Future Generation Computer Systems 100 (2019) 952–981

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Multiobjective feature selection for microarray data via distributed parallel algorithms

Bin Cao [a,b], Jianwei Zhao [a,b,*], Po Yang [c,*], Peng Yang [b], Xin Liu [a], Jun Qi [c], Andrew Simpson [c], Mohamed Elhoseny [d], Irfan Mehmood [e], Khan Muhammad [f]

[a] State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, China
[b] School of Artificial Intelligence, Hebei University of Technology, China
[c] Department of Computer Science, Liverpool John Moores University, UK
[d] Faculty of Computers and Information, Mansoura University, Egypt
[e] Department of Media Design and Technology, Faculty of Engineering and Informatics, University of Bradford, Bradford BD7 1DP, United Kingdom
[f] Department of Software, Sejong University, Seoul 143-747, Republic of Korea

HIGHLIGHTS

- A multi-objective feature selection model is presented and tackled.
- Algorithms with two encoding methodologies are proposed.
- Adaptive technique is explored.
- Explicit feature number threshold and distributed parallelism are employed for efficiency.

ARTICLE INFO

Article history:
Received 7 August 2018
Received in revised form 28 January 2019
Accepted 18 February 2019
Available online 17 May 2019

Keywords:
Microarray dataset
High dimension
Multiobjective feature selection
Distributed parallelism
Feature redundancy

ABSTRACT

Many real-world problems are large in scale and hence difficult to address. Due to the large number of features in microarray datasets, feature selection and classification are even more challenging for such datasets. Not all of these numerous features contribute to the classification task, and some even impede performance. Through feature selection, a feature subset that contains only a small quantity of essential features can be generated to increase the classification accuracy and significantly reduce the time consumption. In this paper, we construct a multiobjective feature selection model that simultaneously considers the classification error, the feature number and the feature redundancy. For this model, we propose several distributed parallel algorithms based on different encodings and an adaptive strategy. Additionally, to reduce the time consumption, various tactics are employed, including a feature number constraint, distributed parallelism and sample-wise parallelism. For a batch of microarray datasets, the proposed algorithms are superior to several state-of-the-art multiobjective evolutionary algorithms in terms of both effectiveness and efficiency.

1. Introduction

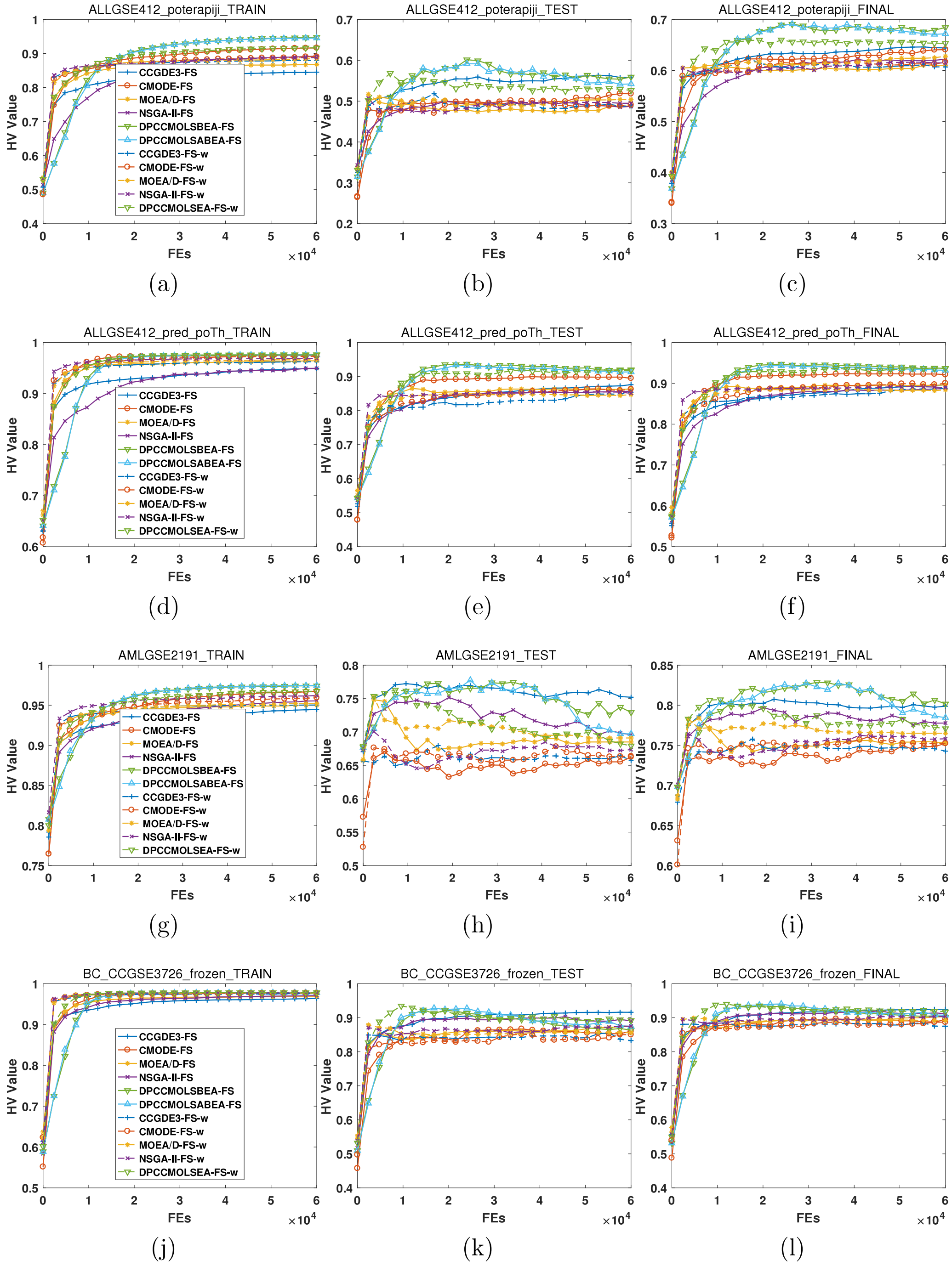
An object can be abstracted as a series of features representing various properties of that object. Based on these features, one can classify object instances and perform other analyses [1–5]. With the emergence of the big data era, many problems are becoming increasingly larger in scale. For the feature selection problem with respect to microarray data [6,7], as the number of genes reaches more than tens of thousands, the problem complexity increases even more rapidly. If the feature selection problem is viewed as a combination problem, then in the case of n features, the number of possible combinations will be $2^n - 1$ due to their exponential relationship. Thus, for microarray datasets, exhaustive enumeration will be very time consuming and intolerable.

Due to high computational costs, the existing research on feature selection for microarray datasets has focused on filter methods [6,7], and studies on wrapper and embedded methodologies are relatively few. However, in filter methods, the classifier is not considered simultaneously, often leading to poor classification performance.

For NP-hard problems and time-consuming combination problems such as those discussed above, heuristic algorithms can be used to explore the search space by means of simple population-based strategies, thus allowing suboptimal solutions to be found

[*] Corresponding authors.
E-mail addresses: 201422102003@stu.hebut.edu.cn (J. Zhao), poyangcn@gmail.com (P. Yang).

https://doi.org/10.1016/j.future.2019.02.030
0167-739X/© 2019 Published by Elsevier B.V.



(a) (b) (c)

(d) (e) (f)

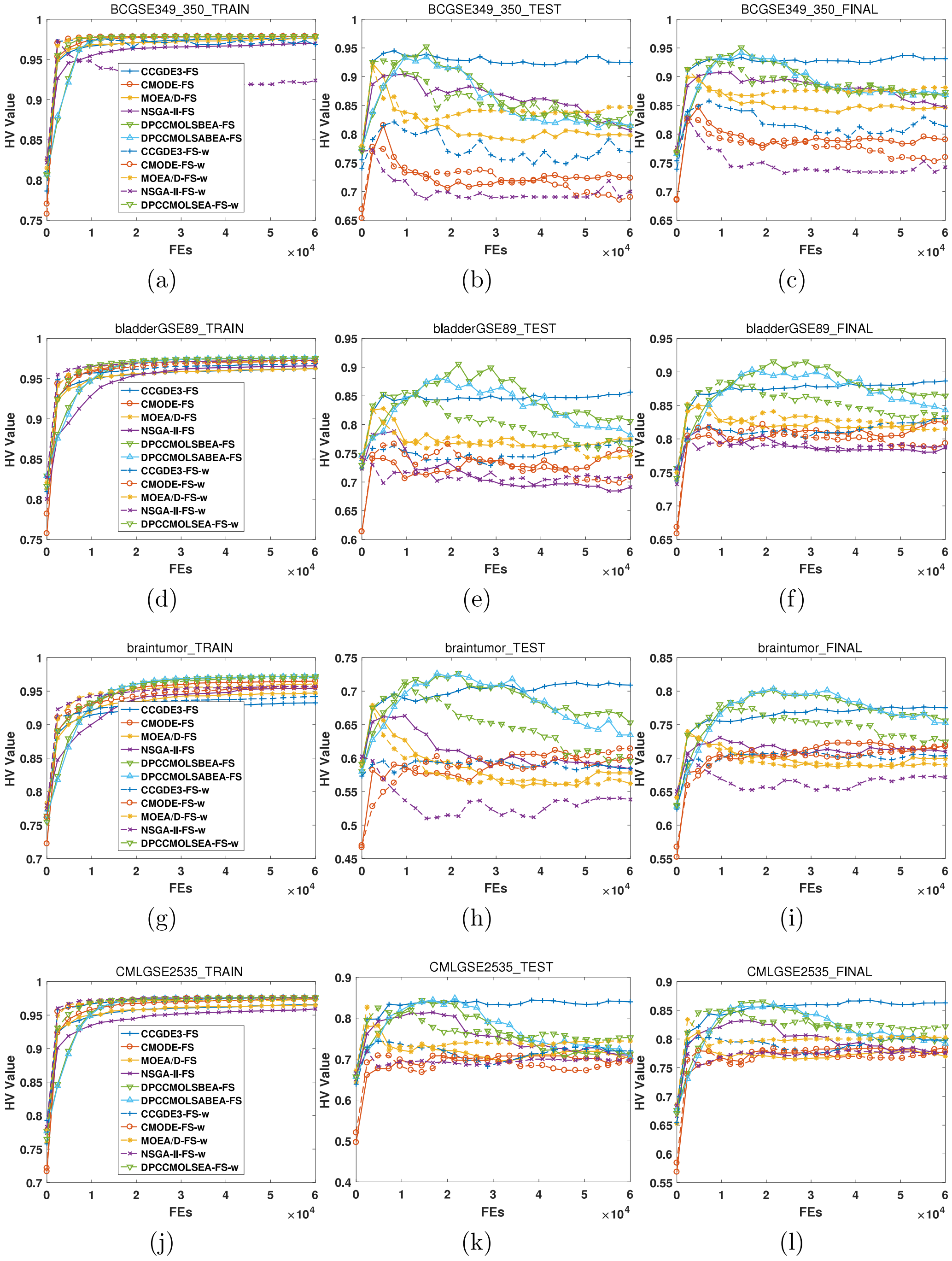
(g) (h) (i)

(j) (k) (l)

Fig. 1. Illustration of HV indicator values during evolution.

within an acceptable amount of time without examining every possible solution. In [8], a simple genetic algorithm (GA) [9] was enhanced with a local improvement strategy, resulting in a powerful evolutionary algorithm (EA) oriented toward feature selection. In each experiment, a desirable feature number was specified, and a penalty was applied to the fitness function. Gu et al. [10] employed the competitive swarm optimizer, a variant of particle swarm optimization (PSO) [11], to generate a feature subset from a large number of features, and a threshold was utilized to select features represented by continuous values. In [12], a feature selection algorithm was proposed by hybridizing PSO and SVM approaches; specifically, the feature selection status and the parameters of the RBF kernel function in the SVM algorithm were simultaneously optimized utilizing PSO. Onan et al. [13] utilized a GA to aggregate the feature rankings from filter methods for feature selection. In [14], the grey wolf optimization was transformed into a binary form, which was shown to outperform PSO and GA methods.

However, in the research discussed above, only one main objective was considered, namely, the classification accuracy, while the feature number was not concerned, a threshold was applied to indicate the desired number of features or a weighted sum was utilized to aggregate the above two together. If, instead, multiple objectives are to be simultaneously considered, multiobjective evolutionary algorithms (MOEAs) are a suitable approach. Thus,



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 2. Illustration of HV indicator values during evolution.

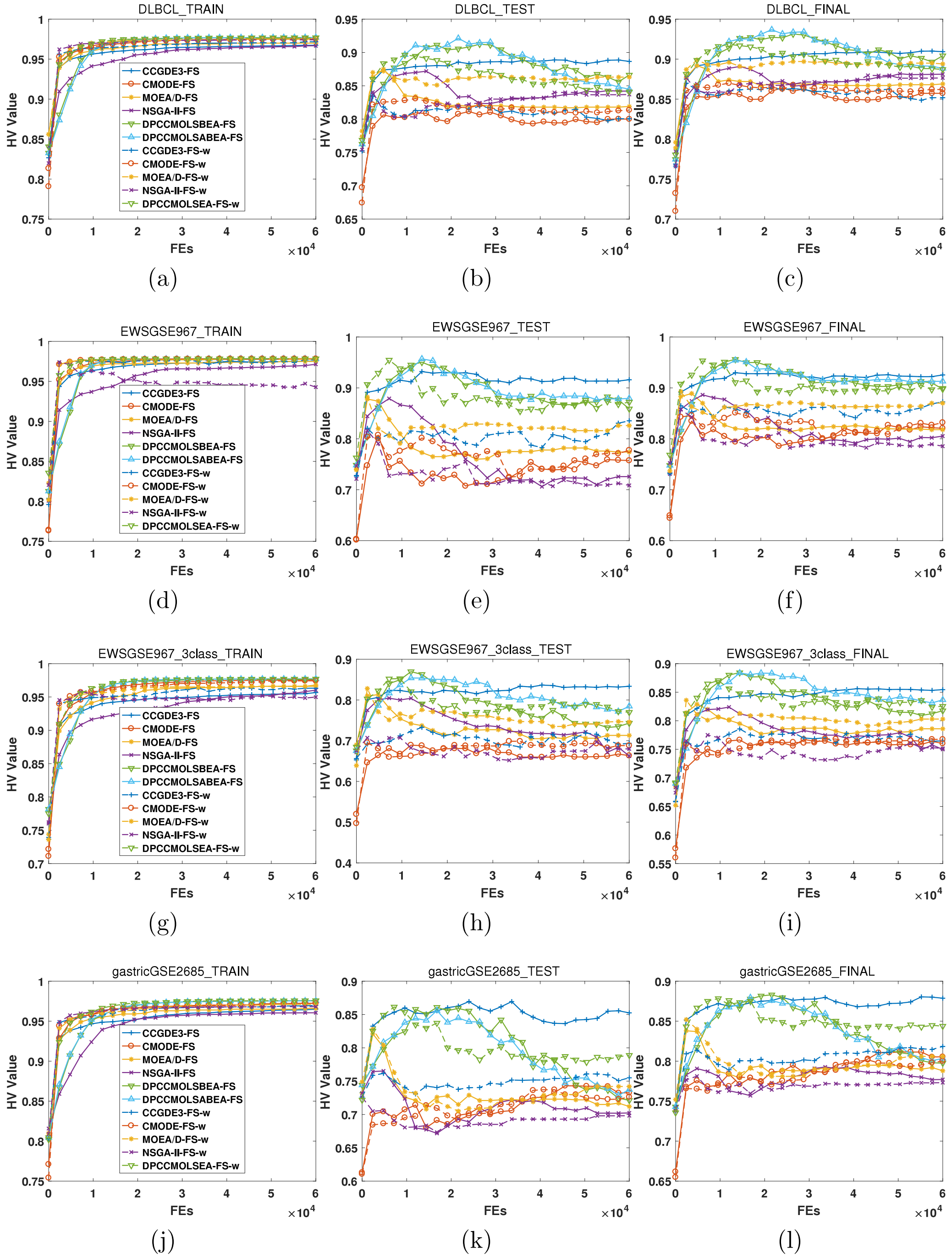
many research efforts have been devoted to the multiobjective feature selection problem [15].

In studies on MOEAs, the classification accuracy is always the main concern, which may be characterized in terms of metrics such as the overall classification accuracy, the true positive rate and the true negative rate [16]. The feature number is also of interest [17]. However, the feature redundancy is rarely considered [18,19]. In this paper, we propose a multiobjective feature selection model that simultaneously considers three objectives: classification error, feature number and feature redundancy.

Microarray datasets contain an extremely large number of features. Although MOEAs are more efficient than brute force methodologies, their time consumption may still be intolerable in some cases. Consequently, feature number constraint and distributed parallel MOEAs [20] will be beneficial. Additionally, for high-dimensional multiobjective problems (MOPs), the cooperative coevolutionary (CC) framework [21] divides the original problem into several low-dimensional tasks by separating the variables into several groups, thus yielding better optimization effectiveness and efficiency.

In summary, the contributions of this paper can be highlighted as follows:

1. A multiobjective feature selection model is proposed that simultaneously considers three objectives: classification error, feature number and feature redundancy.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

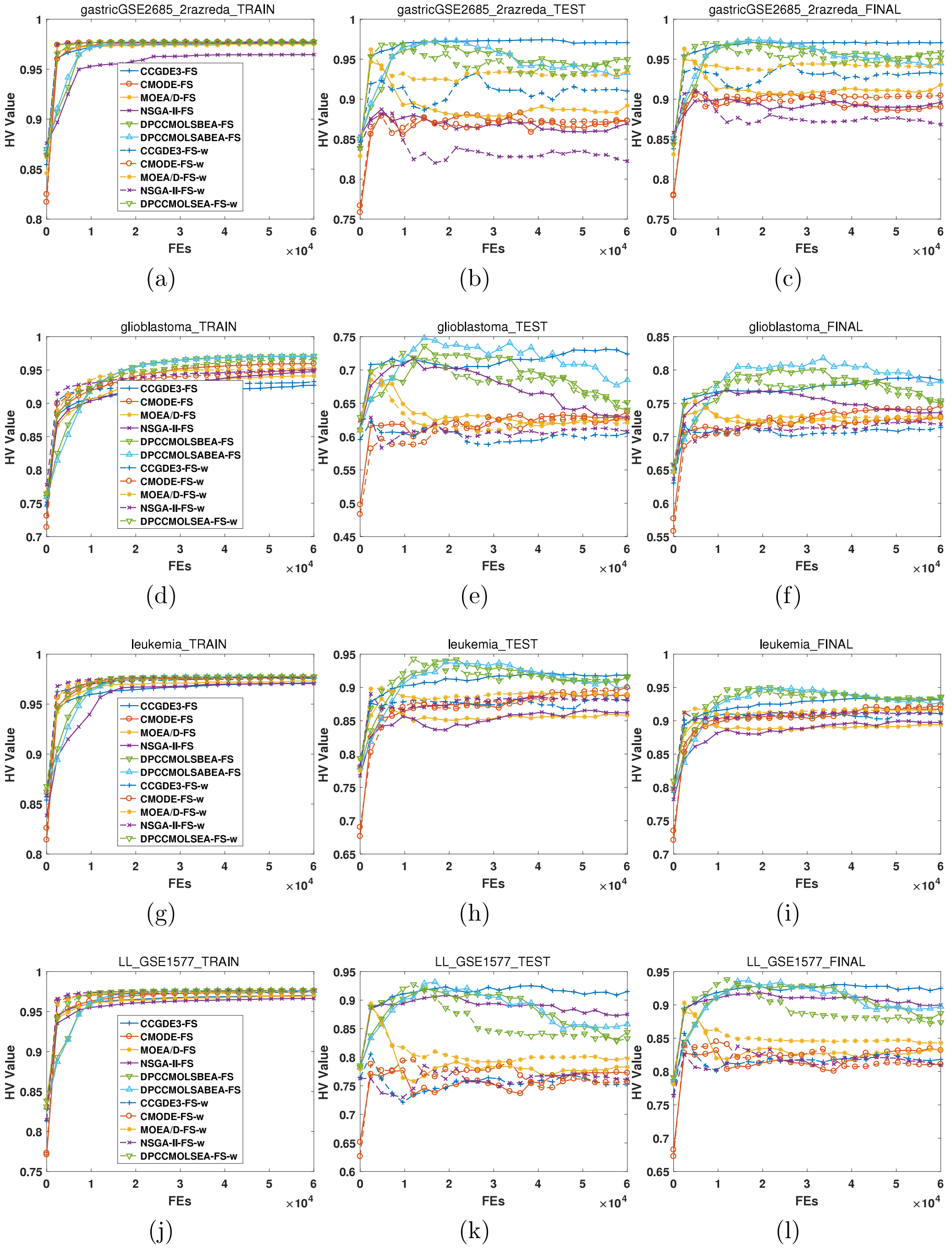
Fig. 3. Illustration of HV indicator values during evolution.

