國 立 高 雄 應 用 科 技 大 學

資 訊 工 程 系 碩 士 班

碩 士 論 文

田口式二進制粒子族群最佳化演算法
應用於疾病預測

Hybrid Taguchi-Binary Particle Swarm Optimization for Disease Prediction

研 究 生：吳 國 銓 **(Kuo-Chuan Wu)**

指導教授：楊 正 宏 **(Cheng-Hong Yang)** 博士

中 華 民 國 九 十 九 年 七 月

田口式二進制粒子族群最佳化演算法應用於疾病預測

Hybrid Taguchi-Binary Particle Swarm Optimization for Disease Prediction

研 究 生：吳 國 銓 **(Kuo-Chuan Wu)**

指導教授：楊 正 宏 **(Cheng-Hong Yang)** 博士

國立高雄應用科技大學
資訊工程系碩士班
碩士論文

A Thesis
Submitted to
Institute of Computer Science and Information Engineering
National Kaohsiung University of Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of
Master of Engineering
in
Computer Science and Information Engineering

July 2010
Kaohsiung, Taiwan, Republic of China
中華民國九十九年七月

田口式二進制粒子族群最佳化演算法應用於疾病預測

學生：吳國銓　　　　　　　　　　　指導教授：楊正宏 博士

國立高雄應用科技大學資訊工程系碩士班

摘要

生物資訊學是一門結合統計、電腦科學應用於分子生物學的學科。基因表現(微陣列)及單核苷酸多型性為生物資訊學範疇之一，透過電腦探索其生物意義。本文利用田口式二進制粒子族群最佳化(特徵選取)及 K 最近鄰居法(分類問題)分析微陣列及單核苷酸多型性資料以利疾病之預測。其中加入田口方法作為區域搜尋以改善二進制粒子族群最佳化。實驗結果顯示，本研究方法能獲得較高的分類正確率及挑選出最重要的特徵。因此本論文方法，能用於其他應用特徵選取方法及分類問題的相關研究領域上。

關鍵字：基因表現、單核苷酸多型性、二進制粒子族群最佳化、K 最近鄰居法、田口方法

Hybrid Taguchi-Binary Particle Swarm Optimization for Disease Prediction

Student：Kuo-Chuan Wu　　　　　Advisors：Prof. Cheng-Hong Yang

Institute of Computer Science and Information Engineering
National Kaohsiung University of Applied Sciences

ABSTRACT

Bioinformatics is a study for the used of statistics and computer science in the area of molecular biology. Gene expression (Microarry) and single nucleotide polymorphism (SNP) are the bioinformatics tasks that are used for the computer to explore their biological information. In this thesis, it represents a disease prediction to analylize microarray and SNP data through machine learning. A feature selection as the binary particle swarm optimization (BPSO) and classification problem as K nearest neighbor (KNN) are used for analyzing both of the microarray and SNP data profiles in machine learning. The Taguchi method is used to improve BPSO for local search called TBPSO-KNN. The experimental results for both of the classification accuracy and the selected numbers of features show that the proposed method has the most important features and the highest accuracy. It is conceivable for implementing the feature selection in any other research projects.

Keywords: gene expression, single nucleotide polymorphism, binary particle swarm optimization, K nearest neighbor, Taguchi method.

誌　　謝

就在王建民站穩大聯盟的同時，正是我在研究所努力的同時。我的指導教授楊正宏博士，常以他做爲我的借鏡勉勵我。在研究領域上，他能以簡單的例子或是別的角度引導我思考；在爲人處事上，他能提供他的人生哲理或是社會實例導正我向善。因此，我在此衷心感謝楊教授細心指導。

由衷感謝洪集輝教授、莊麗月教授、張學偉教授及林威成教授費心於學生的論文，並能撥冗參加學生的畢業口試且給予寶貴的意見。感謝研究室的成員：世煉、煜輝、崇睿、譯嫺、怡伶、長軒、妍竹、育唐、兆軒、榮杰、宗牧、昇偉、禹融學長姊們，給予我的細心指導及呵護；我的好同學瑞鴻、智仁能彼此共勉一起努力以及明正、昱達兩位學弟在研究過程中無私的協助。另外感謝環安衛中心成員們及劉建偉大哥，在這段期間裡給予的關懷及照顧。

特別感謝我的親朋好友及我親愛的家人，在背後默默支持我，給予我無盡的包容及無形的支柱。有你們真好，我愛你們！最後以我很喜歡的一部電影「Into the wild」裡的對白"Happiness only real when shared…"，獻給愛我及我愛的人，因爲你們無私的分享讓我擁有真正的幸福！

國銓　謹誌

國立高雄應用科技大學

民國九十九年七月

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Since the DNA discovery first launched in 1953, many biologists began to pay their interest in DNA and gene decoding studies. Globlely, one after another, they investigated the purpose of DNA in so-called the "genetic engineering" area. Along with the completion of the Human Genome Project (HGP) [1], new opportunities and challenges has been presenting uptoday for uncovering the genetic basis of complex diseases via genome-wide association studies.

The goal of the HGP is to provide a tool to help scientists to understand the human genetic map and decipher the genetic code. Finally, the genomic nucleotide sequence can be interpreted and identify all human genes functions. After the completion of the HGP, the researches are entering the post-genomic era and its main task for use of sequence is to find out the information from the genomic nucleotide. This influence brings out a new interpretation for many diseases in medicine. The human DNA has 3 billion encoded base pairs of nucleotide bases are estimated. Thanks to the Bioinformatics discovery, biologists can perform the decoded bases independently with the help of computer science and statistics. For example, gene expression analysis (microarray), single nucleotide polymorphism (SNP), disease prediction, sequence analysis, sequence alignment, sequence prediction, and protein structure prediction etc [2-4] are popular applications for bioinformatics.

The application of microarray data in the classification of cancer types becomes favorable at present. Coupled with statistical techniques, gene expression patterns have been used in screening for potential tumor markers. The differential expressions of genes are statistically analyzed and then assigned into various classes, which is possible to enhance the understanding of the biological processes. The characteristics of microarray data have high dimension and small sample size, which make them difficult for general classification method to obtain the correct data of classification [5-7]. On the other hand, SNPs are known as the most common variant in the human genome, they play an important role in drug development, cancer and genetic disease research. SNPs are defined as single base pair positions in genomic DNA at which with different sequence alternatives (alleles) exist in normal individuals, these occur at appreciable frequency in an abundance of 1% or greater in the human population. The genome-wide SNP discovery, many genome-wide association studies are likely to identify multiple genetic variants that are associated with complicated diseases [8, 9].

The purpose of the classification is to build an efficient and effective model for predicting the class membership of data, which is expected to produce a correct label on the training data, and correctly predict the label on any unknown data. Determining an optimal feature subset is a very complex task, which proves decisive for the outcome of classification accuracy rate. The problem of microarray or SNPs data classification involves feature selection and classifier design. Feature selection is the process of choosing a subset of features from the original feature set and thus can be viewed as a principal pre-processing tool prior to solving the classification problems [10]. The goal of feature selection is to reduce the dimensionality of the problem and to retain the characteristics necessary for recognition, classification and/or the data mining process. A reliable selection method that obtains the relevant genes from the sample data is needed in order to decrease the classification accuracy rates and to avoid incomprehensibility.

Performing an exhaustive search over the entire solution space is not practical since this would require a long computing time associated with high cost. To overcome these feature selection problems, irrelevant and redundant features should be eliminated and only features relevant for the classification process should be considered. Deleting irrelevant features significantly improves the computational efficiency and lowers the classification accuracy rate. As many pattern recognition techniques are originally not designed to deal with the large amount of irrelevant or redundant features, however after combining feature selection techniques they become necessary to enhance pattern recognition efficiency [11-13].

The processed identifying relevant features and removing irrelevant features can be divided into three categories with different evaluation criteria for filters, wrapper and embedded models. The filtering process is separated from the classification process, and calculates a feature weight value for every feature. Based on this value, the better features are chosen to represent the original dataset. However, the filter approach does not account for interactions amongst features. For example: entropy-based method [14], information gain [15], mutual information [16], correlation-based feature selection (CFS) [17], etc., several methods are employed in the filter model for feature selection, The wrapper model uses an optimizing algorithm by adding or deleting features to produce various feature subset and uses a classification algorithm to evaluate the feature subset , such as genetic algorithm (GA) [18], tabu search [19] and particle swarm optimization (PSO) [10]. The embedded model uses the inductive algorithm itself as the feature selector so as the classifier, such as ID3 algorithm [20], C4.5 algorithm [21] and random forest [22].

In similarity of GA, PSO is an optimizer based on population. PSO has memory of its own, the knowledge of good solutions is retained by all the particles and an optimal solution can be found by the swarms following the best particle. In contrary to a GA, PSO does not incorporate the crossover and mutation processes. It has much more profound intelligent background and can be performed more easily. Based on these advantages, PSO is not only suitable for scientific research, but also use in engineering applications [23]. However, the distribution curve of PSO demonstrates two weakness, namely the linearization of the curve attained in steady-state and the location of the median [24].

Many feature selection methods resulting in locally optimal solutions are therefore combined with a local search process to improve their accuracy. For example Oh *et al*. [18] used a local search in their genetic algorithm. In this thesis, one used the Taguchi method as a local search method in PSO. The Taguchi method uses many ideas from statistical experimental design to improve or optimize products, processes and equipment. The Taguchi method uses two major tools: signal-to-noise ratio (SNR) measures the quality and orthogonal arrays (OAs) are used to study many designed parameters simultaneously. It has been successfully applied in machine learning and data mining, e.g., combined data mining and electrical discharge machining [25]. Sohn and Shin used the Taguchi experimental design for the Monte Carlo simulation of classifier combination methods [26]. Kwak and Choi used the Taguchi method for feature selection for classification problems [27]. Chen *et al.* optimized neural network parameter used Taguchi method [28].

The content of this thesis is organized as follow. In section 2, the detail descriptions of machine learning are given. The correlation-based feature selection, particle swarm optimization, Taguchi method, K nearest neighbor and hybrid Taguchi-binary particle swarm optimization are described in chapter 3. Chapter 4 is results and discussion. Finally, conclusion and future works are given in Chapter 5.

2. BACKGROUND

2.1 Machine learning

In recent years, machine learning (ML) is an emerging field and has been widely applied in various areas globlely. It draws on theory from many areas including statistics, mathematics and information science etc. ML is a study of making computers to have learning ability. The learning problem can be described as exploring a rule that utilizes data which are given only from a sample of limited size and limited known experiments [29]. Scientists mistakenly analyzed the process that are trying to build a variety of features in its relevance, it makes difficult to resolve certain problems. However, ML is often successfully solving those problems. Any approach of ML consists of two steps, the selection of a candidate model with the using of the learning algorithms and the estimation model parameter of the available data. In general model, the choosing of combination with parameter estimation are both operated at same time in the iteration. In many cases, the choosing of model is either by intuition or experience and sometimes are both. In other words, the user is based on the learning algorithm to choose model that is utilizing the model parameter of estimation.

ML algorithms can be divided into three categories [29]: 1) Supervised learning: this model is used from existing samples (i.e. training data), by utilizing them to find a deterministic function (model) that maps out the input to the desired output with future input-output minimum disagreement. Training data consist of the input component to the output component. The general output model is a continuous value (i.e. regression analysis) or classification tag (i.e. classification problem). 2) Unsupervised learning:

data clustering is a typical unsupervised learning approach. Given a set of untagged data, these data are assigned to differentiate subsets (i.e. clusters) by using clustering algorithms. Thus each subset has common (or similar) attribute. 3) Reinforcement learning: this model is to learn what to do, i.e. for finding a target strategy in order to define "good" and "bad" from each situation. Through the observation and learning to reward good case and punish bad case, after the continuous feedback, the model is established. In this article, we discussed details for the used of supervised learning in the next section.

2.1.1 Classification problem

According to the process of supervised learning, we generalized a rule for learning process as Figure 1.

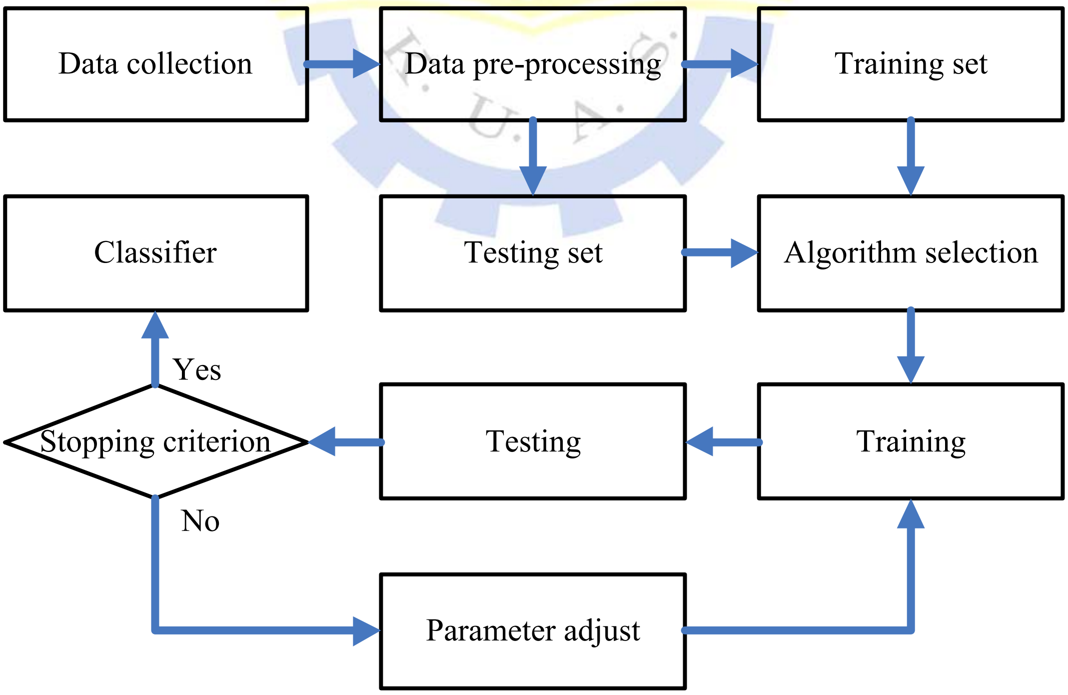


Figure 1 Process of supervised learning

2.1.1.1 Data preparation and data pre-processing

Data preparation includes analysis of original data and producing of higher quality data. The main compositions consist of data collection, data integration, data transformation, data cleaning, data reduction and data discretization. In the process of collecting data, it was necessary to provide what attributes, what characteristics or what function are important. Otherwise the use of the simple approach "brute-force", means to measure the whole related information and attribute weight of all data. However, the data information was obtained by brute-force approach is unsuitable to induce directly. Because these data include noise or missing data, thus it requires a lot of pre-processes. Depending on the difference of circumstances, the researchers had several methods to deal with these missing data or noise data, the general common approaches include: 1) ignoring the missing data; 2) replacing the missing data by experts; 3) replacing by mean or mode; 4) replacing by random [30, 31].

