






A Length-Adaptive Non-Dominated Sorting Genetic Algorithm for Bi-Objective High-Dimensional Feature Selection

Yanlu Gong , Junhai Zhou , Quanwang Wu , *Member, IEEE*,
MengChu Zhou , *Fellow, IEEE*, and Junhao Wen 

Abstract—As a crucial data preprocessing method in data mining, feature selection (FS) can be regarded as a bi-objective optimization problem that aims to maximize classification accuracy and minimize the number of selected features. Evolutionary computing (EC) is promising for FS owing to its powerful search capability. However, in traditional EC-based methods, feature subsets are represented via a length-fixed individual encoding. It is ineffective for high-dimensional data, because it results in a huge search space and prohibitive training time. This work proposes a length-adaptive non-dominated sorting genetic algorithm (LA-NSGA) with a length-variable individual encoding and a length-adaptive evolution mechanism for bi-objective high-dimensional FS. In LA-NSGA, an initialization method based on correlation and redundancy is devised to initialize individuals of diverse lengths, and a Pareto dominance-based length change operator is introduced to guide individuals to explore in promising search space adaptively. Moreover, a dominance-based local search method is employed for further improvement. The experimental results based on 12 high-dimensional gene datasets show that the Pareto front of feature subsets produced by LA-NSGA is superior to those of existing algorithms.

Index Terms—Bi-objective optimization, feature selection (FS), genetic algorithm, high-dimensional data, length-adaptive.

I. INTRODUCTION

IN a big data era, it is increasingly common for a data mining task to deal with high-dimensional data with thousands of or even more features (e.g., in bioinformatics) [1]. In these cases, there are usually many irrelevant or redundant features that not only result in much more training time but also reduce the classification performance. Feature selection (FS), aiming

to select a subset of relevant features from the original ones, can be effective to address this issue and thus plays an important role in data mining [2].

In the past few decades, many FS methods were proposed. They can be roughly divided into filter, wrapper and embedded ones [3]. Filter methods use intrinsic characteristics of a dataset to evaluate features and do not rely on a specific classifier. For example, the Relief method [4] calculates a distance-based weight for each feature, and features with larger weights are regarded to be more relevant to the class label. Wrapper methods utilize a given classifier as a black box to score a feature subset according to its predictive power. They usually achieve higher classification accuracy but incur more runtime on the other side. Their main challenge is how to search a feature subset efficiently from a total of 2^n feature subsets, where n is the number of features in the dataset. For example, the sequential feature selection (SFS) method [5] starts with an empty set and adds one feature individually in each step which gives the highest value for the objective function. Embedded methods perform FS in the process of training. Decision trees [6], [7] and Lasso regression [8] are two classical embedded methods.

Traditionally, FS is mainly resolved as a single-objective optimization problem for classification accuracy. However, it is actually a bi-objective one as it has two essential competing goals: maximizing accuracy and minimizing the number of used features. Up to now, some methods have been proposed based on multi-objective evolutionary algorithm (MOEA) to optimize these two conflicting objectives simultaneously: A population of individuals are evolved iteratively and at the end a set of non-dominated individuals (a.k.a., Pareto front) is returned [9]–[11]. For example, Hamdani *et al.* [9] follows the classical MOEA framework NSGA-II (non-dominated sorting genetic algorithm II) [12] to optimize these two objectives, and the experimental results on five low-dimensional datasets show that it achieves a good balance between minimizing the number of features and maximizing the accuracy.

However, existing MOEA-based bi-objective FS methods adopt a length-fixed individual encoding to represent feature subsets. In other words, the length of each individual is set to the feature dimension and remains constant in the entire evolutionary process. For low dimensional data with tens or hun-

Manuscript received April 27, 2023; accepted May 15, 2023. This work was supported in part by the National Natural Science Foundation of China (62172065, 62072060). Recommended by Associate Editor Qinglai Wei. (Corresponding authors: Quanwang Wu and MengChu Zhou.)

Citation: Y. L. Gong, J. H. Zhou, Q. W. Wu, M. C. Zhou, and J. H. Wen, “A length-adaptive non-dominated sorting genetic algorithm for bi-objective high-dimensional feature selection,” *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 9, pp. 1834–1844, Sept. 2023.

Y. L. Gong, J. H. Zhou, and Q. W. Wu are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: gyl@cqu.edu.cn; zhoujunhai@cqu.edu.cn; wqw@cqu.edu.cn).

M. C. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

J. H. Wen is with the College of Big Data and Software Engineering, Chongqing University, Chongqing 400044, China (e-mail: jhwen@cqu.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2023.123648

dreds of features, such representation may work well. However, when applied to high-dimensional data, it results in a huge amount of memory overhead and training time. In addition, it leads to loss of population diversity and its search efficiency deteriorates because of too huge search space. For bi-objective high-dimensional FS, the number of used features and classification accuracy may be not in complete conflict, i.e., reducing the number of used features may increase accuracy. The length-fixed encoding usually yields subsets with a large number of used features, and is hence unable to make some individuals focus on small fruitful search space.

This paper proposes a length-adaptive non-dominated sorting genetic algorithm (LA-NSGA) based on the NSGA-II framework for bi-objective high-dimensional FS. Since more relevant features are more likely to be selected, it first rearranges features decreasingly according to their relevance to the class label. Based on a length-variable individual encoding, it then initializes a population of individuals of diverse lengths based on correlation and redundancy. Equipped with length-adaptive genetic operators including a dominance-based length change operator, LA-NSGA enables the evolving individuals to concentrate on smaller but more promising search area adaptively. Besides, a local search based on Pareto dominance is introduced for improving promising individuals further. As the first bi-objective FS method with a length-adaptive evolution, LA-NSGA is compared with existing methods based on 12 high-dimensional gene datasets. The experimental results demonstrate that LA-NSGA outperforms them as it can achieve a higher hypervolume value.

In a nutshell, this work aims to make the following novel contributions: 1) This work proposes a novel bi-objective high-dimensional FS method by using a length-adaptive evolution mechanism for the first time; 2) It performs extensive experimental evaluations to verify the significant superiority of the proposed method to existing ones.

The remaining of this paper is organized as follows. Section II reviews related work on FS and introduces preliminaries on MOEA. Section III presents the proposed method and Section IV describes the experimental results and analysis. Section V concludes the paper and points out future work.

II. RELATED WORK AND PRELIMINARIES

FS selects some key features that can retain the valid information of a dataset, and removes irrelevant or redundant ones, thereby reducing the data used for training. As a crucial data preprocessing method, there exist some comprehensive surveys about FS [1], [2] and [13]. In the following, we review literature on high-dimensional feature selection and multi-objective one as the research topic considered in this paper is an intersection of them. In addition, we introduce preliminaries of MOEA.

A. High-Dimensional Feature Selection

In a big data era, the curse of dimensionality has become a severe challenge to the application of data mining technology. FS is a critical method able to address this issue, especially for high-dimensional datasets. Traditional FS methods can be

directly applied to high-dimensional datasets, but their performance is in general worse than that of the FS methods specific to high-dimensional datasets according to [14]. Existing high-dimensional FS methods mainly consider the classification accuracy as their sole optimization objective and can be roughly divided into filter and wrapper methods.

In the area of filter ones, Yu and Liu [15] propose a fast correlation-based filter method (FCBF) for high-dimensional FS. It first sorts features in descending order of correlation, and then removes redundant features based on the introduced concept of predominant features. Sun *et al.* [16] use local learning to decompose an arbitrarily complex nonlinear problem into multiple locally linear ones, and then learn feature relevance globally for high-dimensional data analysis. Bommert *et al.* [17] analyze 22 filter methods based on 16 high-dimensional datasets in terms of runtime and accuracy and the results show that the best filter methods differ among the data sets. Lee *et al.* [18] propose a multivariate feature ranking method for high-dimensional microarray data, where Markov blanket (MB) is adopted to perform redundancy analysis for feature sorting.