2. Several distributed algorithms are presented to address the multiobjective feature selection problem. Specifically, different encodings are considered, resulting in weight-encoded and binary-encoded algorithms with real and bin ary-encoded values, respectively. In addition, an adaptive improvement strategy is tested, yielding two binary-encoded algorithms.
3. By constraining the feature number, the time consumption is greatly reduced. Based on variable grouping and individual allocation, a two-layer distributed parallel structure is constructed. With this structure, a large number of CPU cores can be used to perform the individual evaluation in parallel, significantly reducing the time consumption. In addition, sample-wise parallelism is quite beneficial for reducing the time consumption of the recording process.

The remainder of this paper is organized as follows. Section 2 introduces the multiobjective feature selection model. The proposed algorithms are detailed in Section 3. The experimental analysis follows in Section 4. Discussion and future work are provided in Section 5. Finally, we conclude this paper in Section 6.

2. Multiobjective feature selection model

For the feature selection problem, we simultaneously consider three objectives, namely, classification error, feature number, and



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 4. Illustration of HV indicator values during evolution.

the redundancy among features, with respect to the generated feature subset. These objectives will be detailed in the following subsections.

2.1. Classification error

The classification error possesses the utmost importance, and it can be formulated as follows:

$$f_E = \frac{N_N}{N_N + N_P} \tag{1}$$

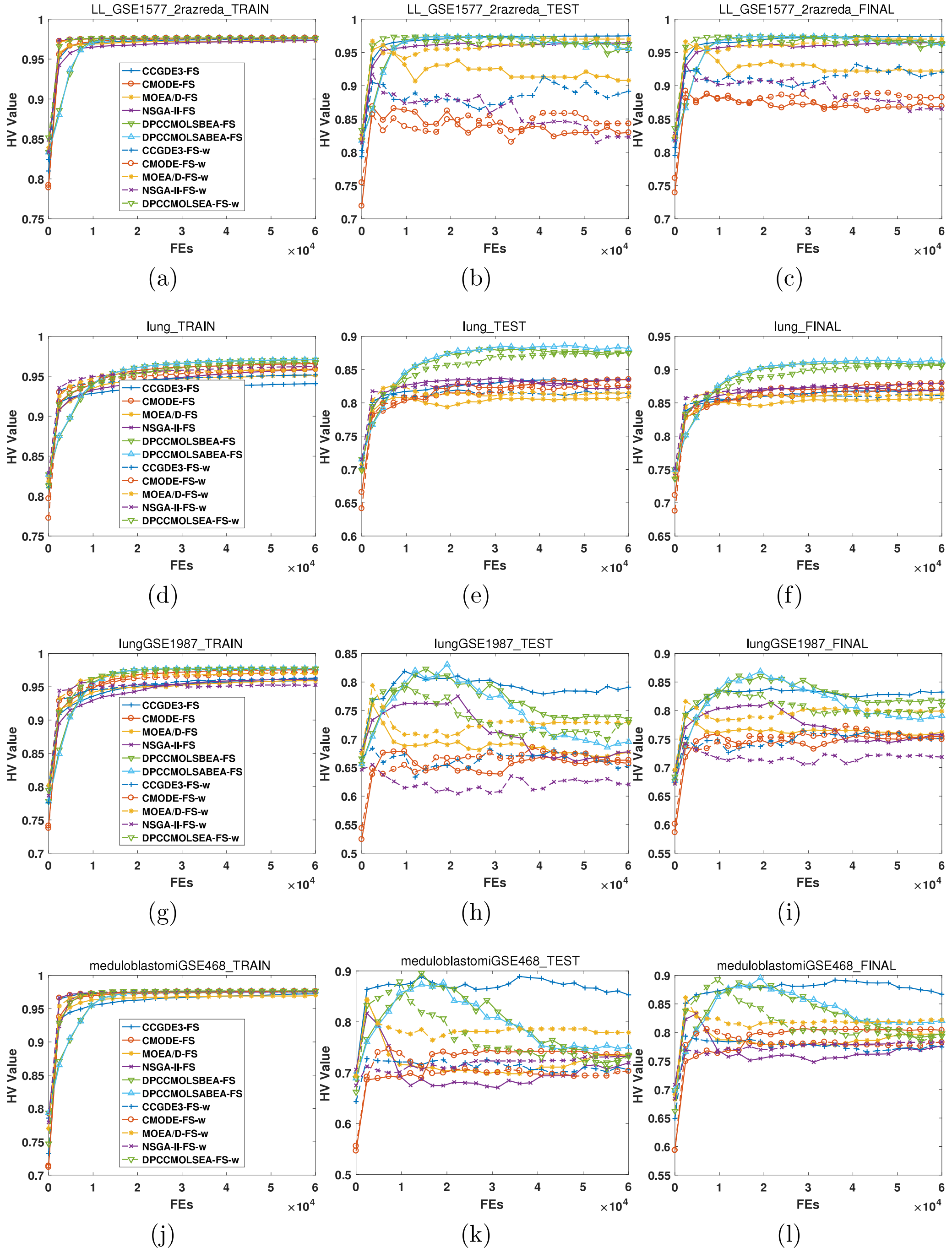
where f_E represents the fitness value of the classification error objective, and N_N and N_P denote the numbers of misclassified and correctly classified samples, respectively.

2.2. Feature number

This objective refers to the number of features in the generated feature subset, as illustrated in the following:

$$f_N = \frac{N_f}{N_F^{th}} \tag{2}$$

where f_N denotes the fitness value of the feature number objective and N_f and N_F^{th} denote, respectively, the feature number



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 5. Illustration of HV indicator values during evolution.

in the generated feature subset and the maximum number of features allowed in a feature subset, thus, $N_f \leq N_F^{th}$.

2.3. Feature redundancy

We utilize the Pearson correlation coefficient, as follows, to measure the correlation between two features:

$$r\left(F_\alpha, F_\beta\right) = \left| \frac{\sum_{i=1}^{N_S} \left(F_\alpha(i) - \overline{F_\alpha}\right)\left(F_\beta(i) - \overline{F_\beta}\right)}{\sqrt{\sum_{i=1}^{N_S}\left(F_\alpha(i) - \overline{F_\alpha}\right)^2}\sqrt{\sum_{i=1}^{N_S}\left(F_\beta(i) - \overline{F_\beta}\right)^2}} \right| \qquad (3)$$

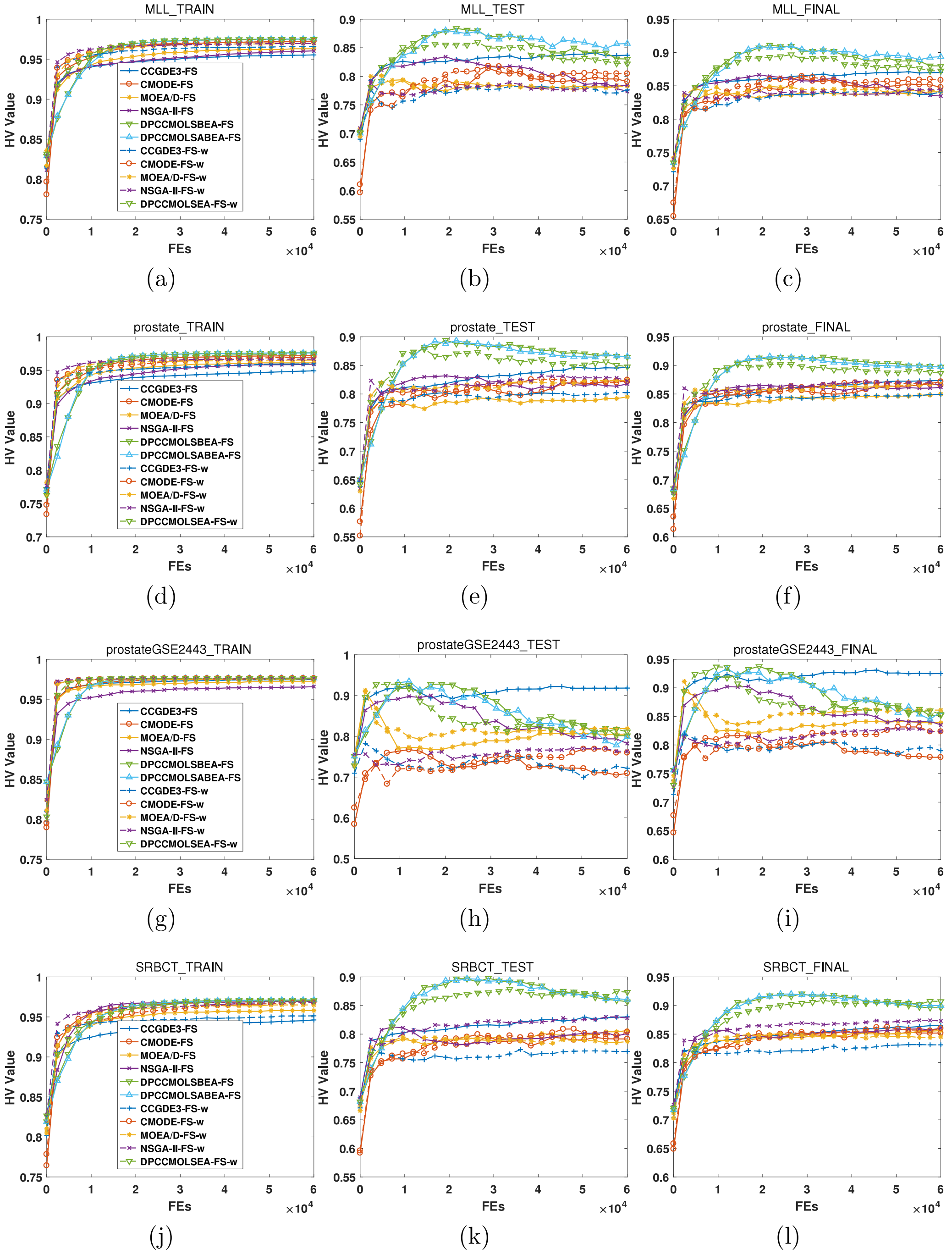
where N_S is the number of training samples, α and β denote the features, $F_\alpha(i)$ $(F_\beta(i))$ represents the value of feature α (β) for the ith sample, $\overline{F_\alpha}$ $(\overline{F_\beta})$ is the average value of feature α (β) over all samples, and $r\left(F_\alpha, F_\beta\right)$ is the correlation between features α and β, which has a positive value in the range of [0.0, 1.0].

Furthermore, the objective value related to the feature redundancy, f_R, is formulated as follows:

$$f_R = \frac{1}{N_f\left(N_f - 1\right)/2} \sum_{\alpha,\beta\in S_f,\ \alpha\neq\beta} r\left(F_\alpha, F_\beta\right) \qquad (4)$$

where S_f denotes the set of selected features and $N_f = \left|S_f\right|$ represents the cardinality of that set. If the generated feature subset contains only one feature, the following formula is applied:

$$f_R = \frac{1}{N_F - 1} \sum_{\beta\neq\alpha} r\left(F_\alpha, F_\beta\right) \qquad (5)$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 6. Illustration of HV indicator values during evolution.

where N_F denotes the total number of features, α represents the selected feature, and β is another feature chosen from among the remaining ones.

2.4. Multiobjective formulation

For an MOEA, the optimization target can be summarized as follows:

$$\min\,(f_E, f_N, f_R) \tag{6}$$

From the above specifications of the three objectives, it is easy to comprehend that their values all lie in the range of [0.0, 1.0]. Our aim is to minimize them simultaneously to improve the feature selection performance.

3. Proposed algorithms

In this section, we introduce the proposed algorithms. First, we illustrate the overall algorithm framework. Subsequently, based on this framework, by utilizing different encoding methods and an additional adaptive strategy, we derive each of the proposed algorithms one by one. Finally, the feature number constraint and the details of the parallel implementation are presented.

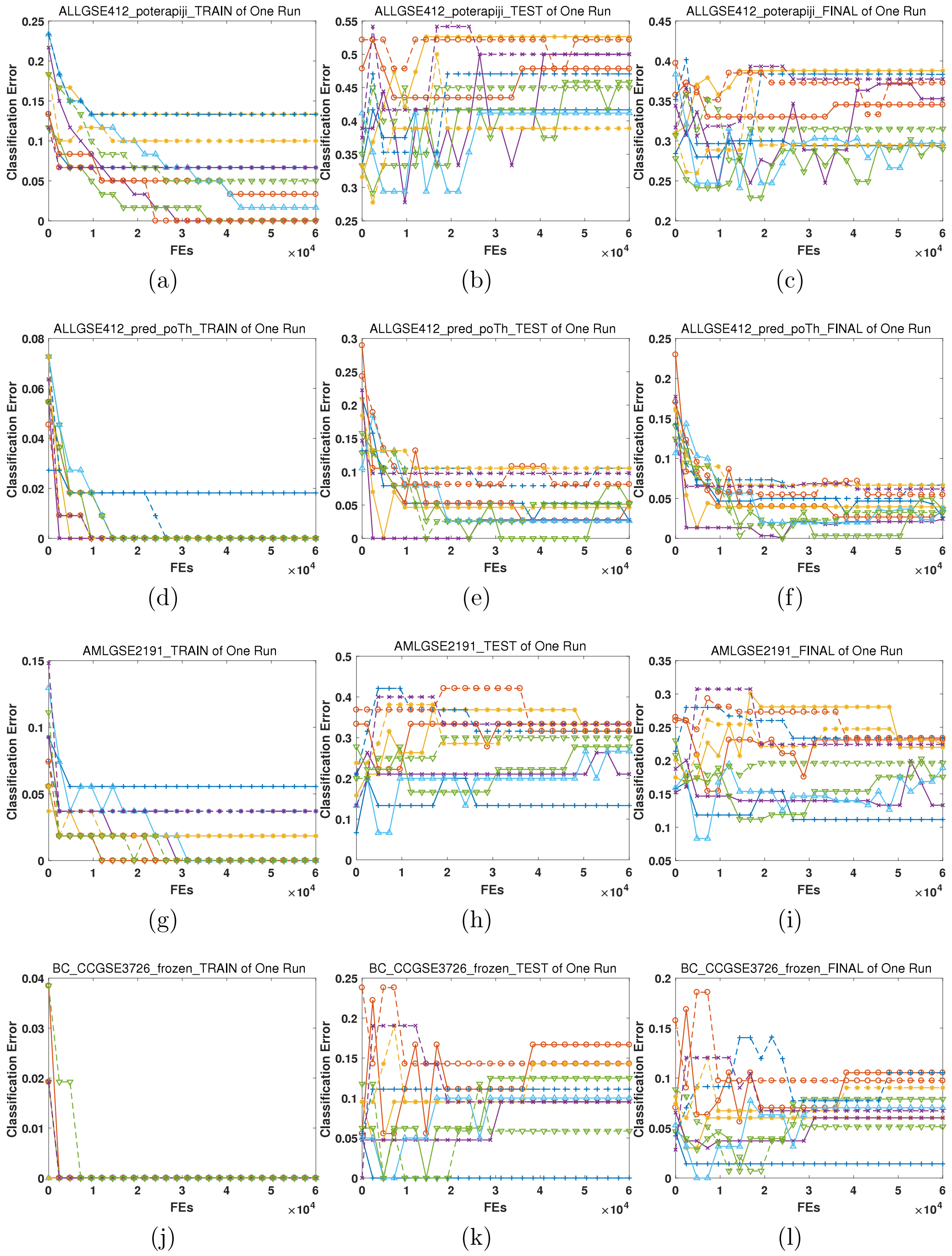

ALLGSE412_poterapiji_TRAIN of One Run
(a)
ALLGSE412_poterapiji_TEST of One Run
(b)
ALLGSE412_poterapiji_FINAL of One Run
(c)
ALLGSE412_pred_poTh_TRAIN of One Run
(d)
ALLGSE412_pred_poTh_TEST of One Run
(e)
ALLGSE412_pred_poTh_FINAL of One Run
(f)
AMLGSE2191_TRAIN of One Run
(g)
AMLGSE2191_TEST of One Run
(h)
AMLGSE2191_FINAL of One Run
(i)
BC_CCGSE3726_frozen_TRAIN of One Run
(j)
BC_CCGSE3726_frozen_TEST of One Run
(k)
BC_CCGSE3726_frozen_FINAL of One Run
(l)

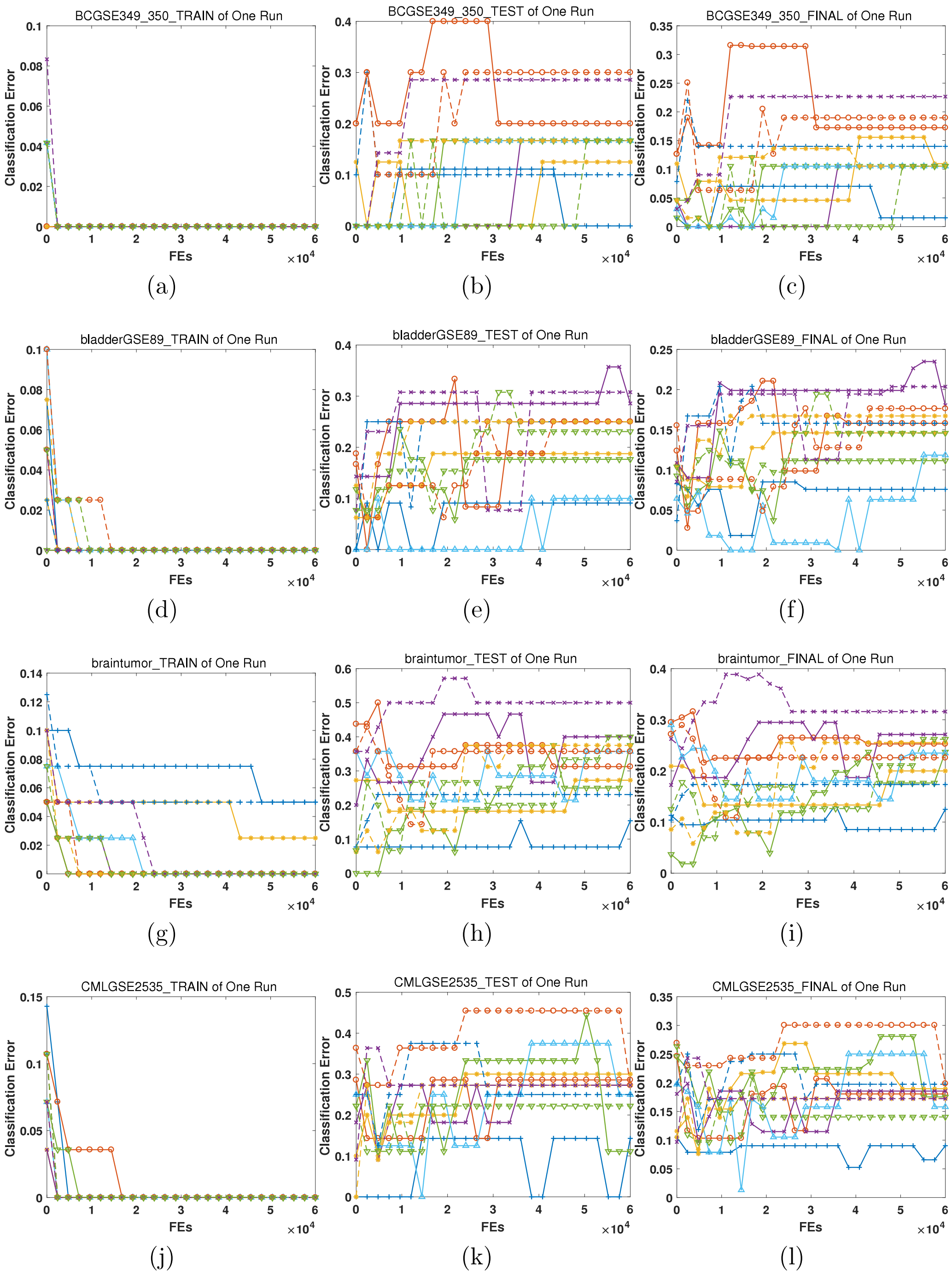
Fig. 7. Illustration of classification error during evolution.