2.1.1.2 Algorithm selection

It is a decisive step to choose the specific learning algorithm. When the classifier is trained, the result shows us the satisfactory. Then the classifier is available for routine usage. Accuracy was used to estimate the classifier that means probability of correct classifying testing samples. There are several common technologies are applied for classifier validation, called cross validation. The details of validation technology are described as following section. Ideally, we would like the accuracy of classifier to be independent for the particular partitioning training data from the randomization process, because it makes much easier to replicate the experimental results to be published.

However, each experiment always has certain sensitivity in partitioning. Usually, ten repetitions are tested at several times from the same data with different random partitioning and then observing the outcome [32].

Several learning algorithms are classified into neural-based learner, rule-based learner and statistical learner as the following [33]:

Neural-based learner

The concept of artificial neural network (ANN) was proposed by Nilsson at mid-1960s. It is an artificial intelligence for pattern recognition based on neural like threshold units. ANN composed the simple elements based on mathematical model or computational model that tries to simulate and inspire by biological nervous systems. ANN needs the training network model to perform a particular function by adjusting the parameters or weights. And when it passes the input system through the network model to compute, it can predict and output into the output system [33, 34].

Rule-based learner

The theories of rule-based learning are usually consist of sets of discrete non-statistical rules. There are many available approaches which are rule-based learning methods for machine learning. One of the approaches is the decision tree that is divide-and-conquer approach or a top-down induction method. The goal of decision tree creates a model that is utilizing several input variables for prediction (i.e. classification). Each node of tree represents to one of the input variables and each leaf is a value of the target from the input variables. The node to the leaf called edge represents each possible problem of input variable [33, 35]. Figure 2 is an example of a typical problem of decision tree for classification. In order to classify a number of questions that have to be answered. This tree would classify by the weather to determine if one is going to play tennis or not, according to the Outlook, Humidity and Windy.

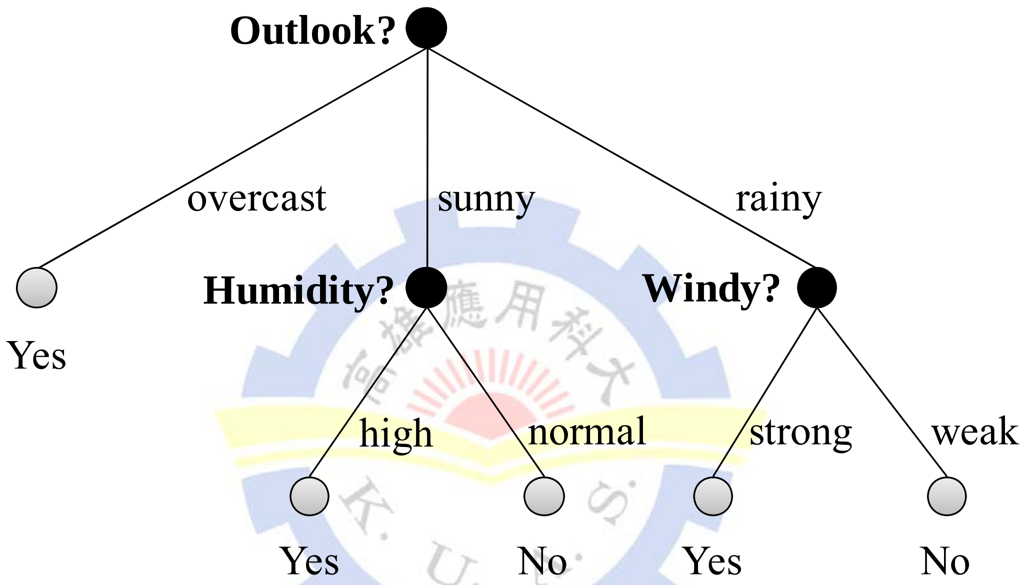


Figure 2 An example of decision tree for classification

Statistical learner

Statistical learning plays an important role in many areas of science. The main goal is to provide a framework for making predictions, decisions or classifications. This statistical framework is constructed from a set of data (i.e. training data) that is an assumption of process about the statistical nature. Recently, the statistical learning theory has received more attention from the pattern recognition especially when the support vector machine (SVM) was developed by Vapnik in the mid-1990s. The basic principle of K nearest neighbor (KNN) is that each unseen sample (e.g. testing data) was compared with the existing sample (e.g. training data) using Euclidean distance to calculate the distance metric (e.g. training model). The closest existing sample was assigned in different classes for the unseen sample. Naive Bayes (NB) is a simple classifier that calculates the maximum posterior probability based on Bayes theorem. The naive Bayes probability model was built from the independent feature. This model combines the maximum of a posteriori decision rule that selects a most probable hypothesis of common rule [33, 36].

2.1.1.3 Cross validation

In data mining like classification problem, a typical task is to construct a model from available data, such a model may be a classifier. We cannot be sure of that if a model can predict the future unseen data well, so the model needs to demonstrate the prediction capability. In statistics, a cross validation is an approach to estimate the generalization performance of prediction. Two or more learning algorithms can be compared through cross validation that can use in a statistical hypothesis test to know if one approach is superior than the another. There are three common cross validation methods including holdout validation, *m*-fold cross validation and leave-one-out cross validation, they are shown as follow [37, 38]:

Holdout cross validation

A simplest kind of cross validation method is called holdout cross validation is to separate the available data into the two non-overlapped sets (i.e. training set and testing set). It is common to split 2/3 of the data as the training sets and remaining 1/3 of data as the test sets. The model maybe a classifier fits in a function for using the training sets. And then the testing sets are used to predict the output for the using of data in the model [37]. The procedure of holdout cross validation is shown in Figure 3.

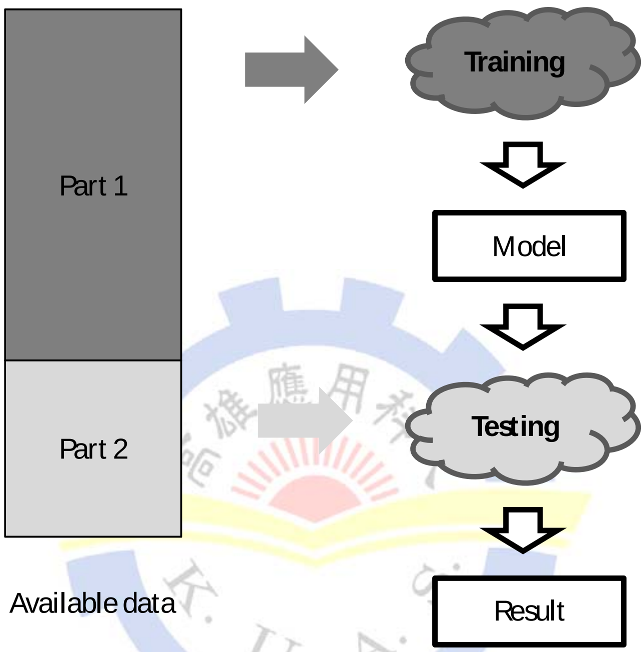


Figure 3 Illustration of procedure of holdout cross validation

m-fold cross validation

An improved cross validation approach from holdout validation method is called *m*-fold cross validation. In *m*-fold cross validation, the available data are separated into *m* non-overlapped and equally sized sets. A variant of these separated sets are randomly dividing the data into the training and testing sets as *m* in different times. The holdout method is repeated for *m* times. One of the *m* subsets is used as the testing sets and the remaining *m*-1 subsets as the training sets. Then the average accuracy across all *m* trials are calculated [37]. The procedure of *m*-fold cross validation is shown in Figure 4, here $m = 3$.

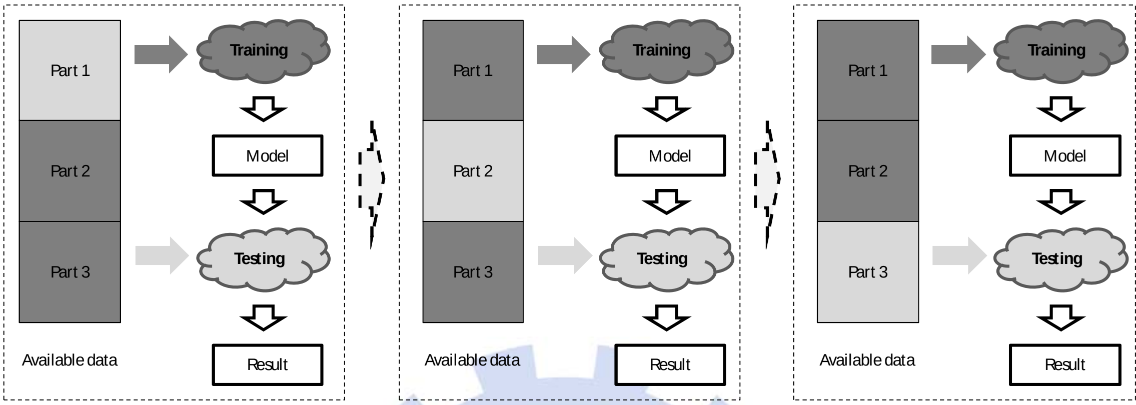


Figure 4 Illustration of procedure of 3-fold cross validation

Leave-one-out cross validation

A special case for *m*-fold cross validation, where *m* equals to the number of available data is called leave-one-out cross validation. The available data are separated and similar to *m*-fold cross validation. According to the previous calculation, the average accuracy across all *m* trials are calculated to estimate the model [37]. The procedure for leave-one-out cross validation is shown in Figure 5.

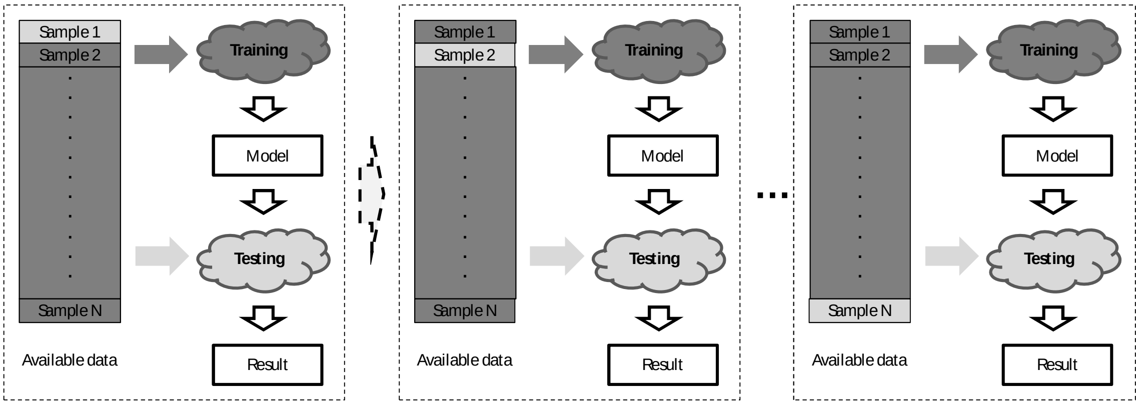


Figure 5 Illustration of procedure of leave-one-out cross validation

2.1.2 Feature selection

Feature selection is an important process of technology for high dimension data analysis. In other word, it selects a subset of d attributes from a set of D attributes based on some criterion, where $d < D$. Feature selection has been successfully applied in many areas of applications for its data sets from tens to hundred thousands of variables available. There are five main objectives of feature selections in pattern classification including: (a) finding the minimal size of feature subset that is successful, necessary and sufficient for the target concept, (b) improving the prediction accuracy performance for the models (maybe classifiers), (c) providing faster and more cost-effective models (maybe classifiers), (d) providing a better understanding of the underlying process that generates the data, (e) avoiding overfitting and improving model performance [11, 12, 39, 40]. Feature selection techniques are organized into three common models: the filter methods, wrapper methods and embedded methods. The three common taxonomy of feature selection techniques for each process of feature selection type are shown in Figure 6. The three common taxonomy of feature selection techniques for each concept of feature selection type are shown in Table 1. The details of three common taxonomy feature selection techniques are shown as following:

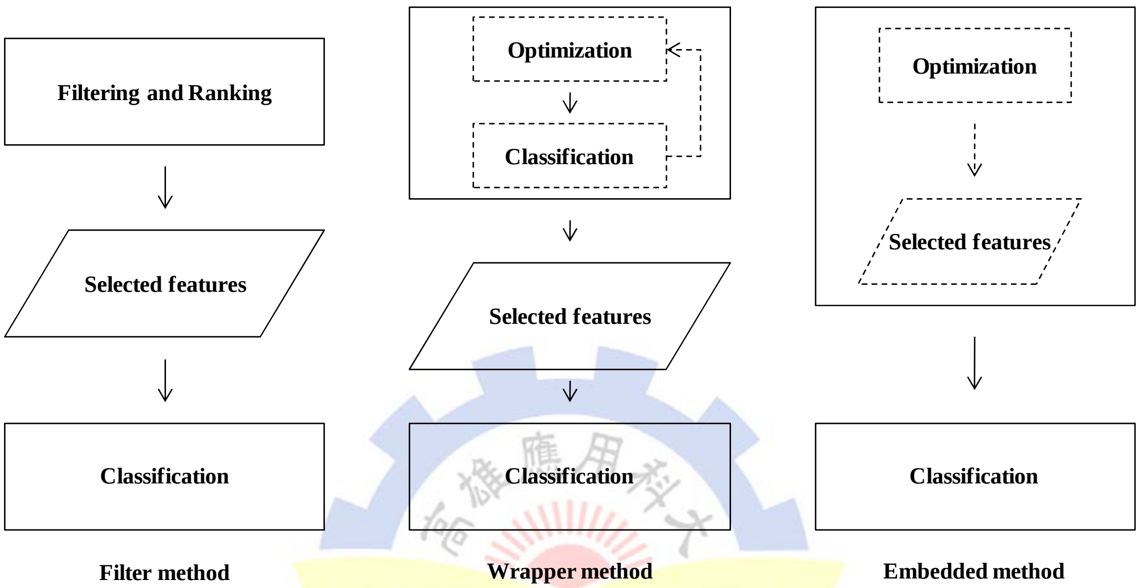


Figure 6 The process of three common model of feature selection

The source of this figure is modified from [7].

Table 1 The three common model of feature selection of overview

Model search	Advantages	Disadvantages	Examples
Filter	Univariate		
	Fast Scalable Independent of the classifier	Ignores feature dependencies Ignores interaction with the classifier	Euclidean distance *i*-test Information gain, Gain ratio
	Multivariate		
	Models feature dependencies Independent of the classifier Better computational complexity than wrapper methods	Slower than univariate techniques Less scalable than univariate techniques Ignores interaction with the classifier	Correlation-based feature selection Markov blanket filter Fast correlation-based feature selection
Wrapper	Deterministic		
	Simple Interacts with the classifier Models feature dependencies Less computationally intensive than randomized methods	Risk of over fitting More prone than randomized algorithms to getting stuck in a local optimum (greedy search) Classifier dependent selection	Sequential forward selection Sequential backward elimination Plus *q* take-away *r* Beam search
	Randomized		
	Less prone to local optima Interacts with the classifier Models feature dependencies	Computationally intensive Classifier dependent selection Higher risk of overfitting than deterministic algorithms	Simulated annealing Randomized hill climbing Estimation of distribution algorithms Genetic algorithms Tabu search Particle swarm optimization
Embedded	Interacts with the classifier Better computational complexity than wrapper methods Models feature dependencies	Classifier dependent selection	Decision trees C4.5 Weighted naive Bayes Feature selection using the weight vector of SVM

The source of this table is modified from literature [40].
The references of examples are also seeing [40].