Evolutionary computing (EC) is widely used for high-dimensional wrapper FS. Garcia-Torres *et al.* [19] propose a variable neighborhood search metaheuristic for high-dimensional FS, where features are grouped through MB to improve search effectiveness. Similarly, a competitive swarm optimizer with an archive technique in [20] and a two-stage hybrid ant colony optimization in [21] are proposed for solving this problem. To enable EC methods to handle high-dimensional datasets more flexibly, Tran *et al.* [22] develop a length-variable particle swarm optimization (PSO) method, where particles with different lengths are first initialized and then shortened gradually in an iterative process. Similarly, a length-variable representation is used for EC-based FS on high-dimensional data in [23], [24]. In [23], an individual is represented with a length-variable list of feature indexes while in [24] it is a length-variable list of binary values. These studies [22]–[24] demonstrate that EC methods with length-variable encoding perform effectively in high-dimensional datasets as it can reduce the search space, thereby saving memory overhead and training time.

B. Multi-Objective Feature Selection

Recently, more and more studies treat FS as a multi-objective optimization problem to optimize accuracy, the number of used features, and some other objectives. They usually employ MOEAs to handle it. In the following we use bi-objective FS to specifically refer to studies only considering accuracy and feature count, and multi-objective FS if over two objectives are considered.

Genetic algorithm-based MOEAs are the most widely used for multi-objective FS [25]. For example, Hamdani *et al.* [9] first uses NSGA-II [12] for bi-objective FS, and the results show that the quality of the Pareto optimal solutions can be ameliorated continuously. Li *et al.* [26] investigate how to select key quality characteristics of unbalanced production data and define an FS problem to optimize 3 performance

measures including geometric mean, F_1 score and accuracy as well as the feature count. Then, a hybrid method combining a genetic algorithm (GA) with direct multi-search is proposed to handle it. Xue *et al.* [27] propose a multi-objective binary GA with an adaptive crossover operator. Specifically, it selects a crossover operator by probability from five candidate crossover operators, and the selection probability for each candidate relies on the number of solutions it produces which survive to the next generation. A problem-specific NSGA-II method (PS-NSGA) is proposed for minimizing three objectives of FS, i.e., accuracy, feature count and a specific distance metric [28]. An accuracy-preferred domination operator is employed, and the most proper feature subset from the obtained Pareto front is returned. In [29], a duplication analysis-based EC method is proposed for bi-objective FS in classification. It improves the NSGA-II framework in three aspects: reproduction, duplication analysis and diversity-based selection.

PSO is also widely used for multi-objective FS. Xue *et al.* [30] propose two multi-objective PSO methods to search for PF based on the ideas of non-dominated sorting, crowding, mutation, and domination. A cost-based FS problem is considered in [31], which aims to maximize the classification performance and minimize the cost associated with features. A multi-objective PSO method integrating probability-based encoding and a hybrid operator is proposed. Amoozegar and Minaei-Bidgoli [32] propose a multi-objective PSO-based method with a feature ranking mechanism to improve the quality of the archive set. A flexible cut-point PSO is proposed in [33] to optimize accuracy, feature count and a distance metric. An arbitrary number of cut-points can be selected for data discretization. Rashno *et al.* [34] propose a PSO-based bi-objective FS method, where both particles and features are ranked to update PSO particles.

Some other metaheuristics are also used, e.g., differential evolution [35], [36], artificial bee colony (ABC) [37], [38], and salp swarm algorithm [39]. Zhang *et al.* [35] propose a binary differential evolution algorithm with a self-learning strategy for multi-objective FS (MOFS-BDE), where a binary mutation operator based on probability difference is used. In [36], a multi-objective FS method based on differential evolution is designed for recognizing facial expression. In [37], a fast multi-objective FS method based on ABC is proposed, where a parameter-free particle update model is embedded. Similarly, an FS method based on a multi-objective ABC algorithm integrated with non-dominated sorting and genetic operators is proposed in [38]. In [39], the salp swarm algorithm is used for multi-objective FS and multiple leader salps are set in addition to multiple subgroups to improve the convergence ability of the optimal solution.

C. Preliminaries of MOEA

Formally, an optimization problem with m objectives needed to be minimized can be defined as

$$\min F(x) = (f_1(x), f_2(x), \dots, f_m(x)), \quad x \in \Omega \quad (1)$$

where x represents decision variables and Ω is decision space.

Generally, it is impossible to find a solution with m objective values all being the minimum because the objectives usually contradict with each other [10]. Hence, we have to balance them and search for the best tradeoffs among the objectives in terms of Pareto dominance. Specifically, a solution x has a Pareto dominance relation over y if

$$\begin{aligned} \forall i \in \{1, 2, \dots, m\} \quad & f_i(x) \leq f_i(y) \\ \wedge \exists j \in \{1, 2, \dots, m\} \quad & f_j(x) < f_j(y). \end{aligned} \quad (2)$$

A Pareto optimal solution means that it is not dominated by any other one, and the Pareto front (PF) consists of all the Pareto optimal solutions. Since it is usually extremely hard to obtain the true PF, MOEAs are widely employed to find an approximate one.

As the most classic Pareto dominance-based MOEA framework, NSGA-II introduces two effective selection mechanisms to form a new population in order to guide the evolution toward the optimal PF, i.e., non-dominated sorting and crowding distance (CD) [12]. The former divides a population of individuals into several ranked non-dominated fronts based on the Pareto dominance relation. Fig. 1 illustrates an example of non-dominated sorting with two objectives, where a population P of 15 individuals are divided into 3 subsets R_1 , R_2 and R_3 through non-dominated sorting. R_1 is the PF of P , R_2 is the PF of the set $P-R_1$ and R_3 is that of $P-R_1-R_2$. The crowding distance of an individual is calculated as the density of individuals surrounding it and is used for preserving diversity (e.g., in Fig. 1, x_i 's CD value is measured via the area of the dotted rectangle). When selecting individuals to form a new population for evolution, NSGA-II prefers the individuals with a lower rank or with a larger CD value when their ranks are the same.

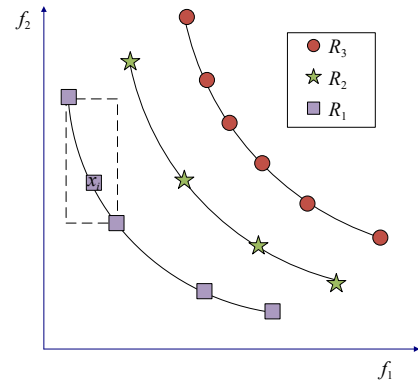


Fig. 1. Example of non-dominated sort.

III. PROPOSED METHOD

FS aims to use as few features as possible to achieve higher classification accuracy. Since a dataset is usually class-imbalanced, we specifically consider the following two objectives for FS: the balanced classification error rate and the proportion of selected features, i.e.,

$$f_1 = \frac{1}{c} \sum_{i=1}^c \eta_i \quad (3)$$

$$f_2 = \frac{|F_{\text{used}}|}{n} \quad (4)$$

where c is the number of classes in a dataset, η_i denotes the false positive ratio for the i th class, F_{used} stands for the set of selected features, and n is the number of all features. We can see that in this problem, a solution x is said to dominate solution y if x has a lower error rate and uses fewer features than y .

This section presents the proposed method LA-NSGA based on NSGA-II for bi-objective FS, as shown in Fig. 2. It first initializes a population P of individuals. Then an evolution process iterates until the termination condition is met (e.g., a specified iteration number is reached). In each iteration, individuals in P are evaluated in terms of the two objectives. Next, non-domination rank and crowding distance are calculated for each individual based on NSGA-II. A tournament selection operator is used to select individuals from P as parents based on their ranks and CD values, and genetic operators including length change, mutation and crossover are carried out to yield offspring. This step repeats until a new population Q with the same size as P is constructed. After that, P and Q are combined and a half of individuals from them are selected to form a new population P by comparing their ranks and CD values. Then, a local search is carried out for improving P . When the termination condition is met, the non-dominated individuals in P are returned as the output.

