Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 9961727, 11 pages
https://doi.org/10.1155/2021/9961727

Research Article

Feature Selection Based on a Large-Scale Many-Objective Evolutionary Algorithm

Yue Li,[1] Zhiheng Sun,[1] Xin Liu,[2] Wei-Tung Chen,[3] Der-Juinn Horng,[3] and Kuei-Kuei Lai[4]

[1] *State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin, China*
[2] *School of Economics and Management, Hebei University of Technology, Tianjin, China*
[3] *Department of Business Administration, NCU, Taoyuan, China*
[4] *Department of Business Administration of Chaoyang University of Technology, Taichung, China*

Correspondence should be addressed to Xin Liu; xinliu10@163.com and Kuei-Kuei Lai; laikk.tw@gmail.com

Received 3 April 2021; Revised 2 June 2021; Accepted 26 June 2021; Published 25 August 2021

Academic Editor: Nian Zhang

The feature selection problem is a fundamental issue in many research fields. In this paper, the feature selection problem is regarded as an optimization problem and addressed by utilizing a large-scale many-objective evolutionary algorithm. Considering the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance, a large-scale many-objective feature selection model is constructed. It is difficult to optimize the large-scale many-objective feature selection optimization problem by using the traditional evolutionary algorithms. Therefore, this paper proposes a modified vector angle-based large-scale many-objective evolutionary algorithm (MALSMEA). The proposed algorithm uses polynomial mutation based on variable grouping instead of naive polynomial mutation to improve the efficiency of solving large-scale problems. And a novel worst-case solution replacement strategy using shift-based density estimation is used to replace the poor solution of two individuals with similar search directions to enhance convergence. The experimental results show that MALSMEA is competitive and can effectively optimize the proposed model.

1. Introduction

Feature selection involves the selection of a specific number of features from existing features to optimize specific objectives [1]. Feature selection can be regarded as a multiobjective optimization problem that can be solved using evolutionary algorithms. Feature selection has attracted the attention of scholars and has been widely used in gene expression analysis [2], face recognition [3], and drug discovery [4]. For example, a two-stage heuristic algorithm minimal redundancy maximal relevance (mRMR) [5] is used to optimize relevance and redundancy simultaneously. A filter-based algorithm [6] is used to consider the entropy-based correlation measure and the combination measure of the redundancy and cardinality of a selected subset. A decomposition algorithm based on a weighted method is utilized to optimize interclass and intraclass distances [7]. Gulsah et al. [8] proposed two algorithms, W-QEISS and F-QEISS, that use nondominated sorting based on classification accuracy, feature number, relevance, and redundancy. Li et al. [9] established a model with feature number, classification performance, interclass distance, and intraclass distance as objectives and proposed a decomposition-based large-scale algorithm (DMEA-FS).

However, some unsolved problems still exist in feature selection using traditional evolutionary algorithms. The first problem is that the selection of a large number of features can be regarded as the optimization of the large-scale optimization problem [1] or the large-scale multiobjective optimization problem (LSMOP) [10], but the traditional evolutionary algorithms cannot effectively solve such problems. The second problem is that feature number and accuracy are two basic objectives, and other objectives are needed to explore the potential information to guide the evolution in feature selection [1]. Correspondingly, more objectives result in many-objective optimization problems (MaOPs) [11, 12].

There are three main types of current algorithms, which are mainly used to solve LSMOPs or MaOPs, but they perform poorly on large-scale many-objective problems (LSMaOPs) [13], which include more than 3 objectives and over 100 decision variables [14, 15].

The first kind of algorithms is based on the Pareto dominance, which improves the convergence pressure by modifying the Pareto dominance relation. The new dominance relations are ε-dominance [16], θ-dominance [17], L-optimality [18], simplex dominance [19], grid dominance [20, 21], etc. The algorithm using shift-based density estimation (SDE) was proposed in the work of [22], which allows individuals with poor convergence to obtain higher density.

The second is based on performance indicators, such as the hypervolume (HV) adaptive grid algorithm (HAGA) [23], the evolutionary algorithm (MaOEA/IGD) using inverted generational distance (IGD) [24], indicator-based algorithm with boundary protection (MaOEA-IBP) [25], and R2 indicator and weight vector-based method (R2-WVEA) [26]. Most of these algorithms are many-objective evolutionary algorithms (MaOEAs), but their computational costs are large.

The third category is composed of decomposition-based methods. The most classic ones are the multiobjective evolutionary algorithm based on decomposition (MOEA/D) [27] and its variants [28–30]. The algorithm based on nondominated sorting approach (NSGA-III) [31] uses evenly distributed reference points to assist the environmental selection. Based on NSGA-III, Gu and Wang [10] introduced an information feedback model to solve LSMaOPs. The reference vector-guided evolutionary algorithm (RVEA) [32] uses reference vectors to guide the optimization.

To more comprehensively describe and better solve the large-scale feature selection problem, this paper studies the existing multiobjective models based on the evolutionary algorithm, combines the existing objectives, constructs the feature selection problem as an LSMaOP, and uses an improved large-scale many-objective evolutionary algorithm (LSMaOEA) for optimization.

The main contributions of this paper are summarized as follows:

(1) A novel worst-case solution replacement strategy based on SDE is proposed. This strategy allows conditional replacement of poor solutions in terms of convergence and diversity compared to other solutions, thereby maintaining a balance between convergence and diversity.

(2) A modified vector angle-based large-scale many-objective evolutionary algorithm (MALSMEA) is proposed, which uses variable grouping-based polynomial mutation instead of naive polynomial mutation to improve the efficiency of solving large-scale problems. In the environmental selection process, the proposed worst solution replacement strategy is used to improve diversity.

(3) A large-scale many-objective feature selection optimization model is constructed, and MALSMEA is used to optimize it. The optimization objectives of this model are the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance.

The remainder of this paper is arranged as follows. Section 2 introduces the related works. Section 3 describes the proposed model and MALSMEA in detail. In Section 4, we compare and analyze the experimental results of MALSMEA and four advanced algorithms in solving benchmark LS-MaOPs, as well as the performance of MALSMEA and three feature selection algorithms in optimizing the proposed feature selection model. Section 5 provides a summary of the full paper and prospects of future research.

2. Related Works

2.1. Large-Scale Many-Objective Optimization Problem. An LSMaOP can be described as

$$\begin{aligned} \min \quad & F(x) = \left(f_1(x), f_2(x), \ldots, f_m(x)\right) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{1}$$

where $\Omega = \prod_{i=1}^{D} [l_i, u_i] \subseteq R^D$ is the decision space, D is the number of decision variables ($D \geq 100$), and l_i and u_i are the lower and upper bounds of decision variables in the ith dimension, respectively. x is the D-dimensional decision vector in Ω, m is the objective number ($m > 3$), and $F(x) \in R^m$ is the objective vector of x. If no other solution dominates x, then x is a Pareto optimal solution [33]. The objective vectors corresponding to all Pareto optimal solutions constitute the Pareto optimal front (PF) [34, 35].

2.2. Shift-Based Density Estimation. We use the SDE [22] with the kth nearest neighbor [36] to estimate the density of all individuals. For an individual x_i, the following method is used to calculate the density value $\mathrm{SDE}(x_i)$.