2.1.2.1 Filter method

Filter techniques do not optimize the classification accuracy of the classifier directly. It assesses the relevance of attributes by looking only at the intrinsic characteristics of the data. In most cases, each feature relevance scores so-called "merits" are computed. The low-scoring features are removed or providing a generic selection of variables (i.e.

score ranking). After the feature is removed, this subset of attribute is presented as the input for the classifier. Several justifications for the filters of the feature selection have been forwarded in some special issues [11]. The advantages of filter, easily scale for very high-dimensional datasets. The algorithms are often simply fast caculated in computation. The filter is independent for the classifier. In contrary, the disadvantages of filter ignore the interaction with the classifier. Most proposed techniques are univalent. [40]. This disadvantage means that each feature is considered or calculated independently, therefore it cases the ignorance of feature dependencies which results in a worse classification accuracy when feature selections are compared. Hence, there are some multivariate filter techniques to overcome the incorporation of feature dependencies. Finally, some filters show the argument for providing a generic selection of variables that is not depending on learning machine. Another compelling justification is that the filter is used as a preprocessing step to eliminate attributes as well as overcoming the overfitting [11, 12, 40].

2.1.2.2 Wrapper method

Wrapper techniques evaluate the selected attribute subset according to their power to improve sample classification accuracy of the classifier. It requires a search space, operators, a search engine, and an evaluation function [12]. Wrapper techniques embed the model hypothesis within the search of feature subset, this good feature subset depends on the model selection found by the search engine. The classical approaches of the search engine include the forward selection and backward elimination. Recently, the evolutionary based of algorithms such as Genetic algorithm (GA) has been proposed as more advanced wrapper algorithm [7]. These search engines are divided in two categories: the deterministic and randomized (like GA) search algorithms [40]. For the evaluation function, it may be used to cross validation as the accuracy estimation criterion to evaluate a specific subset of features. The advantages of wrapper are the algorithms ability that the feature dependencies interaction between feature subset and classifier are taking into account. The algorithms usually obtain a higher classification accuracy. The common drawbacks of wrapper techniques are the algorithms which may cause a higher risk of overfitting than the filter. The algorithms are computationally intensive, which may give poor generalization property on the unseen data classification [7, 11, 40].

2.1.2.3 Embedded method

The embedded techniques use the inductive algorithm, however, the inductive algorithm itself represents the feature selector and the classifier. The embedded techniques search for an optimal subset of features that is built into the classifier construction and combined with space of feature subsets and hypotheses. Examples of these classification trees are ID3, C4.5 and random forest. The advantage of embedded is that the algorithms include the interaction with the classifier as wrapper method. The drawback of embedded is the algorithms that are generally based on greedy, using only the top ranking attributes to perform the sample of classification [7, 40].

2.1.3 Overfitting problem

Overfitting reveals when computational intensive search of algorithms are used. The estimation may be overfitted and yield biased of predictions under these circumstances [41]. If the training data lies too close together, the classifier predictions are shown in poor condition. This occurs when there is insufficient data to train the classifier and the data does not fully cover up the concept that is learnt by the machine. This problem is very common in many real world samples where the available data may rather be noisy [42]. In order to avoid overfitting, some additional techniques are being discussed, such as cross-validation, regularization, and early termination or resampling [43, 44]. However, the best way to avoid overfitting is to use an abundant amount of training data. In this thesis, the microarray data have a high dimension and small sample size, which is subsequently reduced by a filter feature selection. After feature reduction, the LOOCV technique enhances the training data for classification in a wrapper-based feature selection.

3. METHODS

3.1 Correlation-based feature selection

Correlation-based feature selection (CFS) is a filter feature selection method using a heuristic for evaluating the merit of a subset of feature. The heuristic takes the individual features useful for labeling the class and their inter-correlation into account. The hypothesis of CFS is based on the statement *Good feature subsets contain features highly correlated with (i.e., predictive of) the class, yet uncorrelated with (i.e. not predictive of) each other* [17].

This hypothesis is incorporated into the correlation-based heuristic evaluation equation as:

$$Merit_S = \frac{k\overline{\gamma_{cf}}}{\sqrt{k + k(k-1)\overline{\gamma_{ff}}}} \tag{1}$$

where $Merit_S$ is the heuristic merit of a feature subset S containing k features, $\overline{\gamma_{cf}}$ is the average feature and class correlation, and $\overline{\gamma_{ff}}$ is the average feature-feature intercorrelation ($f \in S$). Equation (1) is Pearson's correlation, where all variables have been standardized. General filter methods estimate the significance of a feature individually. CFS is then used to determine the best combination of attribute subsets via score values from the original data sets. The attributes are combined since they would be poor predictors of the class individually. Redundant attributes are discriminated against because they would be highly correlated with one or more of the other attributes [17].

Various heuristic search strategies, such as the best first method [45], are often applied to search the feature subset space in a reasonable time frame. We applied the best-first-method to calculate a matrix of feature-class and feature-feature correlation merits for CFS from the training data. The best-first-search starts with an empty set of features and generates all possible single feature expansions. Given enough time, a best-first-search will explore the entire feature subset space, so CFS uses a stopping criterion when subsets are found [17]. In order to calculate the merit of a feature set, the correlation between features is computed using symmetrical uncertainty (SU):

$$SU = 2.0 \times \left[\frac{H(Y) + H(X) - H(X,\ Y)}{H(Y) + H(X)} \right] \tag{2}$$

where $H(Y)$ and $H(X, Y)$ are defined as:

$$H(Y) = -\sum_{y \in Y} p(y) \log_2(p(y)) \tag{3}$$

where a probabilistic model of a feature Y can be formed by estimating the individual probabilities of the values $y \in Y$ from the training data. If feature Y in the training data is partitioned according to another feature X, then the relationship between features Y and X is given by:

$$H(Y \mid X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y \mid x) \log_2(p(y \mid x)) \tag{4}$$

SU compensates for the information gain's bias toward some attributes; the SU value is in the range [0, 1].

3.2 Binary particle swarm optimization

Particle swarm optimization (PSO) [46] is a population based optimization tool, which was originally introduced as an optimization technique for real-number spaces. In PSO, each particle is analogous to an individual "fish" in a school of fish. A swarm consists of N particles moving around a D-dimensional search space. The process of PSO is initialized with a population of random particles and the algorithm then searches for optimal solutions by continuously updating generations. Each particle makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. The position of the i_{th} particle can be represented by $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The velocity for the i_{th} particle can be written as $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The positions and velocities of the particles are confined within $[X_{\min}, X_{\max}]^D$ and $[V_{\min}, V_{\max}]^D$, respectively. The best previously visited position of the i_{th} particle is denoted its individual best position $p_{\text{i}} = (p_{i1}, p_{i2}, \ldots, p_{iD})$, a value called $pbest_i$. The best value of the all individual $pbest_i$ values is denoted the global best position $g = (g_1, g_2, \ldots, g_D)$ and called $gbest$. At each generation, the position and velocity of the i_{th} particle are updated by $pbest_i$ and $gbest$ in the swarm. However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and between levels of variables. For this reason, Kennedy and Eberhart [47] introduced binary PSO (BPSO), which can be applied to discrete binary variables. In a binary space, a particle may move to near corners of a hypercube by flipping various numbers of bits; thus, the overall particle velocity may be described by the number of bits changed per iteration. In BPSO, each particle is updated based on the following equations:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times \left(pbest_{id} - x_{id}^{old}\right) + c_2 \times r_2 \times \left(gbest_d - x_{id}^{old}\right) \tag{5}$$

$$\text{If } \; v_{id}^{new} \in (V_{mni},\ V_{\max}) \;\; \text{then} \;\; v_{id}^{new} = \max(\min(V_{\max},\ v_{id}^{new}),\ V_{\min}) \tag{6}$$

$$S(v_{id}^{new}) = \frac{1}{1 + e^{v_{id}^{new}}} \tag{7}$$

$$\text{if } r_3 < S(v_{id}^{new}) \text{ then } x_{id}^{new} = 1 \text{ else } x_{id}^{new} = 0 \tag{8}$$

In these equations, w is the inertia weight that controls the impact of the previous velocity of a particle on its current one, r_1, r_2 and r_3 are random numbers between [0, 1], and c_1 and c_2 are acceleration constants that control how far a particle will move in a single generation. Velocities v_{id}^{new} and v_{id}^{old} denote the velocities of the new and old particle, respectively. x_{id}^{old} is the current particle position, and x_{id}^{new} is the new, updated particle position. In formula (6), particle velocities of each dimension are tried to a maximum velocity $V_{\max}$. If the sum of accelerations causes the velocity of that dimension to exceed $V_{\max}$, then the velocity of that dimension is limited to $V_{\max}$. $V_{\max}$ and $V_{\min}$ are user-specified parameters (in our case $V_{\max} = 6$, $V_{\min} = -6$). The position of particles after updating is calculated by the function $S(v_{id}^{new})$ (formula (7)). If $S(v_{id}^{new})$ is larger than r_3, then its position value is represented by {1} (meaning this position is selected for the next update). If $S(v_{id}^{new})$ is smaller than r_3, then its position value is represented by {0} (meaning this position is not selected for the next update). The Pseudo code of BPSO as following:

Begin

Initialize particle swarm by randomly

While(stopping criterion is not met)

Evaluate fitness of particle swarm

Update *pBest* and *gBest*

Update X and V of particle swarm

Next generation until stopping criterion

End

3.3 Taguchi method

The Taguchi method was developed by Genichi Taguchi. It is a statistical method with a robust design. In a robust experimental design [48-50], processes or products can be analyzed and improved by altering relevant design factors. The commonly-used Taguchi method [48-50] provides two mechanisms, an orthogonal array (OA) and a signal-to-noise ratio (SNR), for analysis and improvement. If a particular target (i.e., process or product) has d different design factors, 2^d possible experimental trials will have to be considered in a full factorial experimental design. OAs are principally utilized to decrease experimental efforts associated with the d design parameters. An OA can be considered a fractional factorial experimental design matrix that provides a comprehensive analysis of interactions among all design factors, and fair, balanced and systematic comparisons of the different levels (or options) of each design factor. In the two-dimensional array, each column indicates a specific design parameter and each row represents an experimental trial with a particular combination of different levels for all design factors. The proposed scheme uses a common two-level OA for selecting representative features from the original feature set. A two-level OA can be defined as

$L_h(2^d)$, where d is the number of columns (i.e., the number of design parameters) in the orthogonal matrix, and $h = 2^k$ ($h > d$, $k > \log_2(d)$ and k is an integer) denotes the number of experimental trials; base 2 denotes the number of levels for each design parameter.

The SNR in the Taguchi method is used to determine the robustness of the levels of each design parameter. A "high quality" result for a particular target can be achieved by specifying design parameters at a specific level with a high SNR. The SNR is then utilized to analyze and optimize design parameters for a particular target. In Taguchi method classifies robust parameter design problems into different categories depending on the target of the problem. Typically, the smaller-the-better and larger-the-better SNR types are utilized [50]. Consider a set of t observations $\{y_1, y_2, \ldots, y_t\}$:

For the smaller-the-better characteristic, the SNR is determined as

$$SNR = -10\log\left(\frac{1}{n}\sum_{t=1}^{n} y_t^2\right) \tag{9}$$

For the larger-the-better characteristic, the SNR is determined as

$$SNR = -10\log\left(\frac{1}{n}\sum_{t=1}^{n} \frac{1}{y_t^2}\right) \tag{10}$$

For instance, for a particular target that has 15 design parameters with two levels (i.e., levels 0 and 1), a two-level OA $L_{16}(2^{15})$ can be generated (as shown in Table 2). In this two-level OA, only 16 experimental trials are required for evaluation, analysis and improvement. Conversely, all possible combinations of 15 design factors (i.e., 2^{15}=32768) should be considered in the full factorial experimental design, which is frequently inapplicable in practice. Once an OA is generated, an observation or objective function of each experimental trial can be determined.

Table 2 $L_{16}(2^{15})$ Orthogonal array

Number of experimental trial	Design Factors														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Column Number														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
5	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
6	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1
7	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
8	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0
9	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
10	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
11	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1
12	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0
13	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
14	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0
15	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0
16	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1

Suppose an illustrative example with two sets of observations, A = {78, 89, 86, 99, 85, 90} and B = {85, 90, 99, 98, 82, 92} is given. For the smaller-the-better characteristic, the SNR of sets A and B are,

$$SNR_A = -10\log\left(\frac{1}{t}\sum_{i=1}^{t} y_i^2\right)$$

$$= -10\log\left[\frac{1}{6}\left(78^2 + 89^2 + 86^2 + 99^2 + 85^2 + 90^2\right)\right]$$

$$= -38.90dB$$

and

$$SNR_B = -10\log\left(\frac{1}{t}\sum_{i=1}^{t} y_i^2\right)$$

$$= -10\log\left[\frac{1}{6}\left(85^2 + 90^2 + 99^2 + 98^2 + 81^2 + 92^2\right)\right]$$

$$= -39.19dB$$

Similarly, for the larger-the-better characteristic, the SNR of sets *A* and *B* are,

$$SNR_A = -10\log\left(\frac{1}{t}\sum_{i=1}^{t}\frac{1}{y_i^2}\right)$$

$$= -10\log\left[\frac{1}{6}\left(\frac{1}{78^2}+\frac{1}{89^2}+\frac{1}{86^2}+\frac{1}{99^2}+\frac{1}{85^2}+\frac{1}{90^2}\right)\right]$$

$$= 38.81dB$$

And

$$SNR_B = -10\log\left(\frac{1}{t}\sum_{i=1}^{t}\frac{1}{y_i^2}\right)$$

$$= -10\log\left[\frac{1}{6}\left(\frac{1}{85^2}+\frac{1}{90^2}+\frac{1}{99^2}+\frac{1}{98^2}+\frac{1}{81^2}+\frac{1}{92^2}\right)\right]$$

$$= 39.10dB$$

The SNR is utilized in the Taguchi methods to determine the robustness of all levels of each design parameter. That is, “high quality” of a particular target can be achieved by specifying each design parameter with a specific level having a high SNR. For both the smaller-the-better and larger-the-better characteristics, the SNR of *A* is better than that of *B*.

3.4 K nearest neighbor

The K nearest neighbor (KNN) method is one of the most popular nonparametric methods [51, 52] used for classification of new objects based on attributes and training samples. KNN consists of a supervised learning algorithm which instantly classifies the results of a query instance based on the majority of the KNN category. Classifiers do not use any model for KNN and are determined solely based on the minimum distance from the query instance to the training samples. Any tied results are solved by a random procedure.

Given training data {($\mathbf{x}_1$, y_1), ($\mathbf{x}_2$, y_2), …, ($\mathbf{x}_i$, y_i), ..., ($\mathbf{x}_n$, y_n)} and test data $\mathbf{x}$, where $\mathbf{x}$ is the feature vector of the data, y_i is the class of data $\mathbf{x}_i$, and n is number of data, the distance measure can be defined as $d(x,\ x_i) = \sqrt{\sum_{j=1}^{d}(x_j - x_{ij})^2}$, where d is the dimension of the feature vector. The nearest neighbor rule is nnr($\mathbf{x}$) = y_k，where k = arg min$_i$ $d(\mathbf{x}, \mathbf{x}_i)$. A voting strategy is used if K>1. For example if K=3, three minimal distance measures are calculated; if two points fall into class A and one point falls into class B, class A is chosen.