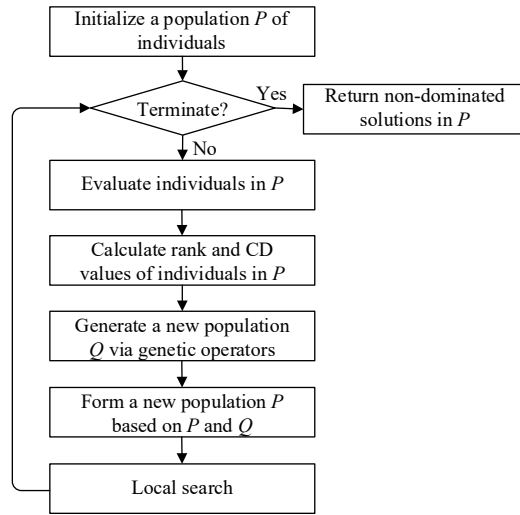


Fig. 2. Flowchart of LA-NSGA.

Length-variable individual encoding and initialization in LA-NSGA are next introduced. Length-adaptive genetic operators are then described. A Pareto dominance-based local search method is explained. Finally, the complexity analysis of LA-NSGA is given. Table I summarizes the key notations used in this section.

A. Individual Encoding and Initialization

Individual encoding in GA determines how an individual represents a candidate solution for a given problem. For FS, a subset of features is encoded as an individual through an array of binary genes, where “1” indicates that the corresponding feature is selected and “0” indicates not. For instance, an indi-

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	Description
F_i	The i th feature of the dataset
C	The class label of the dataset
n	The number of features in the dataset
$SU(F_i, C)$	The symmetric uncertainty between F_i and C
X	An individual in LA-NSGA
P	A population of individuals in LA-NSGA
$L(X)$	The length of an individual in LA-NSGA

vidual with a binary array of 01001 means that the second and fifth features are selected whereas the others are not.

Traditional EC-based methods [9], [30] for bi-objective FS use length-fixed individual encoding. In this manner, the length of all individuals is set to the feature dimension and all features are treated equally by an evolutionary process. However, as features more relevant to the class label are more likely to be selected, it is preferable to differentiate them in evolution, especially for high-dimensional datasets. Hence, length-variable individual encoding based on relevance is adopted in LA-NSGA, where, individuals can have diverse lengths. To support this encoding, features are first rearranged in a decreasing order according to their relevance. Symmetric uncertainty (SU) is adopted to measure relevance here as it well suits for classification. Formally, SU of feature F_i with the class label C is calculated as

$$SU(F_i, C) = \frac{2 \times IG(F_i, C)}{H(F_i) + H(C)} \quad (5)$$

$$IG(F_i, C) = H(F_i) - H(F_i|C) \quad (6)$$

where $H(F_i)$ represents the entropy of F_i , $H(F_i|C)$ is the conditional entropy of F_i given C , and $IG(F_i, C)$ is the information gain of F_i and C . $SU(F_i)$ ranges in $[0, 1]$, and a larger value indicates a higher relevance.

To evaluate an individual, the proportion of selected features can be directly obtained by counting “1” in it and for the error rate objective, any classification algorithm can be utilized for assessment such as K -nearest neighbors (KNN) [40].

1) *Individual Initialization*: To initialize a population of length-variable individuals, two issues need to be addressed: a) how to set the length of an individual; and b) how to set a binary value of each gene in it. For the former, we determine the length of an individual X_j according to its index j in the population, feature dimension n , and population size p , i.e.,

$$L(X_j) = \text{round}(n \times j \times p^{-1}) \quad (7)$$

where function $\text{round}(\cdot)$ maps a real value to its nearest integer. By so doing, the whole search space is divided into multiple subspaces and the search diversity is improved. Moreover, a short individual only considers more relevant features and it is easier for it to find a good solution.

For the latter, a simple solution is to randomly set each binary value to “1” or “0” (i.e., selecting the corresponding feature or not) with a probability of 0.5. However, a high-dimensional dataset usually has a large number of irrelevant

or redundant features, indicating that it is better to assign a different selection probability to each feature for initialization. Hence, we introduce ρ_i to represent the selection probability of F_i . It is set to 0.5 if $i = 1$ and otherwise it is calculated based on F_i 's correlation and redundancy via

$$\rho_i = \begin{cases} \frac{SU(F_i, C)}{SU(F_i, C) + \overline{SU}(F_i, K)}, & \text{if } |K| > 0 \\ \frac{SU(F_i, C)}{SU(F_1, C)}, & \text{otherwise} \end{cases} \quad (8)$$

where K denotes the set of features that have been already selected, and the SU value $\overline{SU}(F_i, K)$, representing the average redundancy between F_i and features in K , is calculated as

$$\overline{SU}(F_i, K) = \frac{\sum_{F_j \in K} SU(F_i, F_j)}{|K|}. \quad (9)$$

Based on the above settings, we can see that when $|K| > 0$, for a feature F_i except F_1 , it is selected with a higher probability if its correlation with the label is larger, and its redundancy is smaller. When $|K| = 0$, it indicates that all features ahead of F_i are not selected. In this case, ρ_i can only be calculated based on F_i 's correlation with C . Specifically, it is normalized by being divided by F_1 's correlation, which is the largest among all after rearrangement.

Algorithm 1 elaborates the individual initialization method in LA-NSGA. The length of an individual is first determined via (7), and then each gene in it is assigned to "1" or "0" according to ρ_i (Lines 3–10). Note that the function $\text{rand}()$ in Line 5 returns a random number between 0 and 1.

Algorithm 1 Individual Initialization

Input: index j , population size P , feature dimension n
Output: an individual X
1: construct a binary array X with a length obtained via (7);
2: $K \leftarrow \{\}$;
3: **for** $i = 1; i \leq L(X); i++$ **do**
4: determine selection probability ρ_i ;
5: **if** $\text{rand}() < \rho_i$ **then**
6: $X[i] \leftarrow 1$, add F_i to K ;
7: **else**
8: $X[i] \leftarrow 0$;
9: **end if**
10: **end for**
11: **return** X ;

B. Genetic Operators

Genetic operators in GA aim to yield offspring individuals based on the current population. Here, we present the length change operator in LA-NSGA and explain how to specifically tailor the traditional mutation and crossover operators in turn.

1) *Pareto Dominance-Based Length Change Operator:* A length change operator is introduced to improve the diversity of a population and encourage individuals to explore a smaller but more fruitful search area adaptively. Its detail is described in Algorithm 2. It uses two individuals X and Y with different lengths as input, and without losing generality we assume that

$L(X) < L(Y)$. First, a length factor l is determined randomly based on the difference of $L(X)$ and $L(Y)$. Next, X and Y are compared for length changing (Lines 2–10): if X is more preferable than Y , it indicates that a short length is more promising, and Y is shortened by l dimensions (Lines 2 and 3); otherwise, X is extended by l dimensions (Lines 4 and 5). Note that in a Pareto dominance-based MOEA framework, individuals are more preferable when they have a lower rank or a larger CD value in the case of the same rank.

