Multiobjective Feature Selection of 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 Computer Science and Engineering, Sejong University, Seoul, Republic of Korea*
[f]*College of Electronics and Information Engineering, Sejong University, Seoul, Republic of Korea*

Abstract

Many real-world problems are large scale and hence difficult to address. Due to the large number of features in microarray datasets, feature selection and classification are even more challenging. Although there are numerous features, not all features contribute to the classification, and some features are even impeditive. Through feature selection, a feature subset that contains only a small quantity of essential features is generated, which can increase the classification accuracy and significantly reduce the time consumption. In this paper, we construct a multiobjective feature selection model that simultaneously considers classification error, feature number and feature redundancy. For this model, we propose several distributed parallel algorithms through different encodings and an adaptive strategy. Additionally, to reduce the time consumption, various tactics are employed, including feature number constraint, distributed parallelism and sample-wise parallelism. For

*Corresponding author
**Corresponding author
Email addresses: 201422102003@stu.hebut.edu.cn (Jianwei Zhao), poyangcn@gmail.com (Po Yang)

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.

Keywords: microarray data set, high dimension, multiobjective feature selection, distributed parallelism, feature redundancy

1. Introduction

An object can be abstracted to a series of features that indicate various properties. Based on these features, we can classify a number of object instances, perform analyses, and so forth [1, 2, 3, 4, 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, its complexity increases even more rapidly. If the feature selection problem is viewed as a combination problem, assuming that there are n features, then the number of possible combinations will be 2^n due to their exponential relationship. Thus, for microarray datasets, the exhaustive enumeration will be very time consuming and intolerable.

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

For NP-hard problems and the aforementioned time-consuming combination problems, heuristic algorithms can explore the search space using simple population-based strategies, finding suboptimal solutions within a tolerable 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) aiming at feature selection. In each experiment, a desirable feature number was introduced, and a penalty was applied to the fitness. 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; specifically, the feature selection status and the parameters of the RBF kernel function

in SVM 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 to a binary form, outperforming PSO and GA.

However, the above research mainly considered one objective, namely, the classification accuracy, while the feature number was fixed and a threshold was applied. To simultaneously consider multiple objectives, multiobjective evolutionary algorithms (MOEAs) are suitable. Thus, many research efforts have been devoted to the multiobjective feature selection problem [15].

In the studies on MOEAs, the classification accuracy is always the main concern, such as the overall classification accuracy, the true positive rate and the true negative rate [16]. The feature number is also an objective [17]. However, the feature redundancy is rarely considered [18, 19]. In this paper, we propose a multiobjective feature selection model by simultaneously considering 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, the time consumption may still be intolerable to some extent. Consequently, feature number constraining and distributed parallel MOEAs [20] will be beneficial. Additionally, for high-dimensional multiobjective problems (MOPs), by separating variables into several groups, the cooperative coevolutionary (C-C) framework [21] divides the original problem into several low-dimensional tasks, 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 by simultaneously considering three objectives: classification error, feature number and feature redundancy.
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 binary-encoded values, respectively. In addition, an adaptive improvement is tested, yielding two binary-encoded algorithms.
3. By constraining the feature number, time consumption is greatly reduced. Based on variable grouping and individual allocation, a two-layer distributed parallel structure is constructed. A large number of CPU cores can perform the individual evaluation in parallel, signifi-

cantly reducing time consumption. 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. 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 the redundancy among features, with respect to the generated feature subset, which 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 describes the feature number of the generated feature subset, 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} are respectively the feature number 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 among features:

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

where N_S is the number of training samples, α and β denote features, $F_\alpha\left(i\right)$ ($F_\beta\left(i\right)$) represents the α (β) feature value of the i-th 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 value, which is a positive value in the range of $[0.0, 1.0]$, between features α and β.

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

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

where S_F denotes the set of features selected and $N_f = |S_F|$ represents the cardinality of the set. If the generated feature subset only contains one feature, f_R cannot be calculated as above; therefore, the following formula is applied:

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

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

2.4. Multiobjective View

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

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

From the above specifications of these three objectives, it is easy to comprehend that their value ranges all lie in $[0.0, 1.0]$, and 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 framework of the algorithm. Subsequently, based on this framework, by utilizing different encoding methods and an additional adaption improvement, we describe the proposed algorithms one by one. Finally, the feature number constraint and parallelism details are provided.

3.1. Overall Framework

3.1.1. Grouping

In microarray datasets, there are numerous features, and each feature 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, whereas by separating variables into several groups and optimizing each variable group under the CC framework [21], the original problem can be separated into several low-dimensional ones, which can be addressed more effectively and efficiently. For this purpose, we randomly separate all variables into N_G uniform groups.

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

3.1.2. Evolution of Variables in the Current Group

For each parent individual, to generate the variables in the current group for an offspring individual, another single individual (for binary-encoded algorithms) or two individuals (for the weight-encoded algorithm) are selected from the parent population; then, crossover or adaptive differential evolution (DE) [23, 24] will be applied, which will be detailed in Sections 3.2.1 and 3.3.1.

3.1.3. Evolution of Remaining Variables in the Other Groups

As in DPCCMOLSEA [22], for the remaining variables in an offspring, they are the crossover result of its parent and another two randomly selected individuals in the parent population. The crossover rate adopts the adaptive behavior as in JADE [25], which can be detailed as follows:

$$u_j^{g+1,i} = \begin{cases} x_j^{g,i}, & \text{if } r \leq CR^i \\ x_j^{g,r_1}, & \text{else if } r_0 \leq 0.5 \\ x_j^{g,r_2}, & \text{otherwise} \end{cases} \quad \text{s.t. } j \notin S_k^G \tag{7}$$

where g, i and j denote the generation number, the individual index and the variable index, respectively, thus, $x_j^{g,i}$ represents the 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 generated in the range of $[0.0, 1.0)$, S_k^G is the variable set 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 according to the Gaussian distribution with the location parameter of μ and the scale parameter of σ, here, CR^i is bounded 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 as 0.5, 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 successfully updated in the prior generation and its cardinality, respectively.

3.1.4. Mutation

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

3.2. Weight-Encoded Algorithm

By encoding each feature via a weight value, we propose the distributed parallel cooperative coevolutionary multiobjective large-scale evolutionary

algorithm for feature selection with weight encoding, denoted as DPCCMOLSEA-FS-w. In this algorithm, the selection priority of one feature is represented by its weight; in other words, the higher the weight is, the more likely it is to be selected.

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

$$\begin{gathered} \underbrace{x_0, \ldots, x_{N_F-1}}_{\text{Feature weights}}, \quad \underbrace{x_{N_F}}_{\text{Feature number}} \\ \text{s.t. } x_j \in [0.0, 1.0], j = 0, \ldots, N_F. \end{gathered} \tag{10}$$

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 selected feature number 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.

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 JADE [25], formulated as follows:

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

where $r_3 \neq r_4 \neq i$ are two randomly selected individuals and F^i denotes the scaling factor of individual i, which, similar to JADE [25], 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 according to the Cauchy distribution with the location parameter of μ and the scale parameter of σ, here, F^i is restricted in the range of $(0.0, 1.0]$, thus, it will be generated again if

$F^i \leq 0.0$ and will be truncated to 1.0 if $F^i > 1.0$. Initially, $\mu_F = 0.5$, and it will be updated as follows:

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

where $c = 0.1$ denotes the learning factor and S_F represents the set of indices of the individuals successfully updated in the prior generation.

3.2.2. Mutation

Polynomial mutation (PM) [26] is utilized to adjust the variable values with the 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, $x^{g+1,i}, i = 1, \ldots, NP$, will be generated. Here, NP is the population size.

3.3. Binary-Encoded Algorithms

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

$$\begin{aligned} &\underbrace{x_0, \ldots, x_{N_F-1}}_{\text{Binary encoding}} \\ &\text{s.t. } x_j \in \{0, 1\}, j = 0, \ldots, N_F - 1 \end{aligned} \tag{15}$$

3.3.1. Evolution

To evolve the variables in the currently considered group as mentioned in Section 3.1.2, for both DPCCMOLSBEA-FS and DPCCMOLSABEA-FS, the process can be described as follows:

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

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 of individual i. The difference between DPCCMOLSBEA-FS and DPCCMOLSABEA-FS depends on the value of CR_B^i:

1. For DPCCMOLSBEA-FS, $CR_B^i = 1.0$ for all $i = 1, \ldots, NP$.
2. For DPCCMOLSABEA-FS, CR_B^i is generated adaptively, similar to CR^i, detailed 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 initial value of μ_{CR}^B is 0.9, and it is updated as follows:

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

where $c = 0.1$ denotes the learning factor and S_{CR_B} and $|S_{CR_B}|$ represent the set of indices of individuals successfully updated in the prior generation and its cardinality, respectively.

3.3.2. Mutation

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

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

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 at most N_F^{th} features can be selected to form a feature subset. In this type of binary-encoded algorithm, from the generated offspring, the cardinality of the corresponding feature subset can exceed N_F^{th} or be less than 1; thus, an adjustment procedure is applied.

The adjustment procedure includes two phases, as follows:

1. Reset the feature number: set the feature number, 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 above N_F^{th}, then randomly 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 Parallelism Details

MOEAs are based on population and iteration. During the evolution, numerous generations of populations will be produced, and a large number of fitness evaluations (FEs) are performed. To reduce the time consumption, 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, compared to which the evolution of the population and other objectives are very efficient. Thus, the classification error objective is considered to be the only time-consuming procedure for analysis. In this study, the nearest neighbor classifier (1-NN) is employed; thus, the time consumption is proportional to the selected feature number. For the considered microarray data, the feature number can reach more than tens of thousands, for which the cardinality of the feature subset can be quite high and the time consumption will be intolerable. By applying the constraint, only a small number of features are considered in the objective evaluation; thus, the time consumption can be greatly reduced.

B) Distributed parallelism: The former strategy reduces the time consumption of each FE; however, the evaluations of all individuals in the offspring population are conducted in serial. By taking advantage of the variable

groups and the population-based evaluation, we construct the following distributed parallel structure:

a) Assume that there are N_C CPU cores and that the group number is N_G. For each group, we form a population with NP individuals, and we divide the CPU cores uniformly to these populations, as follows:

$$N_C^i = \frac{N_C}{N_G} \quad \text{s.t. } i = 1, \ldots, N_G. \tag{20}$$

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

b) Then, the individuals in each population are separated to the CPUs in the population for FEs, as follows:

$$N_C^{i,j} = \frac{NP}{N_C^i} \quad \text{s.t. } i = 1, \ldots, N_G, j = 1, \ldots, N_C^i. \tag{21}$$

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 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, all CPUs are utilized — each CPU evaluates the individuals allocated, and all CPUs operate in parallel. Thus, the evaluation process is parallel at the individual level.

C) Sample-wise parallelism: To observe the evolution behavior of MOEAs during the optimization process, every predefined number of generations, we record the fitness values of the individuals of the current population. In addition, we also test the individuals on the test set and record the results. Although the number of recordings is extremely small compared to the overall generation number, if the evaluation on the test set is performed in serial, its time consumption exceeds that of the optimization process with distributed parallelism.

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

subset, which is broadcast to all other N_C CPUs. Then, the classification burdens of all test samples are uniformly allocated to all CPUs, and all the CPUs perform their own tasks in parallel. Finally, the root CPU gathers the classification results from all CPUs.

After this parallelism, the overall time consumption is significantly reduced, and the benefit of the parallelism of the optimization process is not impeded by the recording process.

4. Experimental Analysis

4.1. Microarray Datasets

Few years ago in the twentieth century, the study of genes was very low in efficiency with only one or few genes checked at one time. While in the living things, there are substantial numbers of genes, and for an instance, we humans own approximately $20,000$ genes. Consequently, the investigation process can take a scientist's lifetime. Fortunately, by the aid of microarray technology, the expression situations of numerous genes can be investigated at once. Compared to healthy cells, there seems to be something wrong with the gene expression. Via microarray the expression levels of numerous genes of healthy and cancer cells (or different cancer cells) can be obtained; then through feature selection and classification, the potential genes causing the cancer (or the relationship among cancers) can be detected, facilitating the study of the mechanism. Additionally, by comparing the difference of gene expression before and after a therapy, the mechanism of treatment 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 feature number and low sample instance number. For each dataset, the data are normalized with respect to each feature; then, we generate a training set using the stratified bootstrap. 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`.

Table 1: Details of the Datasets

Dataset	File name	#Gene	#Sample	#Class
childhood ALL	(ALLGSE412_poterapiji)	8280	60	4
childhood ALL	(ALLGSE412_pred_poTh)	8280	110	2
AML prognosis	(AMLGSE2191)	12625	54	2
breast & colon cancer	(BC_CCGSE3726_frozen)	22283	52	2
breast cancer	(BCGSE349_350)	12625	24	2
bladder cancer	(bladderGSE89)	5724	40	3
brain tumor	(braintumor)	7129	40	5
CML treatment	(CMLGSE2535)	12625	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)	12625	50	4
leukemia	(leukemia)	5147	72	2
lymphoma & leukemia	(LL_GSE1577)	15434	29	3
lymphoma & leukemia	(LL_GSE1577_2razreda)	15434	19	2
lung	(lung)	12600	203	5
lung cancer	(lungGSE1987)	10541	34	3
medulloblastoma	(meduloblastomiGSE468)	1465	23	2
MLL	(MLL)	12533	72	3
prostate	(prostate)	12533	102	2
prostate cancer	(prostateGSE2443)	12627	20	2
SRBCT	(SRBCT)	2308	83	4

4.2. Utilized Algorithms and Parameter Settings

For comparison, four algorithms are utilized, 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].

For a fair comparison, the population size of all algorithms is fixed to 120. In particular, for CCGDE3, there are two swarms, each of which has 60 individuals; for CMODE, there are three swarms for three objectives, with a size of 20 for each of them, 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, in which $F = 0.5$ and $CR = 1.0$, and the same settings are adopted in DE employed in MOEA/D. For CMODE, adaptive DE variants are utilized, and 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, denoted as CCGDE3-FS, CCGDE3-FS-w, CMODE-FS, CMODE-FS-w, MOEA/D-FS, MOEA/D-FS-w, NSGA-II-FS and NSGA-II-FS-w. For the binary encoding, the feature adjustment in Section 3.3.3 is also added. 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 the binary representation issue, a threshold (i.e., the mid-value) is utilized.

For the proposed algorithms, for DE, F follows the adaptive strategy in JADE [25], while CR is fixed to 1.0 or adaptive as in JADE. The number of variable groups is simply set to 5. For PM, its 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 which they have the same parameter settings.

4.3. Analysis

In the following images (Fig. 1 to Fig. 24), there are three columns, 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, the test set and the weighted sum of the former two, 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} — classification errors on the training set and the test set, respectively. Nevertheless, the remaining two objectives are only related to the inherit property of the feature subset.