(i) First, the standardized objective vectors of other individuals in population P are shifted.

(ii) Then, the Euclidean distances between other shifted normalized objective vectors and the considered individual are calculated, expressed as $d(x_i, x_k)$.

(iii) Next, the kth minimum value $\lambda(x_i)$ in the set $\{d(x_i, x_k), x_k \in P \cap x_k \neq x_i\}$ is found, where $k = \sqrt{N}$ and N is the size of the population.

(iv) Finally, $\mathrm{SDE}(x_i)$ is calculated as follows:

$$\mathrm{SDE}(x_i) = \frac{1}{\lambda(x_i) + 2}. \tag{2}$$

Through the above process of estimating the individual density, we can observe that the smaller the individual density is, the better the performance of the individual.

Therefore, this paper uses this strategy, considering both diversity and convergence, to judge a pair of individuals with similar search direction, so as to delete the individual with poor performance.

2.3. Information Theory Criterion Based on Entropy. The feature selection model uses an entropy-based information theory criterion [8] to measure correlation and redundancy. For a given discrete random variable A, its entropy $E(A)$ is determined as follows:

$$E(A) = -\sum_{a\in A} p(a)\log\ p(a), \tag{3}$$

where $p(a) = \Pr(A = a)$, A is the set of all possible values of A, $a \in A$. Then, the joint entropy of A and B is determined as follows:

$$E(A,B) = -\sum_{a\in A}\sum_{b\in B} p(a,b)\log\ p(a,b), \tag{4}$$

where B is a discrete random variable, $p(a,b) = \Pr(A = a, B = b)$, $a \in A$, and $b \in B$. Then, the mutual information between A and B is determined as follows:

$$M(A,B) = E(A) + E(B) - E(A,B). \tag{5}$$

Symmetric uncertainty is used to scale the value range of mutual information to $[0,1]$ [37], which is defined as follows:

$$\mathrm{SU}(A,B) = \frac{2M(A,B)}{E(A)+E(B)}. \tag{6}$$

3. Proposed Model and Algorithm

3.1. Model Design. The optimization objectives of the feature selection model include the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance, which are described as follows:

(1) *The Number of Selected Features.* It is minimized to ensure the simplification of feature selection:

$$F_1(S) = |S|, \tag{7}$$

where $|S|$ represents the cardinality of feature set S.

(2) *Accuracy.* The accuracy of the learning algorithm is measured by the classification performance. The higher the classification performance is, the greater the accuracy. In this paper, the extreme learning machine (ELM) classifier [8] is used to calculate the accuracy:

$$F_2(S) = \frac{\mathrm{tn}+\mathrm{tp}}{\mathrm{fn}+\mathrm{fp}+\mathrm{tn}+\mathrm{tp}}, \tag{8}$$

where tn, tp, fn, and fp represent the true negative, true positive, false negative, and false positive, respectively.

(3) *Relevance.* The relevance between features and categorical variables reflects the recognition ability of the selected features. The greater the correlation is, the stronger the recognition ability is:

$$F_3(S) = \sum_{x_i\in S} \mathrm{SU}(x_i, y), \tag{9}$$

where x_i represents the ith feature and y represents the target categorical variable. This objective is normalized according to $F_3(S) = F_3(S)/\max F_3(S)$.

(4) *Redundancy.* The redundancy is used to quantify the level of similarity between selected features. The smaller the redundancy is, the smaller the similarity:

$$F_4(S) = \sum_{x_i,x_j\in S, i<j} \mathrm{SU}\left(x_i, x_j\right), \tag{10}$$

where x_j represents the jth feature. This objective is normalized according to $F_4(S) = F_4(S)/\max F_4(S)$.

(5) *Interclass Distance.* The interclass distance represents the distance between the mean sample of each class and the average of mean samples of all classes, which reflects the recognition ability of samples of different classes. In the evolutionary process, a better sample distribution is obtained by maximizing the distance between classes:

$$F_5(S) = \sum_{i=1}^{L}\left(m_i - \frac{1}{L}\sum_{i=1}^{L} m_i\right)^2, \tag{11}$$

where L is the total number of classes and m_i is the average value of all samples with feature S in class i. This objective is normalized according to $F_5(S) = F_5(S)/\max F_5(S)$.

(6) *Intraclass Distance.* By calculating the distances between the samples with the selected feature and the mean of all samples of the same kind, this value reflects the cohesion of the same kind of samples and can improve the accuracy to a certain extent:

$$F_6(S) = \sum_{i=1}^{L}\sum_{a_{ij}\in L_i}\left(a_{ij} - m_i\right)^2, \tag{12}$$

where a_{ij} is the jth sample in class i. This objective is normalized according to $F_6(S) = F_6(S)/\max F_6(S)$.

Therefore, the definition of the feature selection optimization model in this paper is as follows:

$$\min\left(F_1(S), -F_2(S), -F_3(S), F_4(S), -F_5(S), F_6(S)\right). \tag{13}$$

3.2. The Proposed Algorithm: MALSMEA. In this paper, a modified vector angle-based large-scale many-objective evolutionary algorithm is proposed, termed as MALSMEA. MALSMEA mainly uses a mutation operator based on variable grouping and the environment selection method of VaEA [38]. Figure 1 shows the program flowchart of MALSMEA. The main process of MALSMEA is as follows:

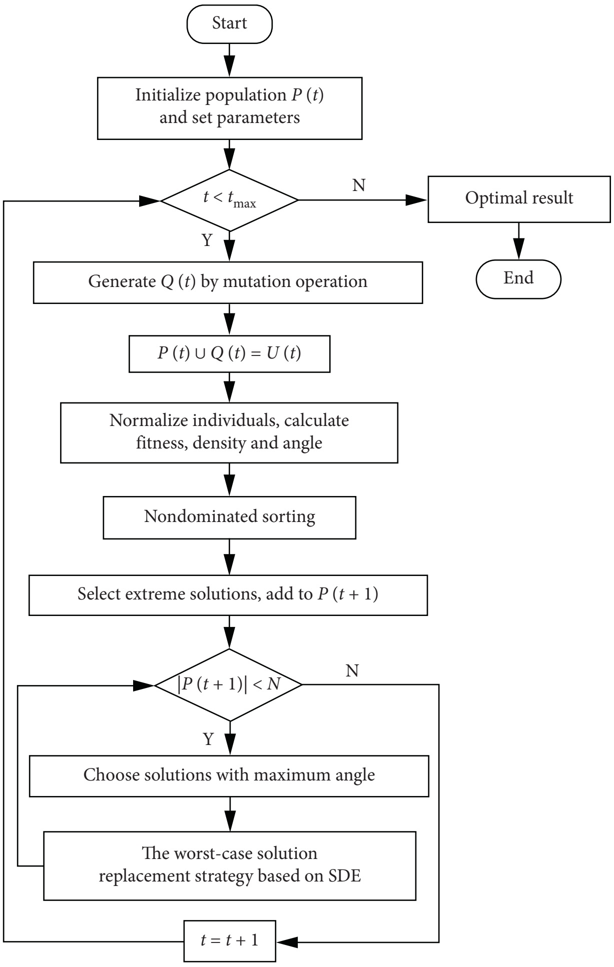
Start
Initialize population $P(t)$ and set parameters
$t < t_{\max}$
N
Y
Optimal result
End
Generate $Q(t)$ by mutation operation
$P(t) \cup Q(t) = U(t)$
Normalize individuals, calculate fitness, density and angle
Nondominated sorting
Select extreme solutions, add to $P(t+1)$
$|P(t+1)| < N$
N
Y
Choose solutions with maximum angle
The worst-case solution replacement strategy based on SDE
$t = t + 1$

Figure 1: Program flowchart of MALSMEA.