Algorithm 2 Length Change Operator

Input: individuals X and Y
Output: a new individual
1: $l \leftarrow \text{rand}() \times (L(Y) - L(X))$;
2: **if** X is more preferable than Y **then**
3: shorten Y by l dimensions;
4: **else**
5: extend X by l dimensions;
6: **end if**
7: **return** X or Y ;

To shorten X by l dimensions, we can simply cut off X 's last l genes. To extend X , a similar implementation is to append l genes with random binary values. However, this implementation neglects the information of the population on these l features. To make full use of the information from the population, we introduce a competition mechanism to extend an individual, in which the value of each newly added gene is determined by competition. Precisely, to append the i th gene to X , we first randomly pick two distinct individuals Y_1 and Y_2 from the population whose length is not shorter than i . Then, the i th gene of the better one between Y_1 and Y_2 is copied and appended to X . This method is realized in Algorithm 3.

Algorithm 3 Competitive Length Extension

Input: individual X , length l
Output: a new individual
1: **for** $i = L(X) + 1$ to $L(X) + l$ **do**
2: randomly pick two individuals Y_1 and Y_2 which are not shorter than i ;
3: **if** Y_1 or Y_2 is null **then**
4: add a random binary gene to X ;
5: **else if** Y_1 is more preferable than Y_2 **then**
6: add the i th gene of Y_1 to X ;
7: **else**
8: add the i th gene of Y_2 to X ;
9: **end if**
10: **end for**
11: **return** X ;

Fig. 3 gives two examples of competitive length extension, where the dimension of an individual X is expanded from 3 to 4. Firstly, two longer individuals Y_1 and Y_2 are randomly selected from the population. In Fig. 3(a), $Y_2.\text{Rank} < Y_1.\text{Rank}$, and thus the 4th gene of Y_2 is copied and added to X . In Fig. 3(b), Y_1 and Y_2 have the same rank but $Y_1.CD$ is larger, and thus the 4th gene of Y_1 is copied and added to X .

2) *Mutation and Crossover Operators:* In GA, the mutation

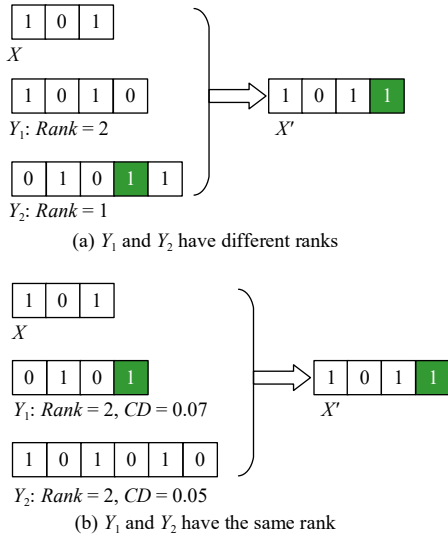


Fig. 3. Example of competitive length extension.

operator is used to maintain genetic diversity in the population and the crossover operator combines the genetic information of parent individuals to produce offspring individuals. A standard one-point mutation is adopted in LA-NSGA: A gene in an individual is randomly selected and its value is flipped, i.e., “1” is flipped to “0” and vice versa. In LA-NSGA, when two parents are fortunate to be of the same length l , a standard one-point crossover is employed: a crossover point μ is randomly chosen from 1 to l , and genes to the right of μ on the two parents are swapped to generate offspring. Nevertheless, it is more common that the lengths of the two parents X and Y are different. Assuming that $L(Y) \geq L(X)$, μ is randomly chosen from 1 to $L(X)$ and then a one-point crossover is carried out between them.

C. Pareto Dominance-Based Local Search

To improve the convergence of a population towards the true PF, local search based on Pareto dominance is devised in LA-NSGA as realized in Algorithm 4. For each non-dominated individual X in P , a new individual X' is generated based on it via an individual local search, which adds or removes features via flipping as done in [22] and [24]. If X dominates X' , it means that this attempt of individual local search fails (Lines 5 and 6). By contrast, if X' dominates X , X is replaced by X' as the latter is better than X (Lines 7 and 8). If they do not dominate each other, X' is added to a temporary individual set T (Lines 9 and 10). The above procedure is repeated for several times for each non-dominated individual and when it finishes, P is combined with T and the best individuals from them are obtained via non-dominated sorting and CD to form a new population P .

Fig. 4 gives an example of this local search procedure, where the red circles represent non-dominated individuals in P . The four new individuals X_1 , X_2 , X_3 and X_4 (yellow circles) are generated via performing individual local search on X . In this case, X_1 is used to replace X , X_2 is directly discarded because of being dominated by X , and X_3 and X_4 are put into T as candidates.

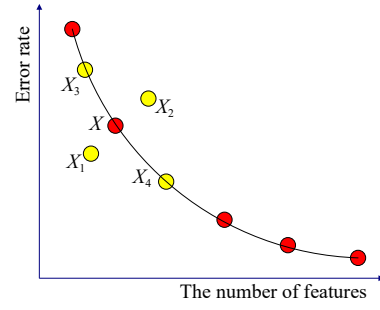


Fig. 4. Example of local search.

Algorithm 4 Pareto Dominance-Based Local Search

Input: population P
Output: a new population
1: $T = \{\}$;
2: **for** each non-dominated individual X in P **do**
3: **while** attempt criterion is met **do**
4: perform individual local search on X to have X' ;
5: **if** X dominates X' **then**
6: **continue**
7: **else if** X' dominates X **then**
8: $X \leftarrow X'$;
9: **else**
10: add X' to T ;
11: **end if**
12: **end while**
13: **end for**
14: form a new population P from $P \cup T$;
15: **return** P ;

D. Complexity Analysis

Let m and n represent the number of examples, and the number of features in a dataset, respectively, and n' , g and p represent the average length of an individual, maximum iteration count, and population size in LA-NSGA, respectively. The computational complexity of LA-NSGA mainly consists of individual evolution and individual evaluation. For the former, the genetic operators of crossover, mutation and length change require a time complexity of $O(n')$, and in each iteration calculating non-dominated ranks and CD values of individuals require a time complexity of $O(p^2)$. For the latter, assuming that KNN ($K = 1$) is used to evaluate the classification performance, it takes $O(m^2 \times n')$ to use a KNN model for evaluating error rates, and $O(n')$ to obtain the number of used features. Hence, we can conclude that error rate evaluation is the main contributing factor, and without considering the local search, the overall complexity of LA-NSGA is $O(g \times p \times m^2 \times n')$.

Note that a length-fixed EC-based FS method requires a complexity of $O(m^2 \times n)$ for error rate evaluation. In LA-NSGA n' is much less than n for a high-dimensional dataset, and thus it is able to outperform length-fixed methods in terms of computation time.

IV. EMPIRICAL STUDIES

In this section, the performance of LA-NSGA is tested

against existing bi-objective FS methods on high-dimensional datasets. Experimental settings and results are described in turn.

A. Experimental Setup

The proposed LA-NSGA is implemented via the Weka platform¹, and evaluation experiments are run on a PC with Intel Core i7-9700K CPU@3.6 GHz and 16 GB RAM, Windows 10, and Java SE 10.

Twelve publicly available high-dimensional gene datasets are used to conduct experiments². Table II lists their detailed information including the numbers of features, instances and class labels as well as the percentages of smallest and largest classes. We can see that these high-dimensional datasets have thousands or tens of thousands of features but only a few instances, and most of them are class-imbalanced. The above characteristics make it quite challenging to carry out FS on these datasets.

TABLE II
THE DESCRIPTION OF DATASETS

Dataset	#Features	#Inst.	#Classes	%Smallest class	%Largest class
SRBCT	2308	83	4	13	35
Breast	2905	168	2	34	66
Leuk1	5327	72	3	13	53
DLBCL	5469	77	2	25	75
9Tumors	5726	60	9	3	15
Brain1	5920	90	5	4	67
NCI60	6114	61	9	7	15
Brain2	10 367	50	4	14	30
Prostate	10 509	102	2	49	51
CLL_SUB	11 340	111	3	10	46
11Tumors	12 533	174	11	4	16
Prostate2	12 625	88	2	43	57

We further use the t-distributed stochastic neighbor embedding algorithm (t-SNE) [41] to visualize some of the datasets (i.e., SRBCT, Leuk1, DLBCL, 9Tumors, CLL_SUB and NCI60) in 2 dimensions in Fig. 5, where class labels of instances are distinguished by color. We can see that the class labels in SRBCT, Leuk1 and DLBCL seem relatively easy to be separated while the distributions of 9Tumors, CLL_SUB and NCI60 are roughly chaotic, making FS and classification for these datasets extremely challenging.