4.3.1. Hypervolume Indicator

In Figs. 1 to 6, we illustrate the hypervolume (HV) indicator [27] values, which can simultaneously measure the distribution and convergence of the obtained nondominated solutions during the evolution. For the reference point in HV, because all objective values are not above 1, we set it as $(1, 1, 1)$.

For the first column, we observe that the HV indicator values are monotonously increasing, indicating that the qualities of the obtained nondominated solutions are being ameliorated. Specifically, for the first approximately 10^4 FEs, the performance of all algorithms is improved rapidly; however, during the following FEs, the improvement is minimal. For the second column, the HV indicator curves are not as monotonous because the classification error is derived based on the test set. At the beginning, the performance is improving quickly; 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 situations become quite worse for some cases. Finally, by simultaneously considering the training and testing performance, the final optimization results are illustrated in the third column. Due to the nonmonotonic behavior of the results on the test set, the final HV indicator values are also not monotonic during the evolutionary process.

Regarding the optimization performance of different algorithms, we have the following:

1. For the training results, the differences among various algorithms are minimal. However, the proposed algorithms can always achieve the best results. For the two types of encodings, the weight-encoded ones converge faster than the binary-encoded ones. The ranking of other algorithms depends on the considered dataset.

2. For the test results, the performance varies. In most cases, overfitting occurs. The common trend is that the indicator value increases rapidly to the maximum value within a very small number of FEs; then, overfitting occurs, and the indicator value decreases or fluctuates. The only exception is the CCGDE3 algorithm, as its evolutionary curve only has slight fluctuations without a large drop, while the binary-encoded C-CGDE3 is much better than the weight-encoded one. However, with the view of the whole process, the peak HV indicator values are always obtained by the proposed algorithms.
3. For the final results, the proposed algorithms can generally obtain the peak HV indicator values.

4.3.2. Classification Error

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

1. For the training data, in most cases, most algorithms can achieve zero classification error. Especially, for 12 datasets of 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 almost during the whole evolutionary process. The difference among all algorithms is the convergence issue, which is trivial. For the proposed algorithms, with respect to all datasets, there exists at least one algorithm that achieves zero classification error, and the same situation is also true for CMODE.
2. For the test data, the minimum classification error fluctuates violently during evolution, which is more complex with respect to that on the training data; thus, we focus on the lowest value over the whole evolution. For 20 out of 24datasets, zero classification error can be obtained; in particular, for the datasets of braintumor, MLL and prostate, only the proposed algorithms achieve zero classification error. On the contrary, the classification errors obtained by CMODE are usually quite high, indicating the serious overfitting problem.
3. Comprehensively considering the training and test data, zero classification error indicates that the corresponding feature subset classifies all samples correctly in both the training data and the test data,

which is the expected result. For the following 13datasets of ALL-GSE412_pred_poTh, BC_CCGSE3726_frozen, BCGSE349_350, bladderGSE89, DLBCL, EWSGSE967, EWSGSE967_3class, gastricGSE2685_2razreda, LL_GSE1577, LL_GSE1577_2razreda, MLL, prostateGSE2443 and S-RBCT, zero classification errors are observed. Specifically, for the datasets of BC_CCGSE3726_frozen and bladderGSE89, only the proposed algorithms can obtain zero classification error, while for all other datasets, the proposed algorithms are also generally not worse than their counterparts. And the binary-encoded ones are better than the weight-encoded one.

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

4.3.3. Feature Number

For 15 datasets of ALLGSE412_pred_poTh, BC_CCGSE3726_frozen, BCGSE349_350, CMLGSE2535, DLBCL, EWSGSE967, EWSGSE967_3class, gastricGSE2685, gastricGSE2685_2razreda, leukemia, LL_GSE1577, LL_GSE1577_2razreda, lung-GSE1987, meduloblastomiGSE468 and prostateGSE2443, illustrated in Figs. 13 to 18, all algorithms can generate a subset within 10 features ($f_N = \frac{N_f}{N_F^{th}} = \frac{10}{50} = 0.2$), except some special cases of CCGDE3. For the other datasets, the feature number fluctuates along the evolution. With respect to the training data and test data, however, in contrast to the status analyzed in the previous sections, the evolutionary curves share similar characteristics.

4.3.4. Feature Redundancy

For the feature redundancy objective, the evolutionary curves in Figs. 19 to 24 are characterized by undulations, which is subtle to comprehend. By formulating the feature redundancy as an objective, during the evolution, it will lead the algorithm to form good-performing feature subsets with 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 consumption over all datasets is listed in the second line from the bottom, and the speedup ratios are the values in parenthesis. With respect to the proposed algorithms, all

other MOEAs are at least one magnitude slower, except for the simple algorithm of CCGDE3-FS. For the proposed distributed algorithms, the number of utilized CPU cores is 60. Additionally, the speedup ratios with respect to 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.

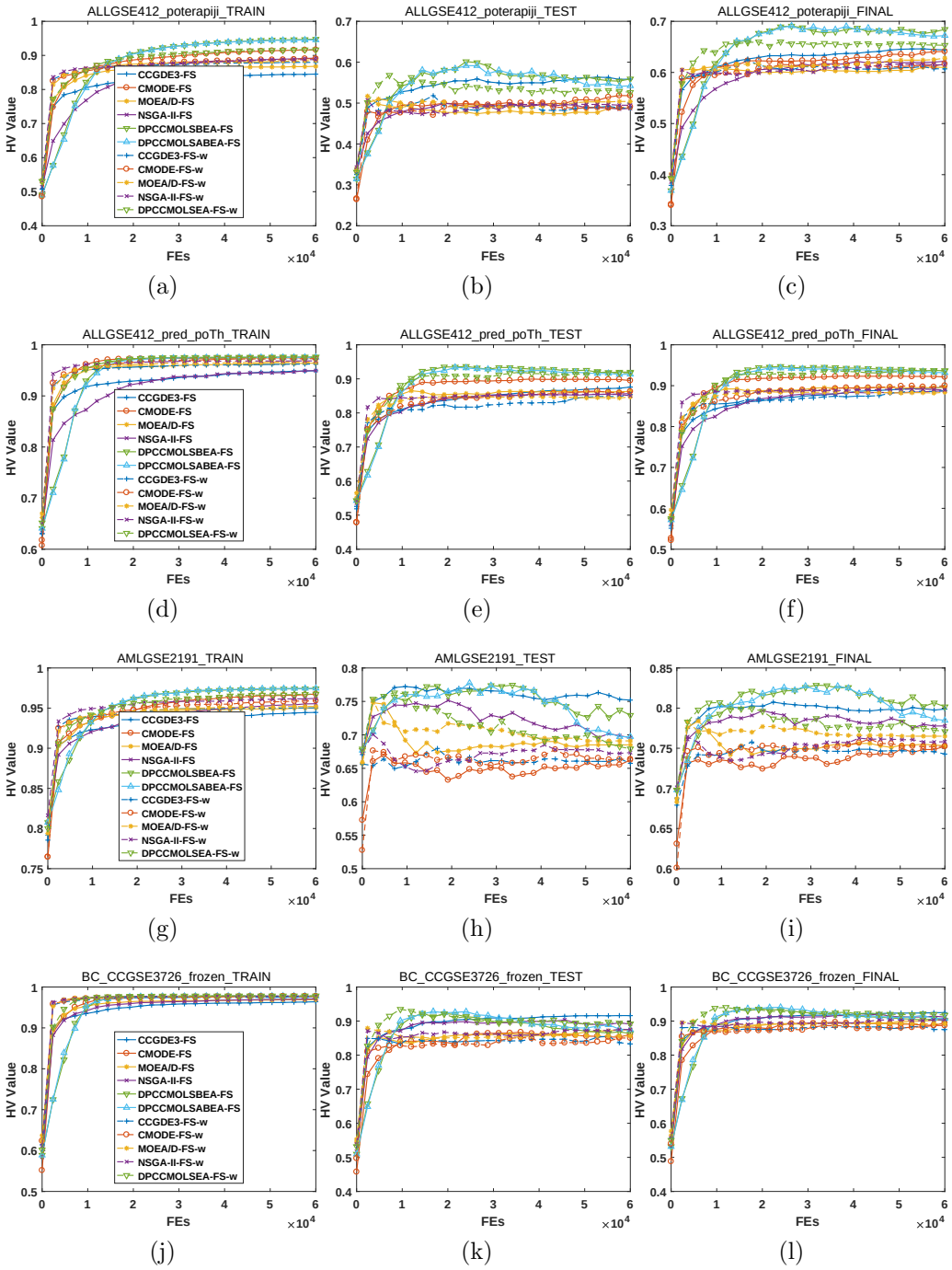

Figure 1: Illustration of HV during evolution.

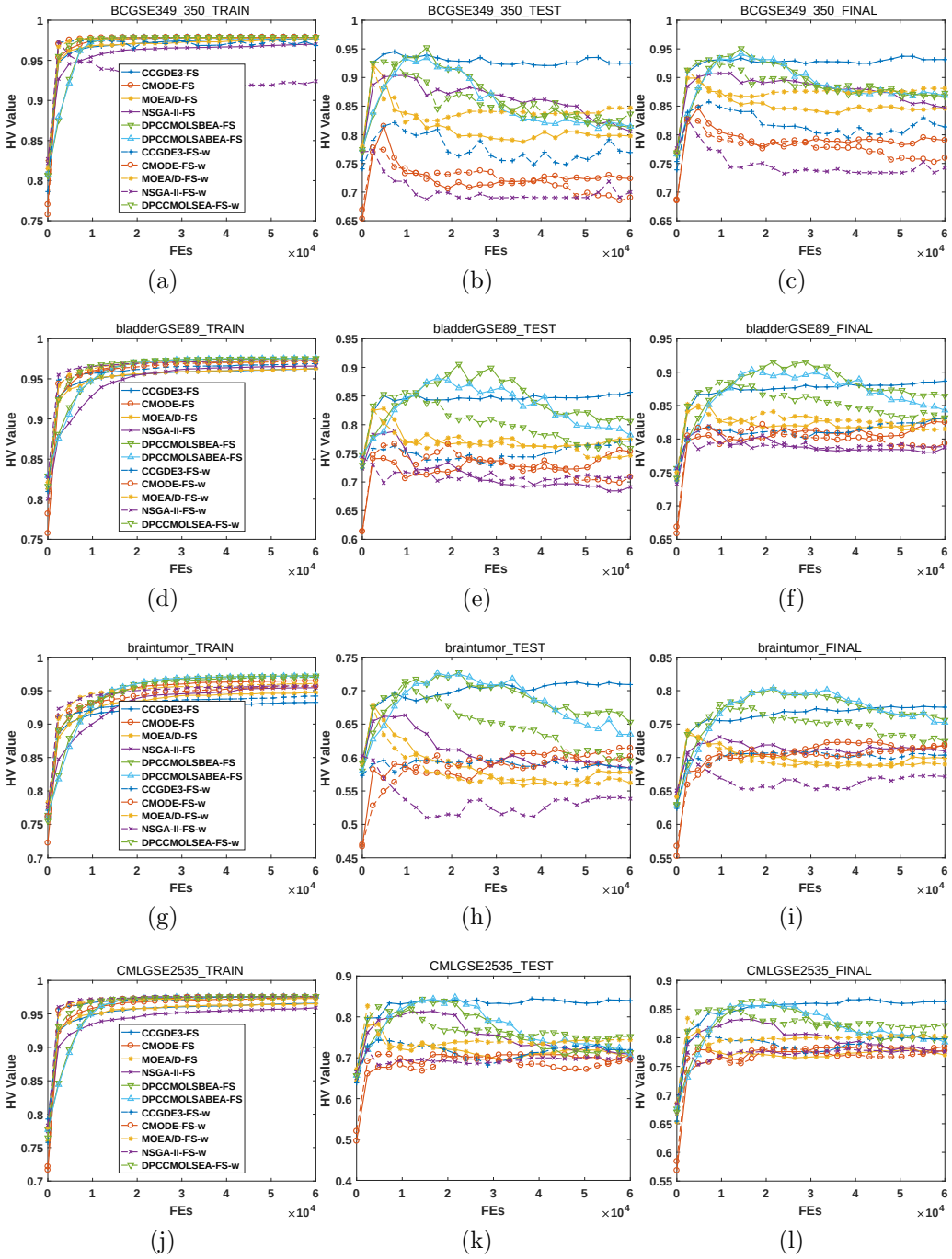

Figure 2: Illustration of HV during evolution.

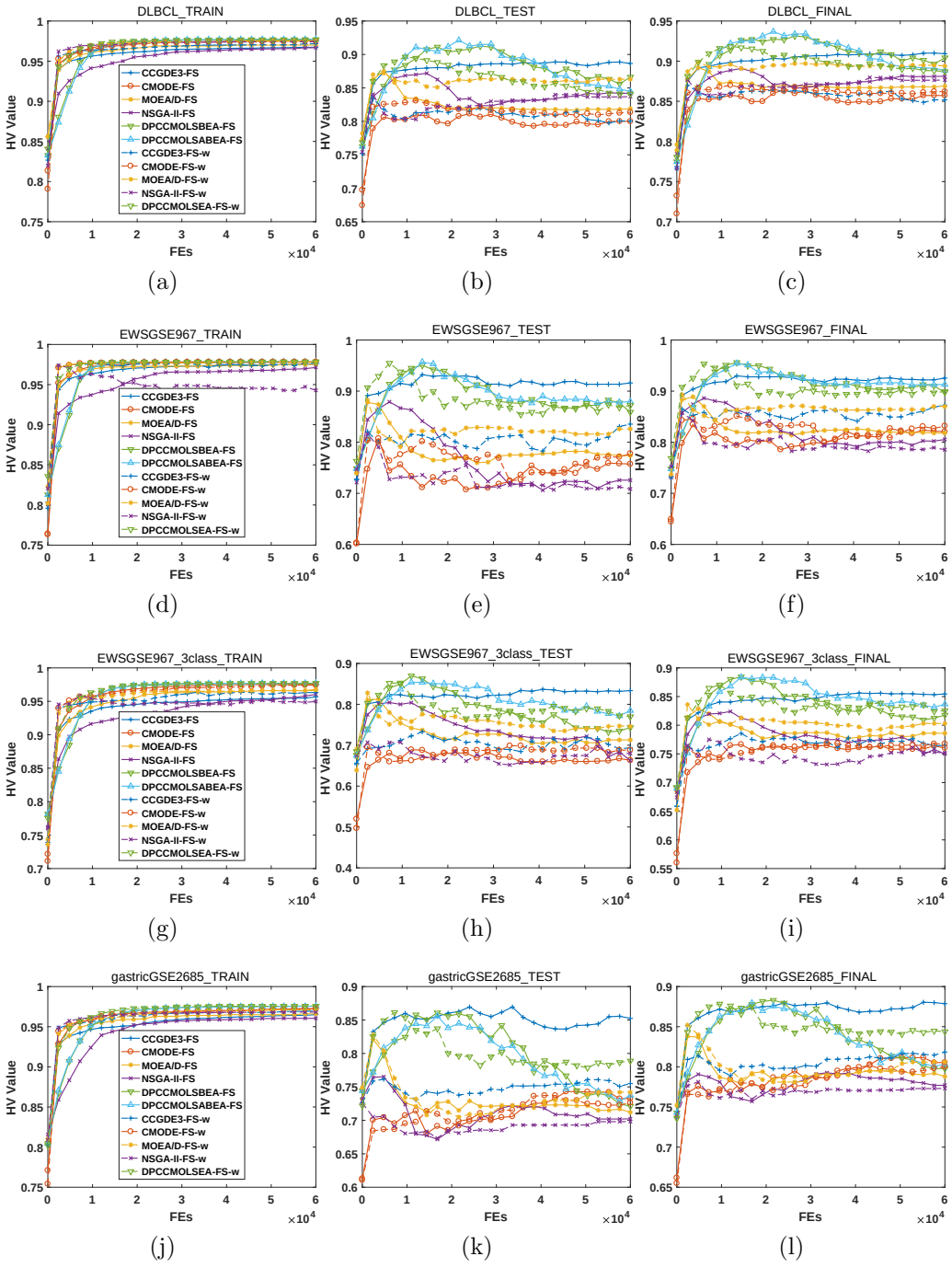

Figure 3: Illustration of HV during evolution.