(i) *Step 1*. Initialize a population $P(t)$ with N individuals randomly in the whole decision space Ω, and set parameters.

(ii) *Step 2*. The mutation operator based on variable grouping is used to mutate the population $P(t)$, in which the grouping method is ordered grouping, to generate the offspring population $Q(t)$.

(iii) *Step 3*. Combine the offspring population $Q(t)$ with the parent population $P(t)$ and obtain the joint population $U(t)$. Then, the environmental selection in steps 4–9 is adopted to select N promising individuals from $U(t)$.

(iv) *Step 4*. Normalize the individuals in the population $U(t)$, and calculate the fitness and density values of each individual as well as the vector angle between every two individuals.

(v) *Step 5*. Use the nondominated sorting method to rank, and determine the last layer $F(l)$.

(vi) *Step 6*. According to the vector angle between any two individuals in layer $F(l)$ and the fitness value of each individual, m individuals with the largest vector angle and m individuals with the smallest fitness value are selected to join $P(t+1)$ to ensure the diversity.

(vii) *Step 7*. If $|P(t+1)| < N$, select the individual with the largest vector angle in $F(l)$ to join the new population $P(t+1)$ by calculating the vector angles between the individuals in $F(l)$ and the individuals in $P(t+1)$; otherwise, go to step 9.

(viii) *Step 8*. To maintain the balance between convergence and diversity, the worst individual replacement strategy is used to replace the poor individual with other individuals. Repeat from step 7 if $|P(t+1)| < N$.

(ix) *Step 9*. Obtain the new population $P(t+1)$.

(x) *Step 10*. Repeat from step 2, and stop when the maximum number of generations $t_{\max}$ is reached.

3.3. The Worst-Case Solution Replacement Strategy Based on SDE. As the extreme individuals have been selected according to the vector angle and fitness value, for the worst individual replacement strategy in the process of environmental selection, we use the SDE strategy to calculate the density of individuals. The SDE strategy can consider the convergence and diversity of individuals simultaneously. Using this method, we can replace the poor individuals with similar search directions. The specific process is as follows: if the angle between an individual a in $F(l)$ and an individual b in $P(t+1)$ is less than the angle between two solutions of N ideal solutions, that is, $\theta = ((\pi/2)/N + 1)$, where N is the population size, then they have similar search directions. In this case, if $\mathrm{SDE}(a) < \mathrm{SDE}(b)$, then individual b is replaced by a. After replacement, the angle between each individual $a \in F(l)$ and the new population $P(t+1)$ is updated.

3.4. The Wrapper Structure of MALSMEA. MALSMEA is applied to the feature selection model, and the pseudocode of the wrapper structure of MALSMEA is shown in Algorithm 1. The main steps are as follows:

(i) First, the input dataset DS is divided into training and test datasets.

(ii) Then, in the initialization process, MALSMEA allocates the random feature vector W_S selected from the data feature matrix W. The selected feature vector W_S is encoded as solutions by using the coding technology of [9] to reduce the amount of computation in the evolutionary process, and the mask of W_S is regarded as the decision variables, and the population P is formed.

(iii) Then, in the wrapper structure, the population P is evaluated via six objective functions to obtain objective vectors and obtain the evaluated population $P(t)$. The feature number is calculated according to the decision variables of the solutions. The accuracy can be obtained from the decoded feature subset and the corresponding ELM classifier [8], and other objectives can be calculated according to the corresponding equations.

(iv) Then, the population is optimized by MALSMEA.

(v) Finally, the optimal set P_S is obtained.

3.5. Time Complexity Analysis. The time complexity of MALSMEA is composed mainly of the following parts: the time complexity of the mutation operation in MALSMEA is $O(D^2N/K)$, where K is the number of groups, the time complexity of nondominated sorting is $O(N \log^{m-2} N)$ [31], the worst-case solution replacement strategy based on SDE has the time complexity of $O(mN^2)$, and the time complexity of other operations is $O(mN^2)$. Therefore, the time complexity of MALSMEA is $\max\{O(D^2N/K), O(N \log^{m-2} N), O(mN^2)\}$. Compared with the four algorithms, the time complexity of the grouped and linked polynomial mutation operator (GLMO) is $\max\{O(D^2N/K), O(mN^2)\}$ [39], linear combination-based search algorithm (LCSA) is $O(mN^2)$ [40], vector angle-based evolutionary algorithm (VaEA) is $\max\{O(N \log^{m-2} N), O(mN^2)\}$ [38], and RVEA is $O(mN^2)$ [32]. Thus, the time complexity of MALSMEA is similar to that of GLMO but greater than that of the other three algorithms.

4. Experimental Studies

In this section, DTLZ1-DTLZ6 in the Deb, Thiele, Laumanns, and Zitzler (DTLZ) test suite [41] and LSMOP1-LSMOP9 in the Large-Scale Multi- and Many-Objective Problems (LSMOP) test suite [42] are selected to evaluate the performance of MALSMEA, and four datasets in the University of California at Irvine (UCI) machine learning library [43] are selected to evaluate the ability of MALSMEA to optimize the proposed feature selection model, among which Heart is a two-class dataset, Zoo and Iris are two multiclass datasets, and Musk1 is a high-dimensional dataset. For LSMaOPs, MALSMEA is compared with GLMO [39], LCSA [40], VaEA [38], and RVEA [32]. GLMO and LCSA are large-scale multiobjective evolutionary algorithms. GLMO uses mutation operators based on variable grouping, and LCSA uses a linear combination to reduce dimensionality. VaEA and RVEA are many-objective evolutionary algorithms that use vector angles and reference vectors, respectively. For the proposed six-objective feature selection model, MALSMEA is compared with W-MOSS [44], W-QEISS, and F-QEISS [8].

In the next sections, we introduce the performance indicators and set the parameters in the experiments. Then, for all algorithms, when the objective numbers m are 5 and 10, the population sizes N are 126 and 275, and the numbers of decision variables D are 500 and 1000, respectively. Each algorithm runs 20 times independently and stops when the number of function evaluations (FEs) reaches 90,000. The performance of MALSMEA is verified by comparing the average IGD values obtained by five algorithms. In each test instance, the best average IGD value is highlighted in bold. Finally, in four datasets, MALSMEA and three feature selection algorithms are utilized to deal with the proposed six-objective feature selection optimization model, for which $N = 100$, the maximum number of FEs is 100, and each algorithm runs independently for 10 times. The optimization ability of MALSMEA is verified by comparing the HV indicator and optimization results.