3.1. Overall framework

3.1.1. Grouping

Microarray datasets have numerous features, each of which is encoded by one variable; thus, the dimensionality of the multiobjective feature selection problem will be quite high. For high-dimensional problems, simultaneously optimizing all variables will be ineffective; however, by separating variables into several groups and optimizing each group of variables under the CC framework [21], the original problem can be separated into several low-dimensional subproblems, which can be solved more effectively and efficiently. For this purpose, we randomly separate all variables into N_G uniform groups.

The proposed algorithms are based on the work done in our previous study [22], in which we proposed the distributed parallel cooperative coevolutionary multiobjective large-scale evolutionary algorithm (DPCCMOLSEA). There are two types of variables, namely, the variables in the group that is currently optimized and the variables in the other groups. The evolutionary strategies for these two types of variables are different, as detailed below.


BCGSE349_350_TRAIN of One Run

BCGSE349_350_TEST of One Run

BCGSE349_350_FINAL of One Run

(a)

(b)

(c)

bladderGSE89_TRAIN of One Run

bladderGSE89_TEST of One Run

bladderGSE89_FINAL of One Run

(d)

(e)

(f)

braintumor_TRAIN of One Run

braintumor_TEST of One Run

braintumor_FINAL of One Run

(g)

(h)

(i)

CMLGSE2535_TRAIN of One Run

CMLGSE2535_TEST of One Run

CMLGSE2535_FINAL of One Run

(j)

(k)

(l)

Fig. 8. Illustration of classification error during evolution.

3.1.2. Evolution of variables in the current group

For each parent individual, to generate the values of the variables in the current group for an offspring individual, either another single individual (for a binary-encoded algorithm) or two individuals (for the weight-encoded algorithm) are selected from the parent population. Then, crossover or adaptive differential evolution (DE) [23,24] is applied, as detailed in Sections 3.2.1 and 3.3.1.

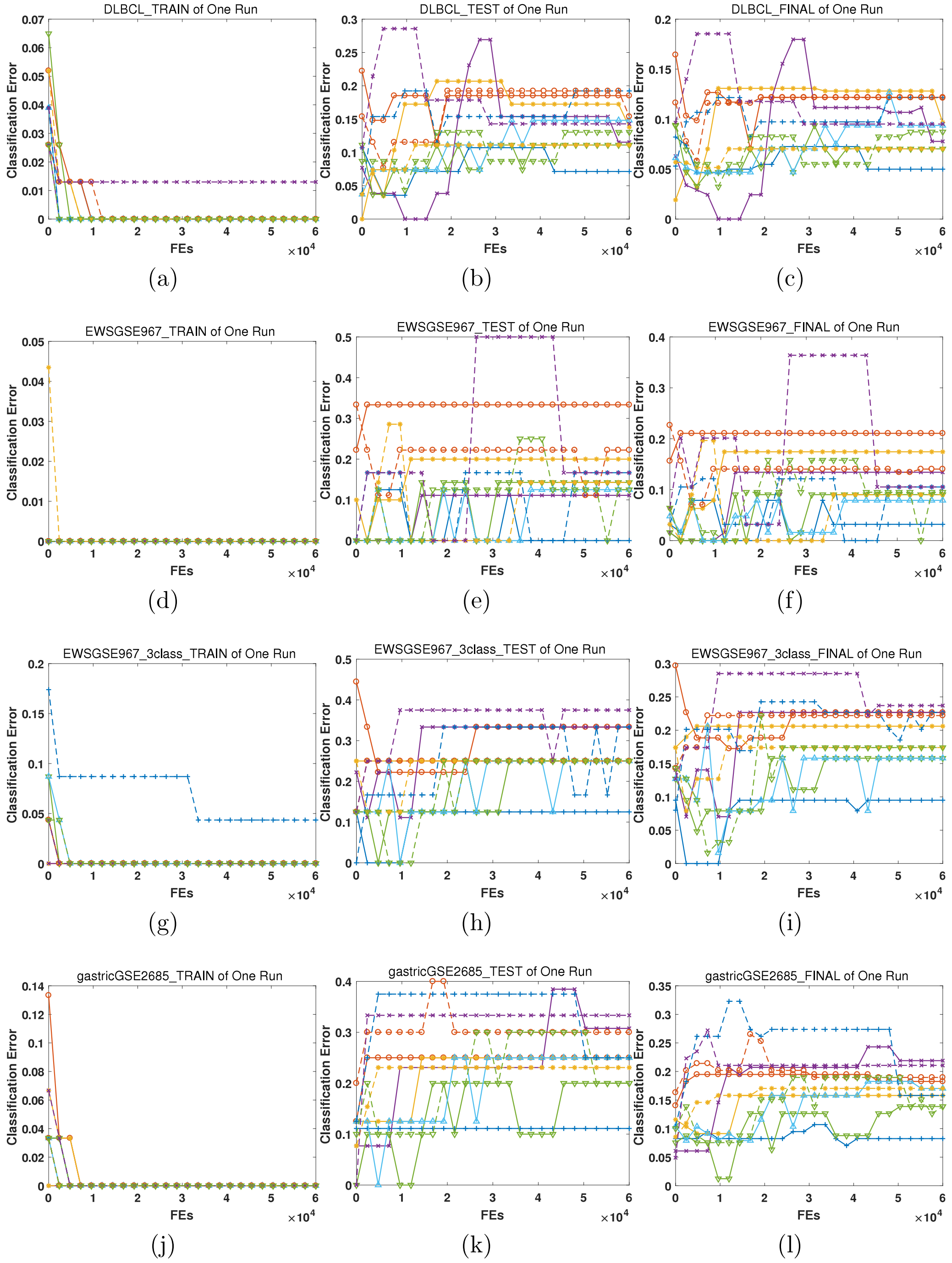
3.1.3. Evolution of variables in the other groups

As in DPCCMOLSEA [22], to obtain the values of the remaining variables (not in the current group) for an offspring individual, the crossover operation is applied to the parent individual and two additional randomly selected individuals in the parent population. The crossover rate is determined using the adaptive strategy applied in JADE [25], which can be described as follows:

$$u_j^{g+1,i} = \begin{cases} x_j^{g,i}, & \text{if } r \leq CR^i \\ x_j^{g,r_1}, & \text{else if } r_0 \leq 0.5 \\ x_j^{g,r_2}, & \text{otherwise} \end{cases} \tag{7}$$

s.t. $j \notin S_k^G$

where g, i and j denote the generation number, the individual index and the variable index, respectively, thus, $x_j^{g,i}$ represents variable j of individual i in the population of generation g, u is the trail vector, r and r_0 denote two random numbers uniformly



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 9. Illustration of classification error during evolution.

generated in the range of [0.0, 1.0), $r_1 \neq r_2 \neq i$ denote another two individuals selected in the parent population, S_k^G is the set of variables in the currently considered group k, and the crossover rate, CR^i, is formulated as follows:

$$CR^i = Gauss\left(\mu_{CR}, 0.1\right) \tag{8}$$

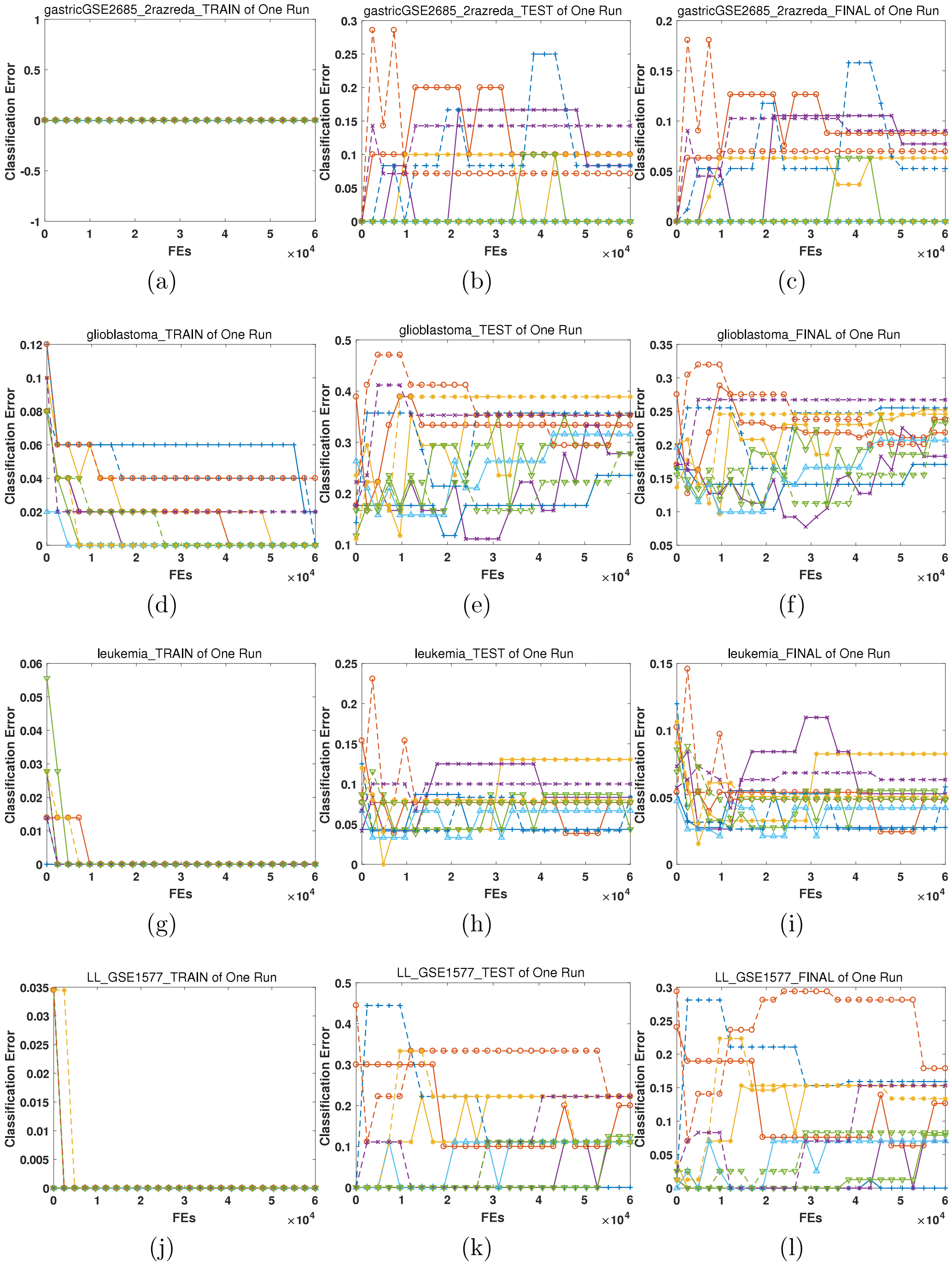
where $Gauss\left(\mu, \sigma\right)$ generates a random value in accordance with a Gaussian distribution with a location parameter of μ and a scale parameter of σ, CR^i is restricted to the range of [0.0, 1.0], thus, it will be truncated to 0.0 or 1.0 if $F^i < 0.0$ or if $F^i > 1.0$, respectively. Initially set to 0.5, the value of μ_{CR} is updated as follows:

$$\mu_{CR} = (1.0 - c) \times \mu_{CR} + c \times \frac{\sum_{i \in S_{CR}} CR^i}{|S_{CR}|} \tag{9}$$

where $c = 0.1$ represents the learning factor and S_{CR} and $|S_{CR}|$ denote the set of indices of the individuals that were successfully updated in the previous generation and the cardinality of this set, respectively.

3.1.4. *Mutation*

After the generation of the values of variables in both the current group and the other groups, an offspring individual is preliminarily generated. To increase the probability of escaping from local optima, mutation is then performed. The details are presented in Sections 3.2.2 and 3.3.2.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 10. Illustration of classification error during evolution.

3.2. Weight-encoded algorithm

By encoding each feature via a weight value, we derive the distributed parallel cooperative coevolutionary multiobjective large-scale evolutionary algorithm for feature selection with weight encoding, abbreviated as DPCCMOLSEA-FS-w. In this algorithm, the selection priority of a feature is represented by its weight; in other words, the higher the weight is, the more likely the feature is to be selected.

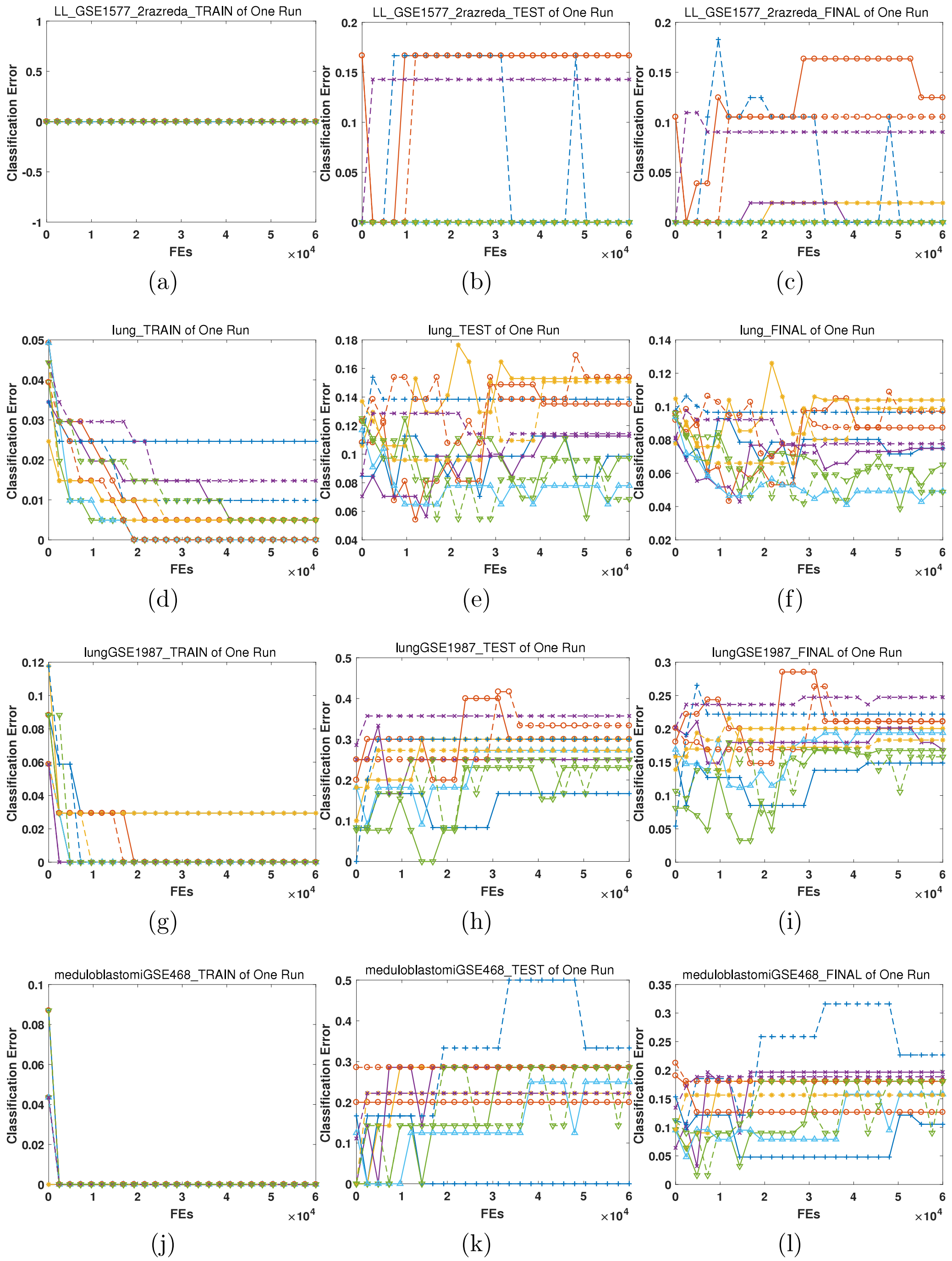
To this end, each gene (variable) is treated as a weight value for the corresponding feature. Additionally, another variable is employed to control the number of features. We can illustrate the encoding of each individual as follows:

$$\underbrace{x_0, \ldots, x_{N_F-1},}_{\text{Feature weights}} \quad \underbrace{x_{N_F}}_{\text{Feature number}} \tag{10}$$

s.t. $x_j \in [0.0, 1.0], j = 0, \ldots, N_F.$

where x_0 to x_{N_F-1} encode the selection weights of all N_F features and x_{N_F} controls the feature number. Specifically, the number of selected features is $N_f = x_{N_F} \times N_F^{th} + 1$ (will be truncated to N_F^{th} if $N_f > N_F^{th}$), which is an integer from 1 to N_F^{th}.

Then, to form the feature subset, the top N_f features with higher weights are selected.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 11. Illustration of classification error during evolution.