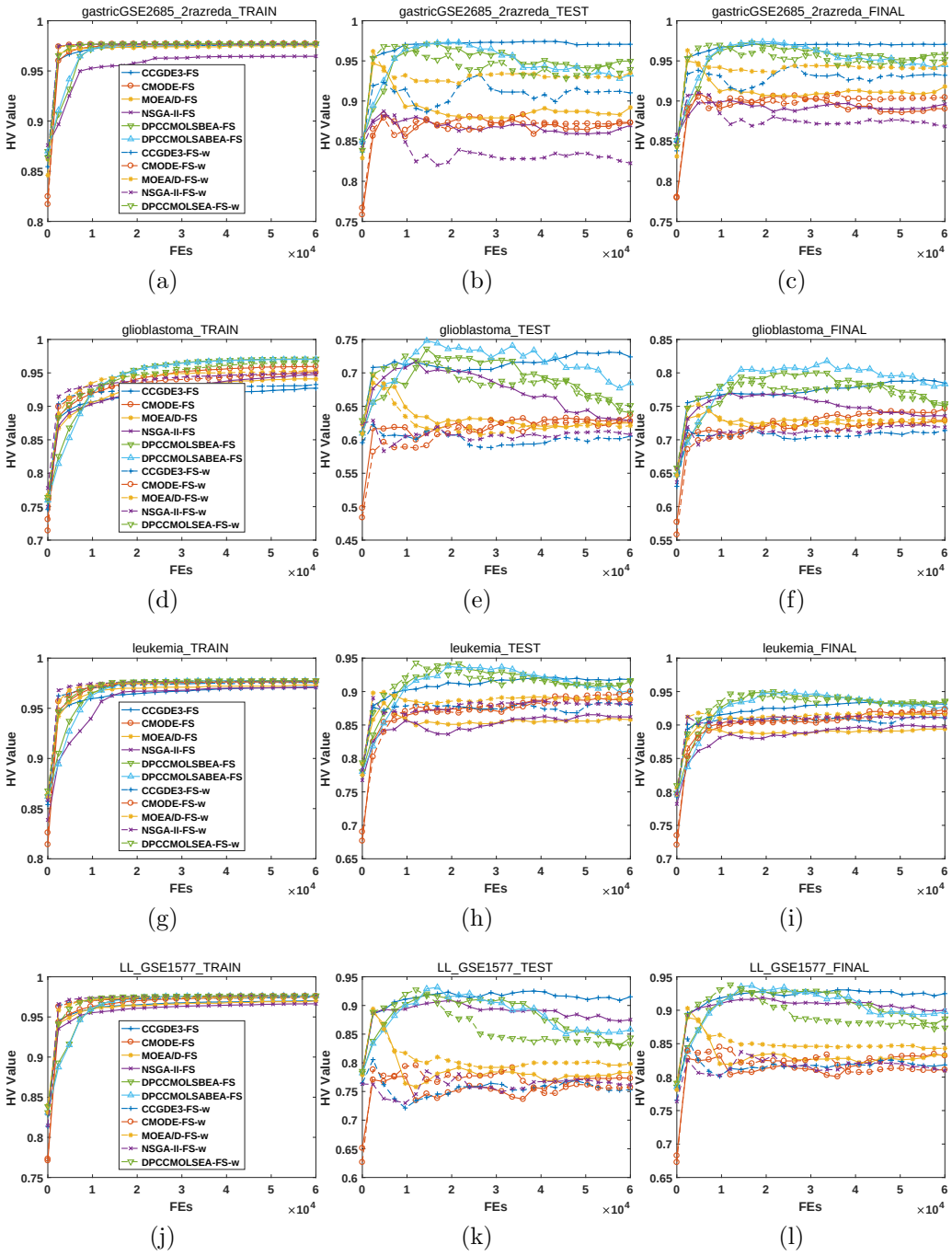

Figure 4: Illustration of HV during evolution.

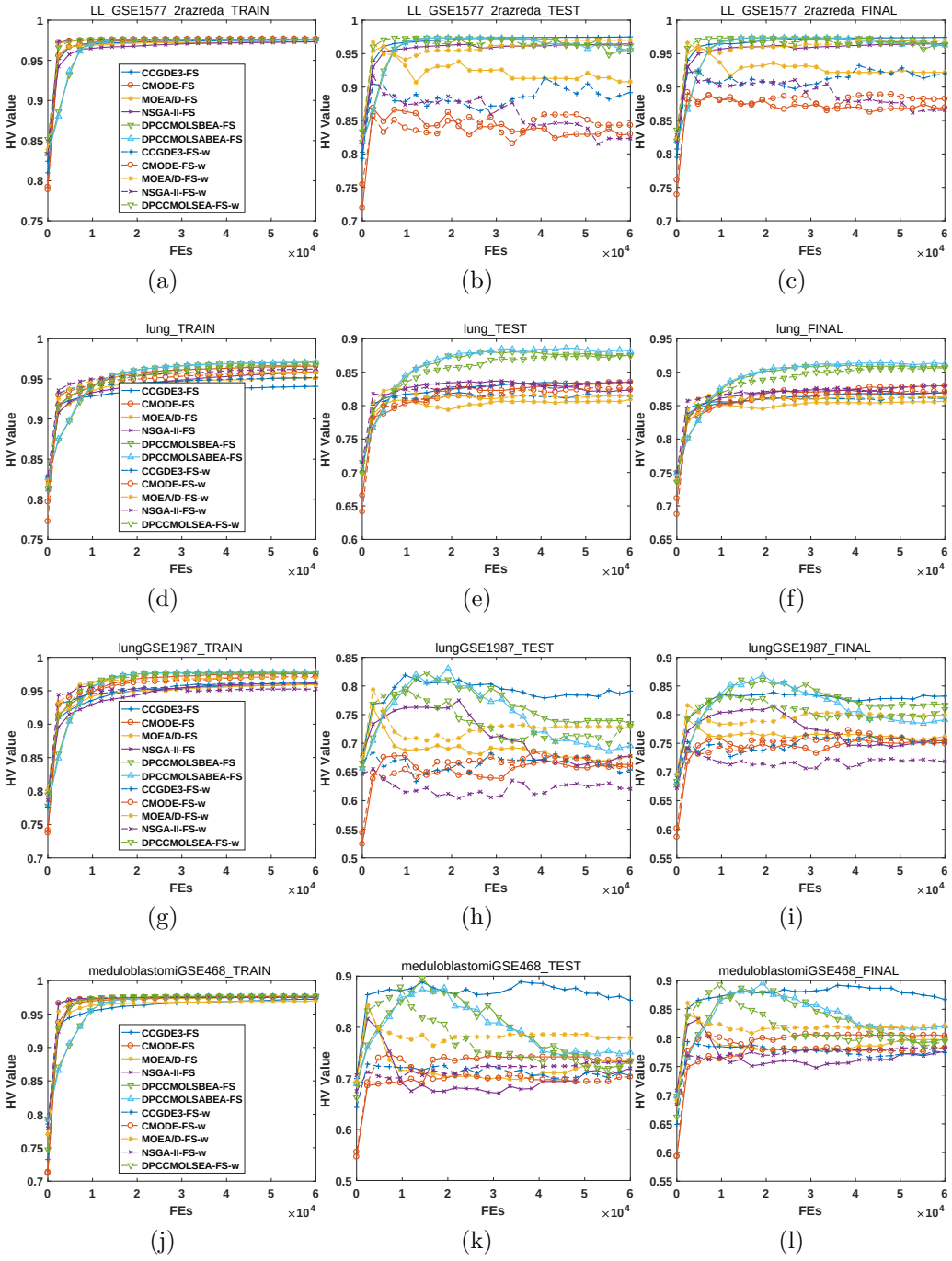

Figure 5: Illustration of HV during evolution.

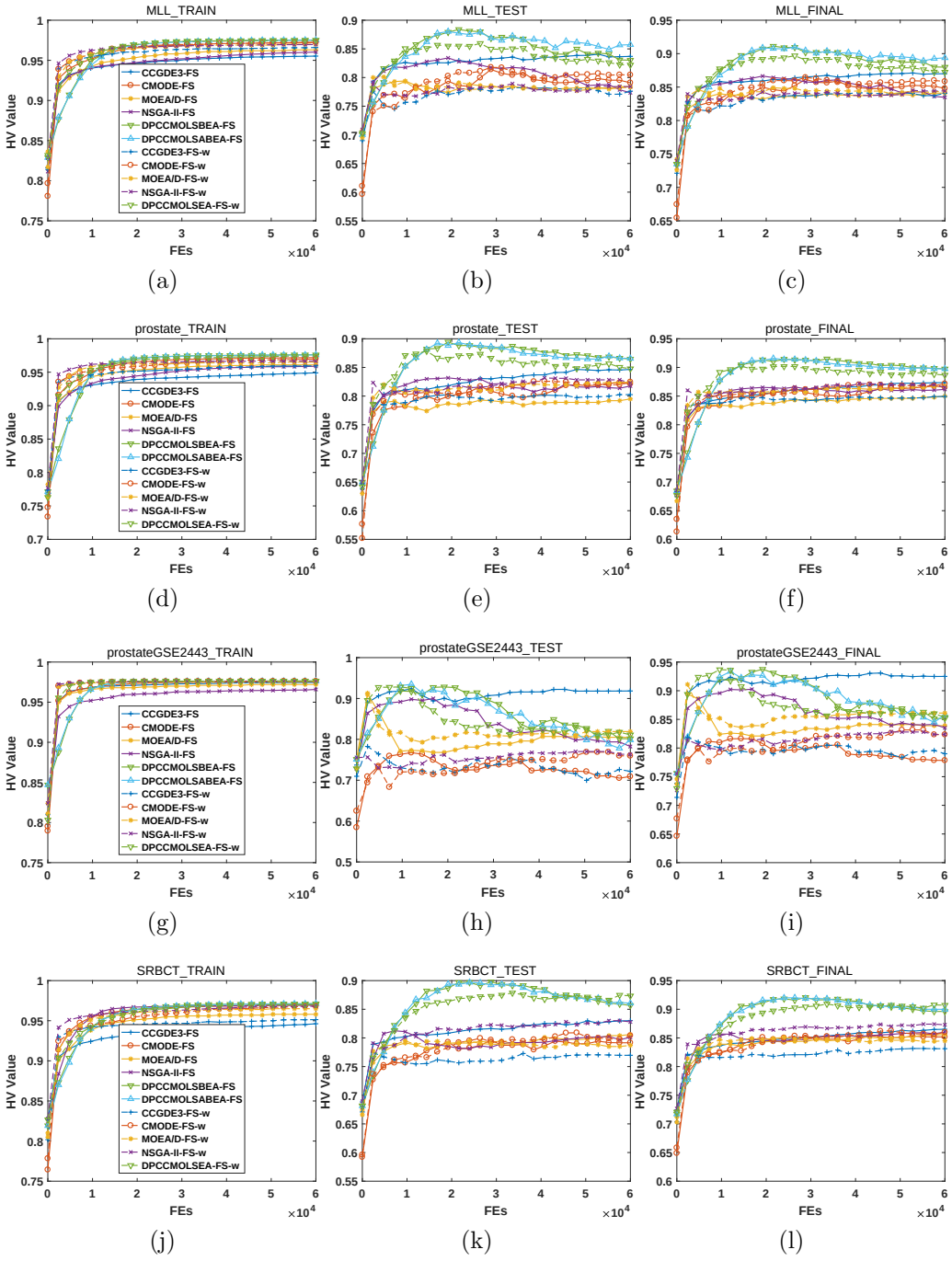

Figure 6: Illustration of HV during evolution.