Begin
    For *i* = 1 to number of test set
        For *j* = 1 to number of train set
            Calculating distance of test with train set
        Next *j*
    Next *i*
    For *k* = 1 to number of parameter K
        Determine class of test set by vote strategy
    Next *k*
    Determine the classification accuracy
End

3.5 Hybrid Taguchi-Binary Particle Swarm Optimization

3.5.1 TBPSO without parameter optimization

This section introduces a correlation-based feature selection method to implement a gene selection preprocess, and then combines it with a Taguchi-binary particle swam optimization. The K-NN with the LOOCV method serves as a classifier to calculate the classification accuracy. The flowchart of CFS-TBPSO is shown in Figure 7 and a detailed description of the individual steps is given below.

Step 1) A feature subset is generated by CFS using Weka [38].

Step 2) Initialize population of particles with random position X ($X \in \{x_1, x_2, ..., x_N\}$) and velocities V ($V \in \{v_1, v_2, ..., v_N\}$) where N is the number of particles; each position of a particle is a candidate for feature subsets CS.

Step 3) Calculate the fitness for each particle and determine the average classification accuracy for training set T (denoted ACC(T, S_j) where S_j is the feature subset) using the K-NN classification rule with the LOOCV technique.

Step 4) Update the individual best solution *pBest*, and global best solution *gBest* according to the fitness evaluation results (i.e., accuracy). The number of selected features is also considered.

Step 5) If *gBest* stays unchanged for m times go to next step. Otherwise go to Step 7.

Step 6.1) Two particles, denoted b_1 and b_2, are randomly selected from the population. Consider that b_1 and b_2 have w different bits ($w \leq n$). Then the Taguchi method is employed on these two particles.

Step 6.2) Generate an "extended" two-level OA with respect to the above particular w bits (i.e., features or factors) of b_1 and b_2. The level of feature i in the OA will be replaced by the corresponding bit of b_1 if the original level is 0. Conversely, the level of feature i in the OA will be replaced by the corresponding bit of b_2 if the original level is 1. Notably, the levels of the remaining $(n - w)$ bits in the two-level OA are the same as the corresponding bits of b_1 and b_2. In each experimental trial j, levels 1 or 0 in each column i of the extended two-level OA indicate whether feature i is selected or not selected in the corresponding feature set S_j for pattern classification.

Step 6.3) ACC(T, S_j) is considered an observation or objective function of the experimental trial j in the extended two-level OA. ACC(T, S_j) is the same as in Step 3. This process, called function value, is used to measure the quality of each feature set or solution S_j.

Step 6.4) Calculate the corresponding SNR for each level (i.e., levels 1 and 0) of the particular w bits according to observations from all experimental trials in the extended two-level OA.

Step 6.5) Generate a better solution *t_best* based on the results in the extended two-level OA. For all w bits in *t_best*, each bit is determined by value 1 if its corresponding SNR for level 1 is greater than that for level 0, and vice versa. Notably, the remaining $(n - w)$ bits of *t_best* are the same as those of b_1 and b_2.

Step 6.6) Repeat Step 6.1-6.5 until each particle has finished the local search process.

Step 6.7) Update *gBest* and *pBest* the same was as in Step 4.

Step 7) Update the velocity and position of each particle according to formulas (5) to formula (8).

Step 8) Repeat Steps 3-7 until a certain number of iterations have been completed. Consequently, the best feature subset is obtained.

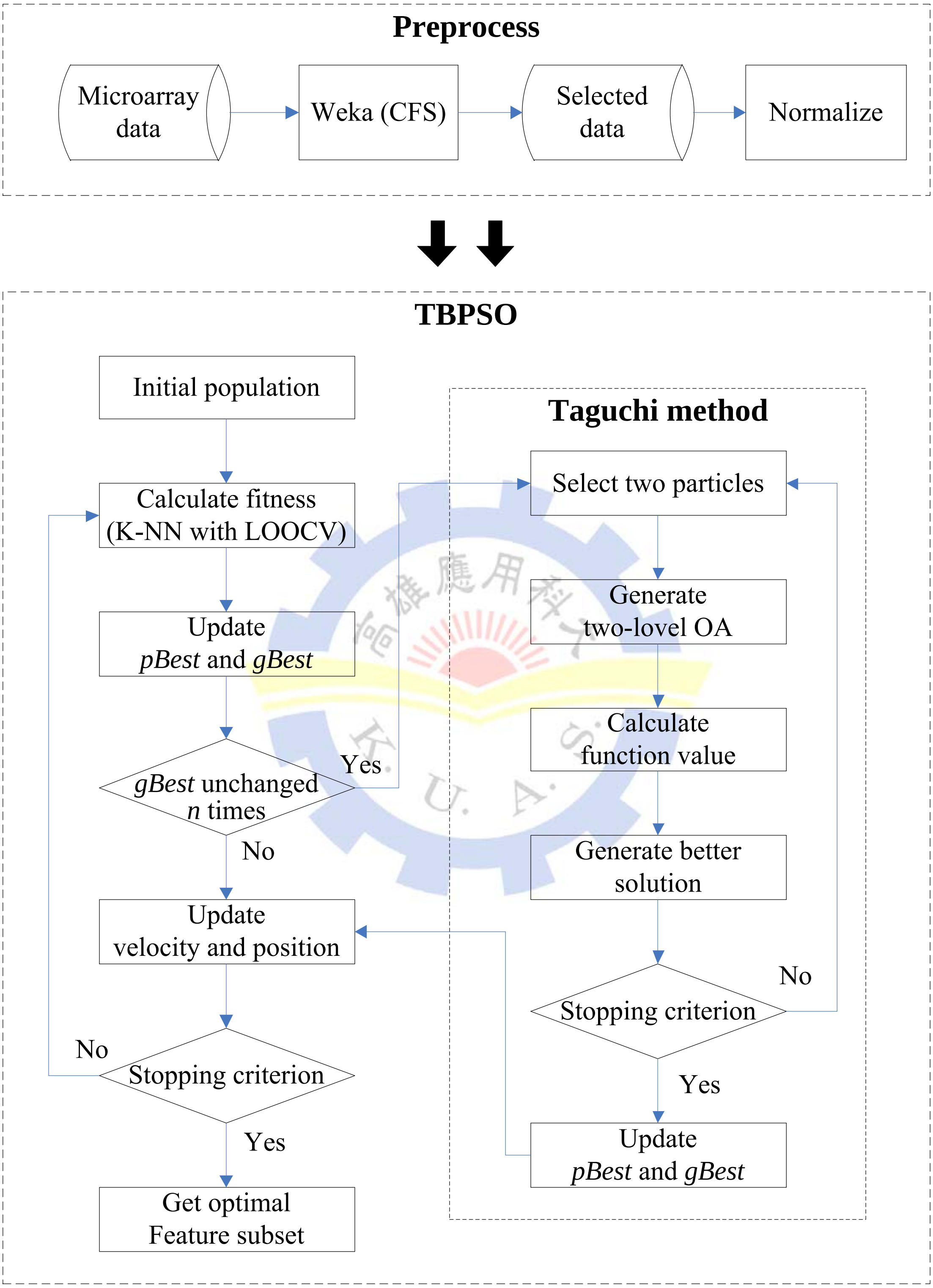


Figure 7 Flowchart of CFS-TBPSO on microarray data

3.5.2 TBPSO with parameter optimization

This section introduces an optimizer algorithm to implement a SNP selection and the classifier parameter optimization proeprocess. The holdout cross validation be used in outer loop that separates two parts: training set and testing set. By running BPSO processing on as many training sets. *m*-fold cross validation be used in the inner loop to guide the search of the feature selection and parameter optimization process on the training data. The flowchart of BPSO is shown in Figure 8 and how the steps executed is described at detailed below.

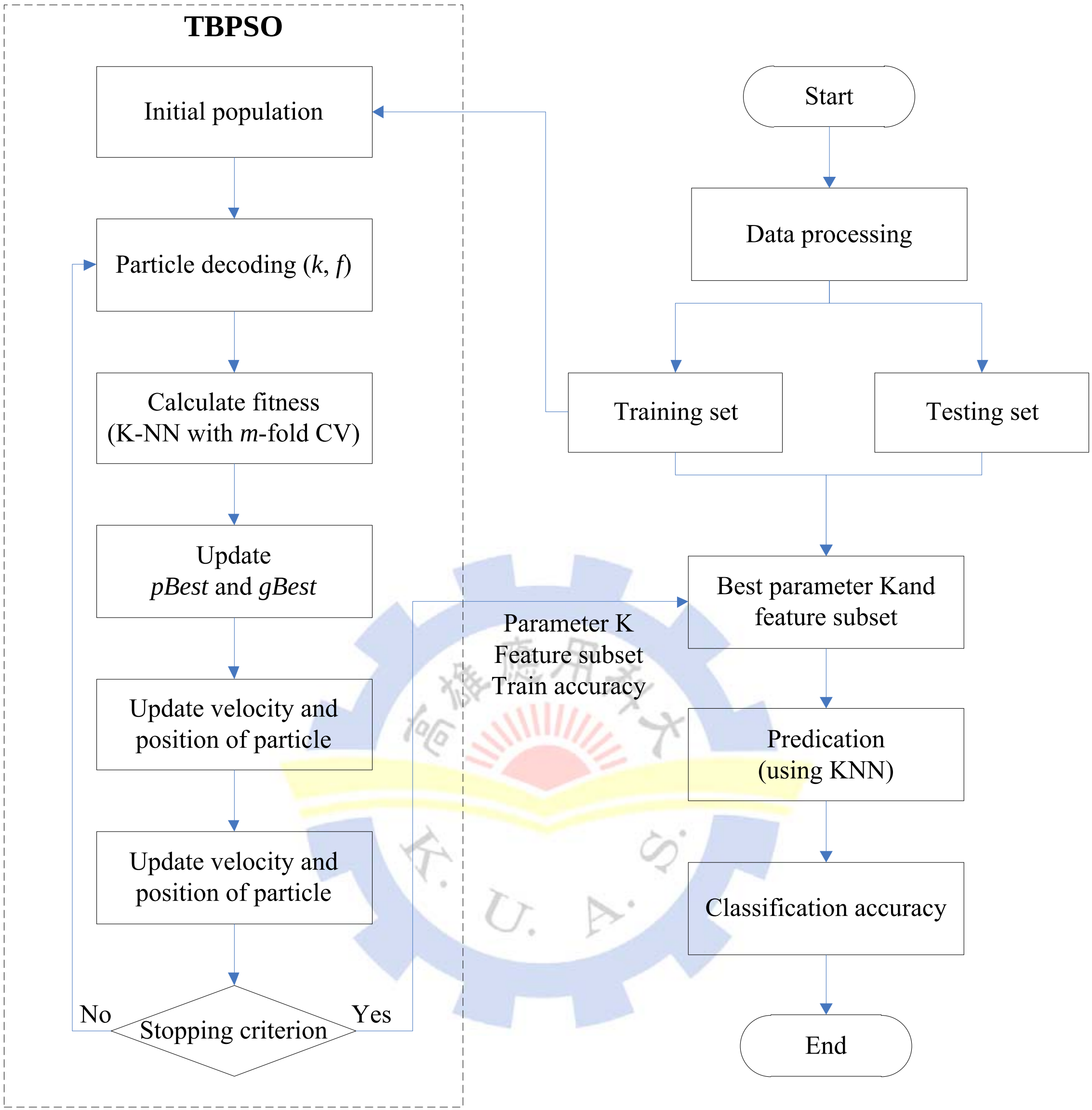


Figure 8 Flowchart of TBPSO-KNN on SNPs data

3.5.3 Particle encoding design

To implement our proposed approach, this research used KNN classifier because the parameter K depends on the specific data. The parameter K and features used as input attributes must be optimized. Hence, the particle encoding desing comprises two parts, K and the features mask. In order to sutable feature selection problem, the binary coding system was used to represent the particle. Figure 9 shows the binary particle representation of our design. The $X_k^1 \ldots X_k^i \ldots X_k^{n_k}$ represents the value of parameter K, $X_f^1 \ldots X_f^i \ldots X_f^{n_f}$ represents the feature mask. n_k is the number of bits representing parameter K. n_f is the number of bits representing the features. Here, the n_k are set to 5, in order to the parameter K be an odd positive integer. Through conversion the $K \in \{1, 3, \ldots, 43\}$, for example the when $X_k^1 \ldots X_k^i \ldots X_k^{n_k}$ = 00000 then K = 1, when $X_k^1 \ldots X_k^i \ldots X_k^{n_k}$ = 00001 then K = 3; $X_k^1 \ldots X_k^i \ldots X_k^{n_k}$ = 11111, K = 63. n_f equals the number of features varying from the datasets. For particle representing the feature mask, the bit with value '1' represents the feature is selected, and '0' indicates feature is not selected.

$X_k^1 \ldots X_k^i \ldots X_k^{n_k}$	$X_f^1 \ldots X_f^i \ldots X_f^{n_f}$

Figure 9 The diagram of particle coding design

3.5.4 Fitness function

A position of particle as a solution, which is comprise a feature subset and parameter K. Classification accuracy and the number of selected features are the two criteria used to design a fitness. The accuracy was obtained by KNN classifier with *m*-Fold cross validation to estimate the individual feature subset and parameter K. The fitness function was shown as follows:

$$fitness(x_{id}) = Accuracy_{\text{KNN with } m\text{-Fold cross validation}} \tag{11}$$

3.5.5 Scheme and process

Step 1) In data processing, the SNP data of osteoporosis were normalized to [-1, 1] and separate into training set and testing set using holdout cross validation.

Step 2) Initialize population of particles with random position X ($X \in \{x_1, x_2, ..., x_N\}$) and velocities V ($V \in \{v_1, v_2, ..., v_N\}$) where N is the number of particles; each position of a particle is a candidate for feature subsets CS.

Step 3) Calculate the fitness for each particle and determine the average classification accuracy for training set T (denoted ACC(T, S_j) where S_j is the feature subset) using the K-NN classification rule with the *m*-fold cross validation technique.

Step 4) Update the individual best solution *pBest*, and global best solution *gBest* according to the fitness evaluation results (i.e. accuracy). Here we also consider the number of selected features.

Step 5) Update the velocity and position of each particle according to formula (5) to formula (8).

Step 6) Taguchi method is used be local search as chapter 4.

Step 7) Repeat Steps 3-6 until a certain number of iterations have been completed. Consequently, the best feature subset is obtained.

Step 8) To predict using the testing data set with the best feature subset and K into K-NN classifer.

3.5.6 Illustrative example

This section provides an example that illustrates the details, in particular the steps regarding the Taguchi method (Steps 6.1-6.6 of section 3.1.5) of the proposed CFS-TBPSO feature selection method. In the Breast-Cancer pattern classification problem [53] with 683 instances, each instance x_e has a set of 10 attributes, denoted {A, B, C, D, E, F, G, H, I, J}. Each specific feature subset is encoded as a string of ten binary digits (or bits). Each feature can be described by a binary digit with the value 1 or 0, which indicates whether the feature is selected or not selected in the corresponding feature subset.