4.1. Experimental Settings

(1) *Performance Indicator*. In the experiment, IGD [45] and HV [46] are used as evaluation indicators. The smaller (larger) the IGD (HV) indicator value is, the better the performance of the algorithm. The IGD indicator evaluates the algorithm by calculating the average of minimum distances between all sampled individuals on the actual PF and the obtained solution set. The HV indicator quantifies the algorithm performance by calculating the volume between the obtained nondominated solution set and the reference point.

(2) *Parameter Settings for the Crossover and Mutation Operators*. In the performance verification experiment of MALSMEA, MALSMEA and GLMO use the mutation operator based on variable grouping to generate offspring. Other algorithms use simulated binary crossover (SBX) [32] and polynomial mutation [47]. The crossover probability is $p_c = 1.0$, the mutation probability is $p_m = 1/D$, and the distribution indicator is $\eta_m = 20$, where D is the number of decision variables. In the experiment to verify the superiority of MALSMEA with respect to the proposed model, according to [9], $p_c = 0.8$, $p_m = 0.2$.

(3) *Other Parameter Settings for Algorithms*. In MALSMEA and GLMO [39], the number of groups K is set to 4, and the ordered grouping method is adopted. For RVEA [32], the index α and the frequency f_r are set to 2 and 0.1, respectively. The parameters in W-QEISS and F-QEISS are set according to [8], and the searching method is based on r-NSGA-II [48]. The parameters in W-MOSS are set according to [44].

(4) *Datasets*. The details of 4 UCI datasets utilized are shown in Table 1.

(5) *ELM Classifier*. For the proposed model, the ELM classifier [8] is utilized to evaluate the accuracy of the current solution, which follows the criterion given in [46]: the activation function is $g(x) = 1/(1 + e^{(-x)})$ in the hidden layer, and the number of neurons is set to $n_h = 10$. The target classification variable and the (input) features are normalized into ranges [0, 1] and [−1, 1] in each dataset, respectively. To minimize the accuracy deviation, the k-fold cross validation approach is utilized with $k = 10$, and the average accuracy is used for comparison [9].

4.2. Performance Comparison of Algorithms on DTLZ. Table 2 describes the IGD indicator values obtained by the five algorithms on the 5- and 10-objective DTLZ1-DTLZ6 with 500 and 1000 decision variables. As shown in Table 2,

```
Input: Datasets with labels, DS; the maximal number of generations, $t_{max}$; the population size, $N$;
Output: The Pareto subset, $P_S$;
divide DS into training and test datasets;
$[W, Y]$ = Segment (training datasets);
$S$ = Encoding ($W_S$); $W_S$ = Feature Select ($W$);
$P(t)$ = Evaluate Six Objectives ($P$); $P$ = Initialize ($N, S$);
$P(t)$ = MALSMEA ($P(t)$);
$P_S \leftarrow P(t)$;
```

Algorithm 1: The wrapper structure of MALSMEA.

Table 1: The information of four UCI datasets.

Dataset	Classes	Features	Instance
Heart	2	13	270
Zoo	7	16	101
Iris	3	4	150
Musk1	2	166	476

MALSMEA is competitive with the other four algorithms. Specifically, MALSMEA produces 18 best results out of 24 test instances, and its performance on the 10-objective DTLZ is significantly better than that of the other algorithms. The experimental results are analyzed in detail as below.

DTLZ1 reflects the convergence of the algorithm. MALSMEA outperforms the other algorithms on the 5- and 10-objective DTLZ1. These results demonstrate that MALSMEA has better convergence on the large-scale high-dimensional DTLZ1. DTLZ2 is generally used to test the scalability of algorithms with respect to the number of objectives. The performance of MALSMEA on the 5-objective DTLZ2 is better than that of LCSA but slightly inferior to that of GLMO, VaEA, and RVEA. The performance of MALSMEA on the 10-objective DTLZ2 is better than that of the other four algorithms. Thus, MALSMEA has better scalability to the objective number.

DTLZ3 is a highly multimodal problem similar to DTLZ1. MALSMEA obtains the smallest IGD indicator value on DTLZ3 with 500 and 1000 decision variables. DTLZ4 is used to test the ability of the algorithm to ensure the diversity of the population. MALSMEA obtains the smallest IGD indicator value on the 10-objective DTLZ4 with 500 and 1000 decision variables. For the 5-objective DTLZ4, VaEA outperforms other algorithms on DTLZ4 with 500 and 1000 decision variables. MALSMEA exhibits greater diversity on the large-scale 10-objective DTLZ4.

For the 5-objective DTLZ5, MALSMEA outperforms LCSA on DTLZ5 with 500 and 1000 decision variables, but inferior to GLMO, VaEA, and RVEA. For the 10-objective DTLZ5, MALSMEA outperforms its counterparts. For DTLZ6, the overall performance of MALSMEA is optimal on instances with up to 1000 decision variables.

To further test the performance of MALSMEA, the nonparametric Friedman test [49] is employed. According to the average IGD indicator values of the five algorithms on DTLZ, Table 3 indicates the average ranking of the five algorithms. The average ranking of MALSMEA is the smallest, which indicates that MALSMEA performs the best. The average ranking of LCSA is the largest, so its performance is the worst.

To verify the efficiency of MALSMEA, Table 4 presents the running time of MALSMEA and the four other algorithms on the 10-objective DTLZ1 with 1000 decision variables. The running times of MALSMEA and GLMO are quite similar but greater than those of other algorithms.

4.3. Performance Comparison of Algorithms on LSMOP.

LSMOP is proposed to test the performance of the algorithm in LSMaOPs. Table 5 lists the IGD indicator values obtained by five algorithms on 5- and 10-objective LSMOP1-LSMOP9 with 500 and 1000 decision variables. MALSMEA produces 26 best results out of 36 test instances. Therefore, compared with the other four algorithms, MALSMEA has better performance in solving LSMaOPs.

Specifically, for the LSMOP test suite with 500 decision variables, MALSMEA outperforms the other algorithms on the 5- and 10-objective LSMOP2, LSMOP4, LSMOP5, LSMOP8, and LSMOP9. MALSMEA is inferior to LCSA on LSMOP3. MALSMEA outperforms the other algorithms on the 10-objective LSMOP1 and LSMOP7, but LCSA obtains the smallest IGD indicator value on the 5-objective LSMOP1 and LSMOP7. MALSMEA obtains the smallest IGD indicator value on the 5-objective LSMOP6, while RVEA performs better on the 10-objective LSMOP6.

For the LSMOP test suite with 1000 decision variables, MALSMEA outperforms the other algorithms on the 5- and 10-objective LSMOP2, LSMOP4, LSMOP5, LSMOP8, and LSMOP9. MALSMEA is inferior to LCSA on LSMOP3. LCSA obtains the best performance on the 5-objective LSMOP1 and LSMOP7, and MALSMEA outperforms the other algorithms on the 10-objective LSMOP1 and LSMOP7. The performance of MALSMEA on the 5-objective LSMOP6 is better than that of the other algorithms, but it is slightly inferior to that of LCSA and RVEA on the 10-objective LSMOP6.

4.4. Comparison of the Optimization Results on the Proposed Model.

Table 6 shows the HV indicator values and objective values of the four algorithms after optimization on four datasets. The results demonstrate that MALSMEA obtains the maximum HV indicator values, showing that MALSMEA has certain advantages in feature selection. As noted in

Table 2: Performance comparison between MALSMEA and four algorithms with respect to the average IGD values on the DTLZ1-DTLZ6 (gray values represent the best values in each row).