The hypervolume (HV) metric (a.k.a., an S metric or Lebesgue measure) [42] is adopted to compare the performance of bi-objective FS methods here as it is the most widely-used one for multi-objective optimization evaluation. HV of a Pareto front Φ measures the portion of the objective space that is dominated by Φ collectively. Based on a reference point r which is dominated by all solutions in Φ , it is computed as

$$HV(\Phi) = \delta(\cup_{i=1}^T v_i) \quad (10)$$

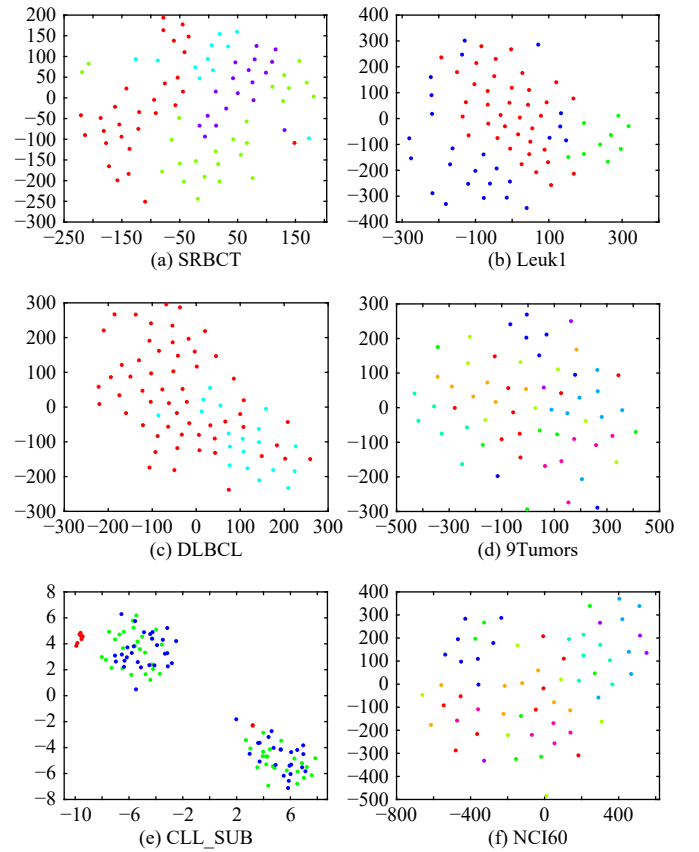


Fig. 5. t-SNE results of some datasets.

where δ denotes the Lebesgue measure, a standard method for assigning a length, area, or volume to a subset of Euclidean space, and v_i represents Euclidean space formed by the i th solution of Φ and r . In our experiments, r is set to (1.0, 1.0), indicating a solution with the highest error rate and all features used. Therefore, the range of HV is [0, 1] and a larger value is preferable, indicating that the non-dominated solutions are closer to the true PF.

10-fold cross-validation is used to divide a dataset into training and test sets. Hence, in the FS process, 9 folds are used as the training set, and the remaining 1 fold is used as the test set. KNN ($K = 1$) is used as the classifier to obtain the balanced error rate.

LA-NSGA is compared with the following three methods:

1) *SFS* [5]: SFS starts with an empty feature subset and in each step the remaining feature whose inclusion results in the best score is included in the subset. This step is repeated until the classification error rate does not decrease any longer. Although SFS is traditionally used as a single-objective FS method, it can be used as a bi-objective one as the solutions that it produces are all Pareto optimal.

2) *NSGA-II* [9]: The NSGA-II framework is used to optimize the error rate and the number of used features in [9].

3) *MOFS-BDE* [35]: It is a recently proposed binary differential evolution algorithm with a self-learning strategy for bi-objective FS. The experimental results [35] show that it is superior to methods in [30], [38]. Similar to other EC-based bi-objective methods [9], it uses a length-fixed individual encoding.

¹ <https://www.cs.waikato.ac.nz/ml/weka/>

² <https://file.biolab.si/biolab/supp/bi-cancer/projections/>

For the parameter settings of LA-NSGA, the maximum iteration number (g) is set to 100. The population size (p) is set to 1/20 of the original number of features, and its lower and upper limits are set to 100 and 300, respectively. Crossover rate (ρ_c), mutation rate (ρ_m), and length change rate (ρ_l) are set to 0.8, 0.2 and 0.3, respectively. Local search is only performed at odd iterations for non-dominated individuals in the population in order to avoid premature convergence and reduce training time. SFS does not require any parameter input, and parameters of NSGA-II and MOFS-BDE are set as suggested in [9], [35]. The detailed parameter settings are listed in Table III. The stochastic MOEA-based algorithms are executed for 20 independent runs, and for each dataset, they are run for 200 times (20 runs \times 10 folds). Then, the average results with standard deviations are reported.

TABLE III
PARAMETER SETTING

Algorithm	Parameters setting
NSGA-II	$p = F /20, g = 100, \rho_c = 0.8, \rho_m = 0.2$
MOFS-BDE	$p = 50, g = 300, \rho_c = 0.3, \sigma = 0.01, tloc=5$
LA-NSGA	$p = F /20, g = 100, \rho_c = 0.8, \rho_m = 0.2, \rho_l = 0.3$

B. Experimental Results and Analysis

Table IV shows the experimental results. The third to fifth columns list the average training time (in minutes), the best HV value and the average one with standard deviation, respectively. The best results in these columns for each dataset are highlighted in bold. The last column S shows the results of the Wilcoxon rank sum test of LA-NSGA against its peers with a significance level of 0.05 [43]. “−”, “=”, and “+” indicate that LA-NSGA performs significantly better, similarly to and significantly worse than the others, respectively.

1) *Performance*: From Table IV we can observe that NSGA-II has the lowest HV value among all. Specifically, NSGA-II only has its best HV value greater than 0.5 on SRBCT, DLBCL, and Prostate, and only has the average HV value greater than 0.5 on SRBCT. By contrast, for SRBCT, the mean HV values of MOFS-BDE, SFS, LA-NSGA are 0.2, 0.3, and 0.4 higher than NSGA-II, respectively. NSGA-II underperforms because it adopts a length-fixed encoding and is difficult to produce solutions with only a few features for high-dimensional dataset.

Compared with NSGA-II, MOFS-BDE achieves slightly better performance and its HV value is higher on all datasets. For datasets with over 10 000 features (i.e., Brain2, Prostate, CLL_SUB, 11Tumors and Prostate2), the HV difference between MOFS-BDE and NSGA-II is less than 0.1 and for the other datasets the difference is larger. This is mainly owing to the self-learning strategy of MOFS-BDE, which further reduces the number of features selected by non-dominated individuals. However, as a length-fixed encoding-based method, its HV value is lower than SFS and LA-NSGA.

