3010e+1 +	6.3561e+1 -	4.9730e+1 -	2.8456e+1
		1000	7.0137e+2 -	3.8752e+2 -	8.7938e+1 +	1.9437e+2 -	1.4833e+2 -	9.3604e+1
	8	100	1.3132e+1 -	2.4668e+1 -	2.6841e+0 +	9.4478e+0 -	9.6136e+0 -	4.3797e+0
		500	1.3672e+2 -	3.2822e+2 -	2.5781e+1 +	5.7244e+1 -	5.6513e+1 -	3.4937e+1
		1000	6.4071e+2 -	1.5828e+3 -	9.6599e+1 +	2.1572e+2 -	1.5557e+2 -	1.0711e+2
	10	100	1.1785e+1 -	1.8781e+1 -	1.9719e+0 +	9.5659e+0 -	8.5923e+0 -	3.4011e+0
		500	2.9596e+2 -	2.6360e+2 -	2.2853e+1 +	6.4751e+1 -	5.3529e+1 -	2.8005e+1
		1000	1.5611e+3 -	1.7375e+3 -	9.5141e+1 +	2.4478e+2 -	1.5304e+2 -	9.8048e+1
	15	100	1.4005e+1 -	2.3279e+1 -	2.6629e+0 +	1.0426e+1 -	8.3770e+0 -	5.7715e+0
		500	5.5020e+2 -	3.6705e+2 -	3.0299e+1 +	7.0077e+1 -	5.7078e+1 -	5.1033e+1
		1000	2.9533e+3 -	3.1159e+3 -	1.1018e+2 +	2.6113e+2 -	1.6797e+2 -	1.3196e+2
SMOP6	3	100	6.2438e+0 -	1.2898e+1 -	1.8830e+0 +	3.4311e+0 -	3.5677e+0 -	2.7566e+0
		500	1.1406e+2 -	1.1527e+2 -	2.4342e+1 +	2.7190e+1 +	2.2407e+1 +	3.0521e+1
		1000	4.6870e+2 -	3.0156e+2 -	7.9764e+1 +	2.5217e+2 -	1.7591e+2 -	1.0795e+2
	5	100	8.8445e+0 -	1.6918e+1 -	2.0630e+0 +	3.9022e+0 -	3.8358e+0 -	3.2908e+0
		500	1.5758e+2 -	1.5191e+2 -	2.5878e+1 +	3.4613e+1 -	3.3233e+1 -	2.9623e+1
		1000	6.7959e+2 -	5.0791e+2 -	8.1878e+1 +	2.5270e+2 -	2.0112e+2 -	1.1476e+2
	8	100	1.1268e+1 -	2.0542e+1 -	2.3697e+0 +	1.0230e+1 -	8.3226e+0 -	4.1875e+0
		500	2.2740e+2 -	2.9376e+2 -	3.5230e+1 +	6.2417e+1 -	5.7967e+1 -	4.3688e+1
		1000	6.1228e+2 -	1.5160e+3 -	1.0717e+2 +	2.9425e+2 -	2.0499e+2 -	1.2960e+2
	10	100	8.7833e+0 -	1.9813e+1 -	1.9683e+0 +	1.0242e+1 -	1.0291e+1 -	3.5219e+0
		500	2.5204e+2 -	2.4597e+2 -	2.5768e+1 +	1.0920e+2 -	7.6521e+1 -	3.3624e+1
		1000	1.2292e+3 -	1.3908e+3 -	8.3476e+1 +	3.0999e+2 -	2.0113e+2 -	1.1747e+2
	15	100	1.2655e+1 -	2.3896e+1 -	2.6814e+0 +	8.8595e+0 -	8.7336e+0 -	5.8598e+0
		500	1.5140e+2 -	3.5864e+2 -	3.6064e+1 +	6.9387e+1 -	5.9700e+1 -	5.7703e+1
		1000	2.4608e+3 -	1.4726e+3 -	1.0112e+2 +	3.3437e+2 -	2.5530e+2 -	1.3466e+2