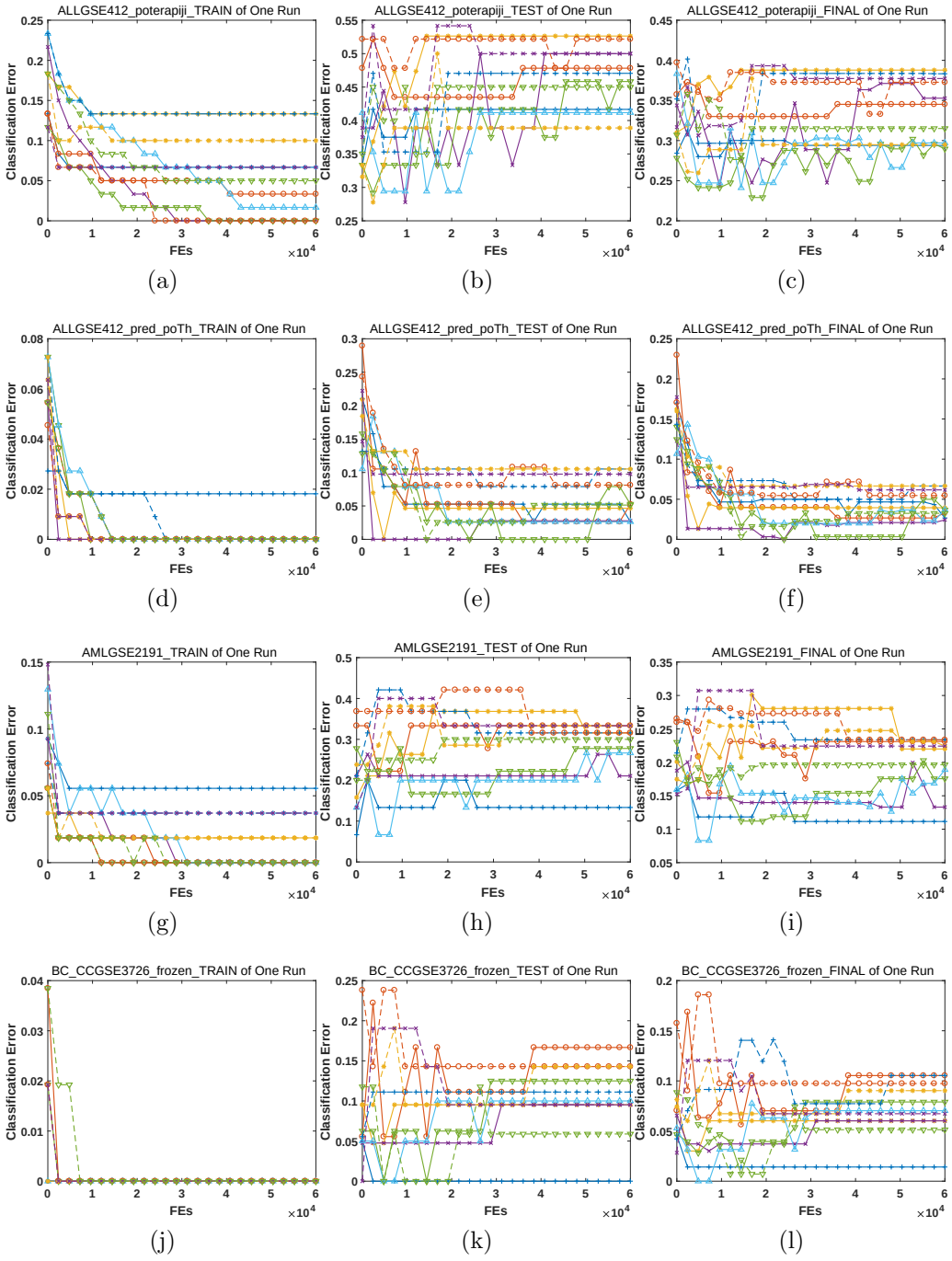

Figure 7: Illustration of classification error during evolution.

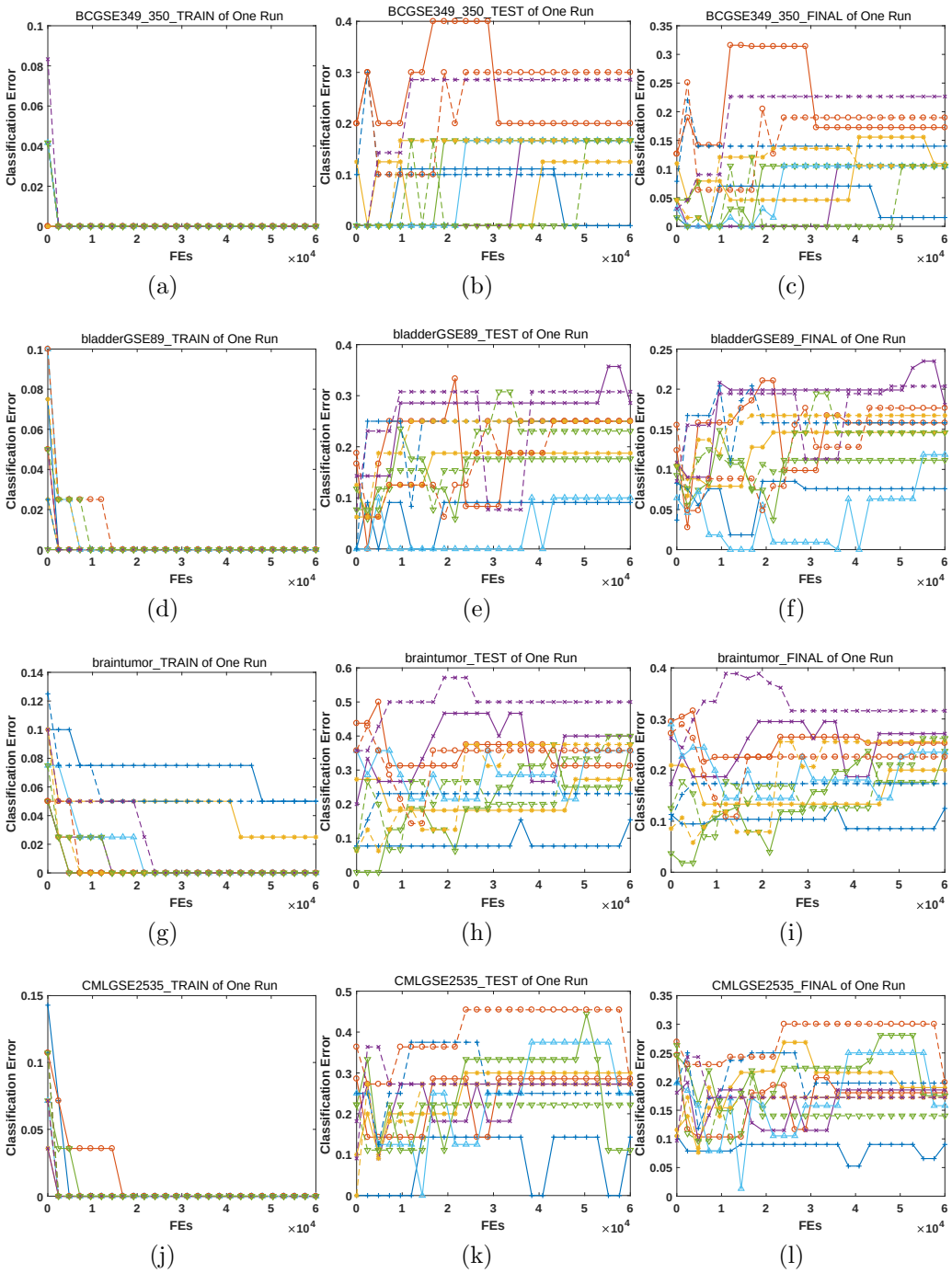

Figure 8: Illustration of classification error during evolution.

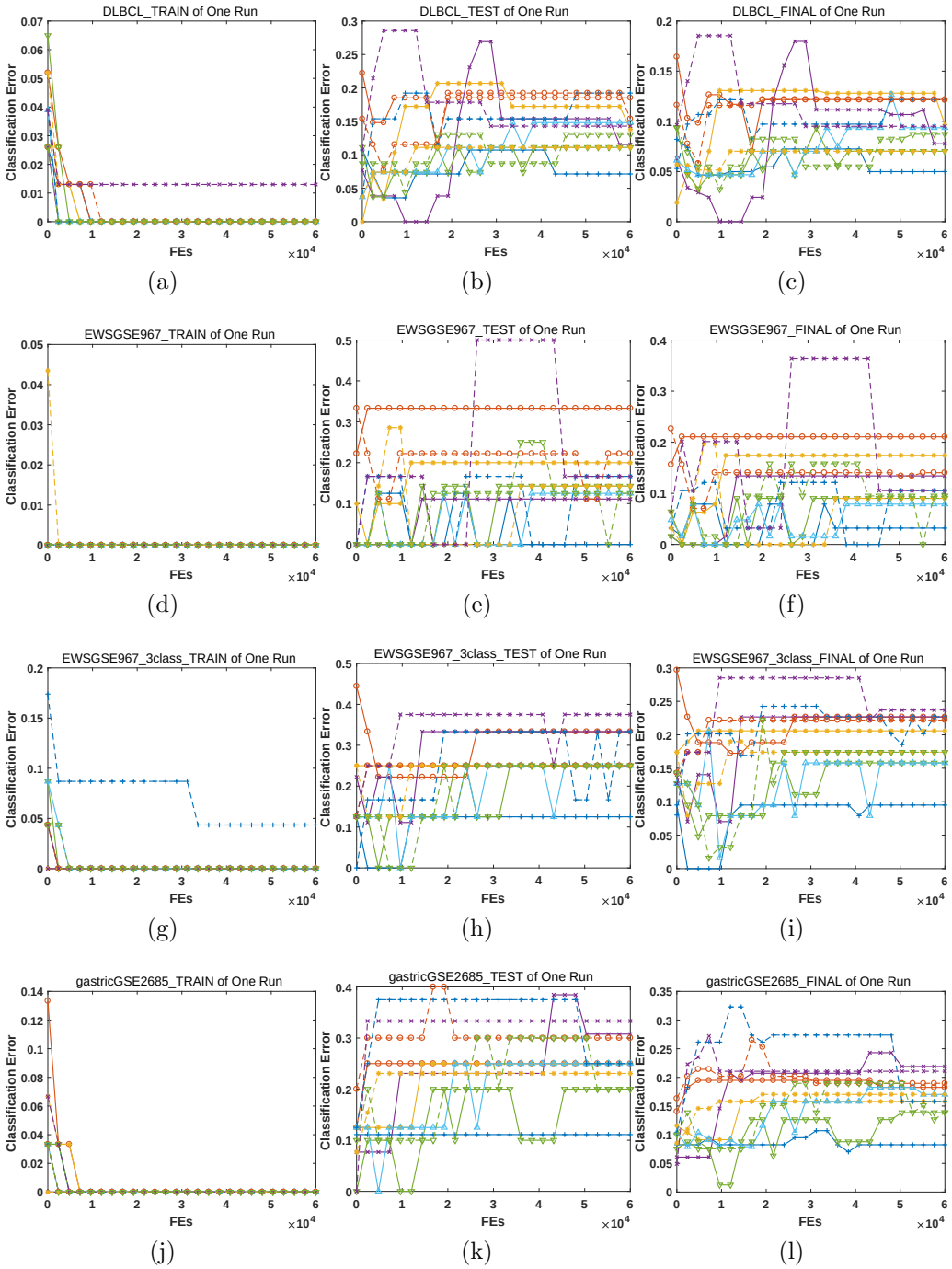

Figure 9: Illustration of classification error during evolution.

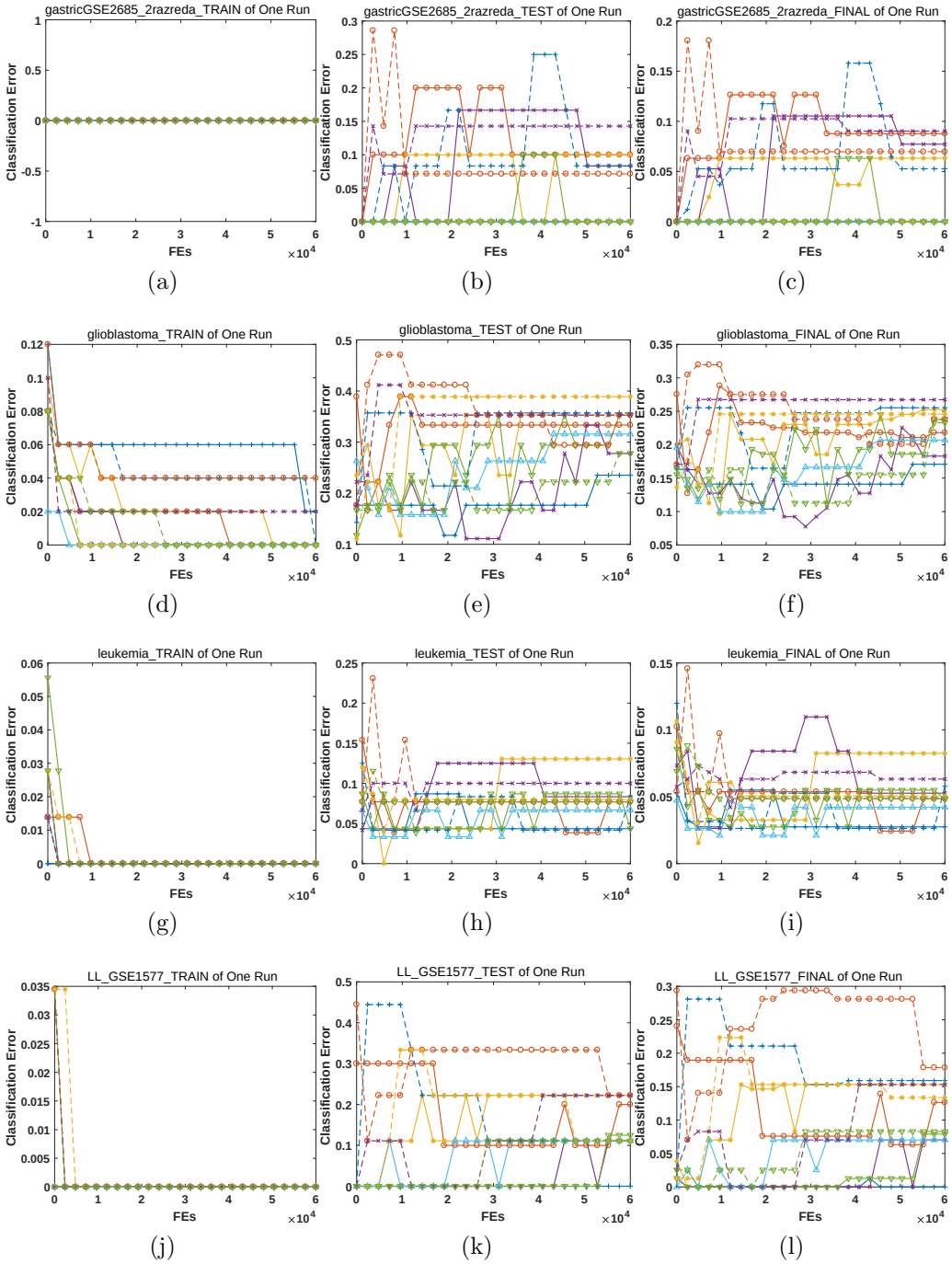

Figure 10: Illustration of classification error during evolution.