For all 12 datasets SFS achieves higher HV values than NSGA-II and MOFS-BDE. This is a rather surprising result, as SFS is not traditionally regarded as a bi-objective FS method but it outperforms the two bi-objective FS methods on

TABLE IV
EXPERIMENTAL RESULTS

Dataset	Algorithm	Time (min)	Best	Mean \pm Std	S
SRBCT	SFS	0.41	0.8617	—	—
	NSGA-II	8.0	0.5601	0.5511 \pm 0.006	—
	MOFS-BDE	8.0	0.7727	0.7543 \pm 0.022	—
	LA-NSGA	0.50	0.9993	0.9917\pm0.003	—
Breast	SFS	0.68	0.6417	—	—
	NSGA-II	35.5	0.3907	0.3837 \pm 0.004	—
	MOFS-BDE	35.4	0.5131	0.4931 \pm 0.014	—
	LA-NSGA	1.4	0.7362	0.7177\pm0.014	—
Leuk1	SFS	1.3	0.9149	—	—
	NSGA-II	44.7	0.4983	0.4841 \pm 0.009	—
	MOFS-BDE	20.9	0.6458	0.6245 \pm 0.015	—
	LA-NSGA	2.6	0.9287	0.9216\pm0.005	—
DLBCL	SFS	1.5	0.8330	—	—
	NSGA-II	48.8	0.5025	0.4907 \pm 0.008	—
	MOFS-BDE	23.1	0.6341	0.6100 \pm 0.014	—
	LA-NSGA	2.2	0.9331	0.9053\pm0.017	—
9Tumors	SFS	3.2	0.4997	—	—
	NSGA-II	31.1	0.2731	0.2621 \pm 0.008	—
	MOFS-BDE	20.2	0.3473	0.3292 \pm 0.010	—
	LA-NSGA	2.7	0.6991	0.6353\pm0.048	—
Brain1	SFS	3.2	0.7829	—	=
	NSGA-II	69.7	0.4391	0.4273 \pm 0.016	—
	MOFS-BDE	31.8	0.5619	0.5403 \pm 0.017	—
	LA-NSGA	3.7	0.8413	0.7825\pm0.033	—
NCI60	SFS	3.6	0.4901	—	—
	NSGA-II	35.3	0.3142	0.3098 \pm 0.004	—
	MOFS-BDE	22.5	0.4502	0.3965 \pm 0.030	—
	LA-NSGA	2.4	0.7185	0.6846\pm0.023	—
Brain2	SFS	6.2	0.7956	—	=
	NSGA-II	64.8	0.4145	0.3763 \pm 0.025	—
	MOFS-BDE	32.5	0.4501	0.4447 \pm 0.003	—
	LA-NSGA	8.0	0.8365	0.8042\pm0.021	—
Prostate	SFS	6.0	0.8831	—	—
	NSGA-II	125.4	0.5024	0.4884 \pm 0.008	—
	MOFS-BDE	57.9	0.5789	0.5757 \pm 0.002	—
	LA-NSGA	6.4	0.9707	0.9593\pm0.009	—
CLL_SUB	SFS	8.9	0.7904	—	—
	NSGA-II	156.5	0.4262	0.4181 \pm 0.004	—
	MOFS-BDE	75.5	0.4789	0.4626 \pm 0.010	—
	LA-NSGA	10.7	0.8875	0.8350\pm0.041	—
11Tumors	SFS	27.5	0.7034	—	—
	NSGA-II	392.6	0.4267	0.4187 \pm 0.005	—
	MOFS-BDE	170.2	0.4810	0.4749 \pm 0.004	—
	LA-NSGA	18.2	0.8924	0.8523\pm0.028	—
Prostate2	SFS	7.0	0.7772	—	—
	NSGA-II	131.1	0.4218	0.4032 \pm 0.012	—
	MOFS-BDE	62.1	0.4503	0.4380 \pm 0.011	—
	LA-NSGA	5.9	0.8949	0.8689\pm0.017	—

high-dimensional datasets. Precisely, its average HV value on Brain2, Prostate, Prostate2 and CLL_SUB is 0.3 higher than that of MOFS-BDE. This is because SFS adopts an incremental search manner for feature subsets and uses accuracy as an evaluation function, thereby making it obtain relatively high accuracy by using only a few features. Hence, in terms of the optimal proportion of selected features (i.e., f_2), SFS performs much better than NSGA-II and MOFS-BDE as it can always achieve the optimum on this objective (i.e., $1/n$, where n is the number of all features) whereas the two length-fixed EC methods get stuck easily in a high-dimensional space. However, for the error rate objective, SFS is prone to falling into a local optimum because of its simple search strategy and its HV value is lower than LA-NSGA. For example, on 9Tumors and NCI60, its HV values are less than 0.5, while those of LA-NSGA are 0.63 and 0.68, respectively.

LA-NSGA achieves the highest best value on all the 12 datasets. According to the Wilcoxon statistical test results, LA-NSGA is significantly better than its peers on 10 datasets, and only shows similar performance to SFS on Brain1 and Brain2. This implies that the Pareto front returned by LA-NSGA has better convergence and diversity than its peers.

It can be concluded that LA-NSGA achieves the best trade-off between the two considered objectives. This is due to the following four mechanisms of LA-NSGA: length-variable encoding, efficient initialization, a dominance-based adaptive length change operator, and dominance-based local search. Based on the length-variable encoding and initialization method, the number of used features in an initialized individual is largely reduced. Based on length change operator and local search, a length-adaptive evolution can be carried out.

2) *Computation Time*: As we can see from the third column of Table IV, NSGA-II incurs the most runtime among the four methods. For example, for 11Tumors, its runtime is 2.3, 14.2, and 21.5 times that of MOFS-BDE, SFS, LA-NSGA, respectively. This is because NSGA-II does not have any heuristic strategy to further reduce the number of used features.

SFS runs fastest among all in most cases because SFS selects a very small number of features, which greatly reduces the training time. However, when the number of original features is very large, SFS needs to perform evaluations of many feature subsets when selecting the next feature for adding, which seriously degenerates its execution efficiency. Therefore, on 11Tumors and Prostate, it runs slower than LA-NSGA.

In short, compared with NSGA-II and MOFS-BDE, the execution efficiency of LA-NSGA is greatly improved owing to its length-adaptive evolution. Moreover, its initialization method enables it to use fewer but more relevant features and the selection mechanism based on NSGA-II it adopts enables short individuals to have a greater survival probability because of their short lengths and relatively low error rate.

3) *Effectiveness Analysis*: For further analysis, Fig. 6 depicts the PF of the four methods on datasets 9Tumors, NCI60, CLL_SUB and 11Tumors, where the x-axis, using a base-10 log scale, represents the number of selected features, the y-axis represents the error rate and a point in each plot represents a feature subset. We can see that NSGA-II and

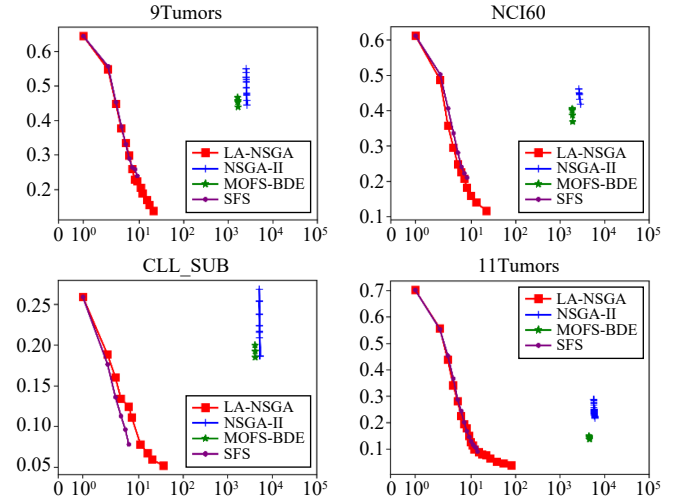


Fig. 6. The PF for each method.

MOFS-BDE perform much worse than SFS and LA-NSGA in terms of the optimal PF, i.e., using much more features does not help reduce the error rate. Compared with SFS, the PF of LA-NSGA has better convergence and diversity. For example, on 9Tumors and NCI60, LA-NSGA largely reduces the error rate by using slightly more features. Note that for a feature subset containing only one feature, SFS always performs the best because it exhaustively tests each feature.