Problem	m	D	MALSMEA	GLMO	LCSA	VaEA	RVEA
DTLZ1	5	500	1.1079e + 3 (4.24e + 2)	9.9478e + 3 (1.61e + 3)	3.9526e + 3 (2.52e + 2)	4.5327e + 3 (2.97e + 2)	7.8347e + 3 (1.99e + 2)
		1000	3.6284e + 3 (1.03e + 3)	1.8810e + 4 (2.97e + 3)	7.7836e + 3 (4.34e + 2)	1.3520e + 4 (5.87e + 2)	1.8532e + 4 (3.84e + 2)
	10	500	2.2202e + 3 (3.47e + 2)	9.4305e + 3 (5.82e + 2)	4.5825e + 3 (3.67e + 2)	8.4640e + 3 (3.85e + 2)	7.2316e + 3 (7.61e + 2)
		1000	4.7828e + 3 (9.45e + 2)	1.8648e + 4 (9.68e + 2)	9.2419e + 3 (4.31e + 2)	1.8151e + 4 (5.38e + 2)	1.6042e + 4 (3.96e + 2)
DTLZ2	5	500	2.8988e + 1 (1.27e + 0)	3.0185e + 1 (4.19e + 0)	2.9554e + 1 (2.79e + 0)	4.0643e + 0 (2.88e - 1)	2.5720e + 0 (1.80e − 1)
		1000	6.6661e + 1 (2.14e + 0)	6.2634e + 1 (6.02e + 0)	7.7019e + 1 (5.98e + 0)	1.8169e + 1 (8.34e − 1)	1.4208e + 1 (6.65e − 1)
	10	500	2.1635e + 1 (3.68e + 0)	3.7764e + 1 (4.64e + 0)	4.2687e + 1 (1.30e + 1)	2.4451e + 1 (1.05e + 0)	2.6761e + 1 (7.91e + 0)
		1000	4.4200e + 1 (7.77e + 0)	7.7862e + 1 (7.30e + 0)	7.9824e + 1 (2.29e + 0)	5.8910e + 1 (1.32e + 0)	5.0172e + 1 (1.07e + 0)
DTLZ3	5	500	4.3020e + 3 (1.57e + 3)	2.3235e + 4 (6.13e + 3)	1.2346e + 4 (7.98e + 0)	1.8602e + 4 (7.53e + 2)	3.1291e + 4 (6.31e + 2)
		1000	1.0670e + 4 (2.63e + 3)	4.3783e + 4 (8.98e + 3)	2.4844e + 4 (1.47e + 1)	6.0220e + 4 (1.81e + 3)	7.6232e + 4 (9.67e + 3)
	10	500	1.3306e + 4 (1.60e + 3)	4.1894e + 4 (2.65e + 3)	1.4213e + 4 (1.03e + 1)	3.8615e + 4 (7.37e + 2)	3.9350e + 4 (7.75e + 2)
		1000	2.5625e + 4 (4.16e + 3)	8.4954e + 4 (6.42e + 3)	2.7420e + 4 (9.93e + 0)	8.5528e + 4 (1.17e + 3)	8.8290e + 4 (1.15e + 3)
DTLZ4	5	500	2.6523e + 1 (1.65e + 0)	3.5059e + 1 (5.11e + 0)	2.6238e + 1 (3.03e + 0)	5.6372e + 0 (3.94e - 1)	5.9172e + 0 (7.12e - 1)
		1000	5.8758e + 1 (3.00e + 0)	6.6782e + 1 (1.20e + 1)	6.9454e + 1 (3.04e + 0)	2.2961e + 1 (9.24e − 1)	2.9794e + 1 (2.77e + 0)
	10	500	2.3290e + 1 (1.42e + 0)	3.3557e + 1 (9.70e + 0)	4.0018e + 1 (1.84e + 0)	2.4352e + 1 (8.56e − 1)	2.5171e + 1 (5.22e − 1)
		1000	4.9795e + 1 (3.90e + 0)	7.0596e + 1 (1.42e + 1)	8.0600e + 1 (2.05e + 0)	5.8910e + 1 (1.22e + 0)	5.6250e + 1 (1.07e + 0)
DTLZ5	5	500	2.8696e + 1 (1.44e + 0)	2.4279e + 1 (7.19e + 0)	3.5655e + 1 (1.04e + 0)	7.8308e + 0 (6.53e − 1)	2.9302e + 0 (2.18e − 1)
		1000	6.3241e + 1 (2.51e + 0)	3.7272e + 1 (1.12e + 1)	7.4941e + 1 (1.88e + 0)	2.7873e + 1 (1.40e + 0)	1.6365e + 1 (4.94e − 1)
	10	500	2.2663e + 1 (4.07e + 0)	2.2874e + 1 (7.97e + 0)	4.0439e + 1 (4.57e + 0)	2.7748e + 1 (1.08e + 0)	2.6317e + 1 (8.24e + 0)
		1000	4.8397e + 1 (7.02e + 0)	4.8756e + 1 (1.63e + 1)	8.1209e + 1 (2.71e + 0)	6.3840e + 1 (1.28e + 0)	4.9904e + 1 (1.13e + 0)
DTLZ6	5	500	8.8879e + 0 (1.39e + 0)	4.2732e + 2 (2.36e + 1)	9.4574e + 0 (8.51e + 0)	3.8495e + 2 (4.90e + 0)	3.6416e + 2 (2.58e + 0)
		1000	1.9706e + 1 (3.29e + 0)	8.9090e + 2 (2.39e + 1)	2.9199e + 1 (1.29e + 1)	8.1717e + 2 (6.01e + 0)	8.0078e + 2 (3.00e + 0)
	10	500	5.4188e + 1 (1.00e + 1)	4.2710e + 2 (1.53e + 1)	7.0212e + 1 (1.23e + 1)	4.1523e + 2 (2.44e + 0)	4.1207e + 2 (2.66e + 0)
		1000	1.0773e + 2 (3.16e + 1)	8.6234e + 2 (4.55e + 1)	1.1227e + 2 (8.94e + 1)	8.5524e + 2 (2.99e + 0)	8.5757e + 2 (2.39e + 0)

Table 3: Average rankings of the Friedman test.

Algorithm	Ranking
MALSMEA	2.1667
GLMO	3.4583
LCSA	3.6667
VaEA	2.9583
RVEA	2.75

Table 4: Comparison of running time between MALSMEA and the other four algorithms.

Algorithm	Time
MALSMEA	2.3113e + 2
GLMO	2.0182e + 2
LCSA	4.3017e + 1
VaEA	1.2587e + 2
RVEA	6.8803e + 1

Table 5: Performance comparison between MALSMEA and four algorithms with respect to the average IGD values on the LSMOP1–LSMOP9 (gray values represent the best values in each row).