Two candidate feature subsets, b_1 and b_2 (0000000111 and 0001111000, respectively) are randomly selected from the population in Step 6.1 (as shown in Table 3). These two candidate feature subsets are comprised of seven different bits, i.e. features D, E, F, G, H, I and J. Accordingly, an "extended" two-level OA with respect to the above *seven* bits of b_1 and b_2 is generated (Table 4). The levels of the remaining *three* features in the "extended" two-level OA are the same in b_1 and b_2.

Table 3 Position of particles

Factors	A	B	C	D	E	F	G
Level 1(particle x_1)	0	0	0	0	1	1	1
Level 2(particle x_2)	1	1	1	1	0	0	0

Table 4 Two-level orthogonal array

Number of experimental trial	Design Factors (Features)						
	A	B	C	D	E	F	G
	Column Number						
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

In Step 6.2, the level of feature i in the OA will be replaced by the corresponding bit of b_1 if the original level is 0. Conversely, the level of feature i in the OA will be replaced by the corresponding bit of b_2 if the original level is 1. Consequently, a new, extended two-level OA (as shown in Table 5) with respect to the above *seven* bits of b_1 and b_2 is obtained. The levels of the remaining *three* features in the two-level OA are the same as the corresponding bits of b_1 and b_2.

In each experimental trial j in the new, extended two-level OA, levels 1 or 0 in each column i indicate whether feature i is selected or not selected in the corresponding feature set S_j. For each feature set S_j, $ACC(T, S_j)$ can be determined using the K-NN classification rule with the LOOCV technique. $ACC(T, S_j)$ is considered an observation or objective function of experimental trial *j* in the new, extended two-level *OA*. For example, the average classification accuracy of feature subset {H, I, J} (i.e., experimental trial 1 in Table 5) is 92.24%. This process, the fitness evaluation, is used to measure the quality of each feature set or solution S_j. The experimental layout and signal-to-noise data of the Breast-Cancer pattern classification problem is summarized in Table 5. The larger-the-better characteristic (formula (10)) is selected for calculating the SNR as maximum classification accuracy is preferred in pattern classification. Next, as shown in Table 5, the corresponding SNR for each level of the particular *seven* features can be calculated according to observations from all experimental trials in the new, extended two-level *OA*. As a result, a better solution *t_best*, encoded 0001111111, can be obtained based on the results in Table 5. For all the *seven* bits in *t_best*, each bit is determined by value 1 if its corresponding SNR for level 1 is greater than that for level 0, and vice versa. The average classification accuracy of the better solution *t_best* is 95.31%, a value that is significantly better than that of the feature subset in each experimental trial in the new, extended two-level OA.

Table 5 Generation of better position from two particles using Taguchi method

	Factors										
				A	B	C	D	E	F	G	
Experiment number	Column number										Function Value
	1	2	3	4	5	6	7	8	9	10	
1	0	0	0	0	0	0	0	1	1	1	92.24
2	0	0	0	0	0	0	1	0	0	0	89.75
3	0	0	0	0	1	1	0	1	0	0	91.51
4	0	0	0	0	1	1	1	0	1	1	94.14
5	0	0	0	1	0	1	0	0	1	0	92.97
6	0	0	0	1	0	1	1	1	0	1	94.86
7	0	0	0	1	1	0	0	0	0	1	92.09
8	0	0	0	1	1	0	1	1	1	0	95.17
E_{F1}				39.26	39.31	39.30	39.29	39.41	39.43	39.40	
E_{F2}				39.44	39.39	39.40	39.41	39.29	39.28	39.30	
Optimal level				x_2	x_2	x_2	x_2	x_1	x_1	x_1	
Optimal position	0	0	0	1	1	1	1	1	1	1	95.31

4. RESULTS AND DISCUSSION

4.1 The data set

4.1.1 Microarray data

The experiment data sets of this study were downloaded from http://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html [5]. They consist of *Leukemia*, *Breast* 2 *class*, *Breast* 3 *class*, *NCI* 60, *Adenocarcinoma*, *Brain*, *Colon*, *Lymphoma*, *Prostate*, and *Srbct*. The data set format was arranged as shown in Table 6, which includes the number of patients, genes, and classes. Generally, feature value scaling can enhance pattern recognition accuracy, hence the data sets were normalized to [0, 1]. The normalization is given by formula (12), where f'_{value} is the scaled value of a feature, f_{value} is the original value of a feature, $value_{\mathrm{MAX}}$ is the upper bound of the feature value, and $value_{\mathrm{MIN}}$ is the lower bound of the feature value.

$$f'_{value} = \frac{f_{value} - value_{MIN}}{value_{MAX} - value_{MIN}} \tag{12}$$

Table 6 Format of ten microarray classification data sets

Method / Data sets	Genes	Patients	Classes	Original reference
Leukemia	3051	38	2	[54]
Breast 2 class	4869	78	2	[55]
Breast 3 class	4869	96	3	[55]
NCI 60	5244	61	8	[56]
Adenocarcinoma	9868	76	2	[57]
Brain	5597	42	5	[58]
Colon	2000	62	2	[59]
Lymphoma	4026	62	3	[60]
Prostate	6033	102	2	[61]
Srbct	2308	63	4	[62]

(1) Genes: number of genes for gene microarray data. (2) Patients: number of patients for gene microarray data. (3) Classes: number of classes for gene microarray data. (4) Original ref.: reference for gene microarray data.

4.1.2 SNP data

Osteoporosis data was approved by the Institutional Review Board of Kaohsiung Medical University, Kaohsiung, Taiwan. All subjects signed the informed consent. No individual was receiving or had previously received hormone replacement therapy. Women with surgical menopause were excluded. Clinical data, including body mass index, smoking history, and blood pressure, were collected. This teaching hospital had 1,500 beds and is located in Southern Taiwan. The characteristics of study subjects were randomly recruited from general health inspection in the Center of Health Examination, Department of Preventive Medicine, Kaohsiung Medical University [73]. Fifty premenopausal (mean age 43 years) and 254 postmenopausal women (mean age 59 years) were involved in this study (Postmenopausal women is defined as more than 6 months without the menstruation occur or more than 59 years old). The attributes are age, menopausal and eleven SNPs (TNFα-857, TGFβ1-509, Osteocalcin, TNFα-308, *BstB* Ⅰ, *Dra* Ⅱ, IL1_ra, HSP70 hom, HSP 70-2, CTR and BMP-4, detail see Table 7) respectively, the number of total feature is thirteen. The type of SNP genotype is symbol, we convert to numerical such as Table 7 shown {-1, 0, 1}.

Table 7 The data type of SNP of osteoporosis

SNP	Chromosome	Gene (location)	rs number	Genotype		
				-1	0	1
1	6	TNFα-857	rs1799724	TT	TC	CC
2	19	TGFβ1-509	rs1800469	TT	TC	CC
3	1	Osteocalcin	rs1800247	CC	CT	TT
4	6	TNFα-308	rs1800629	AA	AG	GG
5	11	PTH (*BstB* Ⅰ)	rs6254	GG	AG	AA
6	11	PTH (*Dra* Ⅱ)	rs6256	AA	AC	CC
7	2	IL1_ra[c]	VNTR[b]	A1A1	A1A2	A1A4
8	6	HSP70 hom	rs2227956	CC	CT	TT
9	6	HSP 70-2	rs1061581	GG	AG	AA
10	7	CTR	rs1801197	CC	CT	TT
11	14	BMP-4	rs17563	CC	CT	TT

[a] Data source [73]; [b] Variable number tandem repeats; [c] IL1_ra genotype: A1, 410 bp; A2, 240bp; and A4, 325 bp.

4.2 Parameter setting

The CFS was implemented under the Weka [38] environment (http://www.cs.waikato.ac.nz/ml/weka/). The parameters, inertia weight w and the acceleration factors c_1 and c_2, need to be considered in BPSO. The balance between the global exploration and local search ability is controlled by w. A large inertia weight facilitates the global search, while a small inertia weight facilitates the local search. c_1 and c_2 control the movement of particles. To avoid premature BPSO convergence, the adjustment should not be too excessive, since this might result in extreme particle movement, which makes impossible to obtain an optimized feature. Hence, suitable parameter adjustment is paramount. In this thesis, we adopted c_1 and c_2 equal to 2 and w was set to 0.8. We set $[v_{\min}, v_{\max}] = [-6, 6]$, which yields a range of [0.9975, 0.0025] using the sigmoid limiting transformation formula (7). The parameters used have the same values as the parameters in Shi and Eberhart [63]. The new standard PSO definition said 50 particles were performed best, and this literature suggests the population size between 20 – 100 particles [74]. Hence, here we set to 100 for microarray data and set to 50 for SNP data. The generation size we set to 30, because in high computational classification algorithm and we can obtain superior performance in few generation sizes. Finally, all Weka parameter of experiments were set to default. Except the population and generation size of GA was the same BPSO that in order to compare with BPSO.

4.3 Experimental results

4.3.1 Experiment of microarray data

Table 8 gives a comparison of classification error rates obtained by methods taken from the literature [5] and the BPSO/1NN, CFS/1NN, CFS-BPSO/1NN, and CFS-TBPSO/1NN methods. The BPSO, CFS-BPSO and CFS-BPSO algorithms were applied to ten microarray data sets and independently executed 10 times for each data set. Four methods from the literature, Random Forest (s.e.=0), Random Forest (s.e.=1), Shrunken centroids (SC.s) and Nearest neighbor variable selection (NN.vs), were used for the comparison. In Table 8, the average classification error rate is 0.162, 0.102, 0.026, and 0.015 for the BPSO/1NN, CFS/1NN, CFS-BPSO/1NN, and CFS-TBPSO/1NN method, respectively. In CFS-TBPSO/1NN, the classification error rate is zero in six out of the ten data sets (Leukemia, NCI 60, Adenocarcinoma, Brain, Colon, Lymphoma, and Srbct). The classification error rates of the other three data sets, Breast 2 class (0.012), Breast 3 class (0.010), and Prostate (0.005), are close to zero. Table 9 shows the number of genes selected by the methods. The number of features selected by the proposed method is lower than the number of features selected by the BPSO/1NN, CFS/1NN, CFS-BPSO/1NN methods. In CFS-TBPSO/1NN, the number of genes in the *Leukemia* and *Adenocarcinoma* microarray data sets were reduced from 3051 genes to 3.9 genes and from 9868 genes to 15.9 genes, respectively (Table 9). This indicates that the proposed approach not only significantly reduces the classification error rate, but also effectively eliminates redundant or unnecessary features. Figure 10 shows the number of genes selected by the proposed method compared to the other methods for the microarray data sets.

Table 8 Classification error rate of feature selection methods for the microarray data

Method / Dataset	RF (s.e. = 0) # Error	RF (s.e. = 1) # Error	SC.s # Error	NN.vs # Error	BPSO 1NN # Error	CFS 1NN # Error	CFS-BPSO 1NN # Error	CFS-TBPSO 1NN # Error
Leukemia	0.087	0.075	0.062	0.056	**0**	**0**	**0**	**0**
Breast 2 class	0.337	0.332	0.326	0.337	0.338	0.182	0.056	**0.026**
Breast 3 class	0.346	0.364	0.401	0.424	0.385	0.368	0.153	**0.097**
NCI 60	0.327	0.353	0.246	0.237	0.251	0.098	0.023	**0**
Adenocarcinoma	0.185	0.207	0.179	0.181	0.158	0.105	0.014	**0.013**
Brain	0.216	0.216	0.159	0.194	0.143	0.048	**0**	**0**
Colon	0.159	0.177	0.122	0.158	0.189	0.129	0.013	**0**
Lymphoma	0.047	0.042	0.033	0.04	0.016	**0**	**0**	**0**
Prostate	0.061	0.064	0.089	0.081	0.100	0.069	0.019	**0.014**
Srbct	0.039	0.038	0.025	0.031	0.044	0.016	**0**	**0**
Average	0.180	0.187	0.164	0.174	0.162	0.102	0.026	**0.015**

(1) Random Forest # Error: classification error rate of Random Forest with s.e. = 0. (2) Random Forest # Error: classification error rate of Random Forest with s.e. = 1. (3) SC.s # Error: classification error rate of shrunken centroids with minimization of error and minimization of features if ties. (4) NN.vs # Error: classification error rate of nearest neighbor with variable selection. (5) CFS # Error: classification error rate of only correlation-based feature selection used. (6) CFS-BPSO # Error: classification error rate of correlation-based feature selection with binary particle swarm optimization. (7) CFS-TBPSO # Error: classification error rate of correlation-based feature selection with Taguchi - binary particle swarm optimization. Lowest classification error rates are in bold-type.

Table 9 Number of genes selected by the feature selection methods for the microarray data

Dataset \ Method	RF (s.e. = 0) # Genes	RF (s.e. = 1) # Genes	SC.s # Genes	NN.vs # Genes	BPSO # Genes	CFS # Genes	CFS-BPSO # Genes	CFS-TBPSO* # Genes
Leukemia	2	2	82	512	1335.8	44	4.1	3.9
Breast 2 class	14	14	31	88	2309.1	62	29.5	28.3
Breast 3 class	110	6	2166	9	2368.6	83	39.0	34.9
NCI 60	230	24	5118	1718	2505.8	97	46.1	43.4
Adenocarcinoma	6	8	0	9868	4599.0	52	19.1	15.9
Brain	22	9	4177	1834	2527.7	146	48.8	45.0
Colon	14	3	15	8	950.7	58	20.0	16.4
Lymphoma	73	58	2796	15	1794.3	229	63.7	63.2
Prostate	18	2	4	7	2963.6	63	23.4	22.8
Srbct	101	22	37	11	1089	98	22.3	23.3

(1) RF # Genes: number of genes selected for Random Forest with s.e. = 0. (2) RF # Genes: number of genes selected for Random Forest with s.e. = 1. (3) SC.s # Genes: number of genes selected for shrunken centroids with minimization of error and minimization of features if ties. (4) NN.vs # Genes: number of genes selected for nearest neighbor with variable selection. (5) BPSO #Genes: number of genes selected for binary particle swarm optimization (6) CFS # Genes: number of genes selected for correlation-based feature selection. (7) CFS-BPSO # Genes: number of genes selected for correlation-based feature selection with binary particle swarm optimization. (8) CFS-TBPSO # Genes: number of genes selected for correlation-based feature selection with Taguchi - binary particle swarm optimization. *: the numbers shown here are average numbers produced over 10 trial runs.