3.2.1. *Evolution*

As mentioned in Section 3.1.2, to evolve the variables in the currently considered group, we use an adaptive strategy similar to that of JADE [25], which is formulated as follows:

$$u_j^{g+1,i} = x_j^{g,i} + F^i \times \left(x_j^{g,r_3} - x_j^{g,r_4}\right) \quad \text{s.t. } j \in S_k^G \tag{11}$$

where $r_3 \neq r_4 \neq i$ are two randomly selected individuals and F^i denotes the scaling factor for individual i. Similar to the scaling factor in JADE [25], F^i has the following form:

$$F^i = Cauchy\left(\mu_F, 0.1\right) \tag{12}$$

where $Cauchy\left(\mu, \sigma\right)$ generates a random value in accordance with a Cauchy distribution with a location parameter of μ and a scale parameter of σ, here, F^i is restricted to the range of (0.0, 1.0], thus, a new value will be generated if $F^i \leq 0.0$, and the value will be truncated to 1.0 if $F^i > 1.0$. Initially, $\mu_F = 0.5$, and it is then updated as follows:

$$\mu_F = (1.0 - c) \times \mu_F + c \times \frac{\sum_{i \in S_F} \left(F^i\right)^2}{\sum_{i \in S_F} F^i} \tag{13}$$

where $c = 0.1$ is the learning factor and S_F represents the set of indices of the individuals that were successfully updated in the previous generation.

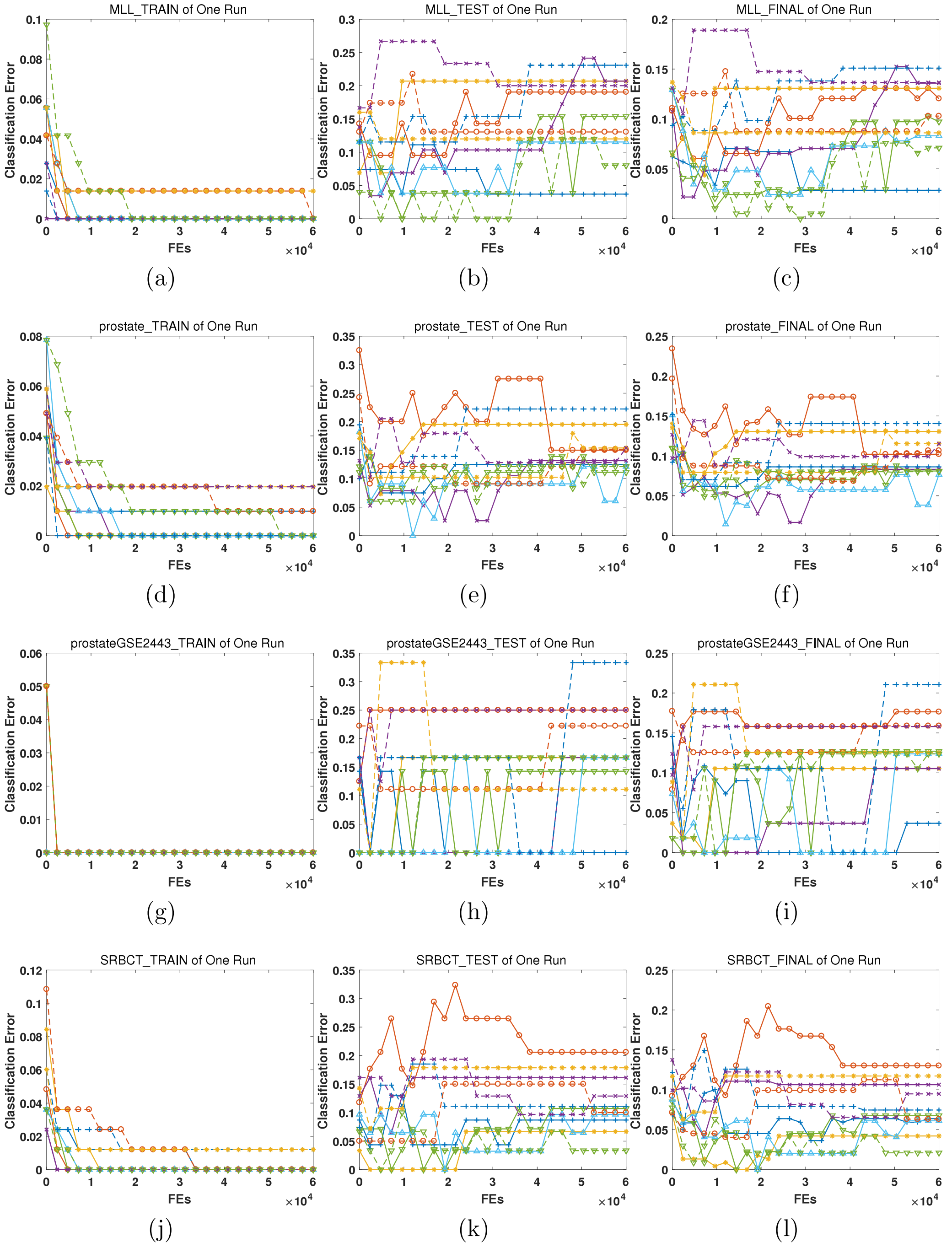


Fig. 12. Illustration of classification error during evolution.

3.2.2. Mutation

Polynomial mutation (PM) [26] is utilized to adjust the variable values with a probability of $p_m = \frac{1}{nDim}$, here, $nDim = N_F + 1$ is the dimensionality of the feature selection problem. The formula is as follows:

$$x^{g+1,i} = PM\left(u^{g+1,i}, p_m\right) \tag{14}$$

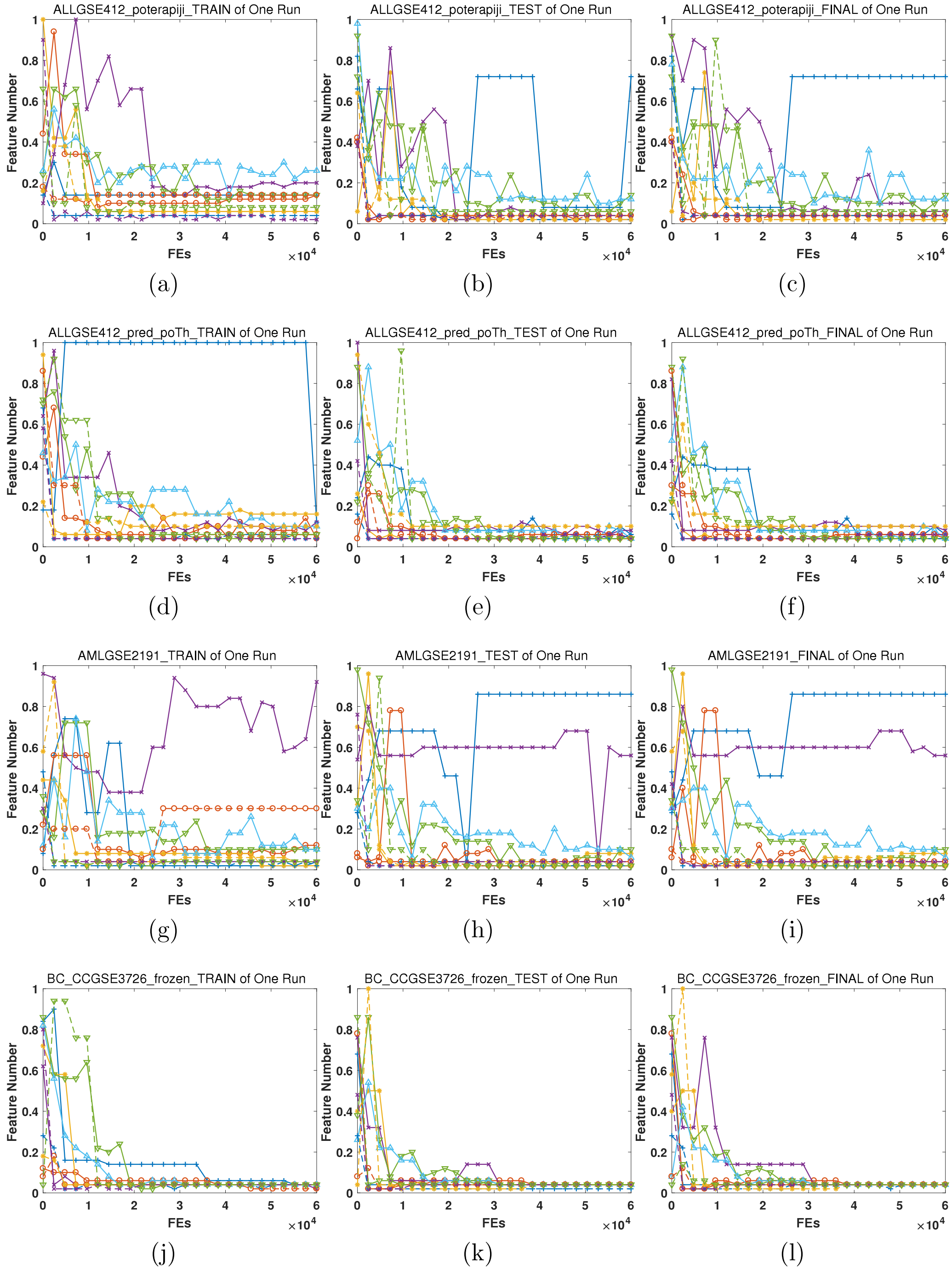
Finally, each offspring individual, $x^{g+1,i}$, $i = 0, \ldots, NP - 1$, will be generated, here, NP is the population size.

3.3. Binary-encoded algorithms

We propose two different binary-encoded algorithms, namely, the distributed parallel cooperative coevolutionary multiobjective large-scale binary evolutionary algorithm for feature selection (DPCCMOLSBEA-FS) and the distributed parallel cooperative coevolutionary multiobjective large-scale adaptive binary evolutionary algorithm for feature selection (DPCCMOLSABEA-FS). In these two algorithms, each feature is represented by a binary value, 1 or 0, which indicates whether the corresponding feature is selected as a part of the feature subset, and there is no additional variable to encode the feature number. The encoding is as follows:

$$\underbrace{x_0, \ldots, x_{N_F-1}}_{\text{Binary encoding}} \tag{15}$$

$$\text{s.t. } x_j \in \{0, 1\}, j = 0, \ldots, N_F - 1$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 13. Illustration of feature number during evolution.

3.3.1. Evolution

The process of evolving the variables in the currently considered group as mentioned in Section 3.1.2, for both DPCCMOLSBEA-FS and DPCCMOLSABEA-FS, can be expressed as follows:

$$u_j^{g+1,i} = \begin{cases} x_j^{g,r_5}, & \text{if } r \le CR_B^i \\ x_j^{g,i}, & \text{otherwise} \end{cases} \tag{16}$$

where $r_5 \neq i$ denotes a randomly selected individual in the parent population, r is a random number uniformly generated in the range of [0.0, 1.0), and CR_B^i represents the crossover rate for individual i. The difference between DPCCMOLSBEA-FS and DPCCMOLSABEA-FS lies in the value of CR_B^i:

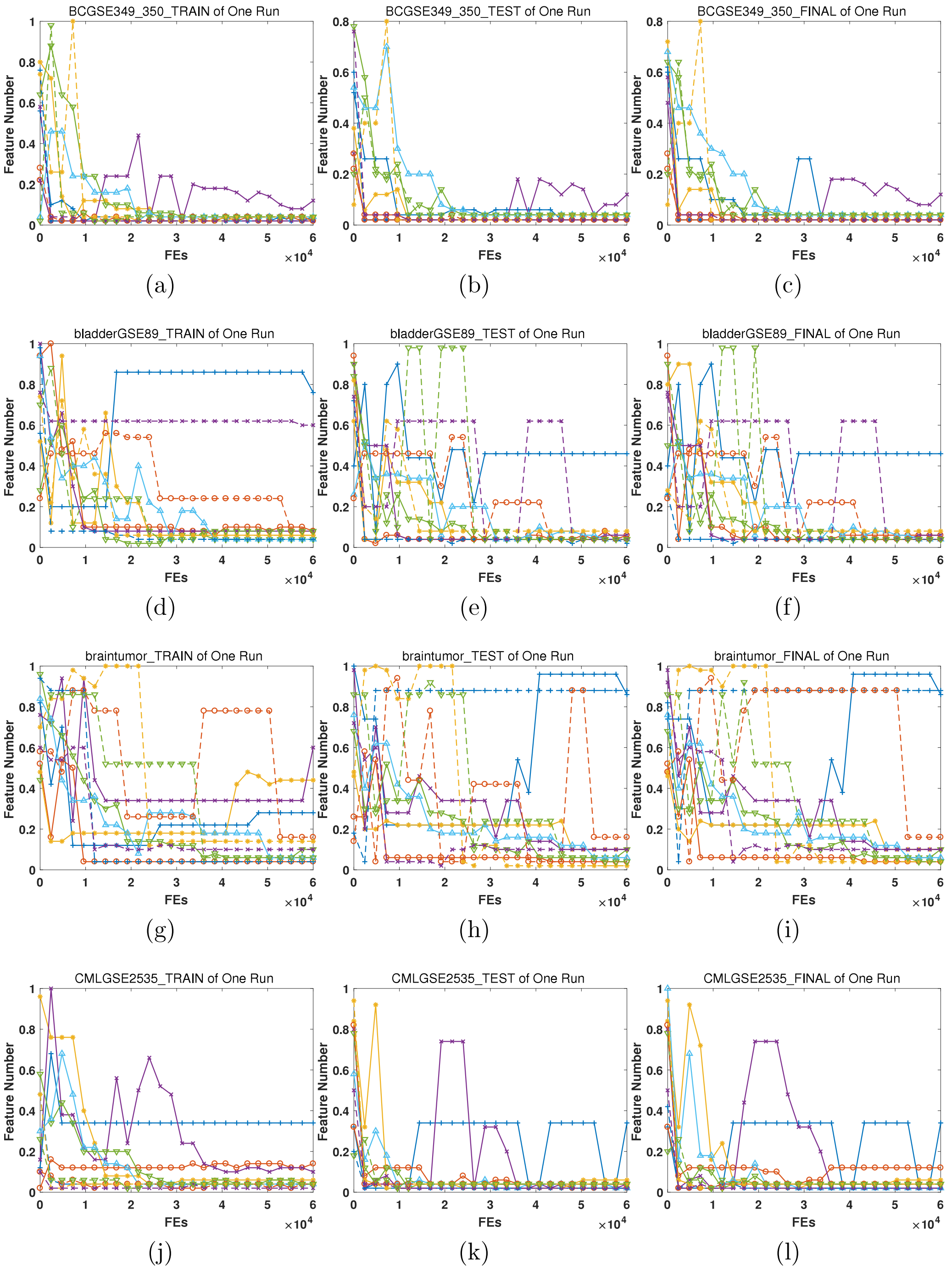
1. For DPCCMOLSBEA-FS, $CR_B^i = 1.0$ for all $i = 0, \ldots, NP - 1$.
2. For DPCCMOLSABEA-FS, CR_B^i is generated adaptively, similar to CR^i, as follows:

$$CR_B^i = Gauss\left(\mu_{CR}^B, 0.1\right) \tag{17}$$

where CR_B^i is truncated to [0.0, 1.0], and the value of μ_{CR}^B is initially set to 0.9 and is then updated as follows:

$$\mu_{CR}^B = (1.0 - c) \times \mu_{CR}^B + c \times \frac{\sum_{i \in S_{CR_B}} CR_B^i}{\left|S_{CR_B}\right|} \tag{18}$$

where $c = 0.1$ denotes the learning factor and S_{CR_B} and $\left|S_{CR_B}\right|$ represent the set of indices of individuals that were



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 14. Illustration of feature number during evolution.

successfully updated in the previous generation and the cardinality of this set, respectively.

3.3.2. Mutation

To mutate a preliminarily generated offspring individual, we apply the following formula:

$$x_j^{g+1,i} = \begin{cases} 1 - u_j^{g+1,i}, & \text{if } r \le p_m \\ u_j^{g+1,i}, & \text{otherwise} \end{cases} \tag{19}$$

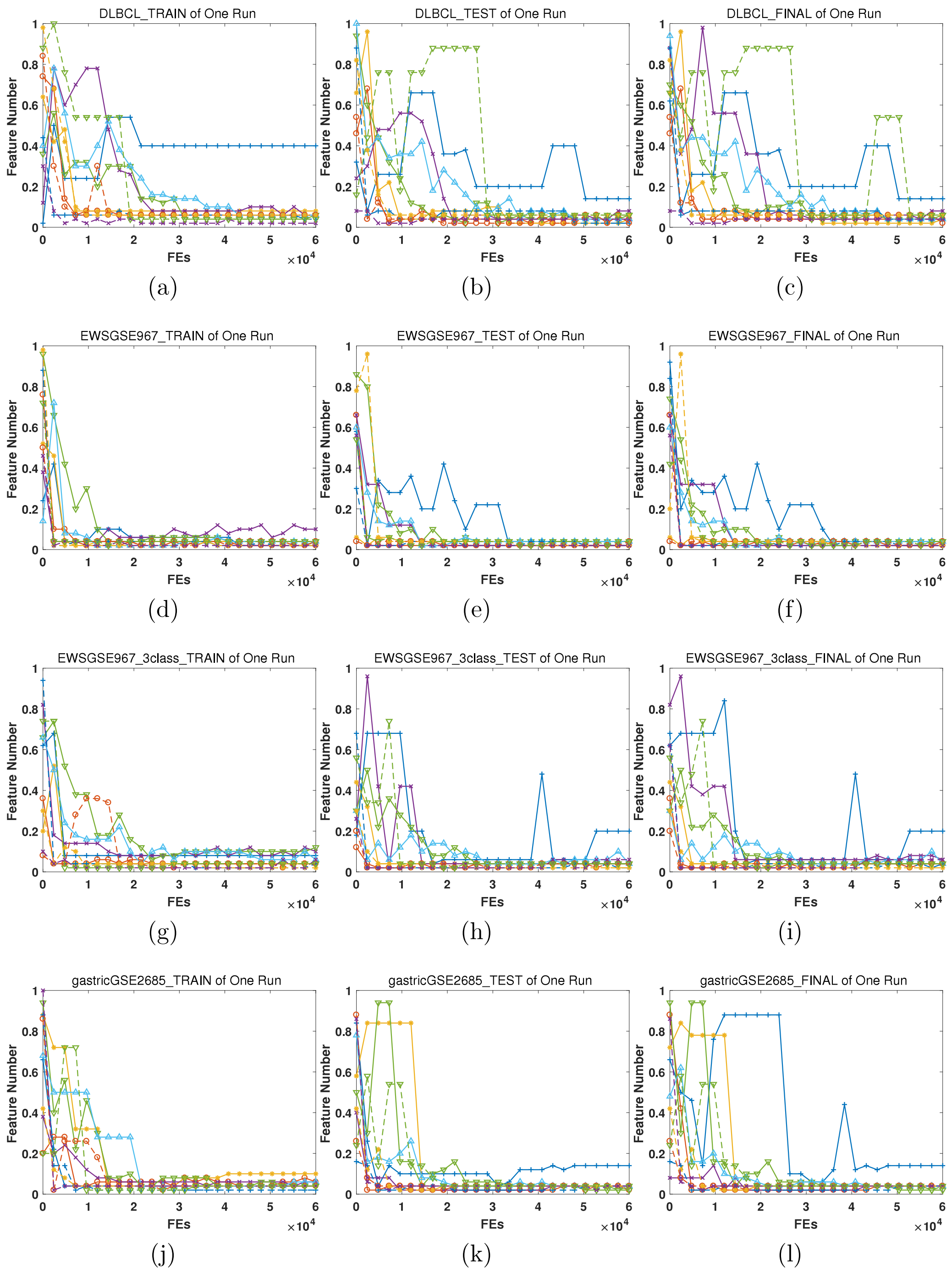
where r is a random number uniformly generated within the range of [0.0, 1.0), and, as mentioned in Section 3.2.2, $p_m = \frac{1}{nDim}$ is the mutation probability, here, $nDim = N_F$.