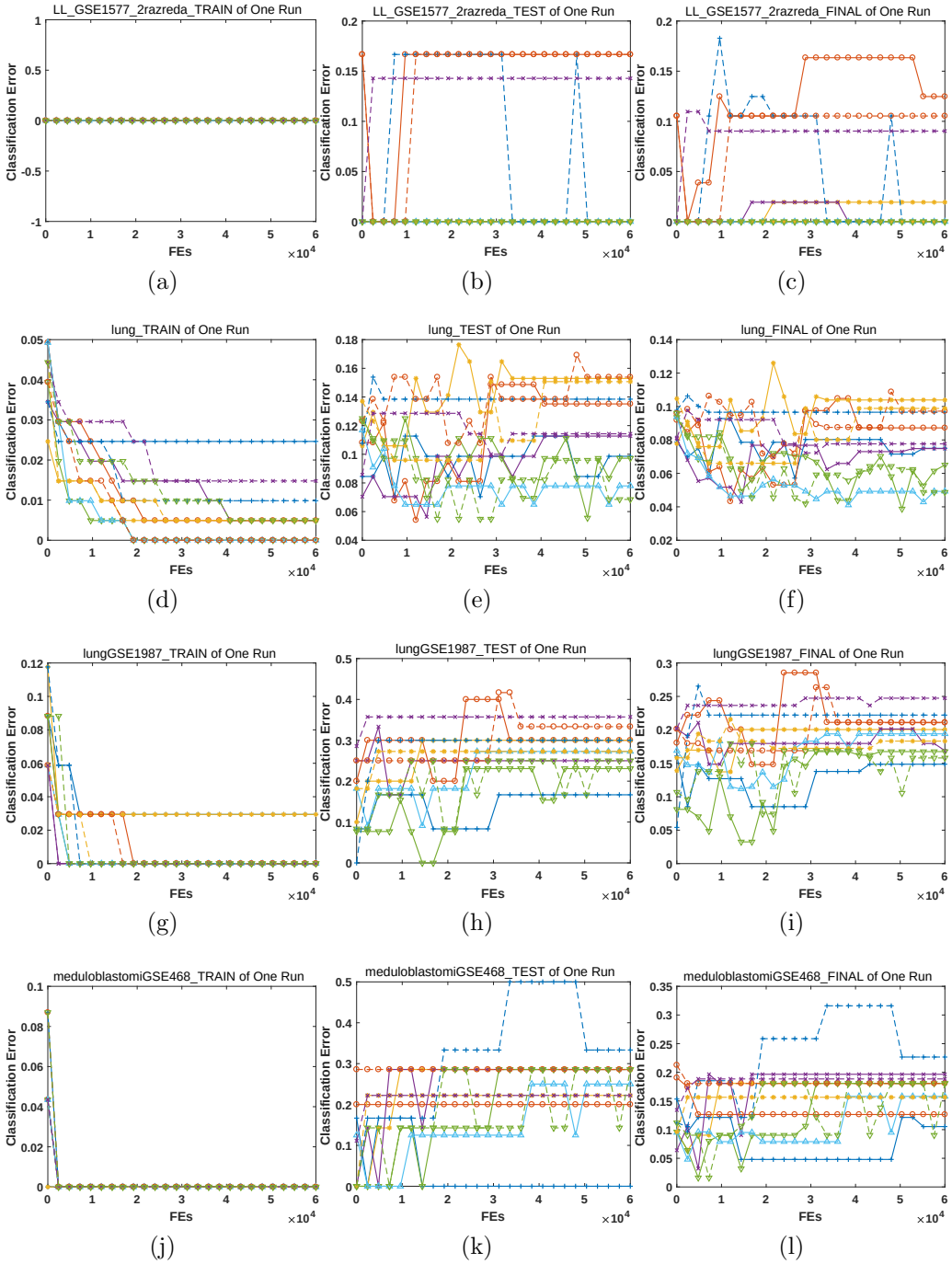

Figure 11: Illustration of classification error during evolution.

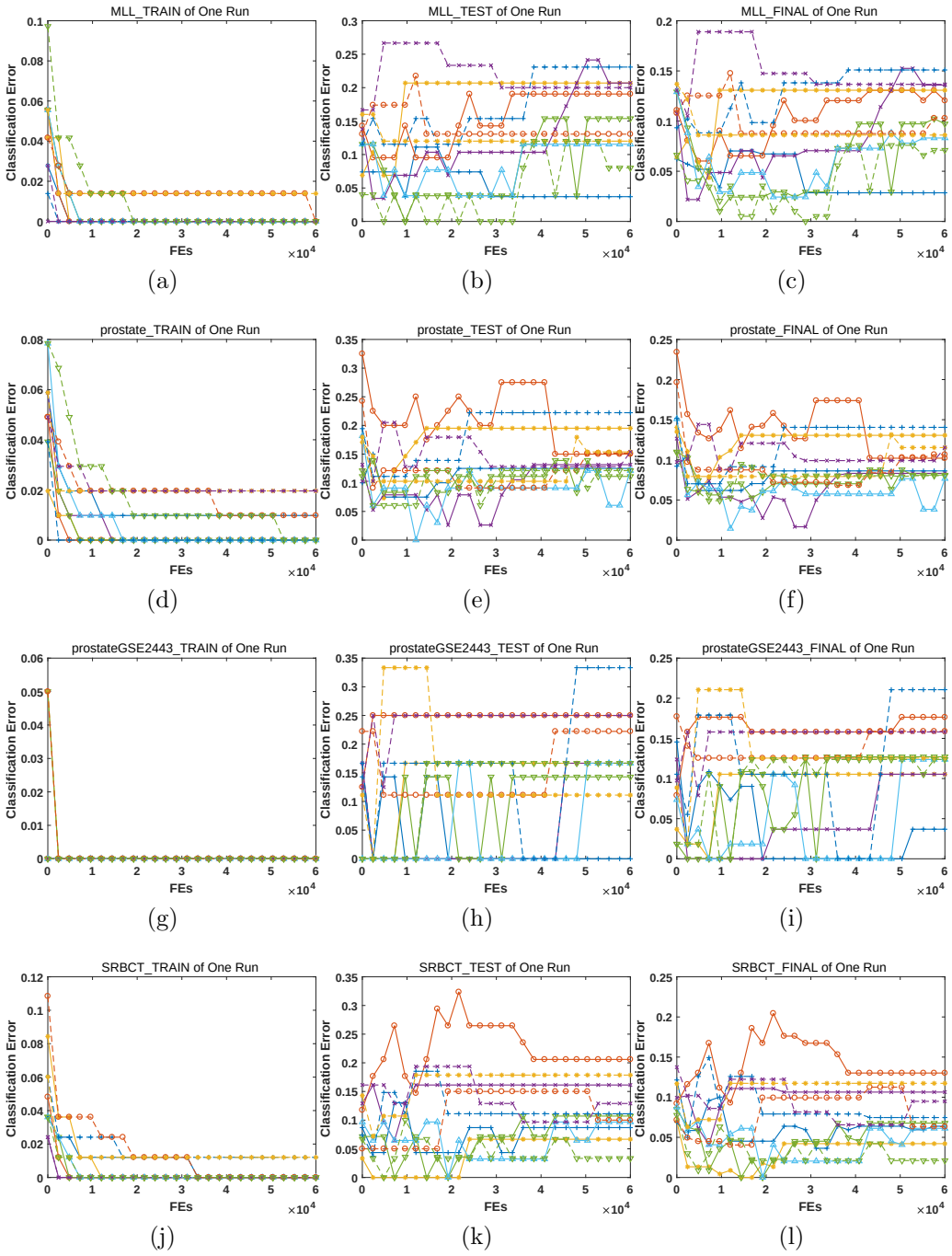

Figure 12: Illustration of classification error during evolution.

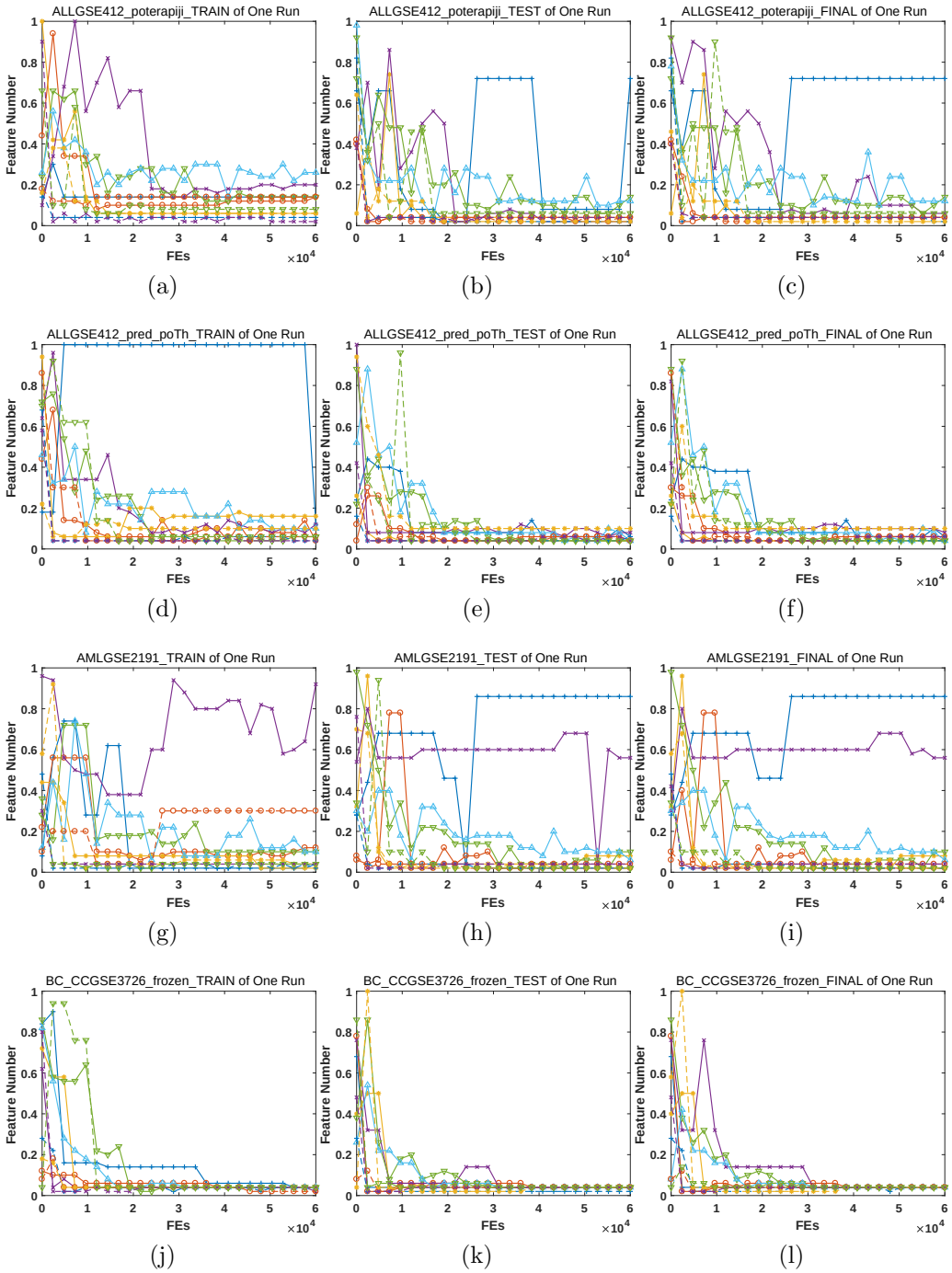

Figure 13: Illustration of feature number during evolution.

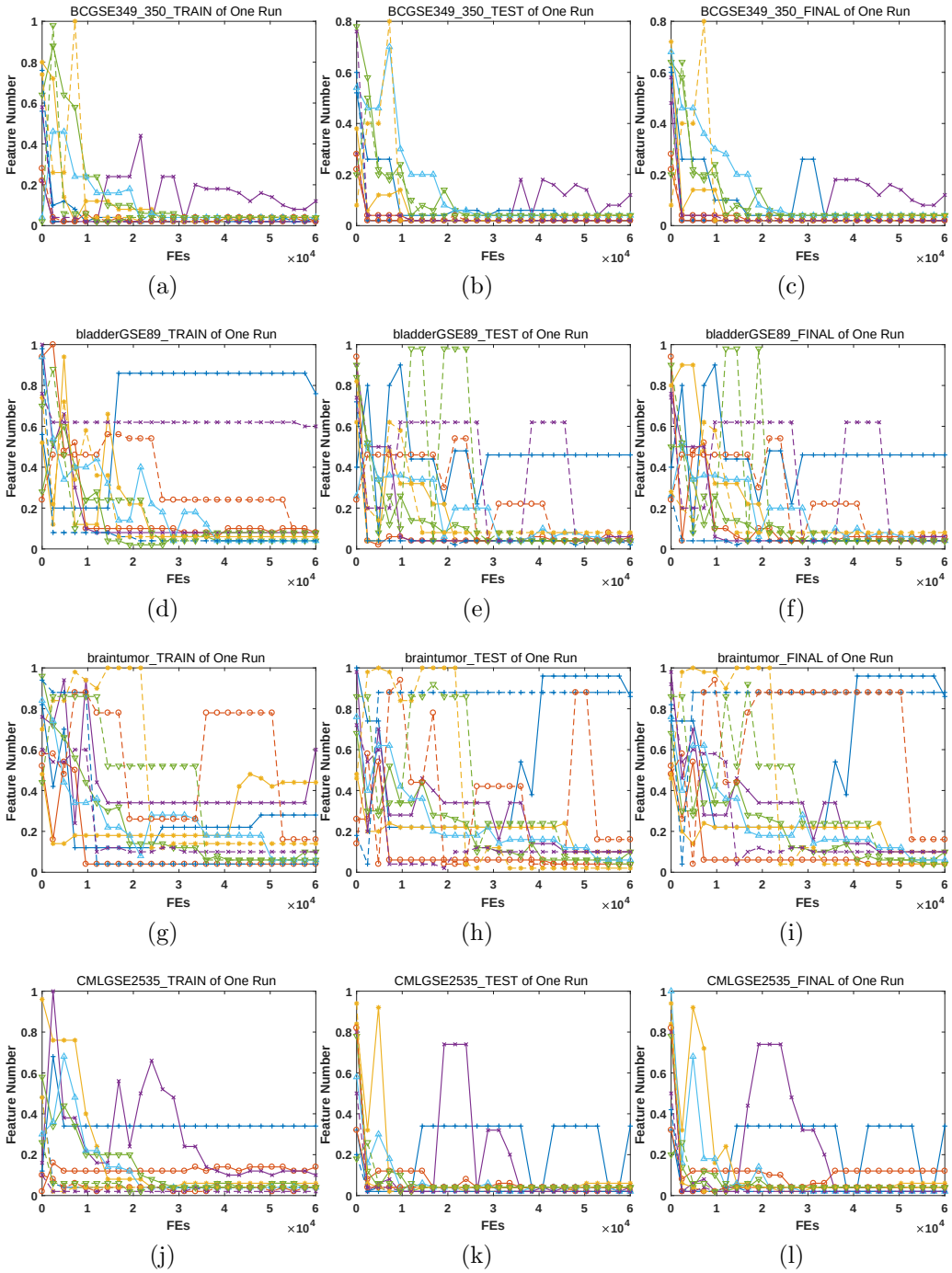

Figure 14: Illustration of feature number during evolution.