Problem	m	D	MALSMEA	GLMO	LCSA	VaEA	RVEA
LSMOP1	5	500	1.3173e + 0 (1.55e − 1)	9.9913 e − 1 (1.05e − 1)	9.3999e − 1 (5.30e − 3)	1.6687e + 0 (2.66e − 1)	1.2713e + 0 (1.54e − 1)
		1000	1.3109e + 0 (1.61e − 1)	1.2099e + 0 (5.21e − 1)	9.3942e − 1 (2.67e − 3)	3.6704e + 0 (4.00e − 1)	2.6898e + 0 (2.09e − 1)
	10	500	1.2008e + 0 (1.89e − 1)	5.9934e + 0 (2.79e + 0)	1.2010e + 0 (1.16e − 3)	4.1745e + 0 (1.28e + 0)	1.6742e + 0 (3.51e − 1)
		1000	1.1728e + 0 (1.53e − 1)	7.9449e + 0 (3.19e + 0)	1.1938e + 0 (2.75e − 3)	7.0153e + 0 (6.50e − 1)	4.0353e + 0 (9.27e − 1)

Table 5: Continued.

Problem	*m*	*D*	MALSMEA	GLMO	LCSA	VaEA	RVEA
LSMOP2	5	500	1.5237*e* − 1 (1.77*e* − 3)	1.8423*e* − 1 (5.16*e* − 3)	1.9821*e* - 1 (6.56*e* − 3)	1.6390*e* - 1 (1.71*e* − 3)	1.6594*e* − 1 (9.99*e* − 4)
		1000	1.3444*e* − 1 (1.08*e* − 3)	1.6139*e* − 1 (4.75*e* − 3)	1.7402*e* − 1 (3.87*e* − 3)	1.4188*e* − 1 (1.73*e* − 3)	1.4299*e* − 1 (8.72*e* − 4)
	10	500	2.8094*e* − 1 (6.69*e* − 3)	3.3525*e* − 1 (7.25*e* − 3)	3.6322*e* − 1 (8.55*e* − 3)	3.1995*e* − 1 (3.89*e* − 3)	2.8197*e* − 1 (3.56*e* − 3)
		1000	2.3979*e* − 1 (2.71*e* − 3)	2.8301*e* − 1 (5.22*e* − 3)	3.0751*e* − 1 (7.90*e* − 3)	2.6900*e* − 1 (1.85*e* − 3)	2.3980*e* − 1 (3.04*e* − 3)
LSMOP3	5	500	1.1955*e* + 1 (3.86*e* + 0)	1.3626*e* + 0 (6.23*e* − 1)	9.5883*e* − 1 (0.00*e* + 0)	1.6636*e* + 1 (4.85*e* + 0)	4.7605*e* + 0 (1.27*e* + 0)
		1000	1.3419*e* + 1 (4.38*e* + 0)	1.4773*e* + 0 (5.34*e* − 1)	9.5883*e* − 1 (0.00*e* + 0)	1.6875*e* + 1 (5.62*e* + 0)	8.7885*e* + 0 (1.03*e* + 0)
	10	500	1.2546*e* + 1 (1.59*e* + 0)	2.1075*e* + 2 (3.43*e* + 2)	1.8733*e* + 0 (1.57*e* − 3)	1.7999*e* + 1 (3.05*e* + 0)	2.4510*e* + 0 (4.99*e* − 1)
		1000	1.3071*e* + 1 (1.29*e* + 0)	1.1423*e* + 4 (1.26*e* + 2)	1.9179*e* + 0 (8.35*e* − 4)	1.9379*e* + 1 (2.80*e* + 0)	4.3816*e* + 1(1.40*e* + 0)
LSMOP4	5	500	2.8356*e* - 1 (8.13*e* - 3)	3.3698*e* − 1 (1.31*e* − 2)	3.2856*e* − 1 (9.98*e* − 3)	3.0856–1 (5.78*e* − 3)	2.8894*e* − 1 (2.96*e* − 3)
		1000	2.1150*e* - 1 (5.31*e* − 3)	2.4674*e* − 1 (7.40*e* − 3)	2.5458*e* − 1 (6.51*e* − 3)	2.1842*e* − 1 (3.10*e* − 3)	2.1661*e* − 1 (1.51*e* − 3)
	10	500	3.3748*e* − 1 (5.61*e* − 3)	3.9190*e* − 1 (1.04*e* − 2)	4.3146*e* − 1 (1.52*e* − 2)	3.7828*e* − 1 (3.79*e* − 3)	3.4044*e* − 1 (3.98*e* − 3)
		1000	2.7003*e* − 1 (2.36*e* − 3)	3.1838*e* - 1 (8.76*e* − 3)	3.5483*e* − 1 (6.41*e* − 3)	3.0457*e* − 1 (3.65*e* − 3)	2.7902*e* − 1 (3.82*e* − 3)
LSMOP5	5	500	4.5817*e* − 1 (5.45*e* − 3)	3.3566*e* + 0 (3.16*e* + 0)	4.6074*e* − 1 (3.81*e* − 2)	4.5633*e* + 0 (3.26*e* − 1)	1.8603*e* + 0 (3.83*e* − 1)
		1000	4.5647*e* − 1 (2.97*e* − 2)	8.3782*e* + 0 (6.28*e* + 0)	4.5874*e* − 1 (1.99*e* − 2)	7.4372*e* + 0 (7.67*e* − 1)	3.3211*e* + 0 (5.08*e* − 1)
	10	500	6.5504*e* − 1 (4.37*e* − 2)	1.6148*e* + 1 (8.45*e* + 0)	1.1132*e* + 0 (8.69*e* − 2)	8.4930*e* + 0 (1.21*e* + 0)	3.0758*e* + 0 (5.69*e* − 1)
		1000	6.6973*e* − 1 (6.22*e* − 2)	1.4246*e* + 1 (6.02*e* + 0)	1.1087*e* + 0 (9.32*e* − 2)	1.0274*e* + 1 (1.04*e* + 0)	6.1324*e* + 0 (5.95*e* − 1)
LSMOP6	5	500	1.2094 *e*+ 0 (1.33*e* − 1)	5.3807*e* + 2 (1.68*e* + 3)	1.2106*e* + 0 (3.67*e* − 2)	1.1135*e* + 1 (5.75*e* + 0)	8.3040*e* + 0 (1.66*e* + 1)
		1000	1.2188*e* + 0 (8.52*e* − 2)	2.5183*e* + 3 (4.35*e* + 3)	1.2549*e* + 0 (5.34*e* − 2)	1.4415*e* + 2 (3.65*e* + 1)	5.3053*e* + 1 (2.99*e* + 1)
	10	500	1.4348*e* + 0 (1.42*e* − 1)	6.0471*e* + 1 (1.88*e* + 2)	1.4179*e* + 0 (8.13*e* − 2)	1.3763*e* + 2 (2.90*e* + 2)	1.2580*e* + 0 (1.09*e* − 1)
		1000	1.4961*e* + 0 (1.46*e* − 1)	7.6272*e* + 2 (3.01*e* + 3)	1.3573*e* + 0(7.95*e* − 2)	1.5136*e* + 0 (8.68*e* − 3)	1.2743*e* + 0 (9.00*e* − 2)
LSMOP7	5	500	1.3323*e* + 0 (6.63*e* − 2)	2.4841*e* + 0 (3.00*e* − 1)	1.0912*e* + 0 (1.46*e* − 2)	2.9317*e* + 0 (1.47*e* − 1)	1.2645*e* + 0 (1.88*e* − 1)
		1000	1.3577*e* + 0 (6.26*e* − 2)	1.7911*e* + 0 (1.01*e* − 1)	1.0321*e* + 0 (1.40*e* − 2)	1.9182*e* + 0 (5.40*e* − 2)	1.1214*e* + 0 (8.68*e* − 2)
	10	500	1.3995*e* + 0 (7.97*e* − 2)	3.5137*e* + 4 (1.36*e* + 4)	1.5578*e* + 0 (5.12*e* − 2)	1.0739*e* + 3 (7.45*e* + 2)	2.6040*e* + 1 (6.95*e* + 0)
		1000	1.4663*e* + 0 (1.11*e* − 1)	3.7805*e* + 4 (1.15*e* + 4)	1.5933*e* + 0 (5.63*e* − 2)	2.7102*e* + 3 (1.09*e* + 3)	1.4501*e* + 2 (3.01*e* + 1)
LSMOP8	5	500	3.8850*e* − 1 (2.43*e* − 2)	1.1661*e* + 0 (7.11*e* − 2)	3.8922*e* − 1 (1.02*e* − 2)	1.1767*e* + 0 (9.67*e* − 3)	9.3066*e* - 1 (1.19*e* − 1)
		1000	3.9206*e* − 1 (3.27*e* − 2)	1.0697*e* + 0 (9.46*e* − 2)	3.9962*e* − 1 (8.72*e* − 3)	1.1544*e* + 0 (1.25*e* − 3)	8.9791*e* - 1 (1.45*e* − 1)
	10	500	6.4152*e* − 1 (4.00*e* − 2)	1.2619*e* + 1 (4.49*e* + 0)	9.6995*e* − 1 (9.27*e* − 2)	2.8446*e* + 0 (5.01*e* − 1)	1.4025*e* + 0 (1.12*e* − 1)
		1000	6.2434*e* − 1 (3.37*e* − 2)	1.1402*e* + 1 (4.19*e* + 0)	1.0886*e* + 0 (1.06*e* − 1)	4.0270*e* + 0 (6.02*e* −1)	2.6957*e* + 0 (3.85*e* − 1)