On the other side, to verify the effectiveness of initialization methods and local search in LA-NSGA, the following two methods are used: NSGA-II and a LA-NSGA variant which disables local search (LA-NSGA-WO). Fig. 7 depicts the average HV value on the training set of the three methods during the evolution process, where x-axis represents the number of iterations, and y-axis is HV. A higher HV value indicates that the method performs better on the training set, and usually also achieves better performance on the test set. As shown in Fig. 7, the HV value of NSGA-II is much lower than the other methods throughout the evolution. LA-NSGA and LA-NSGA-WO have much higher HV values than NSGA-II after initialization, verifying the effectiveness of the proposed initialization method. The HV value of LA-NSGA rises faster than that of LA-NSGA-WO in the early iterations, and it is still higher than LA-NSGA-WO after the whole evolution, revealing that local search can improve the convergence and diversity of PF.

V. CONCLUSIONS AND FUTURE WORK

Aiming to select fewest features to achieve the highest performance, FS can be regarded as a bi-objective optimization problem. This paper proposes a length-adaptive non-dominated sorting genetic algorithm called LA-NSGA for bi-objective high-dimensional feature selection for the first time. Based on the NSGA-II framework, it is characterized by a length-variable encoding and a length-adaptive evolution mechanisms. Specifically, LA-NSGA uses an informative individual initialization method based on correlation and redundancy, and a length change operator to support a length-adaptive evolution in addition to a tailored crossover operator. Moreover, a dominance-based local search is designed and

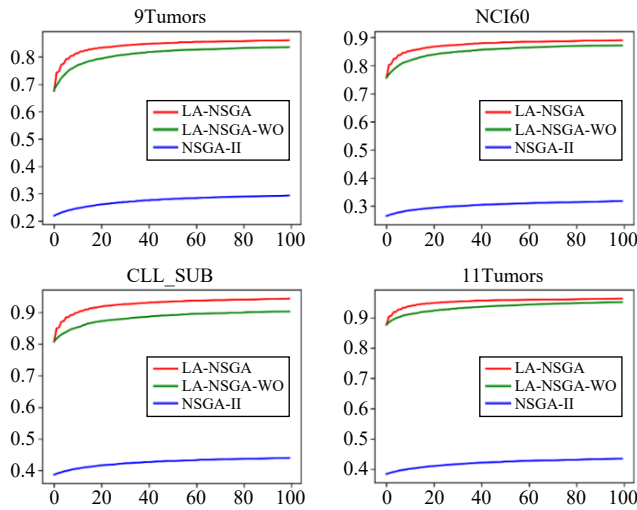


Fig. 7. The average hypervolume value in 100 iterations.

used to further improve the convergence and diversity of the Pareto optimal solutions. Experimental results on 12 high-dimensional gene datasets show that the Pareto front obtained by LA-NSGA is superior to those of some existing methods.

Our future work intends to focus on taking additional factors (e.g., missing labels and noise data) into account, and applying the length adaptive mechanism to some other intelligent optimization frameworks, e.g., indicator-based and decomposition-based ones [44]–[53].

REFERENCES

- [1] J. Li, K. Cheng, S. Wang, *et al.*, “Feature selection: A data perspective,” *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2017.
- [2] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.
- [3] H. Liu, M. Zhou, and Q. Liu, “An embedded feature selection method for imbalanced data classification,” *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, 2019.
- [4] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Machine Learning*, Amsterdam, The Netherlands: Elsevier, 1992, pp. 249–256.
- [5] J. Reunanen, “Overfitting in making comparisons between variable selection methods,” *J. Machine Learning Research*, vol. 3, no. 3, pp. 1371–1382, 2003.
- [6] H. Chen, *et al.*, “Robust decision trees against adversarial examples,” in *Proc. Inter. Conf. Machine Learning*, 2019, pp. 1122–1131.
- [7] X. Luo, X. Wen, M. Zhou, *et al.*, “Decision-tree-initialized dendritic neuron model for fast and accurate data classification,” *IEEE Trans. Neural Networks Learning Syst.*, vol. 33, no. 9, pp. 4173–4183, 2022.
- [8] R. Tibshirani, “Regression shrinkage and selection via the Lasso,” *J. Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.
- [9] T. M. Hamdani, J. M. Won, A. M. Alimi, *et al.*, “Multi-objective feature selection with NSGA II,” in *Proc. Inter. Conf. Adaptive Natural Computing Algorithms*, 2007, pp. 240–247.
- [10] S. Han, K. Zhu, M. Zhou, *et al.*, “Competition-driven multimodal multiobjective optimization and its application to feature selection for credit card fraud detection,” *IEEE Trans. Syst., Man, Cyber.: Syst.*, vol. 52, no. 12, pp. 7845–7857, 2022.
- [11] Z. Wang, S. Gao, M. Zhou, *et al.*, “Information-theory-based nondominated sorting ant colony optimization for multiobjective feature selection in classification,” *IEEE Trans. Cyber.*, 2022. DOI: 10.1109/TCYB.2022.3185554.
- [12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [13] B. H. Nguyen, B. Xue, and M. Zhang, “A survey on swarm intelligence approaches to feature selection in data mining,” *Swarm Evolutionary Computation*, vol. 54, p. 100663, 2020.
- [14] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, *Feature Selection for High-Dimensional Data*. Cham, Switzerland: Springer, 2015.
- [15] L. Yu and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution,” in *Proc. 20th Inter. Conf. Machine Learning*, 2003, pp. 856–863.
- [16] Y. Sun, S. Todorovic, and S. Goodison, “Local-learning-based feature selection for high-dimensional data analysis,” *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 32, no. 9, pp. 1610–1626, 2010.
- [17] A. Bommert, *et al.*, “Benchmark for filter methods for feature selection in high-dimensional classification data,” *Computational Statistics & Data Analysis*, vol. 143, p. 106839, 2020.
- [18] J. Lee, I. Y. Choi, and C.-H. Jun, “An efficient multivariate feature ranking method for gene selection in high-dimensional microarray data,” *Expert Syst. Applications*, vol. 166, p. 113971, 2021.
- [19] M. García-Torres, F. Gómez-Vela, B. Melián-Batista, *et al.*, “High-dimensional feature selection via feature grouping: A variable neighborhood search approach,” *Information Sciences*, vol. 326, pp. 102–118, 2016.
- [20] S. Gu, R. Cheng, and Y. Jin, “Feature selection for high-dimensional classification using a competitive swarm optimizer,” *Soft Computing*, vol. 22, no. 3, pp. 811–822, 2018.
- [21] W. Ma, X. Zhou, H. Zhu, *et al.*, “A two-stage hybrid ant colony optimization for high-dimensional feature selection,” *Pattern Recognition*, vol. 116, p. 107933, 2021.
- [22] B. Tran, B. Xue, and M. Zhang, “Variable-length particle swarm optimization for feature selection on high-dimensional classification,” *IEEE Trans. Evolutionary Computation*, vol. 23, no. 3, pp. 473–487, 2019.
- [23] N. D. Cilia, C. De Stefano, F. Fontanella, *et al.*, “Variable-length representation for EC-based feature selection in high-dimensional data,” in *Proc. Int. Conf. Applications Evolutionary Computation (Part of EvoStar)*, 2019, pp. 325–340.
- [24] J. Zhou, Q. Wu, M. C. Zhou, *et al.*, “LAGAM: A length-adaptive genetic algorithm with Markov blanket for high-dimensional feature selection in classification,” *IEEE Trans. Cybernetics*, 2023. DOI: 10.1109/TCYB.2022.3163577.
- [25] M. Labani, P. Moradi, and M. Jalili, “A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion,” *Expert Systems Applications*, vol. 149, p. 113276, 2020.
- [26] A.-D. Li, B. Xue, and M. Zhang, “Multi-objective feature selection using hybridization of a genetic algorithm and direct multisearch for key quality characteristic selection,” *Information Sciences*, vol. 523, pp. 245–265, 2020.
- [27] Y. Xue, H. Zhu, J. Liang, *et al.*, “Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification,” *Knowledge-Based Systems*, vol. 227, p. 107218, 2021.
- [28] Y. Zhou, W. Zhang, J. Kang, *et al.*, “A problem-specific non-dominated sorting genetic algorithm for supervised feature selection,” *Information Sciences*, vol. 547, pp. 841–859, 2021.
- [29] H. Xu, B. Xue, and M. Zhang, “A duplication analysis-based evolutionary algorithm for biobjective feature selection,” *IEEE Trans. Evolutionary Computation*, vol. 25, no. 2, pp. 205–218, 2021.
- [30] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimization for feature selection in classification: A multi-objective approach,” *IEEE Trans. Cyber.*, vol. 43, no. 6, pp. 1656–1671, 2013.
- [31] Y. Zhang, D.-W. Gong, and J. Cheng, “Multi-objective particle swarm optimization approach for cost-based feature selection in classification,” *IEEE/ACM Trans. Computational Biology Bioinformatics*, vol. 14, no. 1,