Continued

Problem	M	D	Runtime					
			MOEAPSL	PMMOEA	SparseEA	NUCEA	SGECF	SparseEA-AGDS
SMOP7	3	100	5.6398e+0 -	9.2746e+0 -	2.0644e + 0 +	3.1104e+0 -	3.9736e+0 -	2.5693e+0
		500	6.3759e+1 -	8.9387e+1 -	2.1206e + 1 +	3.8100e+1 -	3.0186e+1 -	2.6625e+1
		1000	2.3875e+2 -	1.8494e+2 -	9.1805e + 1 +	1.4949e+2 -	1.2164e+2 -	1.0387e+2
	5	100	5.7211e+0 -	1.2096e+1 -	1.6785e + 0 +	5.3827e+0 -	5.4094e+0 -	2.5203e+0
		500	1.1303e+2 -	1.1182e+2 -	2.2732e + 1 +	5.2225e+1 -	5.1969e+1 -	2.6030e+1
		1000	4.6507e+2 -	3.1281e+2 -	8.3101e + 1 +	1.7765e+2 -	1.6249e+2 -	9.3129e+1
	8	100	1.0999e+1 -	1.6951e+1 -	2.0265e + 0 +	8.2133e+0 -	7.9091e+0 -	3.4175e+0
		500	4.7559e+2 -	2.3732e+2 -	2.4802e + 1 +	5.7303e+1 -	6.3965e+1 -	2.9074e+1
		1000	1.1400e+3 -	9.7345e+2 -	7.8485e+1 -	7.3420e+1 +	6.6444e + 1 +	7.5793e+1
	10	100	1.1626e+1 -	1.6177e+1 -	1.9134e + 0 +	1.6467e+1 -	1.5819e+1 -	2.9245e+0
		500	3.7570e+2 -	4.4064e+2 -	2.1942e + 1 +	1.3802e+2 -	1.0966e+2 -	2.6669e+1
		1000	2.5648e+3 -	1.1887e+3 -	9.6580e+1 +	8.4558e + 1 +	8.4907e+1 +	1.0080e+2
	15	100	1.3814e+1 -	2.5294e+1 -	2.2329e + 0 +	1.1903e+1 -	1.2609e+1 -	6.7621e+0
		500	4.7177e+2 -	7.4669e+2 -	2.2014e + 1 +	7.2028e+1 -	7.2165e+1 -	2.9662e+1
		1000	3.1560e+3 -	4.0188e+3 -	8.8890e+1 +	9.8452e+1 +	8.2838e + 1 +	1.0113e+2
SMOP8	3	100	5.7489e+0 -	8.9388e+0 -	1.7803e + 0 +	4.7592e+0 -	3.2750e+0 -	2.6857e+0
		500	5.5691e+1 -	8.5369e+1 -	2.1177e + 1 +	4.9331e+1 -	3.5263e+1 -	2.7847e+1
		1000	2.1530e+2 -	1.7903e+2 -	6.7233e+1 +	6.8837e+1 +	5.3329e + 1 +	1.0982e+2
	5	100	5.8122e+0 -	1.2578e+1 -	1.6580e + 0 +	8.3261e+0 -	6.5357e+0 -	2.5429e+0
		500	7.9361e+1 -	1.0925e+2 -	2.1933e + 1 +	6.6628e+1 -	5.7389e+1 -	2.4555e+1
		1000	3.4253e+2 -	2.9051e+2 -	8.5634e+1 +	1.0015e+2 -	8.4706e + 1 +	8.7556e+1
	8	100	8.4225e+0 -	1.4592e+1 -	1.9191e + 0 +	7.2070e+0 -	8.9712e+0 -	2.8573e+0
		500	2.0300e+2 -	2.1055e+2 -	2.0010e + 1 +	5.5504e+1 -	5.8818e+1 -	2.4375e+1
		1000	1.0168e+3 -	7.5025e+2 -	7.7915e+1 -	1.0120e+2 -	9.6007e+1 -	7.6754e+1
	10	100	1.0999e+1 -	1.6177e+1 -	1.7802e + 0 +	1.7213e+1 -	1.4135e+1 -	3.0498e+0
		500	3.9049e+2 -	6.7057e+2 -	2.2626e + 1 +	1.1823e+2 -	1.0194e+2 -	2.6470e+1
		1000	2.3218e+3 -	3.0074e+3 -	8.9911e + 1 +	1.2355e+2 -	1.2213e+2 -	9.1478e+1
	15	100	1.2822e+1 -	2.1319e+1 -	2.0964e + 0 +	1.1063e+1 -	1.1337e+1 -	4.2873e+0
		500	4.0274e+2 -	9.2903e+2 -	2.2720e + 1 +	7.6368e+1 -	7.0892e+1 -	3.2900e+1
		1000	3.0744e+3 -	3.8633e+3 -	9.0843e + 1 +	1.4062e+2 -	1.3128e+2 -	1.0789e+2
+/-=			1/119/0	0/120/0	116/4/0	14/106/0	19/101/0	

Table 11. Statistics of wins for eight algorithms on the runtime.