3.3.3. Feature adjustment

In the feature number objective (Eq. (2)), there is a constraint that ensures that at most N_F^{th} features can be selected as part of the feature subset. However, for the offspring generated by this type of binary-encoded algorithm, the cardinality of the corresponding feature subset can exceed N_F^{th} or be less than 1; thus, an adjustment procedure must be applied.

The adjustment procedure consists of two phases, as follows:

1. Reset the feature number: set the number of features, N_f, to a random integer in the range of $\left[1, N_F^{th}\right]$.
2. Randomly add or remove features: if the cardinality of the original feature subset is greater than N_F^{th}, randomly



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

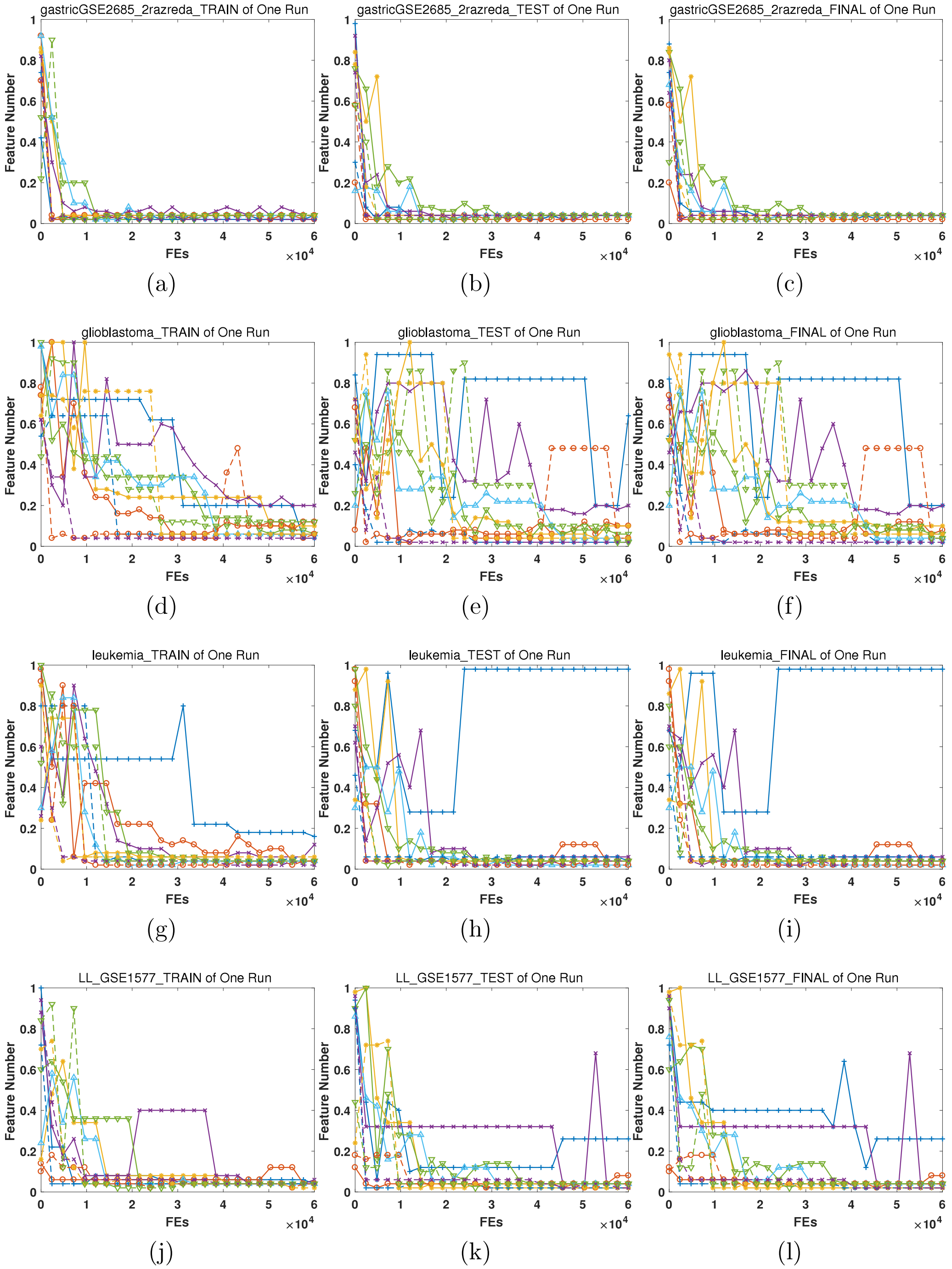
Fig. 15. Illustration of feature number during evolution.

remove $N_F^{th} - N_f$ different features by setting the corresponding variable values to 0; otherwise, randomly add N_f different features by setting the corresponding variable values to 1.

3.4. Feature number constraint and parallel implementation

MOEAs function on the basis of population generation and iteration. During the evolutionary process, numerous generations of populations are produced, and a large number of fitness evaluations (FEs) are performed. To reduce the time consumption of our algorithms, three strategies are applied:

(A) Feature number constraint: From the objective functions (Section 2), it is clear that the time consumption depends on the cardinality of the feature subset and that the classification error objective is the most time-consuming one to address; compared to obtaining this objective, the process of evolving the population and calculating the other objectives are very efficient. Thus, the classification error objective is considered to be the only time-consuming component of the analysis. In this study, the nearest neighbor classifier (1-NN) is employed; thus, the time consumption is proportional to the number of selected features. For microarray data as considered here, the number of features can reach more than tens of thousands; in such a case, the cardinality of the feature subset can be quite high,



gastricGSE2685_2razreda_TRAIN of One Run

(a)

gastricGSE2685_2razreda_TEST of One Run

(b)

gastricGSE2685_2razreda_FINAL of One Run

(c)

glioblastoma_TRAIN of One Run

(d)

glioblastoma_TEST of One Run

(e)

glioblastoma_FINAL of One Run

(f)

leukemia_TRAIN of One Run

(g)

leukemia_TEST of One Run

(h)

leukemia_FINAL of One Run

(i)

LL_GSE1577_TRAIN of One Run

(j)

LL_GSE1577_TEST of One Run

(k)

LL_GSE1577_FINAL of One Run

(l)

Fig. 16. Illustration of feature number during evolution.

and the time consumption will be intolerable. However, with the application of a feature number constraint, only a small number of features will be considered in the objective evaluation; thus, the time consumption can be greatly reduced.

(B) Distributed parallelism: The strategy described above reduces the time consumption for each FE; however, the evaluations of all individuals in the offspring population are conducted in serial. By taking advantage of the variable grouping and a population-based evaluation strategy, we can construct the following distributed parallel structure:

(a) Suppose that there are N_C CPU cores and that the number of groups is N_G. For each group, we form a population of NP individuals, and we divide the CPU cores uniformly among these populations, as follows:

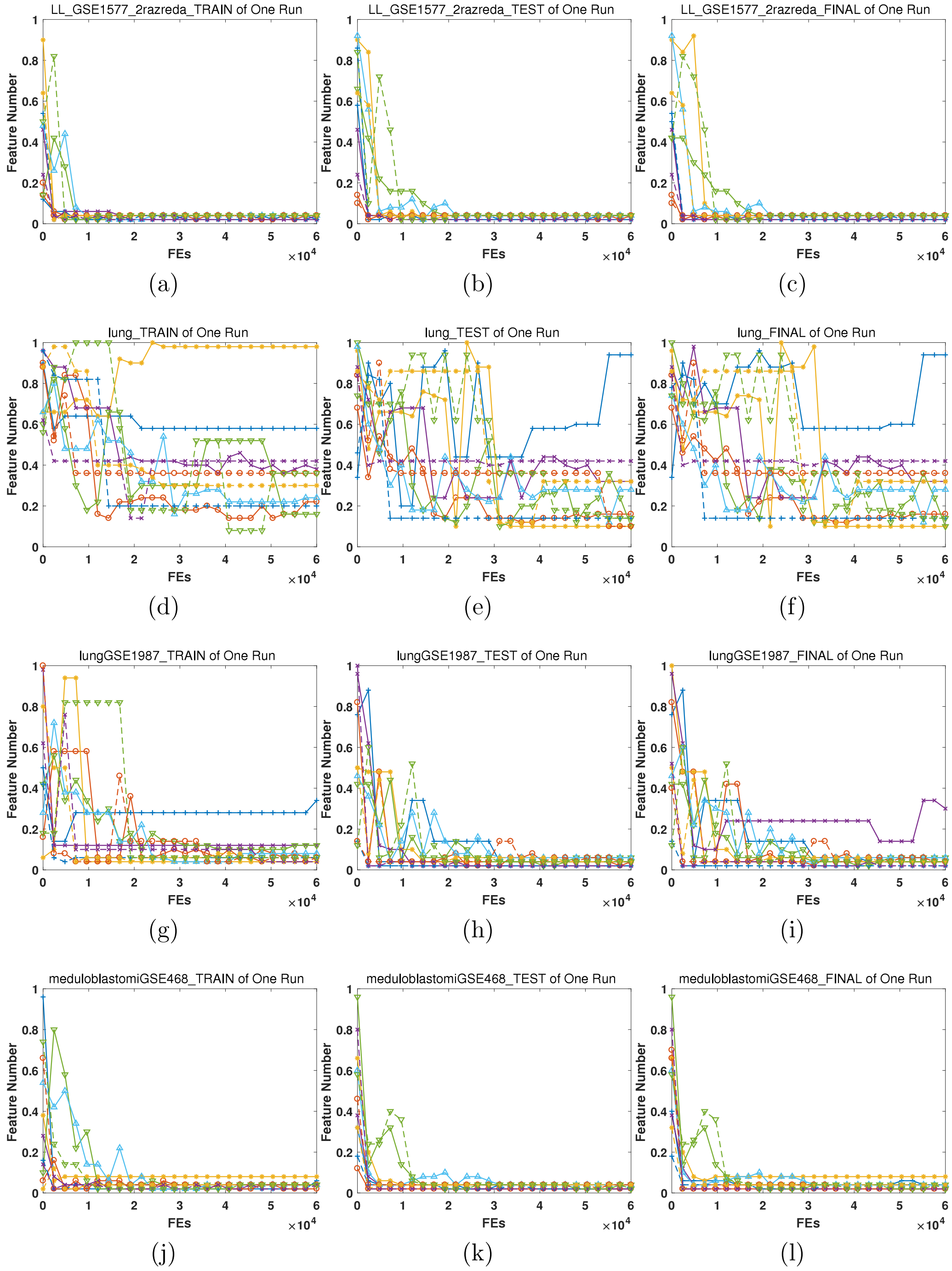
$$N_C^i = \frac{N_C}{N_G} \tag{20}$$

s.t. $i = 1, \ldots, N_G$.

where N_C^i denotes the number of CPU cores allocated to population i.

(b) Then, the individuals in each population are divided among the corresponding CPUs for FEs, as follows:

$$N_C^{i,j} = \frac{NP}{N_C^i} \tag{21}$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 17. Illustration of feature number during evolution.

s.t. $i = 1, \ldots, N_G, j = 1, \ldots, N_C^i$.

where $N_C^{i,j}$ denotes the number of individuals in the charge of CPU j in population i.

(c) In summary, for the evolution of the populations, all individuals are in the charge of one CPU in each population; thus, the evolution is parallel at the population level. For the time-consuming FE process, all CPUs are utilized — each CPU evaluates its allocated individuals, and all CPUs operate in parallel. Thus, the evaluation process is parallel at the individual level.

(C) Sample-wise parallelism: To observe the evolutionary behavior of an MOEA during the optimization process, every time the algorithm has proceeded through a predefined number of generations, we record the fitness values of the individuals in the current population. In addition, we test these individuals on the test set and record the results. Although the number of recordings is extremely small compared to the overall number of generations, if the evaluation on the test set were to be performed in serial, its time consumption would exceed that of the distributed parallel optimization process.

Therefore, we also parallelize this test procedure. Specifically, when evaluating an individual, the root CPU decodes

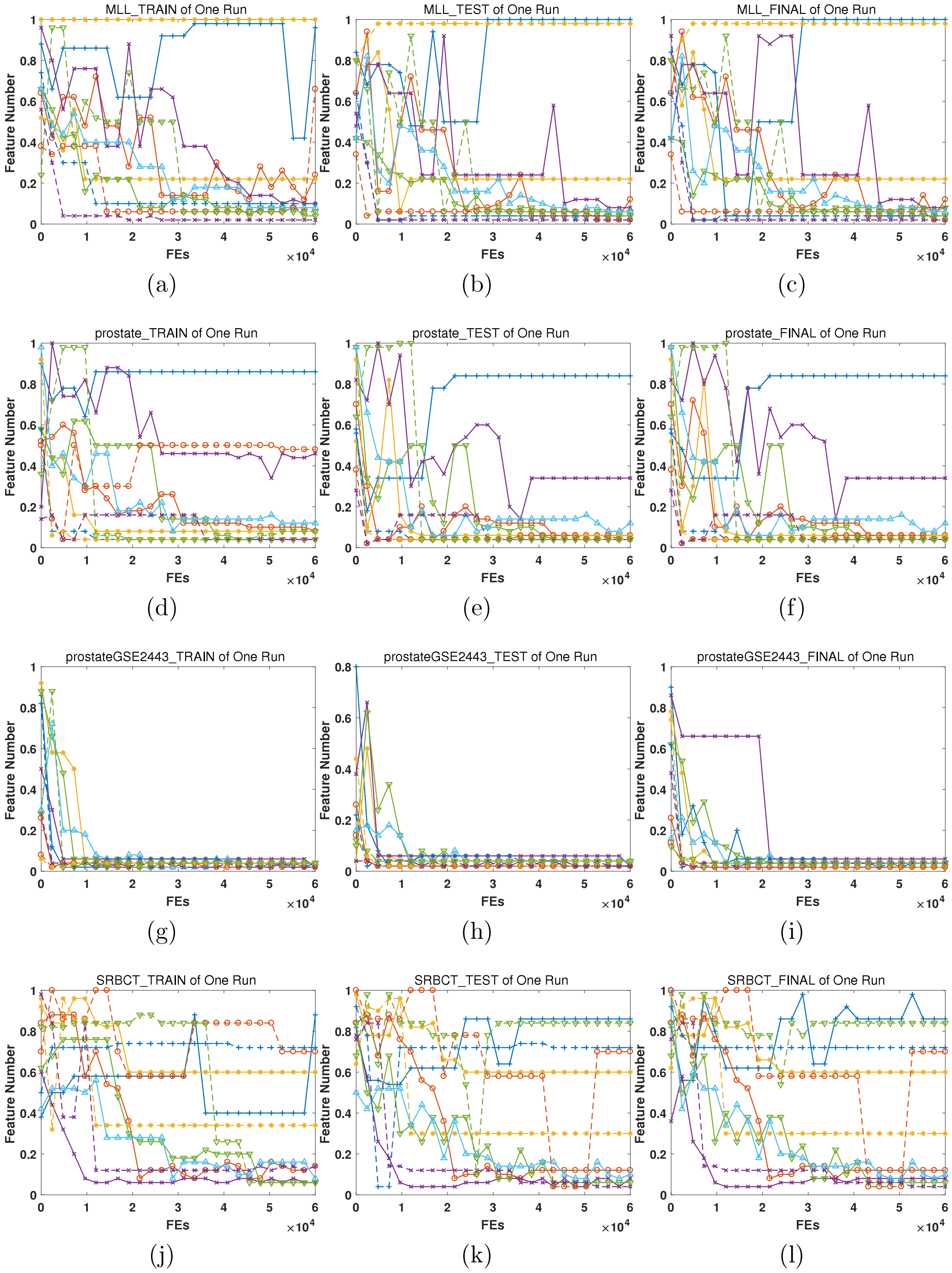


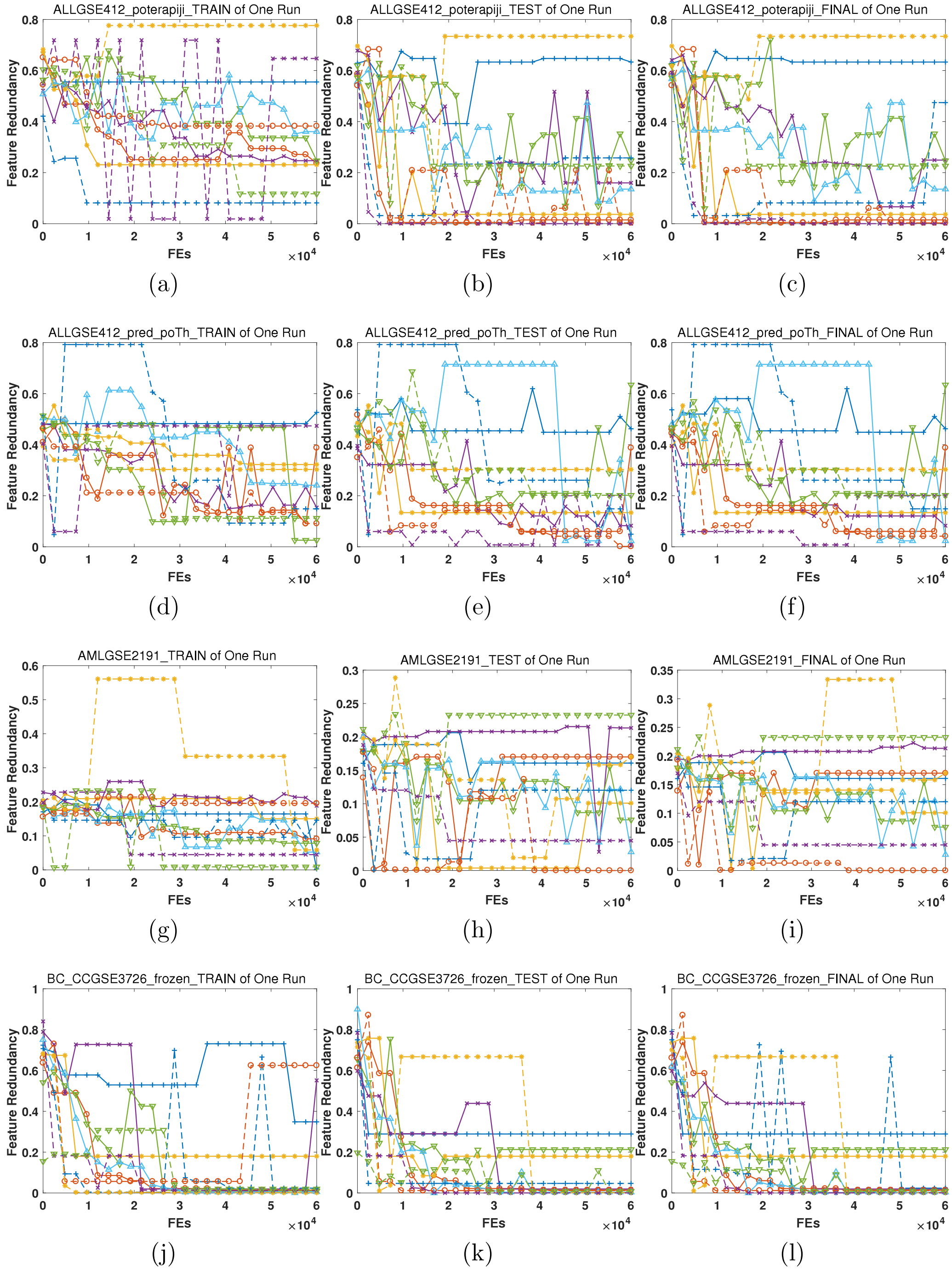
Fig. 18. Illustration of feature number during evolution.

this individual into a subset of features and broadcasts this subset to all other N_C CPUs. Thus, the classification burdens for all test samples are uniformly allocated to all CPUs, and all CPUs perform their own tasks in parallel. Finally, the root CPU gathers the classification results from all CPUs. With this parallelism, the overall time consumption is significantly reduced, such that the benefit achieved by paralleling the optimization process is not lost due to the recording process.