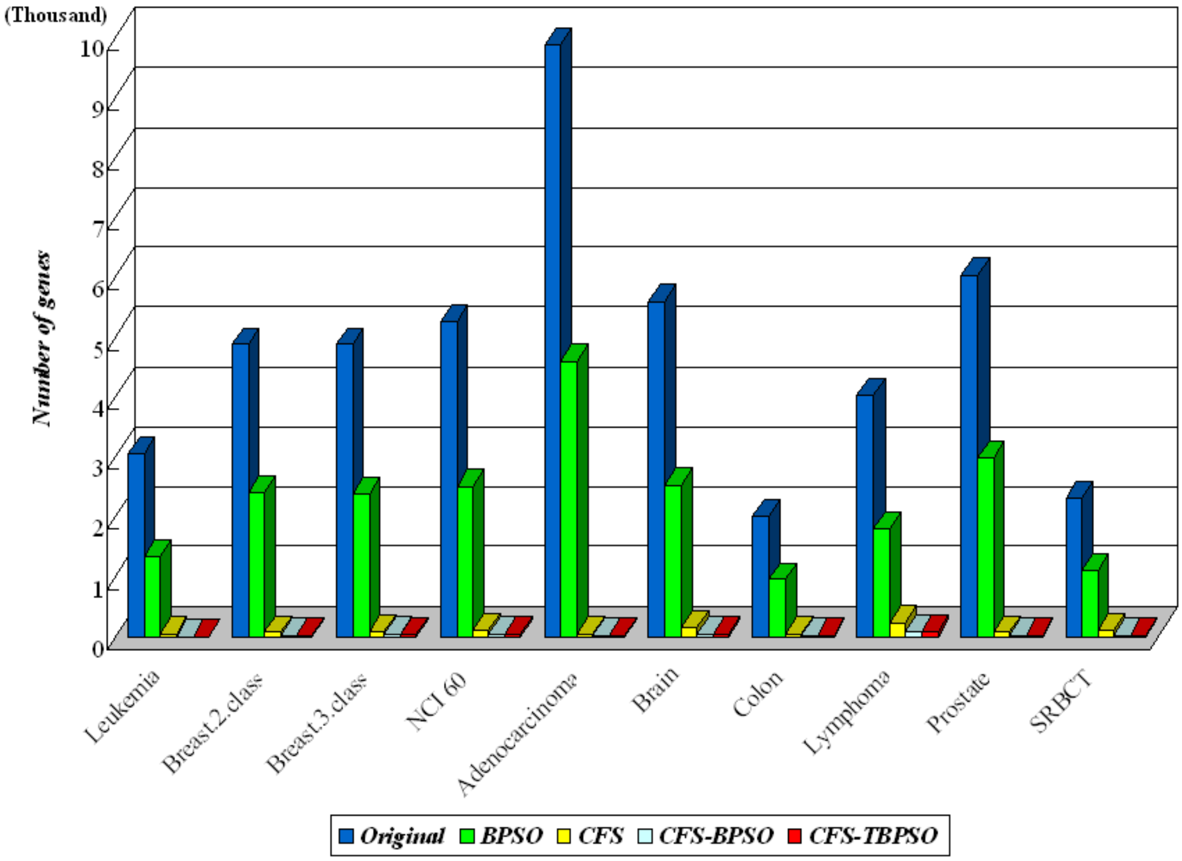


Figure 10 Number of Selected genes

Table 10 shows the best value, mean value, and standard deviation of the error classification rates for ten runs. The number of mean values that are equal to zero for BPSO/1NN, CFS-BPSO, and CGS-TBPSO are one, four, and six, respectively. The number of standard deviations that are equal to zero for BPSO/1NN, CFS-BPSO, and CGS-TBPSO are four, four, and seven, respectively. The other standard deviations of the error classification rates (Breast 2 class, Breast 3 class, and Prostate data set) approach zero. This shows that the proposed method is more stable than either BPS/1NN or CFS-BPSO/1NN.

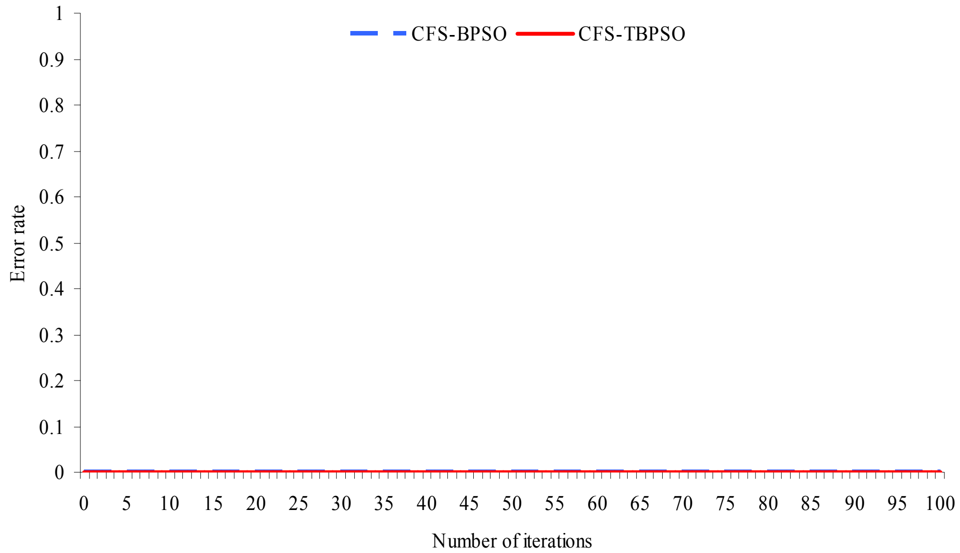
Table 10 Comparison of Best, Mean and SD results for BPSO, CFS-BPSO and CFS-TBPSO

Method / Dataset	BPSO			CFS-BPSO			CFS-TBPSO		
	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
Leukemia	0	0	0	0	0	0	0	0	0
Breast 2 class	0.325	0.338	0.009	0.039	0.056	0.014	0.013	0.026	0.012
Breast 3 class	0.368	0.385	0.007	0.137	0.153	0.009	0.084	0.097	0.010
NCI 60	0.246	0.251	0.008	0.016	0.023	0.009	0	0	0
Adenocarcinoma	0.158	0.158	0	0.013	0.014	0.004	0.013	0.013	0
Brain	0.143	0.143	0	0	0	0	0	0	0
Colon	0.177	0.189	0.011	0	0.013	0.007	0	0	0
Lymphoma	0.016	0.016	0	0	0	0	0	0	0
Prostate	0.088	0.100	0.006	0.010	0.017	0.003	0.010	0.014	0.005
Srbct	0.032	0.044	0.007	0	0	0	0	0	0

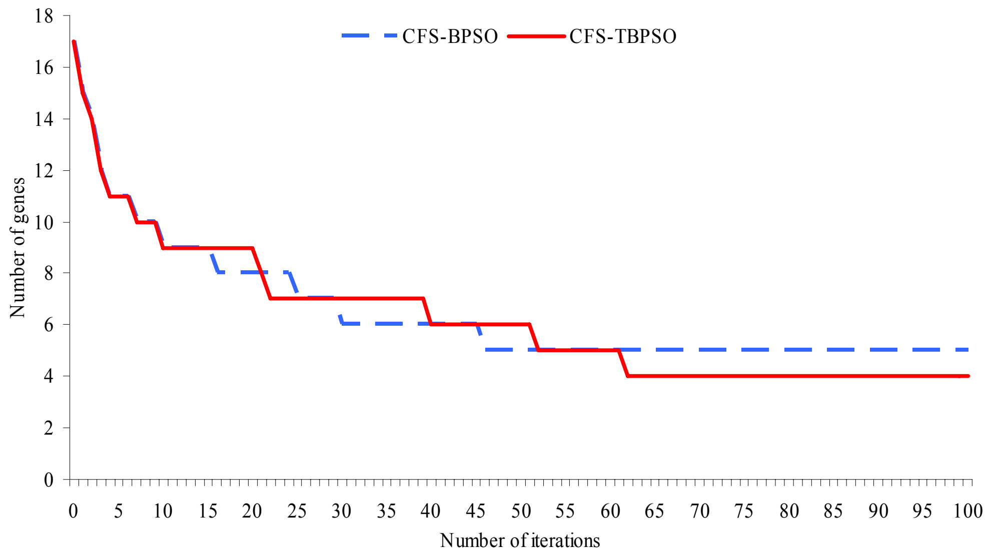
SD: standard deviation.

Figure 11 to Figure 20 show the graphs CFS-BPSO and CFS-TBPSO for 100 generations for the ten microarray data. In these figures, *a* represents the number of iterations vs. the classification error rate, *b* represents the number of iterations vs. the number of genes selected. The dotted line represents CFS-BPSO, and continuous line represents CFS-TBPSO. Figure 21 (a) shows that the Taguchi method effectively avoids a local optimum at the 15th and 44th iteration. Thus, the Taguchi method had a lower classification error rate. Figure 21 (b) shows that although at the 55th iteration performance did not immediately improve, the Taguchi method still led particles beyond the regional solutions during subsequent iterations of the search. Finally, Figure 22 to Figure 26 details the statistical performance of the ten independent runs in BPSO, CFS-BPSO and CFS-TBPSO. It can be observed that CFS-TBPSO obtained the best solution for all data sets, and that its standard deviation was also the smallest. This proves that CFS-TBPSO obtains a globally optimal solution.

CFS-TBPSO produced error rate of zero for some data sets. The main reason for this zero error rate is the two-stage feature selection process for gene expression data. In the first stage, we aim at all features using a filter approach (CFS). CFS calculates a correction-based feature weight for each feature, and thus identifies relevant features. The feature weight is used to set a threshold value for filtering out noise data. In the second stage, a wrapper approach (TBPSO) is implemented to again selected features. Huang *et al.* [64] mention that selection of a minimal number of relevant genes improves classification performance. The experimental results of the proposed method proved that a low could indeed be obtained. The zero error rate produced by an evolutionary algorithms such as a genetic algorithm is not surprising in gene selection and classification problems. For the Leukemia data set, many studies have obtained a zero error rate with evolutionary algorithms, e.g. [65] and [66].

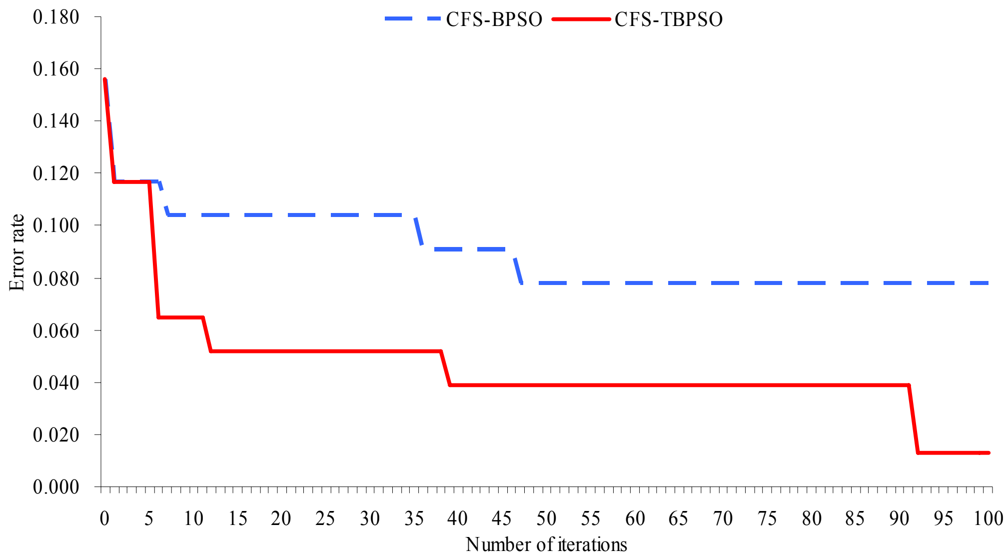


Leukemia (a)

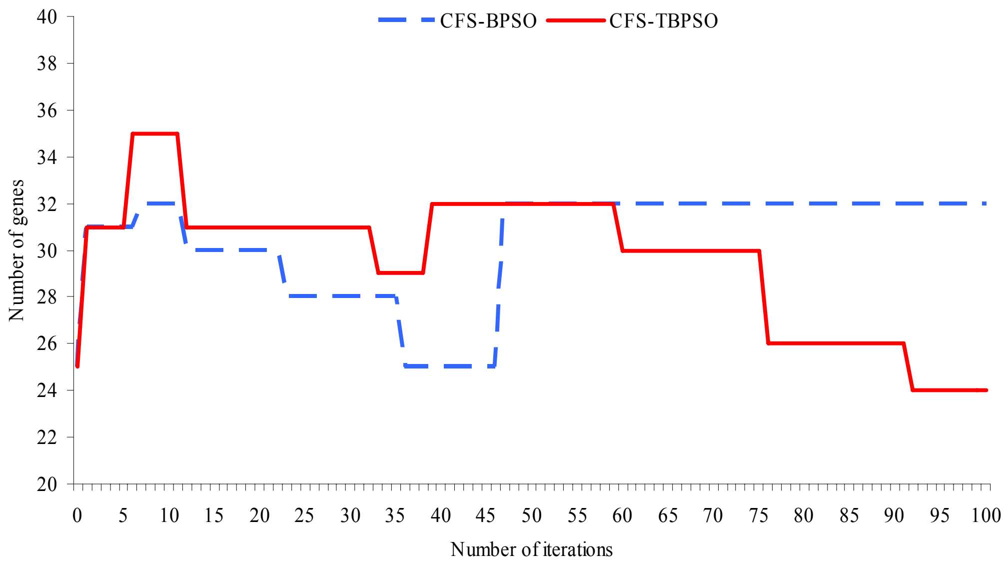


Leukemia (b)

Figure 11 Number of iterations vs. Classification error rate (a) and features (b) in Leukemia of microarray data

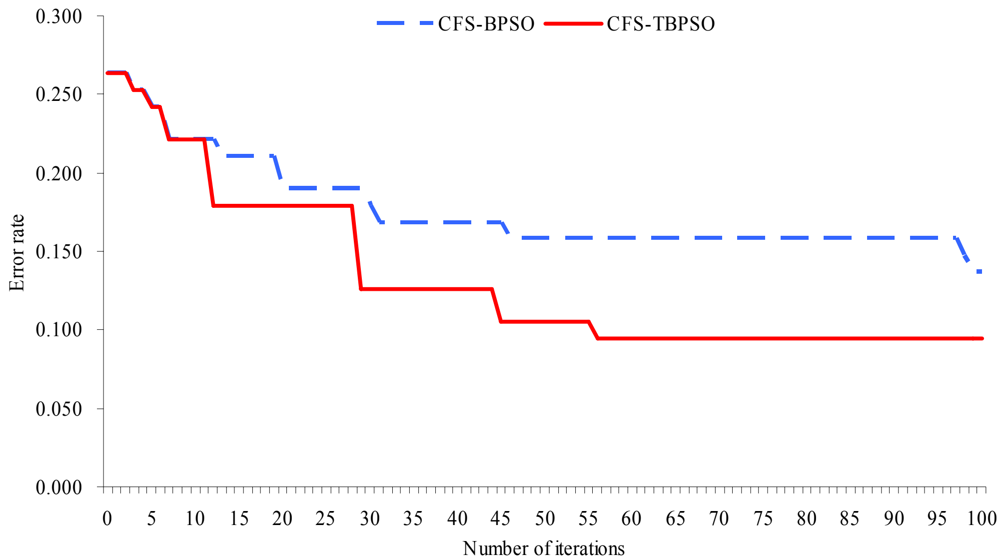


Breast 2 class (a)