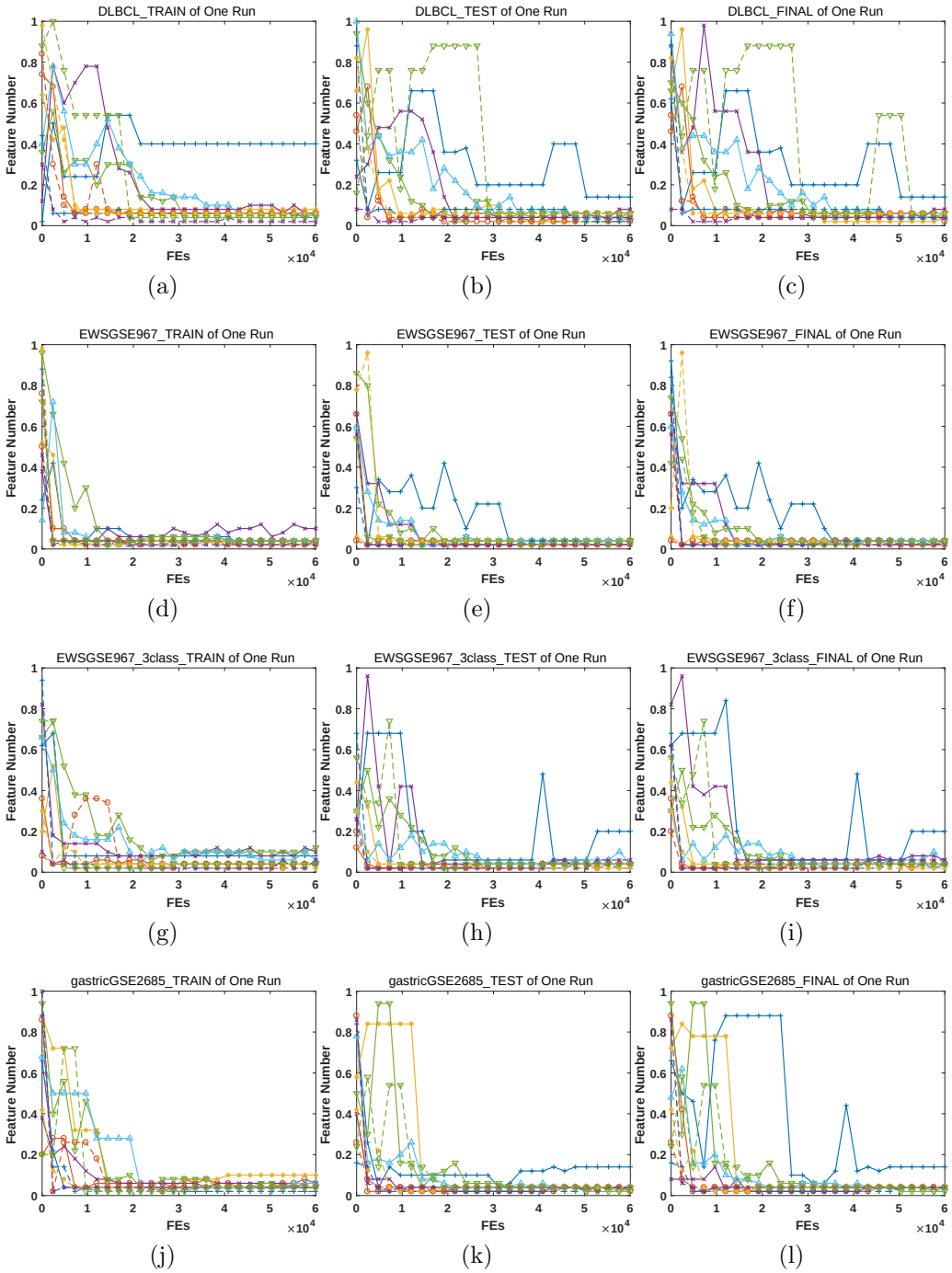

Figure 15: Illustration of feature number during evolution.

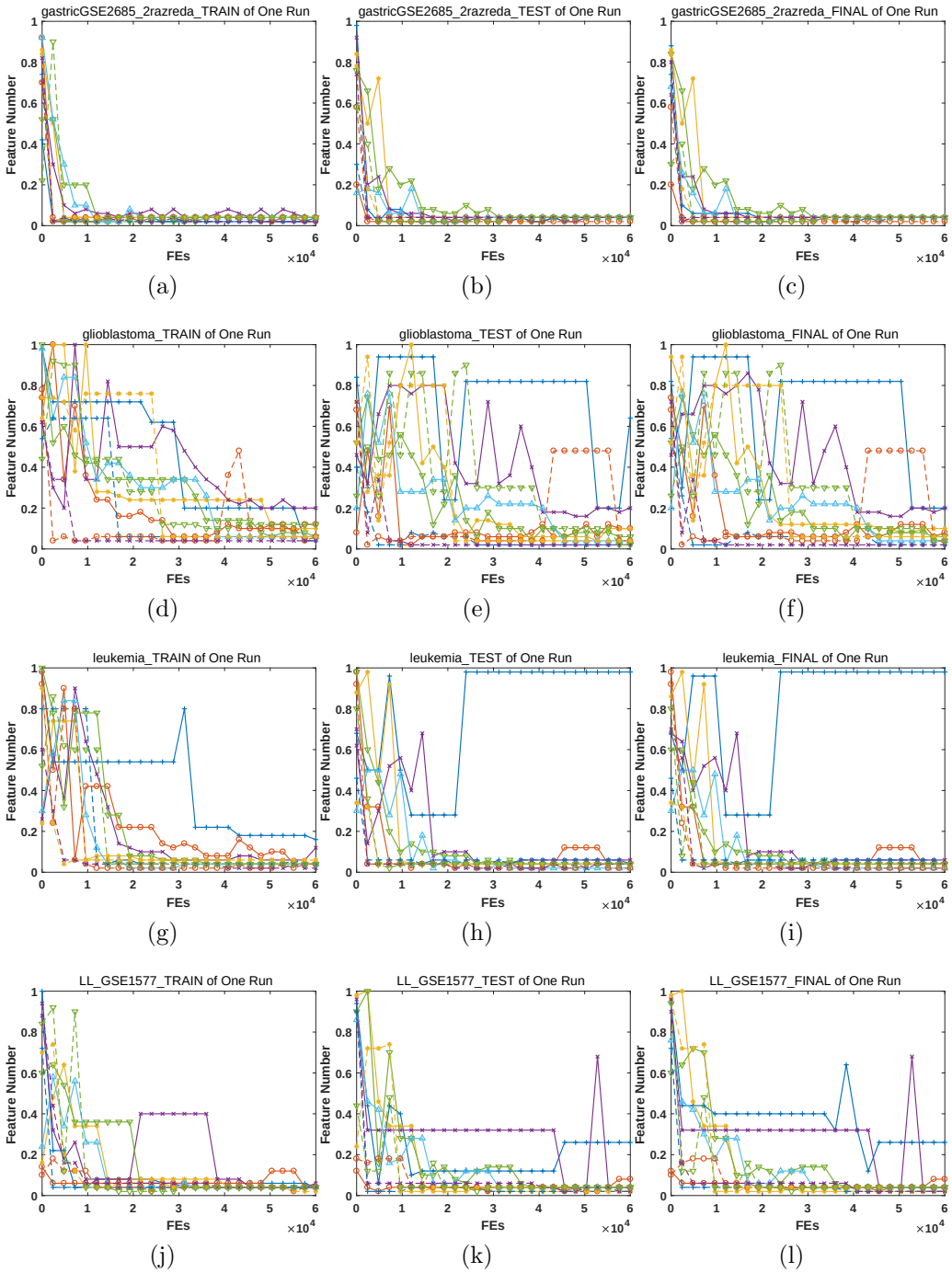

Figure 16: Illustration of feature number during evolution.

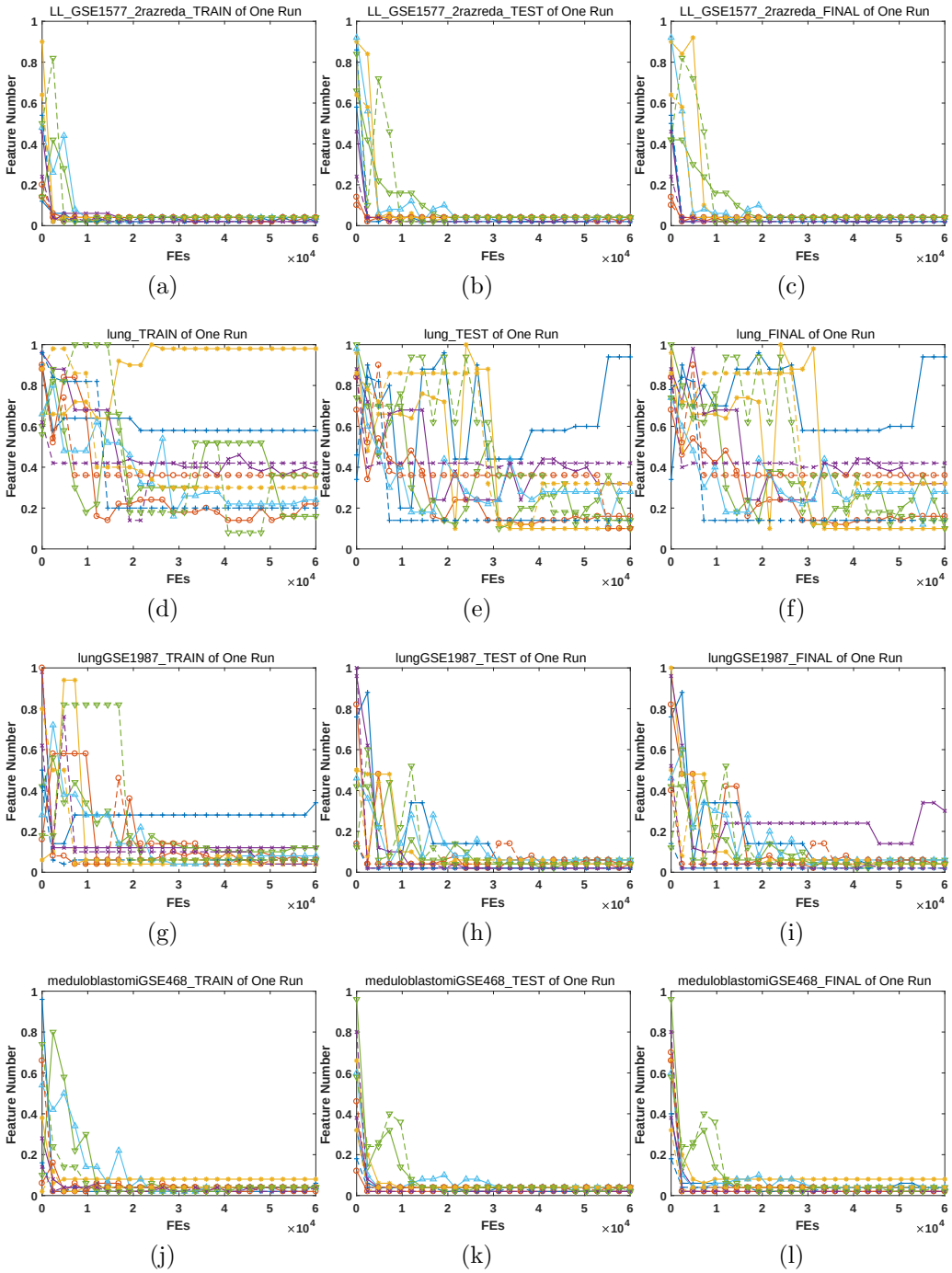

Figure 17: Illustration of feature number during evolution.

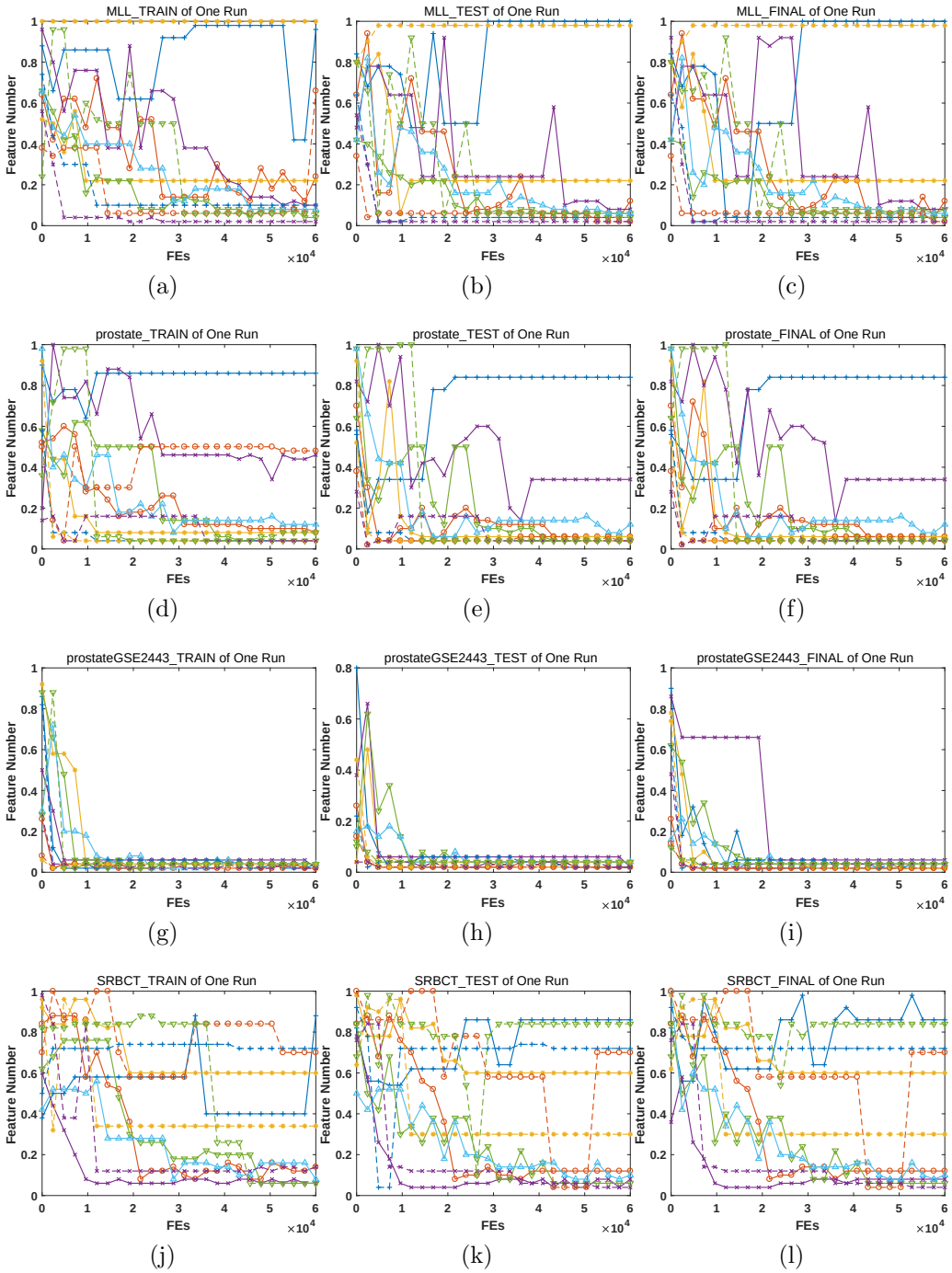

Figure 18: Illustration of feature number during evolution.