Table 5: Continued.

Problem	m	D	MALSMEA	GLMO	LCSA	VaEA	RVEA
LSMOP9	5	500	$2.8005e+0$ $(2.91e-8)$	$2.9775e+0$ $(9.23e-2)$	$2.9985e+0$ $(8.77e-3)$	$1.2971e+1$ $(2.27e+0)$	$2.5483e+1$ $(6.20e+0)$
		1000	$2.9801e+0$ $(9.11e-2)$	$2.9976e+0$ $(9.44e-2)$	$3.0005e+0$ $(0.00e+0)$	$3.5883e+1$ $(3.99e+0)$	$5.5544e+1$ $(1.95e+1)$
	10	500	$6.4182e+0$ $(1.93–1)$	$6.5037e+0$ $(7.63e-1)$	$6.5321e+0$ $(3.65e-15)$	$3.6094e+2$ $(2.89e+1)$	$2.7313e+2$ $(9.11e+1)$
		1000	$6.3652e+0$ $(2.05e-1)$	$6.3891e+0$ $(1.06e+0)$	$6.5321e+0$ $(3.65e-15)$	$5.0223e+2$ $(2.77e+1)$	$3.4370e+2$ $(9.37e+1)$

Table 6: HV values and optimized results of four algorithms (values in bold represent better results).

Dataset	Algorithm	HV	Feature	Accuracy	Relevance	Redundancy	Interclass distance	Intraclass distance
Heart	MALSMEA	**0.9972**	**6**	**0.7979**	0.4615	0.1923	**0.0802**	**0.0123**
	W-MOSS	0.9962	7	0.7667	0.5385	0.2692	0.0769	0.0128
	W-QEISS	0.9943	8	0.7604	**0.6154**	0.3590	0.0764	0.0130
	F-QEISS	0.9980	7	0.7811	0.5385	**0.0256**	0.0798	0.0125
Zoo	MALSMEA	**0.9979**	**5**	**0.9842**	0.3125	0.0833	**0.0637**	**0.0074**
	W-MOSS	0.9975	7	0.9816	0.4375	0.1750	0.0622	0.0085
	W-QEISS	0.9972	7	0.9697	**0.5000**	0.2333	0.0615	0.0076
	F-QEISS	0.9977	6	0.9556	0.3750	**0.0167**	0.0609	0.0083
Iris	MALSMEA	**0.9351**	**2**	**0.9387**	0.5000	**0.1667**	**0.2574**	**0.1667**
	W-MOSS	0.9234	3	0.9071	**0.7500**	0.5000	0.2566	0.1673
	W-QEISS	0.9236	3	0.9049	0.5655	0.1765	0.2571	0.1670
	F-QEISS	0.9247	3	0.9187	**0.7500**	**0.1667**	0.2569	0.1668
Musk1	MALSMEA	**0.9697**	**11**	**0.6173**	0.0663	**0.0045**	$\mathbf{7.3102e-5}$	**0.0060**
	W-MOSS	0.9693	12	0.6130	0.0723	0.0048	$7.3023e-5$	**0.0060**
	W-QEISS	0.9603	13	0.5956	**0.0783**	0.0057	$7.3037e-5$	0.0067
	F-QEISS	0.9627	13	0.6069	**0.0783**	0.0057	$7.3026e-5$	0.0062

Table 6, for the four datasets, the optimization performance of MALSMEA is better on Iris and Musk1. MALSMEA is slightly inferior to the other three algorithms in relevance and redundancy but exhibits better performance in the other four objectives. In addition, W-QEISS and F-QEISS are relatively better than the other algorithms in terms of relevance and redundancy, but they are worse in other objectives.

5. Conclusion

In this paper, a modified vector angle-based large-scale many-objective evolutionary algorithm called MALSMEA is proposed. In MALSMEA, the polynomial mutation based on variable grouping is used to replace the polynomial mutation to improve the efficiency of solving large-scale optimization problems. A novel worst-case solution replacement strategy based on SDE is proposed to replace the worse one of two individuals with similar search directions to increase diversity. In addition, MALSMEA is compared with four typical algorithms to solve the optimization problem with up to 10 objectives and 1000 decision variables. Experimental results indicate that MALSMEA outperforms the four algorithms on the DTLZ and LSMOP test suites. By studying the existing feature selection models, taking the number of selected features, accuracy, relevance, redundancy, interclass distance, and intraclass distance as the optimization objectives, a six-objective optimization model is constructed and solved by using MALSMEA. Compared with the other three feature selection algorithms, MALSMEA has some advantages in solving this model.

Future studies will proceed in two directions. The first direction is to add a parallel strategy to MALSMEA to improve efficiency or to further modify its environmental selection method. Another research direction is to solve LSMaOPs in other fields using MALSMEA.

Data Availability

The details of the four UCI datasets utilized are shown in Table 1.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant no. 61976242, in part by the Fundamental Scientific Research Funds for Interdisciplinary Team of Hebei University of Technology under Grant no. JBKYTD2002, and in part by the Guangdong Provincial Key Laboratory under Grant no. 2020B121201001.