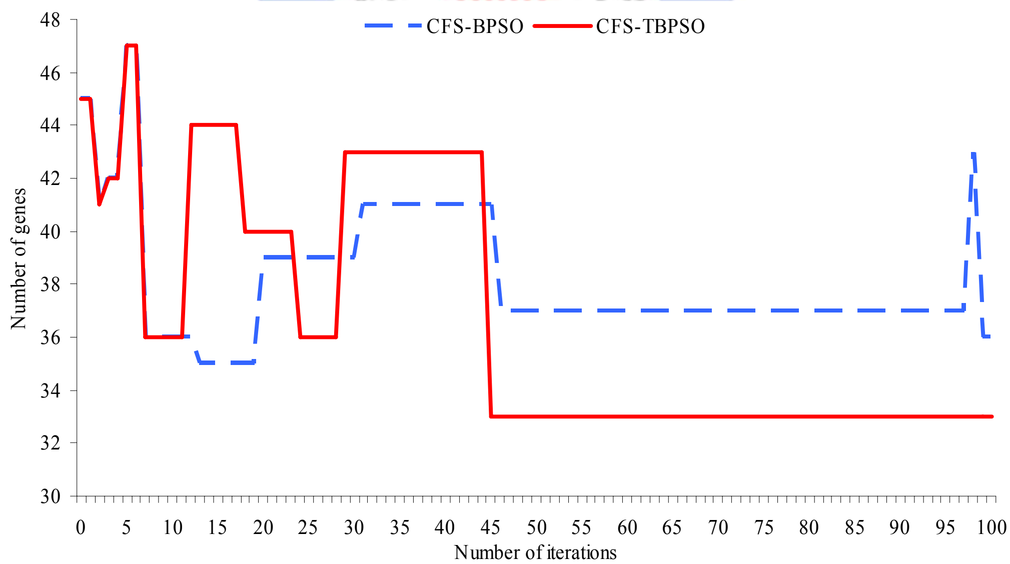


Breast 2 class (b)

Figure 12 Number of iterations vs. Classification error rate (a) and features (b) in Breast 2 class of microarray data

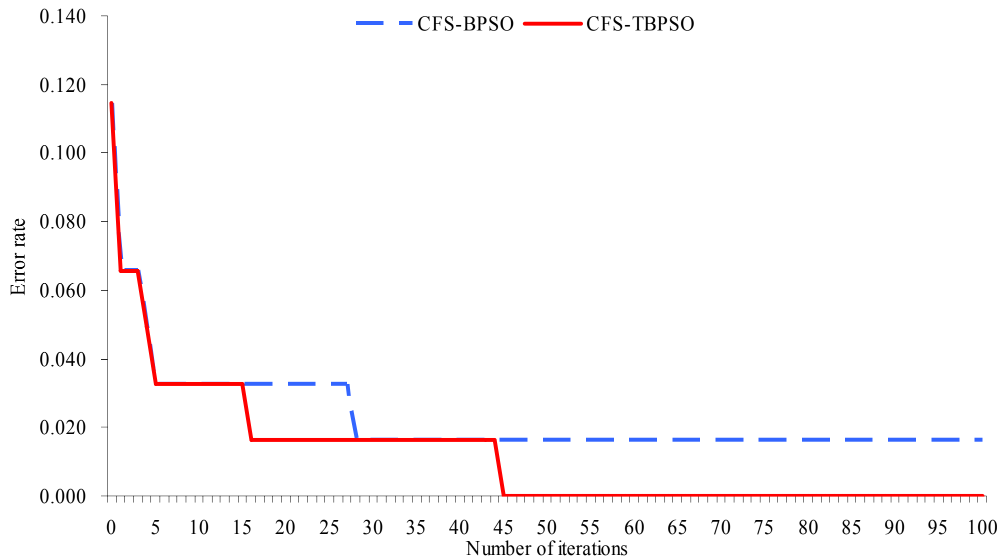


Breast 3 class (a)

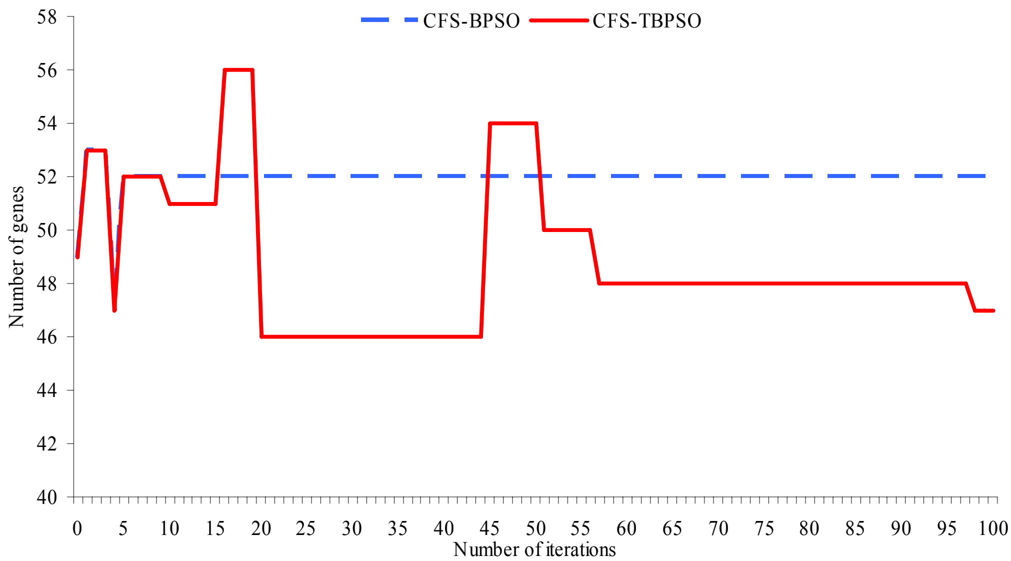


Breast 3 class (b)

Figure 13 Number of iterations vs. Classification error rate (a) and features (b) in Breast 3 class of microarray data

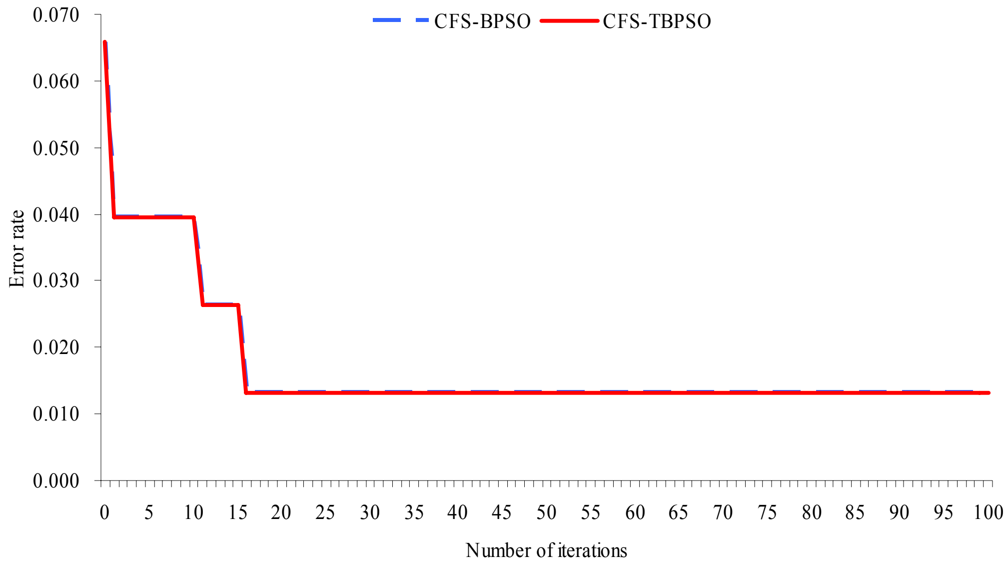


NCI 60 (a)

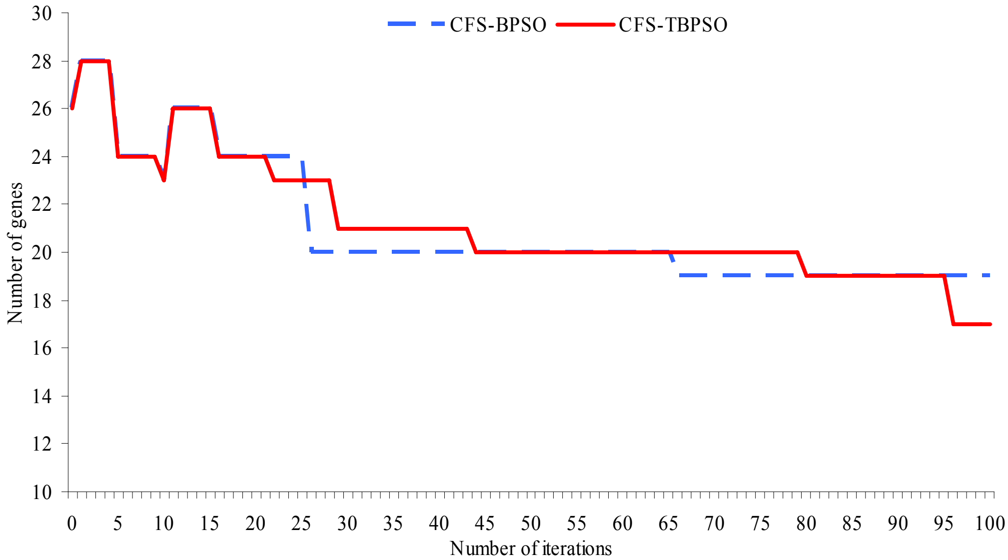


NCI 60 (b)

Figure 14 Number of iterations vs. Classification error rate (a) and features (b) in NCI 60 of microarray data

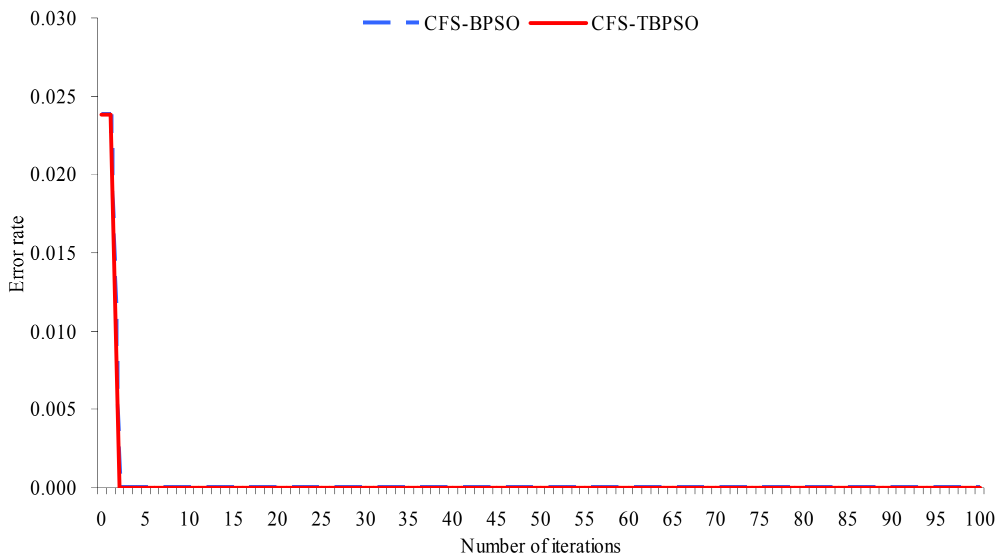


Adenocarcinoma (a)

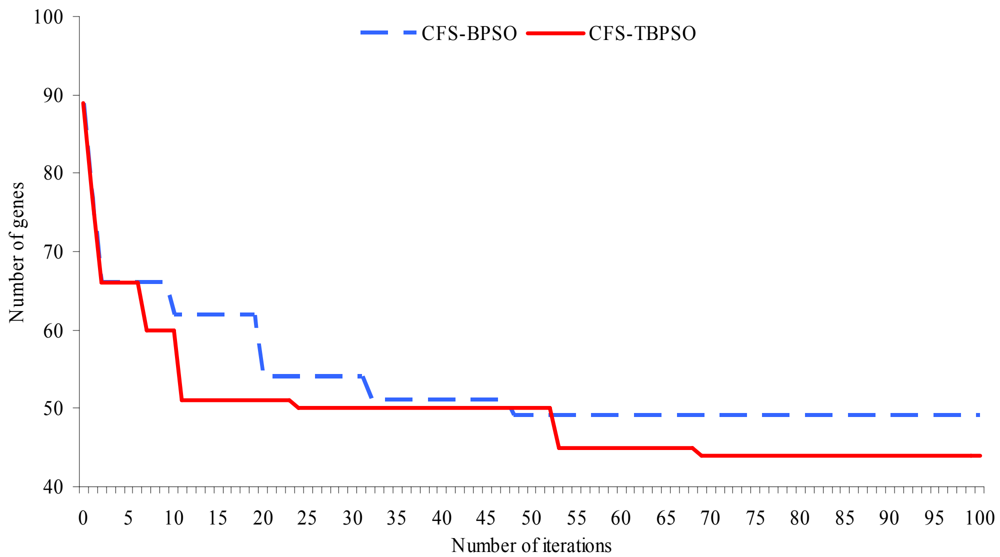


Adenocarcinoma (b)

Figure 15 Number of iterations vs. Classification error rate (a) and features (b) in Adenocarcinoma of microarray data

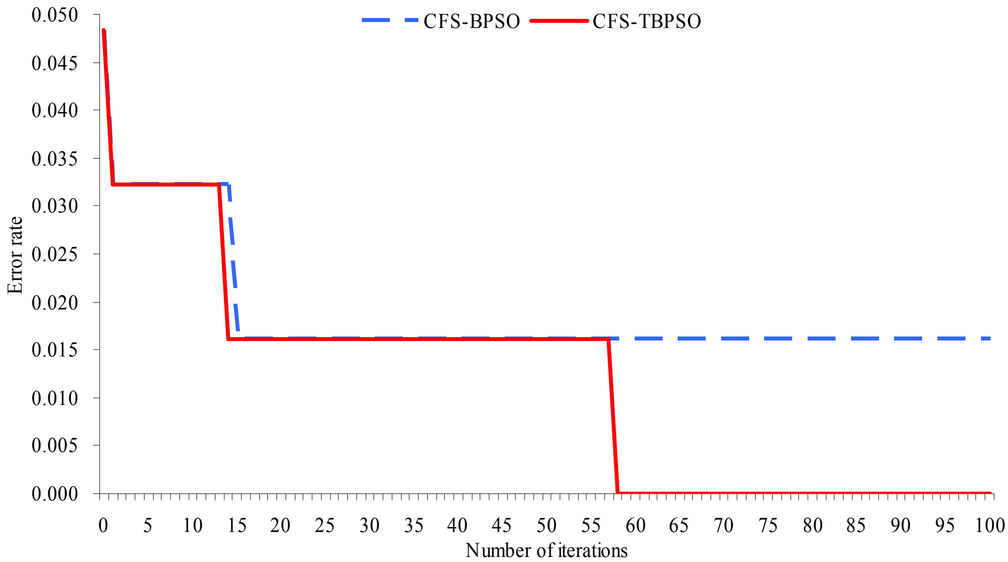


Brain (a)