Data availability

The data presented in this study are available on request from corresponding author.

Received: 20 September 2024; Accepted: 19 February 2025

Published online: 18 March 2025

References

- Pasha, L. et al. Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings. *Adv. Eng. Inform.* **52**, 101623 (2022).
- Omidvar, M. N., Li, X., Mei, Y. & Yao, X. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Trans. Evol. Comput.* **18**(3), 378–393 (2014).
- Yao, X. A review of evolutionary artificial neural networks. *Int. J. Intell. Syst.* **8**(4), 419–440 (1993).
- Anil, K., Jain, M., Murty, N. & Flynn, P. J. Data clustering: A review. *ACM Comput. Surveys.* **31**(3), 264–323 (1999).
- Cai, Q., Gong, M., Ma, L. & Jiao, J. A survey on network community detection based on evolutionary computation. *Int. J. Bio-Inspired Comput.* **6**(5), 279–291 (2014).
- Marinakis, Y. & Marinaki, M. A particle swarm optimization algorithm with path relinking for the location routing problem. *J. Math. Modelling Algorithms.* **7**(1), 59–78 (2008).
- Tian, Y., Zhang, X., Wang, C. & Jin, Y. An evolutionary algorithm for large-scale sparse multi-objective optimization problems. *IEEE Trans. Evol. Comput.* **24**(2), 380–393 (2020).
- Jin, Y. & Sendhoff, B. Pareto-based multi-objective machine learning: an overview and case studies. *IEEE Trans. Syst. Man. Cybernetics Part. C: Appl. Reviews.* **38**(3), 401–410 (2008).
- Lwin, K. & Qu, R. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Appl. Soft Comput.* **14**(12), 1155–1169 (2014).
- Tian, Y., Zhang, X. & Jin, Y. An improved large-scale sparse multi-objective evolutionary algorithm using unsupervised neural network. *IEEE Trans. Cybernetics.* **50**(1), 1–14 (2020).
- Tian, Y. et al. Evolutionary large-scale multi-objective optimization: A survey. *ACM Comput. Surveys.* **54**(8), 1–34 (2021).
- Shao, S. et al. A permutation group-based evolutionary algorithm for car sequencing problems in assembly lines, *International Conference on Data-driven Optimization of Complex Systems.* 1–8 (2023).
- Yin, F. & Cao, B. A two-space-decomposition-based evolutionary algorithm for large-scale multi-objective optimization. *Swarm Evol. Comput.* **101397**, 101397 (2023).
- Fan, W. et al. Two-stage distributionally robust optimization model of integrated energy system group considering energy sharing and carbon transfer. *Appl. Energy.* **331**, 120426 (2023).
- Si, L. et al. Linear subspace surrogate modeling for large-scale expensive single/multi-objective optimization. *IEEE Transactions on Evolutionary Computation*(2023).
- Huang, C. et al. Co-evolutionary competitive swarm optimizer with three-phase for large-scale complex optimization problem. *Inf. Sci.* **619**, 2–18 (2023).
- Tian, Y. et al. Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization. *IEEE/CAA J. Automatica Sinica.* **9**(10), 1801–1817 (2022).
- Liu, S., Lin, Q., Tian, Y. & Tan, K. C. A variable importance-based differential evolution for large-scale multi-objective optimization. *IEEE Trans. Cybernetics.* **52**(12), 13048–13062 (2021).
- Feng, Y., Feng, L., Kwong, S. & Tan, K. C. A multi-variety multi-factorial evolutionary algorithm for large-scale multi-objective optimization. *IEEE Trans. Evol. Comput.* **26**(2), 248–262 (2021).
- Lin, Q. et al. An adaptive two-stage evolutionary algorithm for large-scale continuous multi-objective optimization. *Swarm Evol. Comput.* **77**, 101235 (2023).
- Ding, Z., Chen, L., Sun, D. & Zhang, X. A multi-stage knowledge-guided evolutionary algorithm for large-scale sparse multi-objective optimization problems. *Swarm Evol. Comput.* **73**, 101119 (2022).
- Smith, J. & Brown, A. An evolutionary algorithm based on rank-1 approximation for sparse multi-objective optimization. *Springer* **56**(6), 1581–1586 (2023).
- Wang, H., Fang, D. G. & Chow, Y. L. An evolutionary algorithm for large-scale sparse multi-objective optimization problems. *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* 1561–1568 (2019).
- Tan, Z., Wang, H. & Liu, S. Multi-stage dimension reduction for expensive sparse multi-objective optimization problems. *Neurocomputing* **440**, 159–174 (2021).
- Ye, Q. et al. A cluster-based competitive particle swarm optimizer with a sparse Truncation operator for multi-objective optimization. *Swarm Evol. Comput.* **71**, 101083 (2022).
- Shao, S., Tian, Y. & Zhang, X. A non-uniform clustering based evolutionary algorithm for solving large-scale sparse multi-objective optimization problems, *Bio-Inspired Computing: Theories and Applications, Communications in Computer and Information Science.* 103–116 (2024).
- Tian, Y., Lu, C., Zhang, X. & Cheng, F. Solving Large-Scale Multi-Objective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE Trans. Cybernetics.* **51**(6), 3115–3128 (2021).
- Tian, Y., Lu, C., Zhang, X., Cheng, F. & Jin, Y. A pattern mining-based evolutionary algorithm for large-scale sparse multi-objective optimization problems. *IEEE Trans. Cybernetics.* **52**(7), 6784–6797 (2022).
- Wang, X., Zhang, K., Wang, J. & Jin, Y. An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multi-objective optimization. *IEEE Trans. Evol. Comput.* **26**, 859–871 (2022).
- Cheng, R. & Jin, Y. A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybernetics.* **45**(2), 191–204 (2014).
- Wu, C., Tian, Y., Zhang, Y., Jiang, H. & Zhang, X. A sparsity-guided elitism co-evolutionary framework for sparse large-scale multi-objective optimization. *Proceedings of the IEEE Congress on Evolutionary Computation* 1–8 (2023).
- Li, W., Zhang, T., Wang, R., Huang, S. & Liang, J. Multi-modal Multi-objective optimization: comparative study of the State-of-the-Art. *Swarm Evol. Comput.* **82**, 101253 (2023).
- Goldberg, D. E. & Deb, K. A comparative analysis of selection schemes used in genetic algorithms. *Elsevier* **1**, 69–93 (1991).
- Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002).
- Deb, K. & Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**, 577–601 (2014).
- Tian, Y., Zhang, X., Li, H. & Zhang, X. Efficient large-scale multi-objective optimization based on a competitive swarm optimizer. *IEEE Trans. Cybernetics* **50**(8), 3696–3708 (2019).
- Tian, Y., Cheng, R., Zhang, X. & Jin, Y. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Comput. Intell. Mag.* **12**(4), 73–87 (2017).
- Zhang, X., Zheng, X., Cheng, R., Qiu, J. & Jin, Y. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf. Sci.* **415**, 28–44 (2018).