pp. 64–75, 2017.

- [32] M. Amoozegar and B. Minaei-Bidgoli, “Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism,” *Expert Systems Applications*, vol. 113, pp. 499–514, 2018.
- [33] Y. Zhou, J. Kang, S. Kwong, *et al.*, “An evolutionary multi-objective optimization framework of discretization-based feature selection for classification,” *Swarm Evolutionary Computation*, vol. 60, p. 100770, 2021.
- [34] A. Rashno, M. Shafipour, and S. Fadaei, “Particle ranking: An efficient method for multi-objective particle swarm optimization feature selection,” *Knowledge-Based Systems*, vol. 245, p. 108640, 2022.
- [35] Y. Zhang, D. Gong, X. Gao, *et al.*, “Binary differential evolution with self-learning for multi-objective feature selection,” *Information Sciences*, vol. 507, pp. 67–85, 2020.
- [36] U. Mlakar, I. Fister, J. Brest, *et al.*, “Multi-objective differential evolution for feature selection in facial expression recognition systems,” *Expert Systems Applications*, vol. 89, pp. 129–137, 2017.
- [37] X.-H. Wang, Y. Zhang, X. Y. Sun, *et al.*, “Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size,” *Applied Soft Computing*, vol. 88, p. 106041, 2020.
- [38] E. Hancer, B. Xue, M. Zhang, *et al.*, “Pareto front feature selection based on artificial bee colony optimization,” *Information Sciences*, vol. 422, pp. 462–479, 2018.
- [39] I. Aljarah, M. Habib, H. Faris, *et al.*, “A dynamic locality multi-objective salp swarm algorithm for feature selection,” *Computers & Industrial Engineering*, vol. 147, p. 106628, 2020.
- [40] E. F. Ohata, G. M. Bezerra, J. V. S. das Chagas, *et al.*, “Automatic detection of COVID-19 infection using chest X-ray images through transfer learning,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 1, pp. 239–248, 2021.
- [41] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [42] K. Shang, H. Ishibuchi, L. He, *et al.*, “A survey on the hypervolume indicator in evolutionary multiobjective optimization,” *IEEE Trans. Evolutionary Computation*, vol. 25, no. 1, pp. 1–20, 2021.
- [43] F. Wilcoxon, “Individual comparisons by ranking methods,” in *Breakthroughs Statistics*, New York, USA: Springer, 1992, pp. 196–202.
- [44] Y. Zhang, G. G. Wang, K. Li, *et al.*, “Enhancing MOEA/D with information feedback models for large-scale many-objective optimization,” *Information Sciences*, vol. 522, pp. 1–16, 2020.
- [45] S. Han, K. Zhu, M. C. Zhou, *et al.*, “A novel multiobjective fireworks algorithm and its applications to imbalanced distance minimization problems,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1476–1489, 2022.
- [46] Q. Fan and O. K. Ersoy, “Zoning search with adaptive resource allocating method for balanced and imbalanced multimodal multi-objective optimization,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 6, pp. 1163–1176, 2021.
- [47] Q. Kang, X. Song, M. C. Zhou, *et al.*, “A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms,” *IEEE Trans. Syst., Man, Cybernetics: Syst.*, vol. 49, no. 12, pp. 2416–2423, 2018.
- [48] X. Zhu and M. Zhou, “Multiobjective optimized cloudlet deployment and task offloading for mobile-edge computing,” *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15582–15595, 2021.
- [49] M. Cui, L. Li, M. Zhou, *et al.*, “Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems,” *IEEE Trans. on Evolutionary Computation*, vol. 26, no. 4, pp. 676–689, 2022.
- [50] Z. Lei, S. Gao, Z. Zhang, *et al.*, “MO4: A many-objective evolutionary algorithm for protein structure prediction,” *IEEE Trans. Evolutionary Computation*, vol. 26, no. 3, pp. 417–430, 2022.
- [51] H. Li, B. Wang, Y. Yuan, *et al.*, “Scoring and dynamic hierarchy-based NSGA-II for multiobjective workflow scheduling in the cloud,” *IEEE Trans. Autom. Science Engineering*, vol. 19, no. 2, pp. 982–993, 2022.
- [52] M. Cui, *et al.*, “A bi-population cooperative optimization algorithm assisted by an autoencoder for medium-scale expensive problems,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 11, pp. 1952–1966, 2022.
- [53] Y. Zhou, W. Xu, M. Zhou, and Z.-H. Fu, “Bi-Trajectory Hybrid Search to Solve Bottleneck-Minimized Colored Traveling Salesman Problems,” *IEEE Trans. Autom. Science Engineering*, 2023. DOI: 10.1109/TASE.2023.3236317



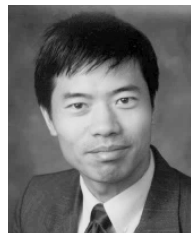
Yanlu Gong received the B.S. and M.S. degrees in computer science and technology from Chongqing Normal University, in 2017 and 2020, respectively. She is currently a Ph.D. candidate in computer science and technology at the College of Computer Science, Chongqing University. Her main research interests include machine learning and computational intelligence.



Junhai Zhou received B.S. degree in internet of things engineering from East China University of Technology in 2019. He is currently a master student in computer science and engineering at Chongqing University. His main research interests include feature selection and computational intelligence.



Quanwang Wu (Member, IEEE) received the B.S., M.S. and Ph.D. degrees in computer science from Chongqing University in 2007, 2010, and 2013, respectively. He was a Special Researcher at the Digital Content and Media Sciences Research Division of the National Institute of Informatics (NII) in Tokyo, Japan from 2014 to 2015. He is currently an Associate Professor with the College of Computer Science, Chongqing University. His interests include services computing, cloud computing and data mining.



Mengchu Zhou (Fellow, IEEE) received B.S. degree in control engineering from Nanjing University of Science and Technology in 1983, the M.S. degree in automatic control from Beijing Institute of Technology in 1986, and the Ph.D. degree in computer & systems from Rensselaer Polytechnic Institute, USA in 1990 and then joined New Jersey Institute of Technology in 1990, and is now a Distinguished Professor. His interests are in Petri nets, automation, Internet of Things, cloud/edge computing, and AI.

He has 1100+ publications including 14 books, 750+ journal papers (600+ in IEEE Transactions), 31 patents and 32 book-chapters. He is Fellow of IFAC, AAAS, CAA and NAI.



Junhao Wen received the Ph.D. degree in computer science and technology from Chongqing University in 2008, where he is a Professor with the College of Big Data and Software Engineering. His research interests include service computing, cloud computing and software dependable engineering.

He has published over 100 refereed journal and conference papers. He has over 30 projects and developed many commercial systems and software tools.