References

[1] M. Komeili, W. Louis, N. Armanfard, and D. Hatzinakos, "Feature selection for nonstationary data: application to human recognition using medical biometrics," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1446–1459, 2018.

[2] P. García-Díaz, I. Sánchez-Berriel, J. A. Martínez-Rojas, and A. M. Diez-Pascual, "Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-seq data," *Genomics*, vol. 112, no. 2, pp. 1916–1925, 2020.

[3] S. L. Marie-Sainte and S. Ghouzali, "Multi-objective particle swarm optimization-based feature selection for face recognition," *Studies in Informatics and Control*, vol. 29, no. 1, pp. 99–109, 2020.

[4] Z.-Z. Liu, J.-W. Huang, Y. Wang, and D.-S. Cao, "ECoFFeS: a software using evolutionary computation for feature selection in drug discovery," *IEEE Access*, vol. 6, pp. 20950–20963, 2018.

[5] H. Hanchuan Peng, F. Fuhui Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[6] H. Xia, J. Zhuang, and D. Yu, "Multi-objective unsupervised feature selection algorithm utilizing redundancy measure and negative epsilon-dominance for fault diagnosis," *Neurocomputing*, vol. 146, no. 25, pp. 113–124, 2014.

[7] S. Paul and S. Das, "Simultaneous feature selection and weighting—an evolutionary multi-objective optimization approach," *Pattern Recognition Letters*, vol. 65, no. 1, pp. 51–59, 2015.

[8] K. Gulsah, G. Stefano, S. A. Damla, and T. Riccardo, "Identifying (quasi) equally informative subsets in feature selection problems for classification: a max-relevance min-redundancy approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1424–1437, 2016.

[9] H. Li, F. He, Y. Liang, and Q. Quan, "A dividing-based many-objective evolutionary algorithm for large-scale feature selection," *Soft Computing*, vol. 24, no. 9, pp. 1–31, 2019.

[10] Z.-M. Gu and G.-G. Wang, "Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization," *Future Generation Computer Systems*, vol. 107, pp. 49–69, 2020.

[11] Q. Lin, S. Liu, K.-C. Wong et al., "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 391–405, 2019.

[12] W. L. Wang, W. Li, and Y. L. Wang, "An opposition-based evolutionary algorithm for many-objective optimization with adaptive clustering mechanism," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 5126239, 27 pages, 2019.

[13] Q. Wang, L. Zhang, S. Wei, and B. Li, "Tensor decomposition-based alternate sub-population evolution for large-scale many-objective optimization," *Information Sciences*, vol. 569, pp. 376–399, 2021.

[14] X. Y. Zhang, Y. Tian, R. Cheng, and Y. C. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 99, pp. 97–112, 2018.

[15] Z. A. Yin, G. G. Wang, K. Q. Li, W. C. Yeh, M. W. Jian, and J. Y. Dong, "Enhancing MOEA/D with information feedback models for large-scale many-objective optimization," *Information Sciences*, vol. 522, pp. 1–16, 2020.

[16] D. Hadka and P. Reed, "Borg: an auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.

[17] C. Zhou, G. M. Dai, and M. C. Wang, "Enhanced θ dominance and density selection based evolutionary algorithm for many-objective optimization problems," *Applied Intelligence*, vol. 48, no. 1, pp. 992–1012, 2018.

[18] X. F. Xiufen Zou, Y. Yu Chen, M. Z. Minzhong Liu, and L. S. Lishan Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1402–1412, 2008.

[19] J. C. Yuan and H. L. Liu, "A new dominance relation based on simplex for many objective optimization problems," in *Proceedings of the 2016 12th International Conference on Computational Intelligence and Security (CIS)*, pp. 175–178, Wuxi, China, December 2016.

[20] J. K. Chong and K. C. Tan, "A novel grid-based differential evolution (DE) algorithm for many-objective optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2776–2783, Vancouver, Canada, July 2016.

[21] X. Cai, Y. Xiao, M. Li, H. Hu, H. Ishibuchi, and X. Li, "A grid-based inverted generational distance for multi/many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 21–34, 2021.

[22] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[23] S. Rostami and F. Neri, "A fast hypervolume driven selection mechanism for many-objective optimisation problems," *Swarm and Evolutionary Computation*, vol. 34, no. 1, pp. 50–67, 2016.

[24] Y. N. Sun, C. C. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2018.

[25] Z. Liang, T. Luo, K. Hu, X. Ma, and Z. Zhu, "An indicator-based many-objective evolutionary algorithm with boundary protection," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–14, 2020.

[26] Y. Liu, J. Liu, T. Li, and Q. Li, "An R2 indicator and weight vector-based evolutionary algorithm for multi-objective optimization," *Soft Computing*, vol. 24, no. 7, pp. 5079–5100, 2019.

[27] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[28] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016.

[29] C. Zhao, Y. Zhou, and Z. Chen, "Decomposition-based evolutionary algorithm with automatic estimation to handle many-objective optimization problem," *Information Sciences*, vol. 546, pp. 1030–1046, 2021.

[30] C. Dai, X. Lei, and X. Q. He, "A decomposition-based evolutionary algorithm with adaptive weight adjustment for many-objective problems," *Soft Computing*, vol. 24, no. 1, pp. 10587–10609, 2020.

[31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with

box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[32] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[33] F. Gu and Y.-M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 211–225, 2018.

[34] M.-G. Dong, B. Liu, and C. Jing, "A many-objective evolutionary algorithm based on decomposition with dynamic resource allocation for irregular optimization," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 8, pp. 1171–1190, 2020.

[35] L. Li, G. G. Yen, A. Sahoo, L. Chang, and T. Gu, "On the estimation of pareto front and dimensional similarity in many-objective evolutionary algorithm," *Information Sciences*, vol. 563, pp. 375–400, 2021.

[36] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, "AnD: a many-objective evolutionary algorithm with angle-based selection and shift-based density estimation," *Information Sciences*, vol. 509, pp. 400–419, 2020.

[37] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Vol. 31, Morgan Kaufmann, Burlington, MA, USA, 2011.

[38] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.

[39] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "Mutation operators based on variable grouping for multi-objective large-scale optimization," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece, December 2016.

[40] H. Zille, *Large-scale multi-objective optimisation: new approaches and a classification of the state-of-the-art*, Ph.D thesis, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany, 2019.

[41] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[42] C. Ran, Y. C. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2017.

[43] K. Bache and M. Lichman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml/, 2019.

[44] T. M. Hamdani, J. M. Won, and A. M. Alimi, "Multi-objective feature selection with NSGA-II," in *Adaptive and Natural Computing Algorithms. ICANNGA 2007*, vol. 4431, pp. 240–247, Springer, Berlin, Germany, 2009.

[45] Y. Zhou, Y. Xiang, Z. Chen, J. He, and J. Wang, "A scalar projection and angle-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2073–2084, 2019.

[46] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[47] G. Chen and J. Li, "A diversity ranking based evolutionary algorithm for multi-objective and many-objective optimization," *Swarm and Evolutionary Computation*, vol. 48, pp. 274–287, 2019.

[48] L. Ben Said, S. Bechikh, and K. Ghedira, "The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 801–818, 2010.

[49] J. Alcalá-Fdez, L. Sánchez, S. García et al., "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2008.