39. Khan, W. & Zhang, Q. MOEA/D-DRA with two crossover operators. *UK Workshop on Computational Intelligence (UKCI)*. 1–6 (2010).
40. Ishibuchi, H., Masuyama, N. & Nojima, Y. Modified distance calculation in generational distance and inverted generational distance. *Lect Notes Comput. Sci.* **319**, 15892–15908 (2015).
41. Wilson, P. Extension of the Wilcoxon rank sum test for complex sample survey data. *J. R. Stat. Soc.* **61**(4), 653–707 (2012).
42. Li, F., Shang, Z., Shen, H., Liu, Y. & Huang, P. Q. Combining modified inverted generational distance indicator with reference-vector-guided selection for many-objective optimization. *Appl. Intell.* **53**, 12149–12162 (2023).

Author contributions

W. Z. wrote a manuscript; X. W. obtained funding sources; Z. D. is responsible for data analysis; J. T. and Y. F. are responsible for proofreading the manuscript. All authors reviewed the manuscript.

Funding

This work was supported by the Basic research project of Science and Technology Department of Yunnan Province (Grant Number 202201AT070021), Open Subject of Yunnan Key Laboratory of Unmanned Autonomous Systems (Grant Number 202408YB07), and the National Natural Science Foundation of China (Grant Number 62161052 and 61963038).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to X.W. or J.-N.T.

Reprints and permissions information is available at www.nature.com/reprints.

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025