4. Experimental analysis

4.1. Microarray datasets

Only a few years ago, in the twentieth century, the study of genes was very low in efficiency, with only one or a few genes being checked at a time. However, living things possess substantial numbers of genes; for instance, humans have approximately 20,000 genes. Consequently, at the previous rate, the investigation process could take a scientist's lifetime. Fortunately, with the aid of microarray technology, the expression of numerous genes can be investigated at once. In cases where there seems to



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

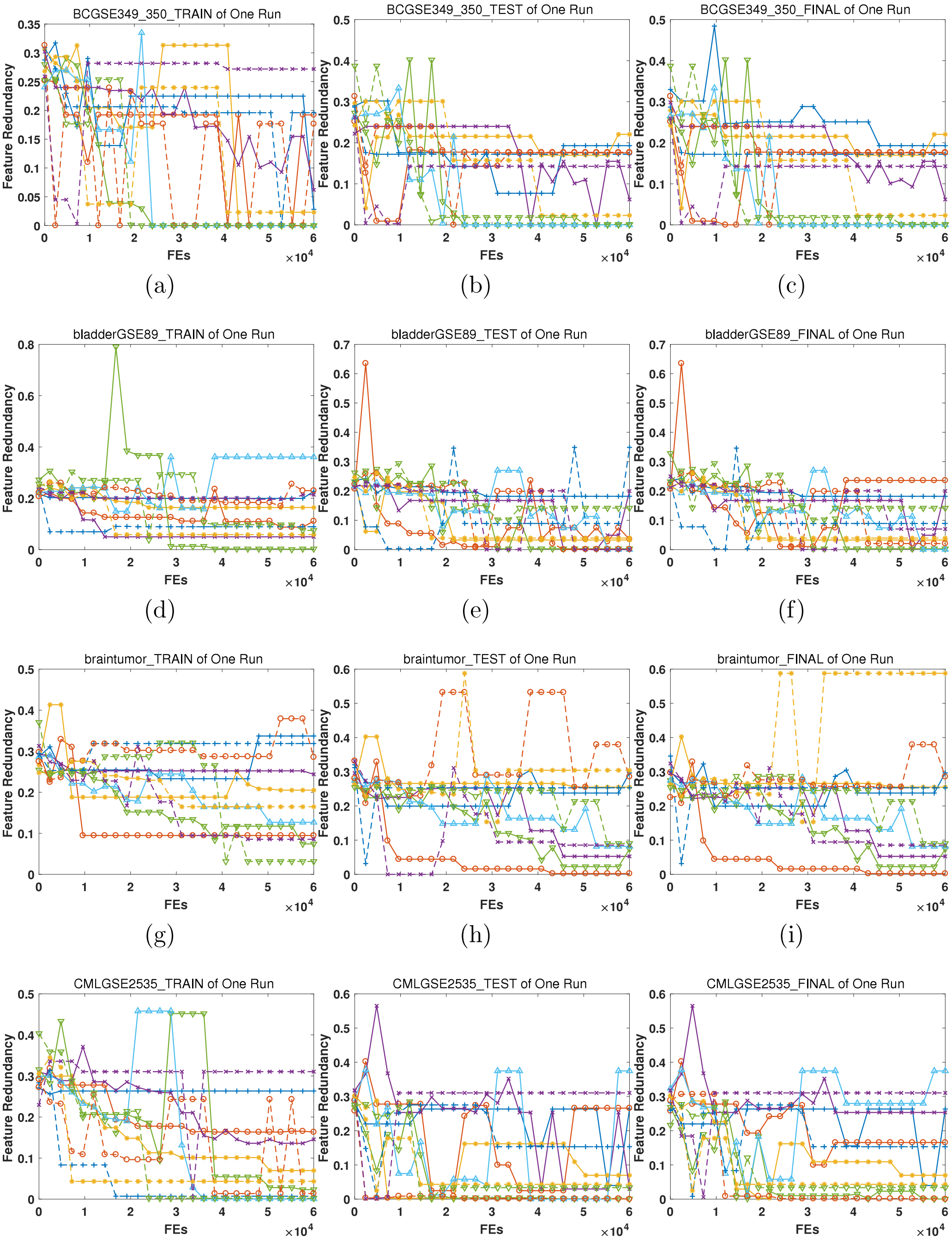
(j) (k) (l)

Fig. 19. Illustration of feature redundancy during evolution.

be something wrong with gene expression compared to healthy cells, microarray technology can be used to obtained the expression levels of numerous genes from healthy and cancer cells (or from different types of cancer cells); then, through feature selection and classification, the potential genes causing the cancer (or the relationship among multiple cancers) can be detected, thus facilitating the study of the related mechanism. Additionally, by comparing the differences in gene expression before and after therapy, the corresponding treatment mechanism and its effectiveness can be examined.

The microarray datasets[1] utilized in this paper are listed in Table 1. There are 24 datasets, each of which is characterized by a very high number of features and a low number of sample instances. For each dataset, the data are normalized with respect to each feature; then, we generate a training set using the stratified bootstrap method. Thus, the class distribution is maintained, and the samples that are not selected form the test set. Furthermore, the leave-one-out (LOO) methodology is employed for calculating the classification error.

[1] The utilized microarray datasets can be downloaded at http://www.biolab.si/supp/bi-cancer/projections/info/SRBCT.html.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 20. Illustration of feature redundancy during evolution.

4.2. Utilized algorithms and parameter settings

Four algorithms are considered for comparison, as follows:

1. Cooperative coevolutionary generalized differential evolution 3 (CCGDE3) [27].
2. Cooperative multiobjective differential evolution (CMODE) [28].
3. Multiobjective evolutionary algorithm based on decomposition (MOEA/D) [29].
4. Nondominated sorting genetic algorithm II (NSGA-II) [30].

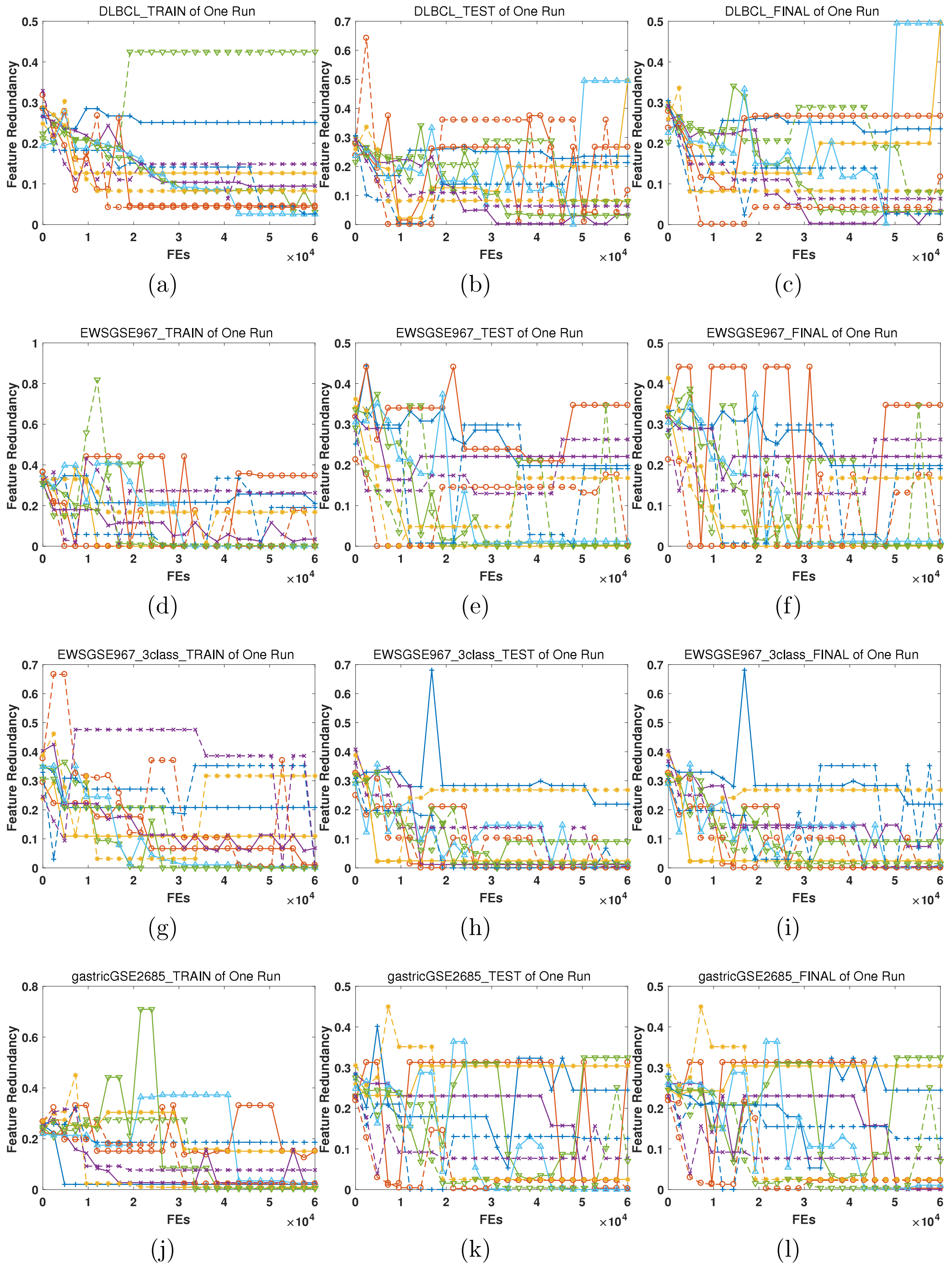
To ensure fair comparisons, the population size of every algorithm is fixed to 120. In particular, for CCGDE3, there are two swarms, each of which contains 60 individuals; for CMODE, there are three swarms, one for each of the three objectives, each with a size of 20, and the size of the archive is 120.

The maximum number of FEs is 6×10^4. For each dataset, each MOEA runs 20 times.

In CCGDE3, DE [23,24] is utilized, with $F = 0.5$ and $CR = 1.0$, and the same settings are adopted for the DE strategy employed in MOEA/D. For CMODE, adaptive DE variants are utilized; their parameter settings can be found in [31] and [25].

In NSGA-II, GA [9] is utilized. The distribution indices for crossover and mutation are both 20, and their probabilities are 1.0 and $\frac{1}{nDim}$, respectively.

For all the above algorithms, different encodings are tested; the corresponding configurations are denoted by CCGDE3-FS,



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 21. Illustration of feature redundancy during evolution.

CCGDE3-FS-w, CMODE-FS, CMODE-FS-w, MOEA/D-FS, MOEA/D-FS-w, NSGA-II-FS and NSGA-II-FS-w. For binary encoding, the feature adjustment strategy described in Section 3.3.3 is also applied. Note that for CCGDE3-FS, CMODE-FS, MOEA/D-FS and NSGA-II-FS, each variable is still encoded as a real value; thus, for binary representation, a threshold (i.e., the mid-value) is utilized.

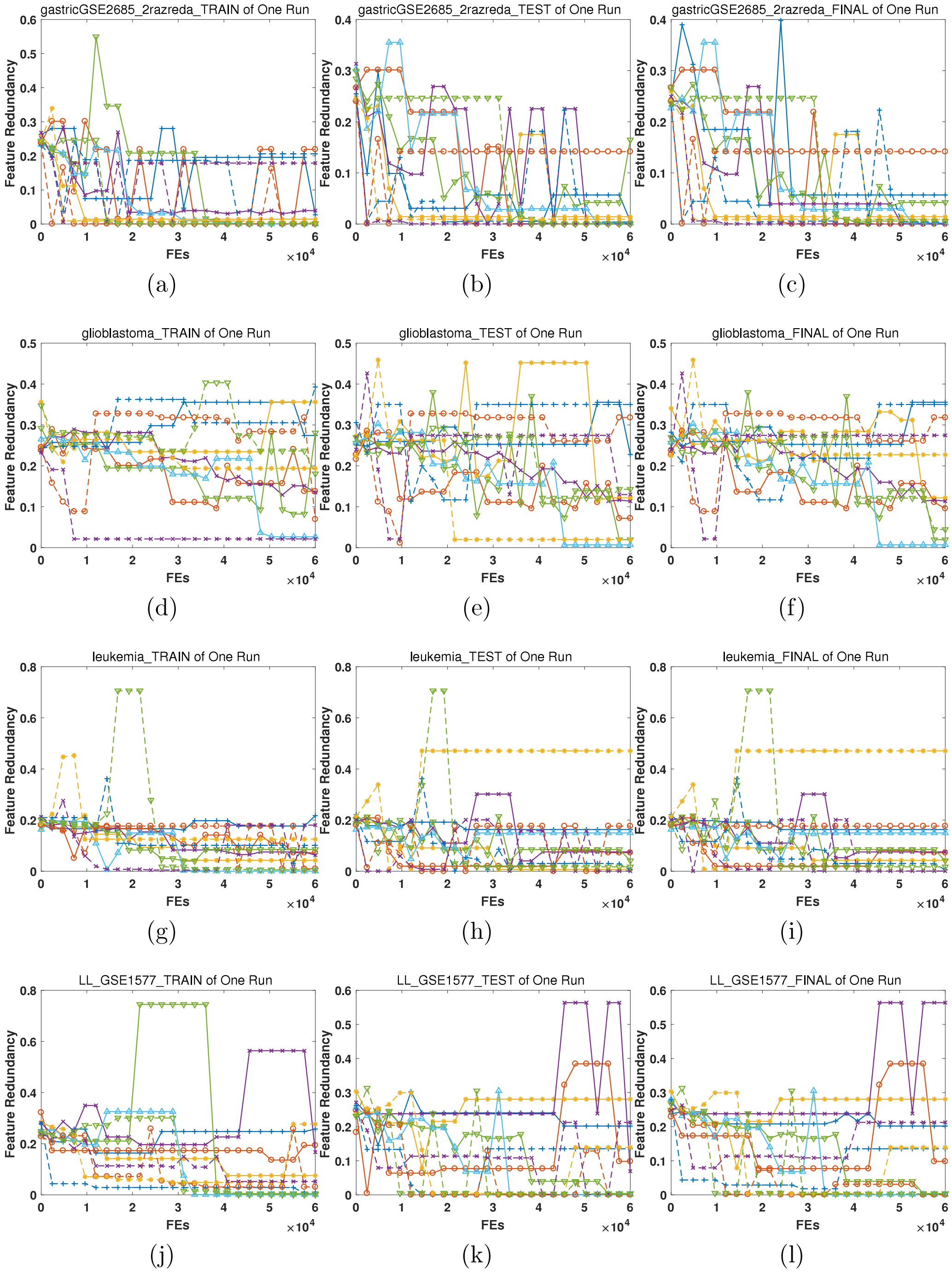
In the proposed algorithms, for DE, the adaptive strategy used in JADE [25] is adopted for F, while CR is either fixed to 1.0 or is adaptive as in JADE. The number of variable groups is simply set to 5. For PM, the distribution index is 20, and the probability is $\frac{1}{nDim}$. Similar to MOEA/D, each individual corresponds to a weight vector in the objective space; for this purpose, the same parameter settings used in MOEA/D are adopted.

4.3. Analysis

In each of the following images (Figs. 1 to 24), there are three columns, one each corresponding to the training results, the test results and the final results. Specifically, for an obtained population, a feature subset is decoded from each individual. The classification error objective is calculated with respect to the training set and the test set separately, and a weighted sum of the resulting objective values is then calculated as follows:

$$f_E^{FINAL} = 0.368 \times f_E^{TRAIN} + 0.632 \times f_E^{TEST} \tag{22}$$

where f_E^{FINAL} denotes the final classification error value, which depends on f_E^{TRAIN} and f_E^{TEST}, the classification errors on the training set and the test set, respectively. Then, the remaining two



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

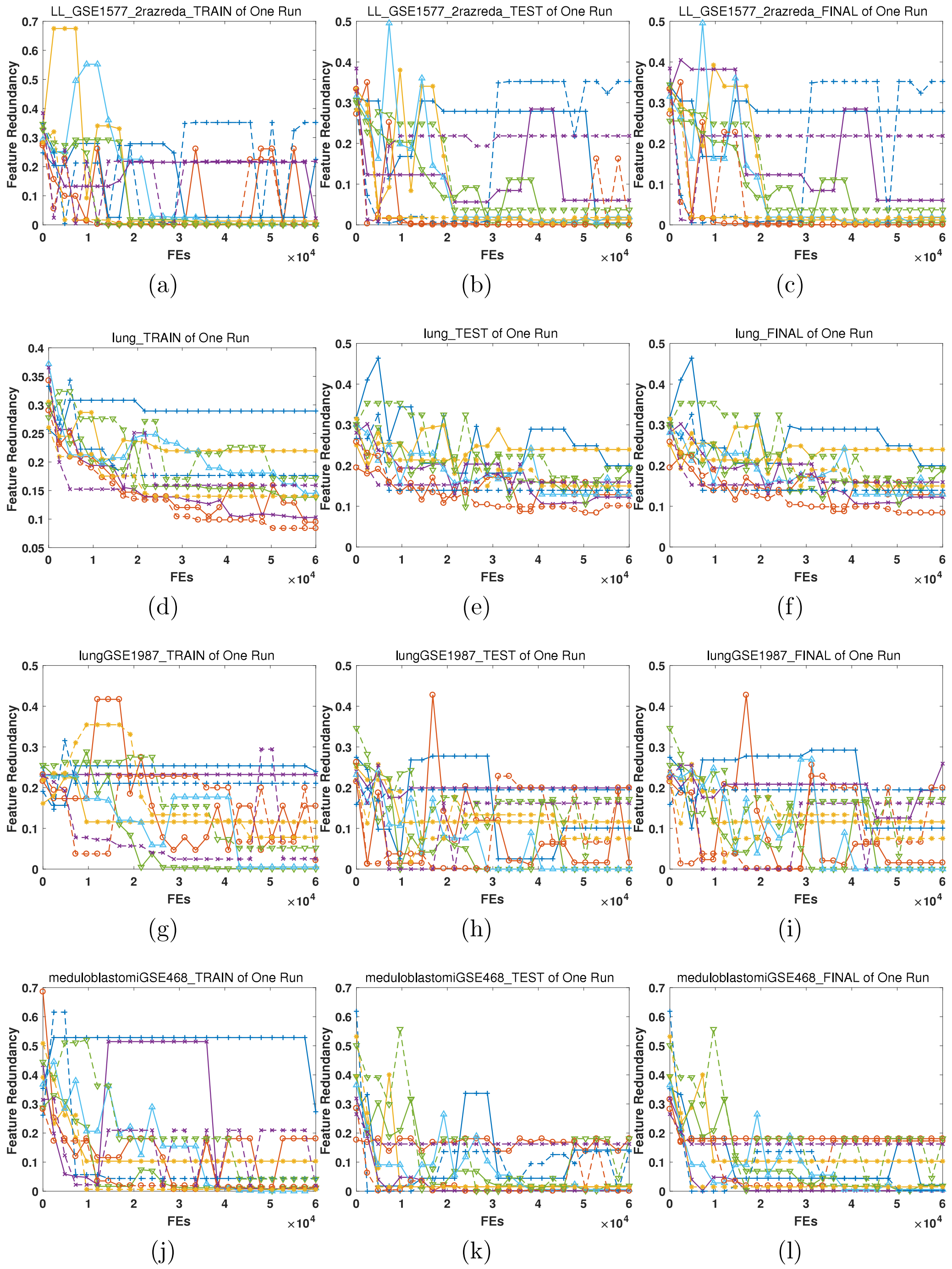
Fig. 22. Illustration of feature redundancy during evolution.

objectives are related only to the inherit properties of the feature subset.