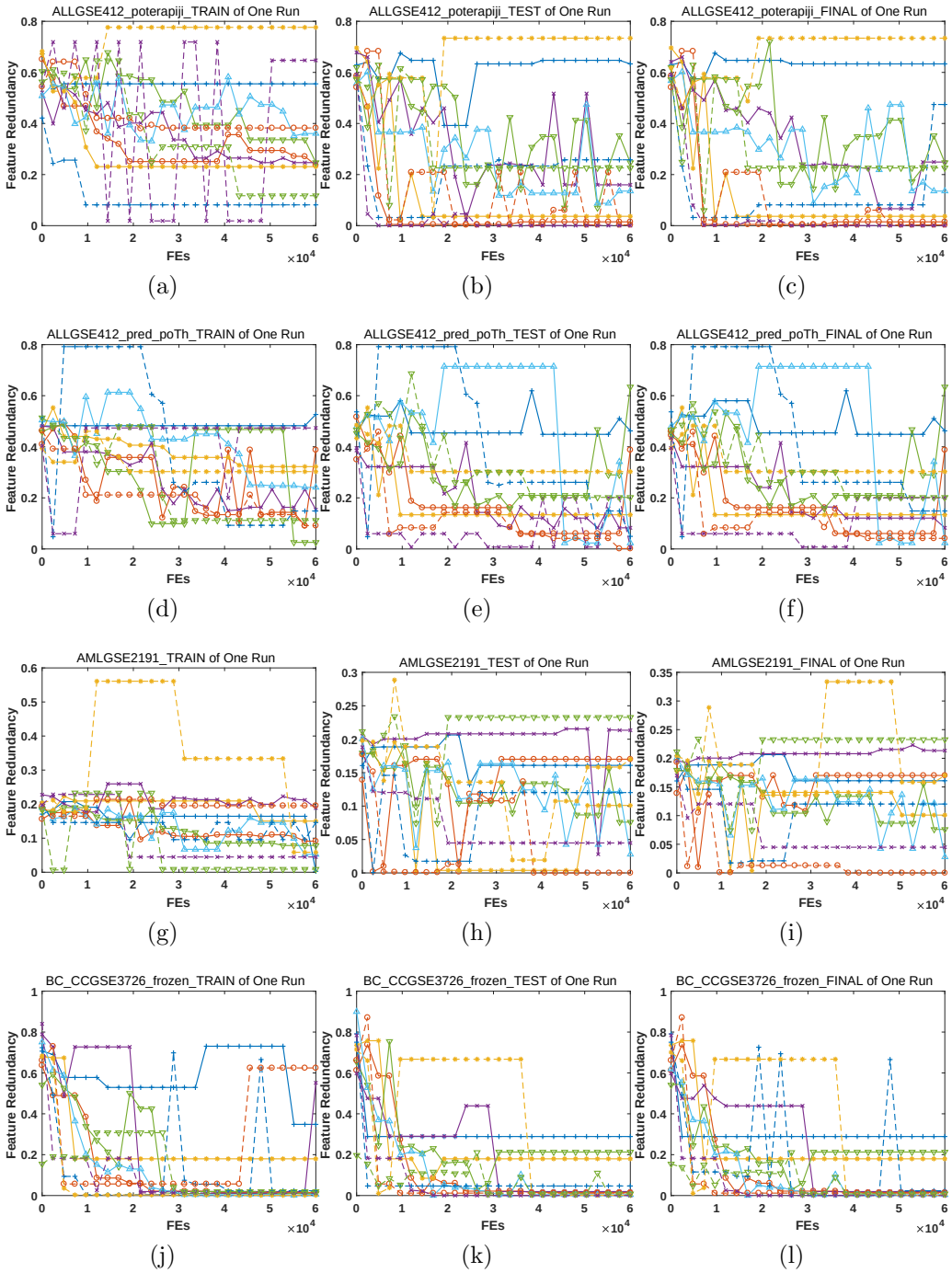

Figure 19: Illustration of feature redundancy during evolution.

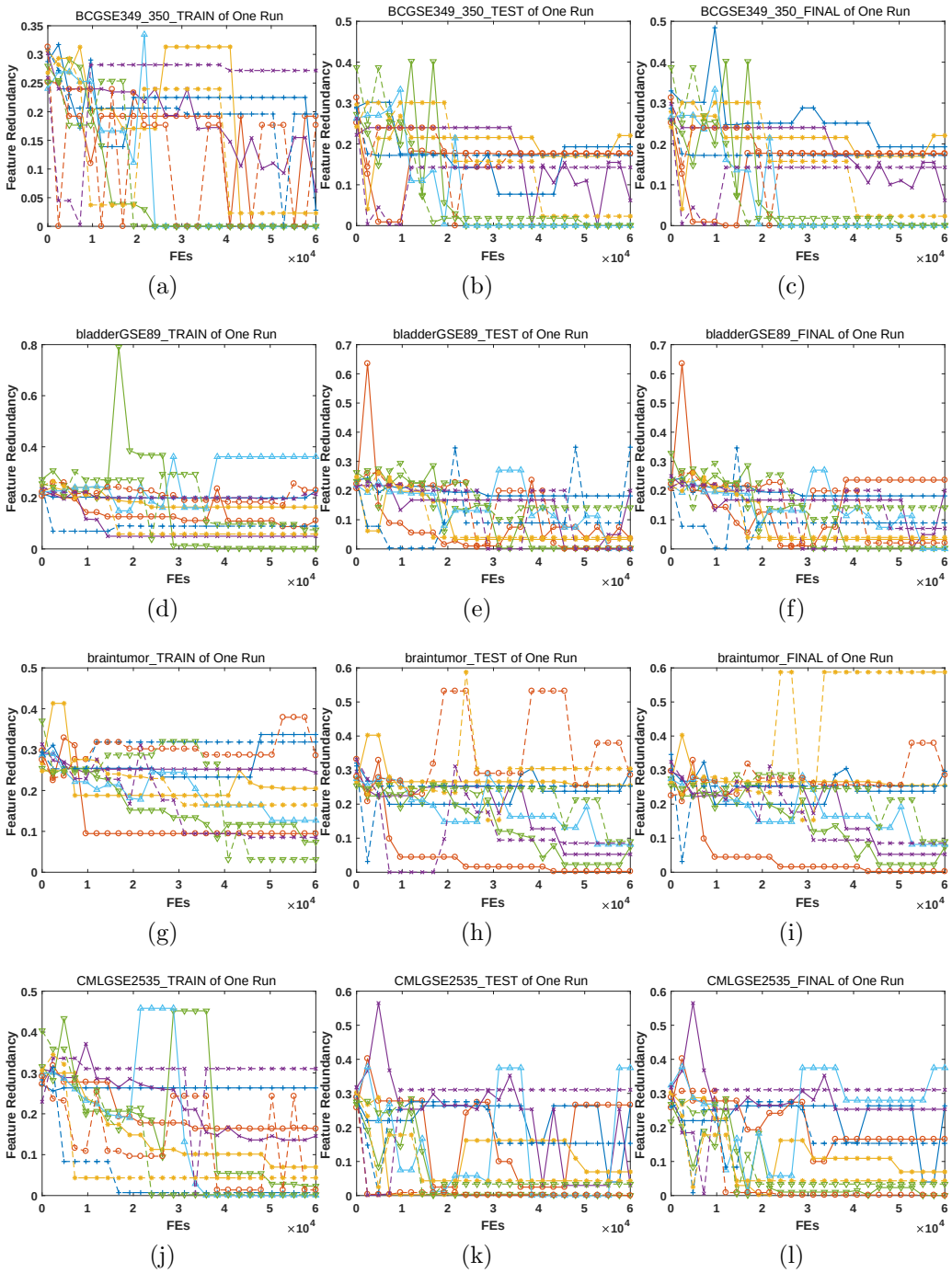

Figure 20: Illustration of feature redundancy during evolution.

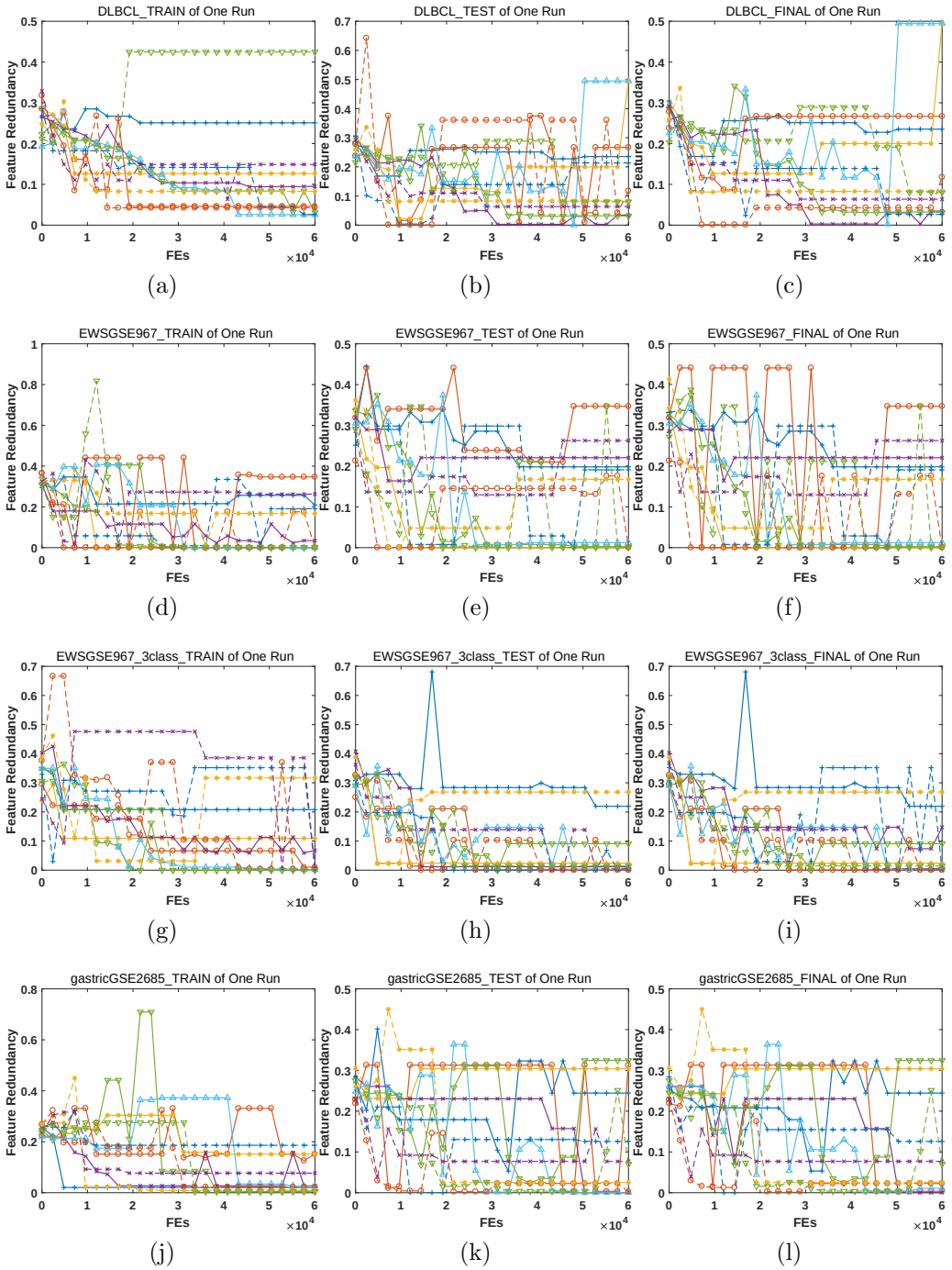

Figure 21: Illustration of feature redundancy during evolution.

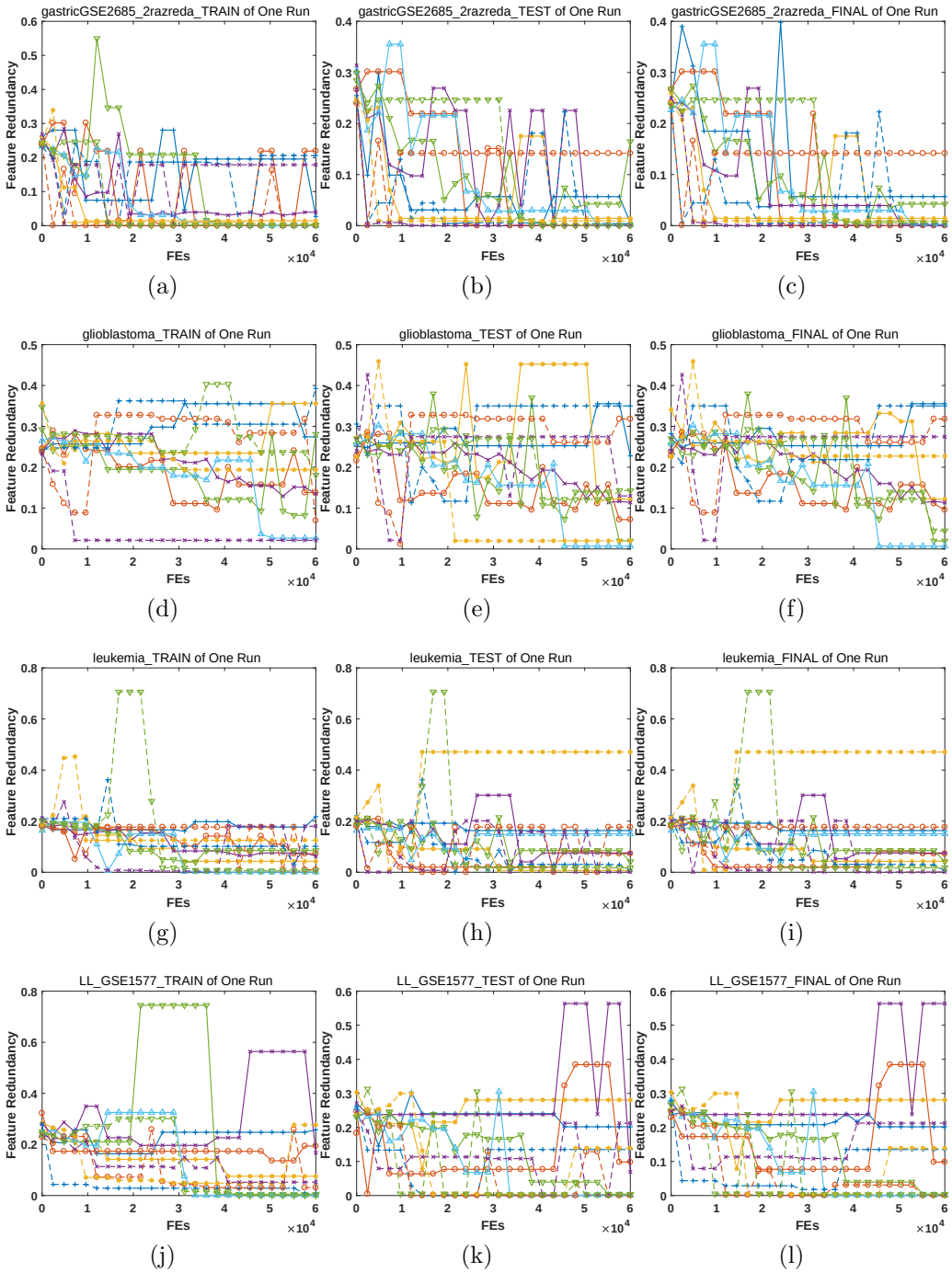

Figure 22: Illustration of feature redundancy during evolution.