4.3.1. Hypervolume indicator

In Figs. 1 to 6, we illustrate the values of the hypervolume (HV) indicator [27], which can simultaneously measure the distribution and convergence of the nondominated solutions obtained during the evolutionary process. Because all objective values are in the range of [0.0, 1.0], we set the HV reference point to (1, 1, 1).

In the first column of each figure, we observe that the HV indicator values are monotonically increasing, indicating that the qualities of the obtained nondominated solutions are improving. Specifically, for the first approximately 10^4 FEs, the performance rapidly improves for all algorithms; however, during the following FEs, the improvement is minimal. In the second column, the HV indicator curves are not as monotonic because the classification error is derived based on the test set. Initially, the performance quickly improves; then, however, for most datasets and algorithms, there are drops in the curves, indicating the occurrence of overfitting. In other words, although the optimization performance on the training set is still good, the validation results on the test set deteriorate to some extent, and the situation becomes considerably worse in some cases. Finally, considering the training and testing performance simultaneously, the final optimization results are illustrated in the third column. Due to the nonmonotonic behavior of the results on the test set, the evolution of the final HV indicator values is also not monotonic.


LL_GSE1577_2razreda_TRAIN of One Run

LL_GSE1577_2razreda_TEST of One Run

LL_GSE1577_2razreda_FINAL of One Run

(a) (b) (c)

lung_TRAIN of One Run

lung_TEST of One Run

lung_FINAL of One Run

(d) (e) (f)

lungGSE1987_TRAIN of One Run

lungGSE1987_TEST of One Run

lungGSE1987_FINAL of One Run

(g) (h) (i)

meduloblastomiGSE468_TRAIN of One Run

meduloblastomiGSE468_TEST of One Run

meduloblastomiGSE468_FINAL of One Run

(j) (k) (l)

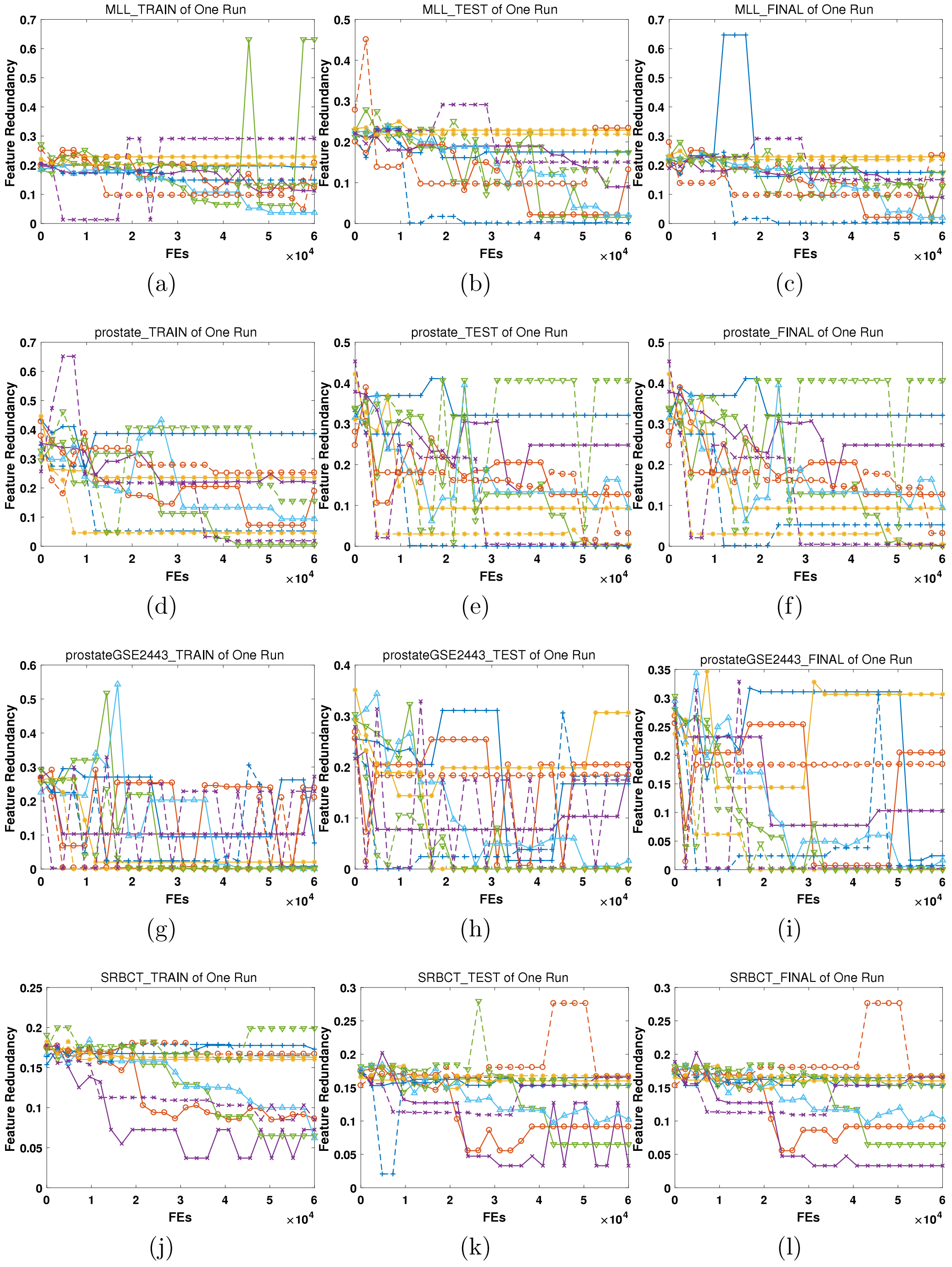
Fig. 23. Illustration of feature redundancy during evolution.

Regarding the optimization performance of the different algorithms, we draw the following conclusions:

1. With regard to the training results, the differences among the various algorithms are minimal. However, the proposed algorithms can always achieve the best results. Between the two types of encodings, the weight-encoded algorithms converge faster than the binary-encoded ones. The ranking of the other algorithms depends on the considered dataset.
2. With regard to the test results, the performance varies. In most cases, overfitting occurs. The common trend is that the indicator value increases rapidly to its maximum within a very small number of FEs; then, overfitting occurs, and the indicator value decreases or fluctuates. The only exception is the CCGDE3-FS algorithm, whose evolutionary curve exhibits slight fluctuations without a large drop; notably, the binary-encoded CCGDE3 is much better than the weight-encoded one. However, from the perspective of the whole process, the proposed algorithms always obtain the highest HV indicator values.
3. With regard to the final results, the proposed algorithms can generally obtain the highest HV indicator values.

4.3.2. Classification error

To better clarify the optimization results, we plot the lowest classification errors observed during the evolutionary process for the run with the median HV indicator value in Figs. 7 to 12. We can summarize the results as follows:



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 24. Illustration of feature redundancy during evolution.

1. On the training data, in most cases, most algorithms can achieve zero classification error. In particular, for 12 data sets, namely, BC_CCGSE3726_frozen, BCGSE349_350, bladderGSE89, CMLGSE2535, EWSGSE967, gastricGSE2685, gastricGSE2685_2razreda, leukemia, LL_GSE1577, LL_GSE1577 _2razreda, meduloblastomiGSE468 and prostateGSE2443, the classification errors remain at 0 throughout almost the whole evolutionary process. The differences among the different algorithms lie in their convergence behavior, but these differences are trivial. Among the proposed algorithms, with respect to all datasets, there is at least one algorithm that achieves zero classification error, and the same is also true for CMODE.
2. On the test data, the minimum classification error fluctuates violently throughout the evolutionary process, showing behavior more complex than that observed on the training data; thus, we focus on the lowest value over the whole evolutionary process. For 20 of the 24 datasets, a classification error of zero can be obtained; in particular, for the braintumor, MLL and prostate datasets, only the proposed algorithms achieve zero classification error. By contrast, the classification errors obtained by CMODE are usually quite high, indicating a serious overfitting problem.
3. When the training and test data are comprehensively considered, a classification error of zero indicates that the corresponding feature subset enables the correct classification of all samples in both the training data and the

Table 1
Details of the Datasets.

Dataset	File name	#Genes	#Samples	#Classes
childhood ALL	(ALLGSE412_poterapiji)	8280	60	4
childhood ALL	(ALLGSE412_pred_poTh)	8280	110	2
AML prognosis	(AMLGSE2191)	12,625	54	2
breast & colon cancer	(BC_CCGSE3726_frozen)	22,283	52	2
breast cancer	(BCGSE349_350)	12,625	24	2
bladder cancer	(bladderGSE89)	5724	40	3
brain tumor	(braintumor)	7129	40	5
CML treatment	(CMLGSE2535)	12,625	28	2
DLBCL	(DLBCL)	7070	77	2
childhood tumors	(EWSGSE967)	9945	23	2
childhood tumors	(EWSGSE967_3class)	9945	23	3
gastric cancer	(gastricGSE2685)	4522	30	3
gastric cancer	(gastricGSE2685_2razreda)	4522	30	2
glioblastoma	(glioblastoma)	12,625	50	4
leukemia	(leukemia)	5147	72	2
lymphoma & leukemia	(LL_GSE1577)	15,434	29	3
lymphoma & leukemia	(LL_GSE1577_2razreda)	15,434	19	2
lung	(lung)	12,600	203	5
lung cancer	(lungGSE1987)	10,541	34	3
medulloblastoma	(meduloblastomiGSE468)	1465	23	2
MLL	(MLL)	12,533	72	3
prostate	(prostate)	12,533	102	2
prostate cancer	(prostateGSE2443)	12,627	20	2
SRBCT	(SRBCT)	2308	83	4

test data, which is the expected result. Classification errors of zero are observed for 13 datasets, namely, of ALLGSE412_pred_poTh, BC_CCGSE3726_frozen, BCGSE349_350, bladderGSE89, DLBCL, EWSGSE967, EWSGSE967_3class, gastricGSE2685_2razreda, LL_GSE1577, LL_GSE1577_2razreda, MLL, prostateGSE2443 and SRBCT. Specifically, for the BC_CCGSE3726_frozen and bladderGSE89 datasets, only the proposed algorithms can achieve zero classification error, while for all other datasets, the proposed algorithms are generally no worse than their counterparts. Notably, the binary-encoded algorithms are better than the weight-encoded one.

For the individuals with the minimum classification errors as discussed above, the corresponding feature numbers and feature redundancy are discussed in the following two subsections and are illustrated in Figs. 13 to 18 and Figs. 19 to 24, respectively.

4.3.3. Feature number

For 15 datasets, namely, ALLGSE412_pred_poTh, BC_CCGSE3726_frozen, BCGSE349_350, CMLGSE2535, DLBCL, EWSGSE967, EWSGSE967_3class, gastricGSE2685, gastricGSE2685_2razreda, leukemia, LL_GSE1577, LL_GSE1577_2razreda, lungGSE1987, meduloblastomiGSE468 and prostateGSE2443, as illustrated in Figs. 13 to 18, all algorithms can generate a subset containing no more than 10 features ($f_N = \frac{N_f}{N_F^{th}} = \frac{10}{50} = 0.2$), except in some special cases for CCGDE3. For the other datasets, the number of features fluctuates throughout the evolutionary process. With respect to the training data and test data, however, in contrast to the cases analyzed in the previous sections, the evolutionary curves share similar characteristics.

4.3.4. Feature redundancy

With regard to the feature redundancy objective, as shown in Figs. 19 to 24, the evolutionary curves are characterized by undulations, the comprehension of which involves some subtleties. When an objective is formulated on the basis of the feature redundancy, during the evolutionary process, this objective will drive the algorithm to generate well-performing feature subsets consisting of relatively few features with little redundancy.

4.3.5. Time consumption

In Table 2, we list the average operating time for each algorithm with respect to each dataset. Additionally, the sum of time consumptions over all datasets for each algorithm is listed at the bottom, and the speedup ratios are shown in parentheses. In comparison to the proposed algorithms, all other MOEAs are at least one order of magnitude slower, with the exception of the simple CCGDE3-FS algorithm. For the proposed distributed algorithms, the number of utilized CPU cores is 60, and the speedup ratios with respect to the binary-encoded and weight-encoded versions of the most time-consuming serial MOEA, CMODE, are $3.77E + 01$ and $4.91E + 01$, respectively, which are quite close to the ideal value of 60. Therefore, we can conclude that the proposed algorithms are able to obtain better optimization results with high efficiency.

5. Discussion and future work

In the experimental study presented above, although the proposed algorithms can outperform their counterparts, serious over fitting occurs for some datasets, leading to significant performance deterioration on the test sets. By contrast, for the binary-encoded CCGDE3 algorithm, overfitting rarely occurs, and for many datasets, its classification errors are below those of most of the other algorithms. Therefore, we can refer to the binary-encoded CCGDE3 algorithm to find possible ways to alleviate the overfitting that occurs in the proposed algorithms.

In the proposed algorithms, only different encodings and other mechanisms with respect to the MOEA approach have been examined. However, there are many traditional methodologies for addressing the feature selection problem, and the outputs of these methodologies can serves as prior knowledge to guide efforts to further enhance the proposed algorithms.

Typically, for the analysis of microarray data, traditional techniques are very time-consuming. Owing to their higher efficiency and better performance, the proposed algorithms can be profitably applied in analyzing microarray data and can thus contribute to the study and treatment of various cancers.

6. Conclusion

To address the feature selection problem, this paper proposes a multiobjective feature selection model that simultaneously considers classification error, feature number and feature

Table 2
Average Operating Times (sec).