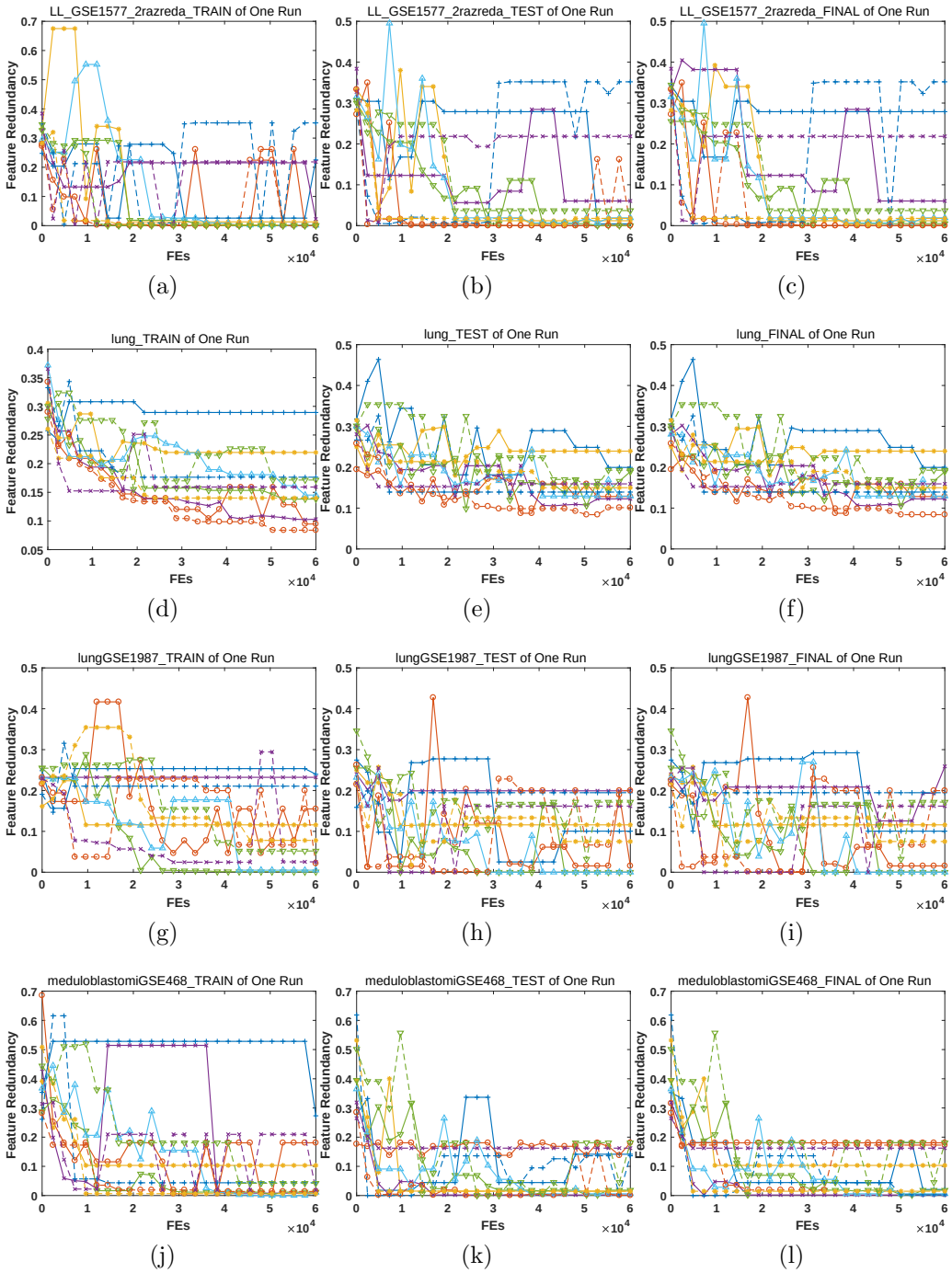

Figure 23: Illustration of feature redundancy during evolution.

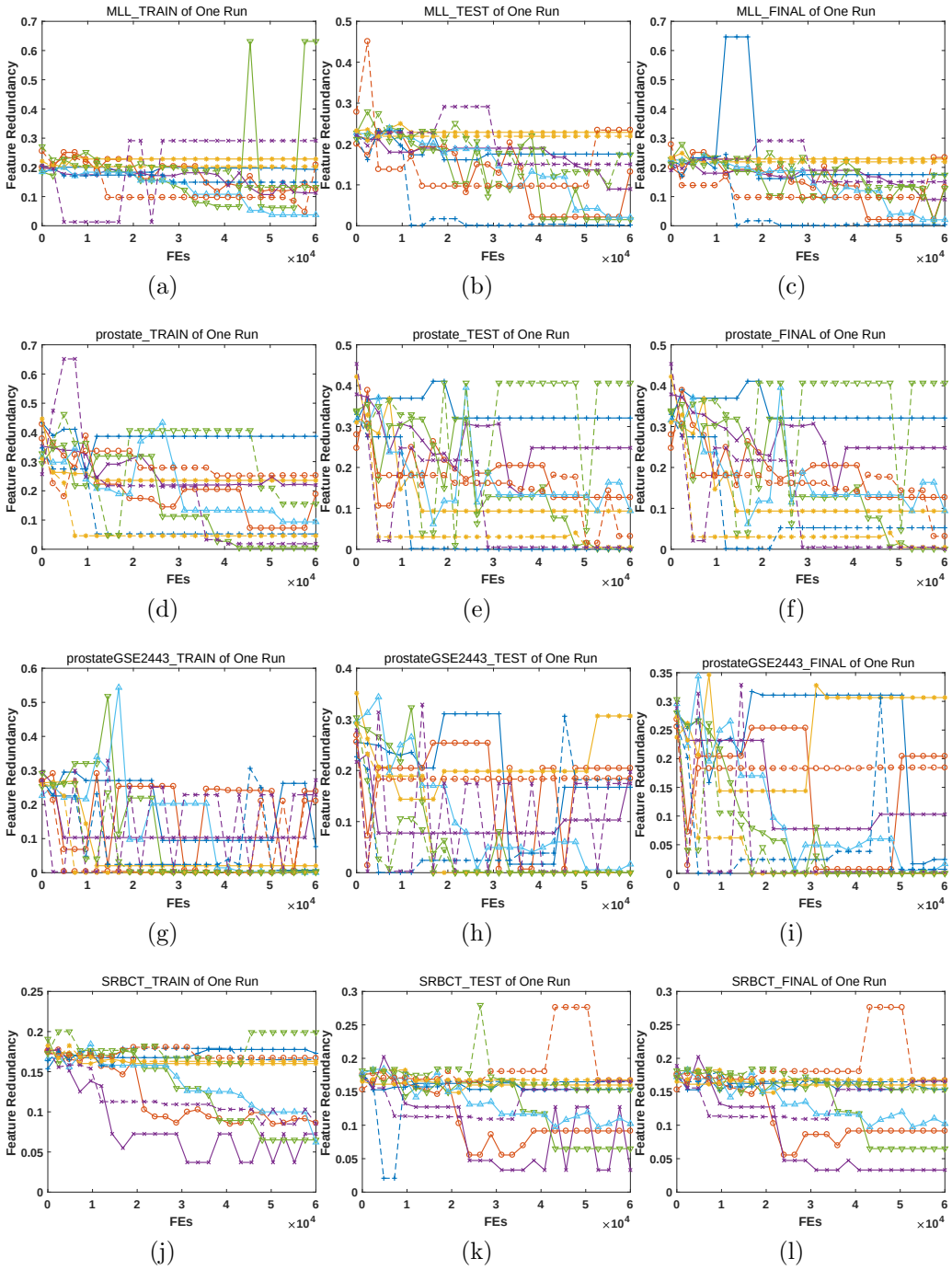

Figure 24: Illustration of feature redundancy during evolution.

Table 2: Average Operating Time (secs).

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

[1] Values in parentheses denote the speedups with respect to those of DPCCMOLSBEA-FS.
[2] Values in bold are related to the parallel algorithms.

5. Discussion and Future Work

In the aforementioned experimental study, though the proposed algorithms can outperform the counterparts, for some datasets, serious overfitting occurs and there are significant performance deterioration in the test sets. Nevertheless, for the binary-encoded CCGDE3, overfitting merely occurs, and for quite a lot of datasets, its classification errors are below most of the other algorithms. Therefore, we can refer to the binary-encoded CCGDE3 to alleviate the occurred overfitting in the proposed algorithms.

In the proposed algorithms, only different encodings and other mechanisms with respect to the MOEA aspect are examined. However, for the feature selection problem, there are many traditional methodologies for analysis, the output of which can act as prior knowledge to further enhance the proposed algorithms.

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

6. Conclusion

To address the feature selection problem, this paper proposes a multi-objective feature selection model by simultaneously considering classification error, feature number and feature redundancy. Then, several distributed parallel algorithms are proposed. Different feature encodings are tested, and an adaptive strategy is examined. With respect to the microarray datasets, the feature number is extremely large; consequently, the time consumption in 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, significantly reducing the time consumption. Moreover, sample-wise parallelism is devised to considerably increase the efficiency of processing the test sets 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.

Acknowledgements

This work was supported in part by the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University under Grant 2018002, in part by the Opening Project of Guangdong High Performance Computing Society under Grant 2017060101, in part by the Foundation of Key Laboratory of Machine Intelligence and Advanced Computing of the Ministry of Education under Grant No. MSC-201602A, and in part by Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (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, GIScience & Remote Sensing 55 (2) (2018) 221–242. doi: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 Transactions on Industrial Informatics 14 (1) (2018) 241–252. doi:10.1109/TII.2016.2617464.

[3] L. Zheng, H. Wang, S. Gao, Sentimental feature selection for sentiment analysis of Chinese online reviews, International Journal of Machine Learning and Cybernetics 9 (1) (2018) 75–84. doi:10.1007/s13042-015-0347-4.
URL https://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 Transactions on Cybernetics 48 (1) (2018) 16–28. doi: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 Conversion and Management 143 (2017) 360 – 376. doi:https://doi.org/10.1016/j.enconman.2017.04.007.

URL http://www.sciencedirect.com/science/article/pii/S0196890417303126

[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 Transactions on Computational Biology and Bioinformatics 9 (4) (2012) 1106–1119. doi: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, Information Sciences 282 (2014) 111 – 135. doi:https://doi.org/10.1016/j.ins.2014.05.042.
URL http://www.sciencedirect.com/science/article/pii/S0020025514006021

[8] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1424–1437. doi: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 Computing 22 (3) (2018) 811–822. doi:10.1007/s00500-016-2385-6.
URL https://doi.org/10.1007/s00500-016-2385-6

[11] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Networks 4 (8) (1995) 1942–1948. doi: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, International Journal of Machine Learning and Cybernetics 9 (1) (2018) 97–111. doi:10.1007/s13042-015-0359-0.
URL https://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, Journal of Information Science 43 (1) (2017) 25–38. arXiv:https://doi.org/10.1177/

0165551515613226, doi:10.1177/0165551515613226.
URL https://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. doi:https://doi.org/10.1016/j.neucom.2015.06.083.
URL http://www.sciencedirect.com/science/article/pii/S0925231215010504

[15] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626. doi: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. doi:10.1016/j.eswa.2009.10.027.
URL 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 Transactions on Cybernetics 43 (6) (2013) 1656–1671. doi: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, International Journal on Artificial Intelligence Tools 22 (04) (2013) 1350024. doi:10.1142/S0218213013500243.

[19] J. R. Vergara, P. A. Estévez, A review of feature selection methods based on mutual information, Neural Computing and Applications 24 (1) (2014) 175–186. doi:10.1007/s00521-013-1368-0.
URL https://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, Applied Soft Computing 34 (2015) 286 – 300. doi:http://dx.doi.org/10.1016/j.asoc.2015.04.061.

URL //www.sciencedirect.com/science/article/pii/S1568494615002987

[21] M. A. Potter, K. A. D. Jong, A cooperative coevolutionary approach to function optimization, in: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, PPSN III, Springer-Verlag, London, UK, UK, 1994, pp. 249–257.
URL http://dl.acm.org/citation.cfm?id=645822.670374

[22] B. Cao, J. Zhao, P. Yang, Z. G. Lv, X. Liu, G. Min, 3D multi-objective deployment of an industrial wireless sensor network for maritime applications utilizing a distributed parallel algorithm, IEEE Transactions on Industrial Informatics (2018) 1–1doi:10.1109/TII.2018.2803758.

[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, Journal of Global Optimization 11 (4) (1997) 341–359. doi:10.1023/A:1008202821328.
URL http://dx.doi.org/10.1023/A:1008202821328

[25] J. Zhang, A. C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 945–958. doi: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 Transactions on Evolutionary Computation 20 (5) (2016) 711–729. doi: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. doi:10.1109/CEC.2013.6557903.

[28] J. Wang, W. Zhang, J. Zhang, Cooperative differential evolution with multiple populations for multiobjective optimization, IEEE Transaction-

s on Cybernetics 46 (12) (2016) 2848–2861. doi:10.1109/TCYB.2015.2490669.

[29] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731. doi:10.1109/TEVC.2007.892759.

[30] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi: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 Transactions on Evolutionary Computation 10 (6) (2006) 646–657. doi:10.1109/TEVC.2006.872133.