	CCGDE3-FS	CMODE-FS	MOEA/D-FS	NSGA-II-FS	DPCCMOLSBEA-FS	DPCCMOLSABEA-FS	CCGDE3-FS-w	CMODE-FS-w	MOEA/D-FS-w	NSGA-II-FS-w	DPCCMOLSEA-FS-w
ALLGSE412_poterapiji	6.89E+01 (7.65E+00)	3.57E+02 (3.96E+01)	1.01E+02 (1.12E+01)	1.19E+02 (1.32E+01)	**9.01E+00**	**9.03E+00** **(1.00E+00)**	1.57E+02 (1.74E+01)	4.43E+02 (4.92E+01)	1.30E+02 (1.44E+01)	2.02E+02 (2.24E+01)	**1.29E+01** **(1.43E+00)**
ALLGSE412_pred_poTh	1.97E+02 (1.61E+01)	4.17E+02 (3.42E+01)	2.75E+02 (2.25E+01)	2.67E+02 (2.19E+01)	**1.22E+01**	**1.22E+01** **(1.00E+00)**	2.79E+02 (2.29E+01)	5.20E+02 (4.26E+01)	3.00E+02 (2.46E+01)	3.85E+02 (3.16E+01)	**1.85E+01** **(1.52E+00)**
AMLGSE2191	6.85E+01 (5.31E+00)	4.97E+02 (3.85E+01)	1.18E+02 (9.15E+00)	1.67E+02 (1.29E+01)	**1.29E+01**	**1.30E+01** **(1.01E+00)**	2.04E+02 (1.58E+01)	6.63E+02 (5.14E+01)	1.86E+02 (1.44E+01)	2.79E+02 (2.16E+01)	**1.90E+01** **(1.47E+00)**
BC_CCGSE3726_frozen	9.28E+01 (3.98E+00)	9.90E+02 (4.25E+01)	2.26E+02 (9.70E+00)	2.62E+02 (1.12E+01)	**2.33E+01**	**2.34E+01** **(1.00E+00)**	3.91E+02 (1.68E+01)	1.27E+03 (5.45E+01)	4.05E+02 (1.74E+01)	5.32E+02 (2.28E+01)	**3.72E+01** **(1.60E+00)**
BCGSE349_350	3.37E+01 (2.98E+00)	5.03E+02 (4.45E+01)	6.40E+01 (5.66E+00)	9.48E+01 (8.39E+00)	**1.13E+01**	**1.13E+01** **(1.00E+00)**	9.21E+01 (8.15E+00)	6.46E+02 (5.72E+01)	9.89E+01 (8.75E+00)	1.71E+02 (1.51E+01)	**1.52E+01** **(1.35E+00)**
bladderGSE89	2.90E+01 (4.92E+00)	2.28E+02 (3.87E+01)	4.29E+01 (7.28E+00)	5.87E+01 (9.97E+00)	**5.89E+00**	**5.95E+00** **(1.01E+00)**	6.51E+01 (1.11E+01)	2.80E+02 (4.75E+01)	5.70E+01 (9.68E+00)	9.62E+01 (1.63E+01)	**7.64E+00** **(1.30E+00)**
braintumor	3.20E+01 (4.47E+00)	2.71E+02 (3.78E+01)	4.84E+01 (6.76E+00)	7.00E+01 (9.78E+00)	**7.16E+00**	**7.20E+00** **(1.01E+00)**	8.19E+01 (1.14E+01)	3.37E+02 (4.71E+01)	6.20E+01 (8.66E+00)	1.14E+02 (1.59E+01)	**8.80E+00** **(1.23E+00)**
CMLGSE2535	3.49E+01 (3.06E+00)	4.96E+02 (4.35E+01)	6.86E+01 (6.02E+00)	9.97E+01 (8.75E+00)	**1.14E+01**	**1.15E+01** **(1.01E+00)**	9.80E+01 (8.60E+00)	6.19E+02 (5.43E+01)	1.04E+02 (9.12E+00)	1.71E+02 (1.50E+01)	**1.54E+01** **(1.35E+00)**
DLBCL	8.13E+01 (9.50E+00)	3.05E+02 (3.56E+01)	1.16E+02 (1.36E+01)	1.24E+02 (1.45E+01)	**8.56E+00**	**8.64E+00** **(1.01E+00)**	1.61E+02 (1.88E+01)	3.79E+02 (4.43E+01)	1.42E+02 (1.66E+01)	2.10E+02 (2.45E+01)	**1.20E+01** **(1.40E+00)**
EWSGSE967	2.59E+01 (2.88E+00)	4.16E+02 (4.63E+01)	5.19E+01 (5.77E+00)	7.29E+01 (8.11E+00)	**8.99E+00**	**9.01E+00** **(1.00E+00)**	6.78E+01 (7.54E+00)	4.91E+02 (5.46E+01)	7.58E+01 (8.43E+00)	1.28E+02 (1.42E+01)	**1.19E+01** **(1.32E+00)**
EWSGSE967_3class	2.60E+01 (2.92E+00)	3.93E+02 (4.41E+01)	4.69E+01 (5.26E+00)	7.03E+01 (7.89E+00)	**8.91E+00**	**8.95E+00** **(1.00E+00)**	6.66E+01 (7.47E+00)	4.87E+02 (5.47E+01)	7.01E+01 (7.87E+00)	1.25E+02 (1.40E+01)	**1.15E+01** **(1.29E+00)**
gastricGSE2685	1.89E+01 (4.15E+00)	1.79E+02 (3.93E+01)	2.82E+01 (6.20E+00)	4.24E+01 (9.32E+00)	**4.55E+00**	**4.56E+00** **(1.00E+00)**	3.81E+01 (8.37E+00)	2.07E+02 (4.55E+01)	3.88E+01 (8.53E+00)	6.51E+01 (1.43E+01)	**5.78E+00** **(1.27E+00)**
gastricGSE2685_2razreda	1.90E+01 (4.12E+00)	1.82E+02 (3.95E+01)	3.23E+01 (7.01E+00)	4.52E+01 (9.80E+00)	**4.61E+00**	**4.62E+00** **(1.00E+00)**	3.87E+01 (8.39E+00)	2.12E+02 (4.60E+01)	4.20E+01 (9.11E+00)	6.64E+01 (1.44E+01)	**6.07E+00** **(1.32E+00)**
glioblastoma	5.21E+01 (4.13E+00)	4.53E+02 (3.60E+01)	9.37E+01 (7.44E+00)	1.41E+02 (1.12E+01)	**1.26E+01**	**1.26E+01** **(1.00E+00)**	1.82E+02 (1.44E+01)	6.49E+02 (5.15E+01)	1.32E+02 (1.05E+01)	2.47E+02 (1.96E+01)	**1.54E+01** **(1.22E+00)**
leukemia	6.33E+01 (9.59E+00)	2.26E+02 (3.42E+01)	9.15E+01 (1.39E+01)	9.79E+01 (1.48E+01)	**6.60E+00**	**6.64E+00** **(1.01E+00)**	1.41E+03 (2.14E+02)	2.91E+02 (4.41E+01)	1.20E+02 (1.82E+01)	1.56E+02 (2.36E+01)	**9.66E+00** **(1.46E+00)**
LL_GSE1577	4.43E+01 (3.21E+00)	5.48E+02 (3.97E+01)	8.10E+01 (5.87E+00)	1.30E+02 (9.42E+00)	**1.38E+01**	**1.39E+01** **(1.01E+00)**	1.23E+02 (8.91E+00)	7.67E+02 (5.56E+01)	1.27E+02 (9.20E+00)	2.10E+02 (1.52E+01)	**1.84E+01** **(1.33E+00)**
LL_GSE1577_2razreda	4.15E+01 (3.07E+00)	4.51E+02 (3.34E+01)	7.22E+01 (5.35E+00)	1.23E+02 (9.11E+00)	**1.35E+01**	**1.36E+01** **(1.01E+00)**	1.26E+02 (9.33E+00)	7.68E+02 (5.69E+01)	9.84E+01 (7.29E+00)	1.81E+02 (1.34E+01)	**1.78E+01** **(1.32E+00)**
lung	9.94E+02 (4.43E+01)	1.36E+03 (3.07E+01)	1.16E+03 (2.62E+01)	1.30E+03 (2.93E+01)	**4.43E+01**	**3.53E+01** **(7.97E−01)**	9.12E+01 (2.06E+00)	1.81E+03 (4.09E+01)	1.21E+03 (2.73E+01)	2.05E+03 (4.63E+01)	**9.45E+01** **(2.13E+00)**
lungGSE1987	3.34E+01 (3.34E+00)	4.19E+02 (4.19E+01)	6.34E+01 (6.34E+00)	8.80E+01 (8.80E+00)	**1.00E+01**	**1.01E+01** **(1.01E+00)**	1.18E+02 (1.18E+01)	5.46E+02 (5.46E+01)	1.07E+02 (1.07E+01)	1.73E+02 (1.73E+01)	**1.35E+01** **(1.35E+00)**
meduloblastomiGSE468	9.66E+00 (5.08E+00)	5.42E+01 (2.85E+01)	1.09E+01 (5.74E+00)	1.71E+01 (9.00E+00)	**1.90E+00**	**1.91E+00** **(1.01E+00)**	1.31E+01 (6.89E+00)	5.83E+01 (3.07E+01)	1.31E+01 (6.89E+00)	2.06E+01 (1.08E+01)	**2.30E+00** **(1.21E+00)**
MLL	7.90E+01 (5.77E+00)	5.14E+02 (3.75E+01)	1.44E+02 (1.05E+01)	1.97E+02 (1.44E+01)	**1.37E+01**	**1.37E+01** **(1.00E+00)**	2.41E+02 (1.76E+01)	6.59E+02 (4.81E+01)	1.83E+02 (1.34E+01)	3.26E+02 (2.38E+01)	**1.80E+01** **(1.31E+00)**
prostate	1.36E+02 (8.50E+00)	5.42E+02 (3.39E+01)	2.78E+02 (1.74E+01)	2.95E+02 (1.84E+01)	**1.60E+01**	**1.59E+01** **(9.94E−01)**	3.10E+02 (1.94E+01)	7.18E+02 (4.49E+01)	3.47E+02 (2.17E+01)	4.54E+02 (2.84E+01)	**2.20E+01** **(1.38E+00)**
prostateGSE2443	3.16E+01 (2.82E+00)	4.88E+02 (4.36E+01)	6.15E+01 (5.49E+00)	9.03E+01 (8.06E+00)	**1.12E+01**	**1.12E+01** **(1.00E+00)**	7.79E+01 (6.96E+00)	6.13E+02 (5.47E+01)	8.32E+01 (7.43E+00)	1.49E+02 (1.33E+01)	**1.46E+01** **(1.30E+00)**
SRBCT	7.70E+01 (1.93E+01)	1.20E+02 (3.02E+01)	7.20E+01 (1.81E+01)	7.13E+01 (1.79E+01)	**3.98E+00**	**4.01E+00** **(1.01E+00)**	6.43E+01 (1.62E+01)	1.33E+02 (3.34E+01)	6.59E+01 (1.66E+01)	8.29E+01 (2.08E+01)	**4.85E+00** **(1.22E+00)**
SUM	2.29E+03 (8.29E+00)	1.04E+04 (3.77E+01)	3.35E+03 (1.21E+01)	4.04E+03 (1.46E+01)	**2.76E+02**	**2.68E+02** **(9.71E−01)**	4.50E+03 (1.63E+01)	1.36E+04 (4.91E+01)	4.20E+03 (1.52E+01)	6.59E+03 (2.39E+01)	**4.13E+02** **(1.49E+00)**

[1] Values in parentheses denote the speedup ratios of DPCCMOLSBEA-FS with respect to the listed run times.
[2] Values in bold are related to the parallel algorithms.

redundancy. Then, several distributed parallel algorithms are proposed. Different feature encodings are tested, and an adaptive strategy is examined. For the microarray datasets, the feature number is typically extremely large; consequently, the time consumption for optimization will be intolerable. Thus, a feature number constraint is applied to reduce the computational complexity. Additionally, by separating variables into several groups and evolving them under the CC framework as well as allocating individuals to numerous CPU cores, a two-layer distributed parallel structure is constructed, thus significantly reducing the time consumption. Moreover, sample-wise parallelism is applied to considerably increase the efficiency of processing on the test set during the recording phase. Compared to several state-of-the-art MOEAs, the proposed algorithms are more effective in terms of optimization performance and more efficient in terms of time consumption.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.future.2019.02.030.

Acknowledgments

This work was supported in part by the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University, China under Grant 2018002, in part by the Opening Project of Guangdong High Performance Computing Society, China under Grant 2017060101, in part by the Foundation of Key Laboratory of Machine Intelligence and Advanced Computing of the Ministry of Education, China under Grant No. MSC-201602A, and in part by Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund, China (the second phase) under Grant U1501501.

References

[1] S. Georganos, T. Grippa, S. Vanhuysse, M. Lennert, M. Shimoni, S. Kalogirou, E. Wolff, Less is more: optimizing classification performance through feature selection in a very-high-resolution remote sensing object-based urban application, GISci. Remote Sens. 55 (2) (2018) 221–242, http://dx.doi.org/10.1080/15481603.2017.1408892.

[2] H. Shi, X. Li, K.S. Hwang, W. Pan, G. Xu, Decoupled visual servoing with Fuzzy Q-learning, IEEE Trans. Ind. Inf. 14 (1) (2018) 241–252, http://dx.doi.org/10.1109/TII.2016.2617464.

[3] L. Zheng, H. Wang, S. Gao, Sentimental feature selection for sentiment analysis of Chinese online reviews, Int. J. Mach. Learn. Cybern. 9 (1) (2018) 75–84, http://dx.doi.org/10.1007/s13042-015-0347-4.

[4] L. Zhang, Q. Zhang, B. Du, X. Huang, Y.Y. Tang, D. Tao, Simultaneous spectral-spatial feature selection and extraction for hyperspectral images, IEEE Trans. Cybern. 48 (1) (2018) 16–28, http://dx.doi.org/10.1109/TCYB.2016.2605044.

[5] C. Zhang, J. Zhou, C. Li, W. Fu, T. Peng, A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting, Energy Convers. Manage. 143 (2017) 360–376, http://dx.doi.org/10.1016/j.enconman.2017.04.007.

[6] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (4) (2012) 1106–1119, http://dx.doi.org/10.1109/TCBB.2012.33.

[7] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, Inform. Sci. 282 (2014) 111–135, http://dx.doi.org/10.1016/j.ins.2014.05.042.

[8] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1424–1437, http://dx.doi.org/10.1109/TPAMI.2004.105.

[9] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, USA, 1992.

[10] S. Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, Soft Comput. 22 (3) (2018) 811–822, http://dx.doi.org/10.1007/s00500-016-2385-6.

[11] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE Int. Conf. Neural Netw. 4 (8) (1995) 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.

[12] S.P. Das, S. Padhy, A novel hybrid model using teaching—learning-based optimization and a support vector machine for commodity futures index forecasting, Int. J. Mach. Learn. Cybern. 9 (1) (2018) 97–111, http://dx.doi.org/10.1007/s13042-015-0359-0.

[13] A. Onan, S. Korukoğlu, A feature selection model based on genetic rank aggregation for text sentiment classification, J. Inf. Sci. 43 (1) (2017) 25–38, http://dx.doi.org/10.1177/0165551515613226.

[14] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381, http://dx.doi.org/10.1016/j.neucom.2015.06.083.

[15] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626, http://dx.doi.org/10.1109/TEVC.2015.2504420.

[16] B. Huang, B. Buckley, T.M. Kechadi, Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications, Expert Syst. Appl. 37 (5) (2010) 3638–3646, http://dx.doi.org/10.1016/j.eswa.2009.10.027.

[17] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Trans. Cybern. 43 (6) (2013) 1656–1671, http://dx.doi.org/10.1109/TSMCB.2012.2227469.

[18] B. Xue, L. Cervante, L. Shang, W.N. Browne, M. Zhang, Multi-objective evolutionary algorithms for filter based feature selection in classification, Int. J. Artif. Intell. Tools 22 (04) (2013) 1350024, http://dx.doi.org/10.1142/S0218213013500243.

[19] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (1) (2014) 175–186, http://dx.doi.org/10.1007/s00521-013-1368-0.

[20] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, J.-J. Li, Distributed evolutionary algorithms and their models: A survey of the state-of-the-art, Appl. Soft Comput. 34 (2015) 286–300, http://dx.doi.org/10.1016/j.asoc.2015.04.061.

[21] M.A. Potter, K.A.D. Jong, A cooperative coevolutionary approach to function optimization, in: Proceedings of the International Conference on Evolutionary Computation, PPSN III, in: The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, Springer-Verlag, London, UK, UK, 1994, pp. 249–257.

[22] B. Cao, J. Zhao, P. Yang, Z. Lv, X. Liu, G. Min, 3D multi-objective deployment of an industrial wireless sensor network for maritime applications utilizing a distributed parallel algorithm, IEEE Trans. Ind. Inf. 14 (12) (2018) http://dx.doi.org/10.1109/TII.2018.2803758, 5487–5495.

[23] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag New York, Inc, Secaucus, NJ, USA, 2005.

[24] R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359, http://dx.doi.org/10.1023/A:1008202821328.

[25] J. Zhang, A.C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958, http://dx.doi.org/10.1109/TEVC.2009.2014613.

[26] Q. Lin, J. Chen, Z.H. Zhan, W.N. Chen, C.A.C. Coello, Y. Yin, C.M. Lin, J. Zhang, A hybrid evolutionary immune algorithm for multiobjective optimization problems, IEEE Trans. Evol. Comput. 20 (5) (2016) 711–729, http://dx.doi.org/10.1109/TEVC.2015.2512930.

[27] L.M. Antonio, C.A.C. Coello, Use of cooperative coevolution for solving large scale multiobjective optimization problems, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2758–2765, http://dx.doi.org/10.1109/CEC.2013.6557903.

[28] J. Wang, W. Zhang, J. Zhang, Cooperative differential evolution with multiple populations for multiobjective optimization, IEEE Trans. Cybern. 46 (12) (2016) 2848–2861, http://dx.doi.org/10.1109/TCYB.2015.2490669.

[29] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.

[30] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[31] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657, http://dx.doi.org/10.1109/TEVC.2006.872133.

Bin Cao received the Ph.D. degree in computer application technology from Jilin University in 2012. He is currently in the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. From 2012 to 2014, he was a Postdoc in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include intelligent computation with its applications to big data, cyber–physical system, graphics and visual media; high performance computing and cloud computing.

Jianwei Zhao received the Master's degree from Hebei University of Technology in 2018. He is currently working toward the Ph.D. degree in the School of Artificial Intelligence of Hebei University of Technology, Tianjin, China. His main research interests include intelligent computation with its applications to big data, cyber–physical system, graphics and visual media; high performance computing and cloud computing.

Po Yang received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2004; the M.Sc. degree in computer science from Bristol University, Bristol, U.K., in 2006; and the Ph.D. degree in electronic engineering from the University of Staffordshire, Stafford, U.K., in 2011. He is currently a Senior Lecturer with the Department of Computer Science, Liverpool John Moores University, Liverpool, UK. He has been a Postdoctoral Research Fellow with the Department of Computing, Bedfordshire University, Luton, U.K. Before he joined Bedfordshire University, he was a Research Assistant with the University of Salford, Salford, U.K. His main research interests include radio-frequency identification and sensor networking, document image processing, computer vision, GPU, and parallel computing.

Peng Yang received the Ph.D. degree from Hebei University of Technology in 2001. He is currently a professor and the dean of School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. His main research interests include intelligent control techniques with applications to intelligent rehabilitation system, computer-aided design and cyber–physical system.

Xin Liu received the Master's degree from Jilin University, Changchun, China, in 2012. She is currently at the Hebei University of Technology, Tianjin, China. Her current research focuses on intelligent computation techniques with applications to big data, cyber–physical system, graphics and visual media, and high performance computing and cloud computing.

Jun Qi is currently pursuing her Ph.D. in the department of Computing Sciences at Liverpool John Moores University, UK. She received the M.Sc. degree and B.Sc. degree in Computer Science and technology at Changzhou University, China, in 2010 and 2013, respectively. She was research associate in the department of Computing and Mathematics at University of Ulster, UK, in 2014. Her research interests include machine learning, expert system, image processing, lifelogging and software development.

Andrew Simpson is a Lecturer within the Department of Computer Science at Liverpool John Moores University. In the past Andrew has worked in the games industry as a programmer for some of the most well-known game studios in the UK including: Bizarre Creations, Traveler's Tales, Lucid Games and Eurocom. He has worked on many different cross platform games including the BAFTA nominated Geometry Wars 3: Dimensions. Andrew is a generalist programmer and has experience working in many different areas of games programming including: gameplay, artificial intelligence, user interface, multiplayer, analytics, localization and tools.

Mohamed Elhoseny is currently an Assistant Professor at the Faculty of Computers and Information, Mansoura University and a researcher at CoVIS Lab, Department of Computer Science and Engineering, University of North Texas. Dr. Elhoseny is the Director of Distributed Sensing and Intelligent Systems Lab, Mansoura University, Egypt. Collectively, Dr. Elhoseny authored/co-authored over 100 ISI Journal articles, Conference Proceedings, Book Chapters, and (9) books published by Springer and Taylor and Francis. His research interests include Sensors Technologies, Network Security, Internet of Things, and Artificial Intelligence Applications. Dr. Elhoseny serves as the Editor-in-Chief of International Journal of Smart Sensor Technologies and Applications, IGI Global. Besides, he is an Associate Editor of prestigious journals such as IEEE Access (Impact Factor 3.5), IEEE Future Directions, PLOS One journal (Impact Factor 2.7), Remote Sensing (Impact Factor 3.5), and International Journal of E-services and Mobile Applications, IGI Global (Scopus Indexed). Also, he is an Editorial Board member in several journals such as Applied Intelligence, Springer (Impact Factor 1.9). Dr. Elhoseny guest-edited several special issues at many journals published by IEEE, Elsevier, Hindawi, Springer, Inderscience, and MDPI. Moreover, he served as the co-chair, the publication chair, the program chair, and a track chair for several international conferences published by IEEE and Springer. Dr. Elhoseny has many collaborative scientific activities with international teams in different research projects. As a result, Dr. Elhoseny is the Co-PI of three international funded projects in USA, and China. Besides, he has four more proposals for international funded projects under review in (USA, China, India, and Egypt).

Dr. Elhoseny is the Editor-in-Chief of the Studies in Distributed Intelligence Springer Book Series, the Editor-in-Chief of The Sensors Communication for Urban Intelligence CRC Press-Taylor and Francis Book Series, and the Editor-in-Chief of The Distributed Sensing and Intelligent Systems CRC Press-Taylor and Francis Book Series. He has been awarded the Egypt National Prize for Young Researchers in 2018 and the best Ph.D. thesis in Mansoura University in 2015. Besides, he is a TPC Member or Reviewer in 50+ International Conferences and Workshops. Furthermore, he has been reviewing papers for 80+ International Journals including IEEE Communications Magazine, IEEE Transactions on Intelligent Transportation Systems, IEEE Sensors Letters, IEEE Communication Letters, Elsevier Computer Communications, Computer Networks, Sustainable Cities and Society, Wireless Personal Communications, and Expert Systems with Applications.

Irfan Mehmood is currently a Lecturer in Applied Artificial Intelligence, School of Media, Design and Technology, Faculty of Engineering and Informatics, University of Braford, United Kingdom. He has been involved in IT industry and academia in Pakistan, South Korea and UK for a decade. His sustained contribution at various research and industry-collaborative projects gives him an extra edge to meet the current challenges faced in the field of multimedia analytics. Specifically, he has made significant contribution in the areas of video summarization, medical image analysis, visual surveillance, information mining, and deep learning in industrial applications. He is an active member of IEEE. He has also provided editorial services in various special issues in top ranked Journals of reputed publishers: Elsevier, IEEE, Springer and Wiley. He is also serving as a professional reviewer for numerous journals and conferences.

Khan Muhammad received the Ph.D degree in Digital Contents from Sejong University, South Korea. He is currently working as an Assistant Professor at Department of Software and lead researcher of Intelligent Media Laboratory, Sejong University, Seoul. His research interests include medical image analysis (brain MRI, diagnostic hysteroscopy and wireless capsule endoscopy), information security (steganography, encryption, watermarking and image hashing), video summarization, computer vision, fire/smoke scene analysis, and video surveillance. He has published over 60 papers in peer-reviewed international journals and conferences in these research areas with target venues as IEEE COMMAG, TII, TIE, TSMC-Systems, IoTJ, Access, TSC, Elsevier INS, Neurocomputing, PRL, FGCS, ASOC, IJIM, SWEVO, COMCOM, COMIND, JPDC, PMC, BSPC, CAEE, Springer NCAA, MTAP, JOMS, and RTIP, etc. He is also serving as a professional reviewer for over 40 well-reputed journals and conferences.