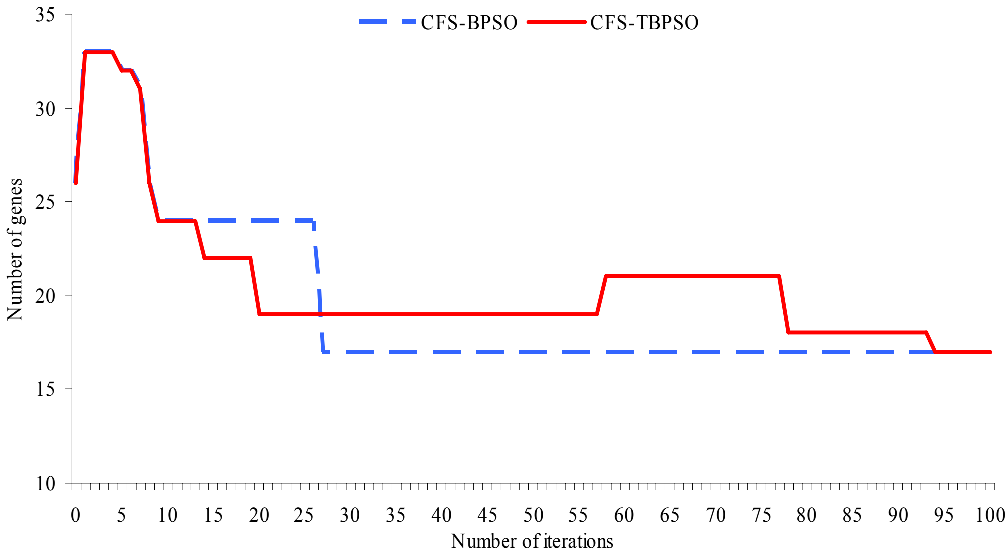


Brain (b)

Figure 16 Number of iterations vs. Classification error rate (a) and features (b) in Brain of microarray data

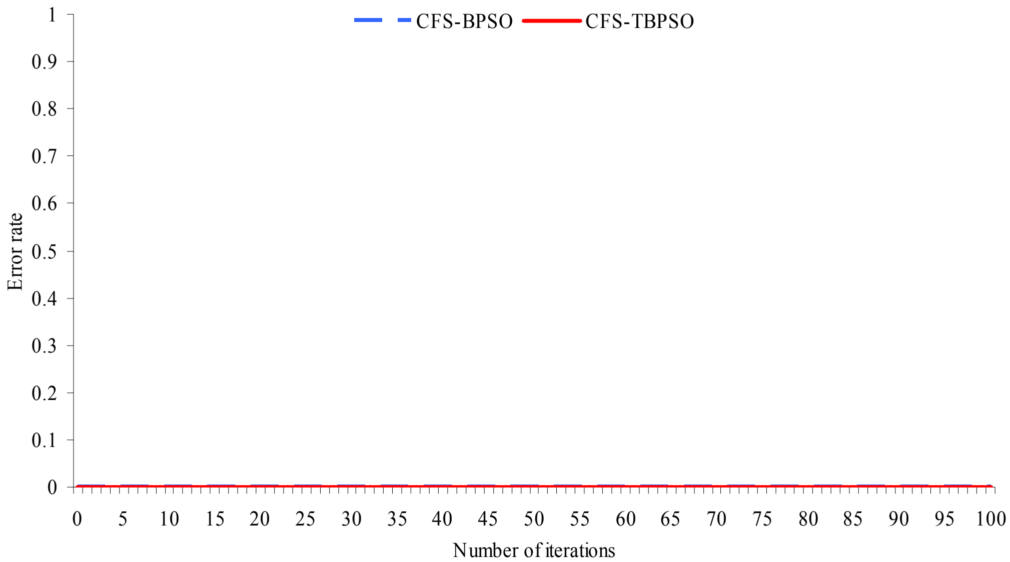


Colon (a)

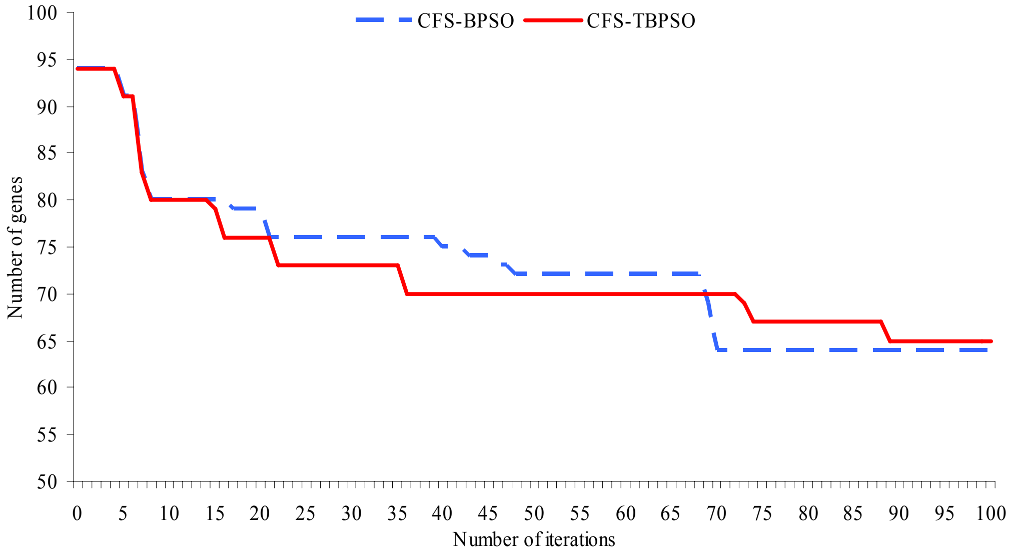


Colon (b)

Figure 17 Number of iterations vs. Classification error rate (a) and features (b) in Colon of microarray data

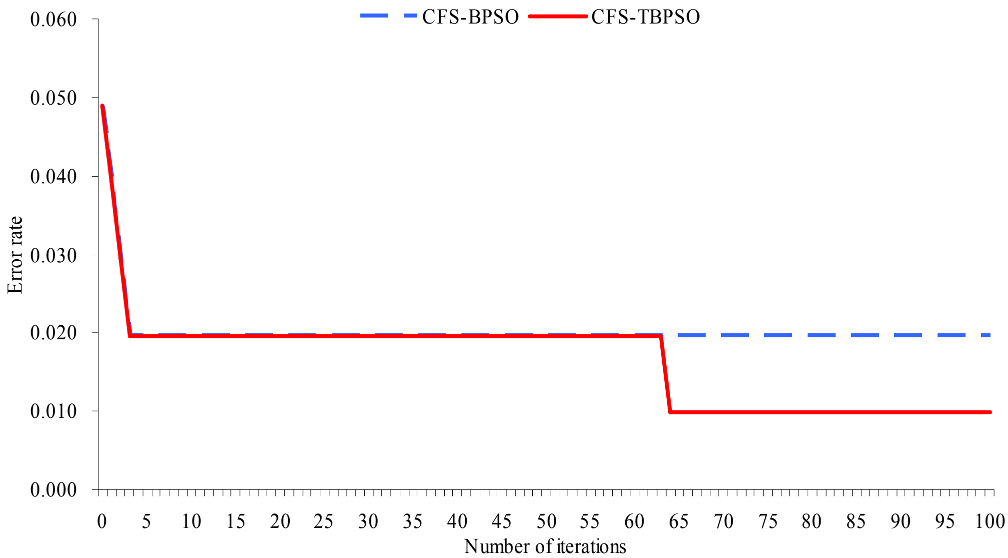


Lymphoma (a)

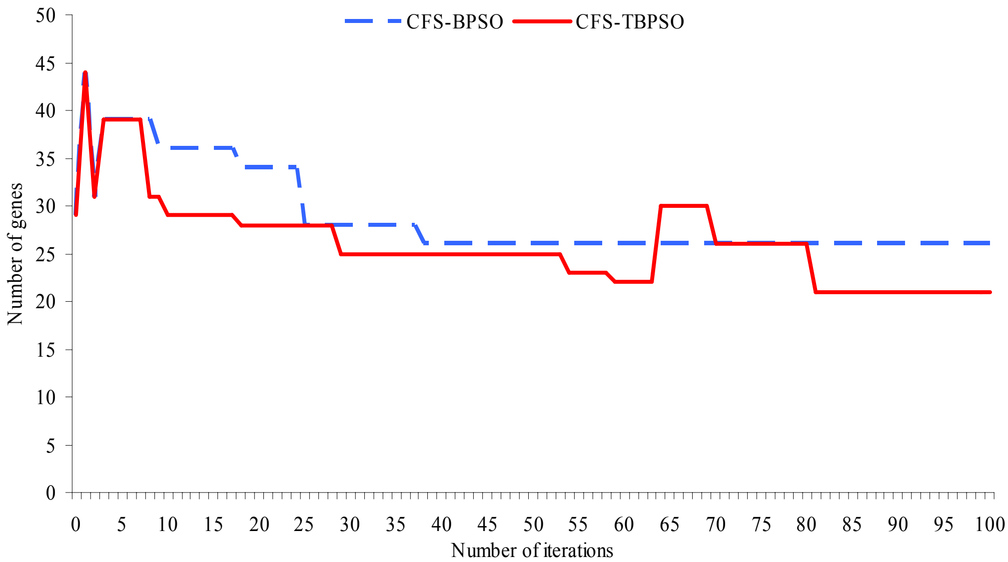


Lymphoma (b)

Figure 18 Number of iterations vs. Classification error rate (a) and features (b) in Lymphoma of microarray data

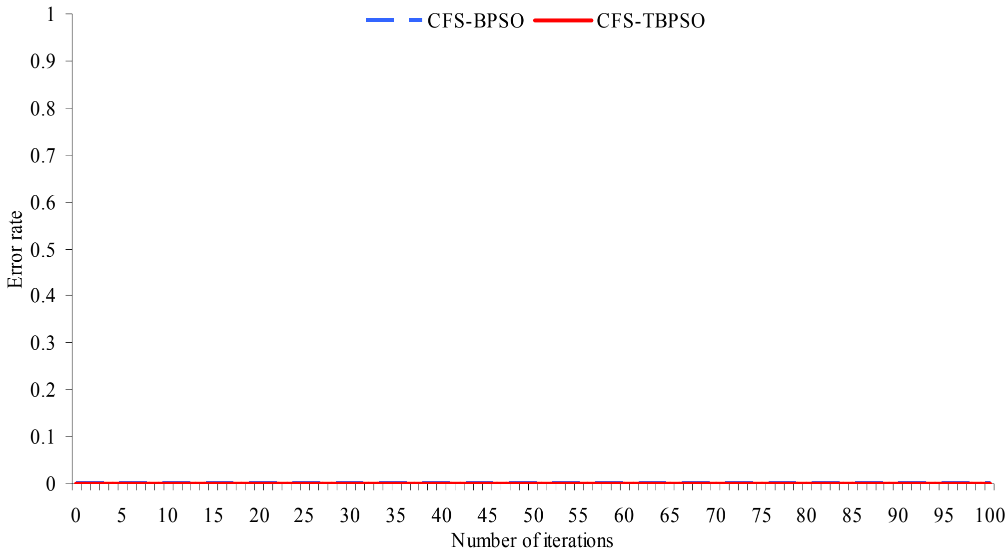


Prostate (a)

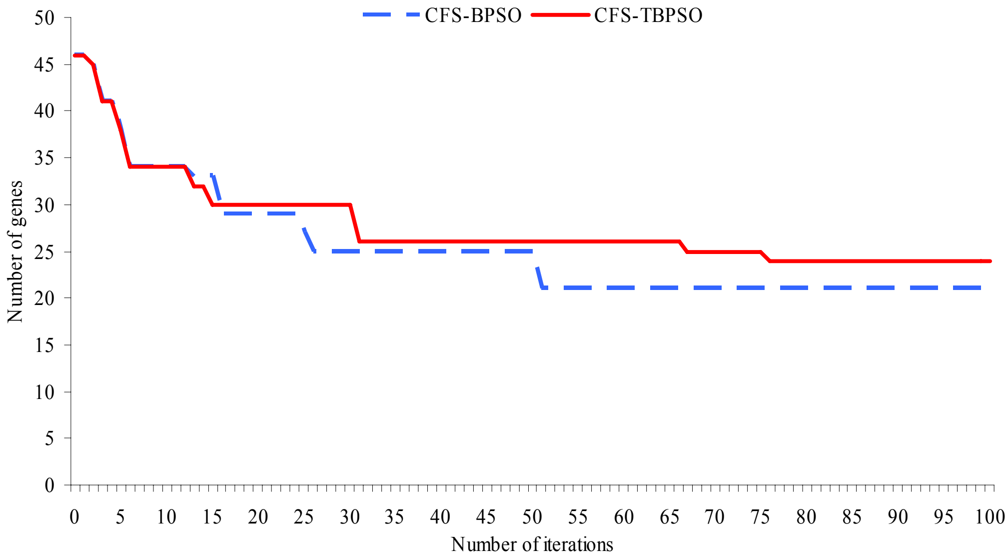


Prostate (b)

Figure 19 Number of iterations vs. Classification error rate (a) and features (b) in Prostate of microarray data

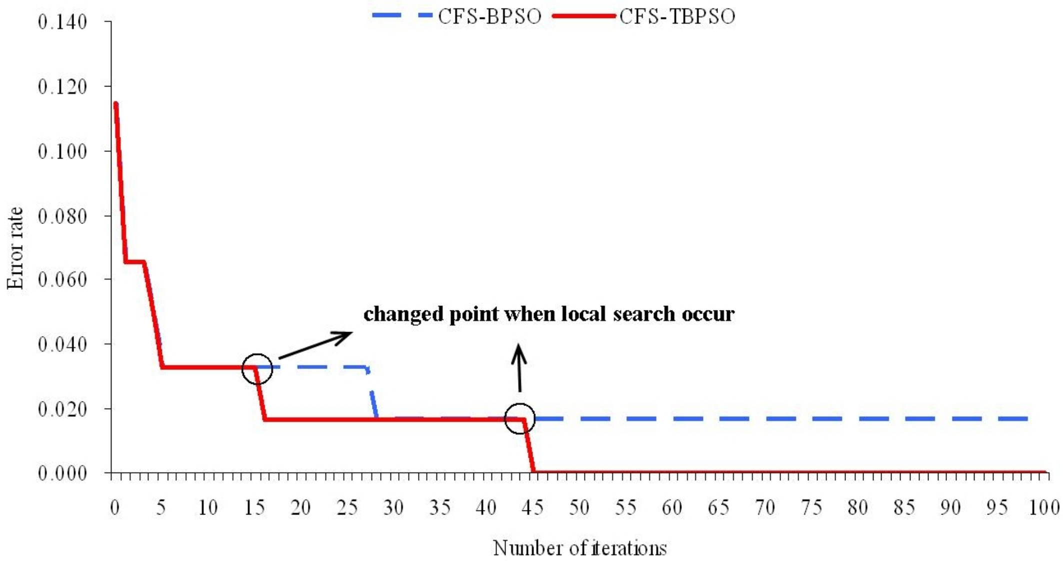


Srbct (a)

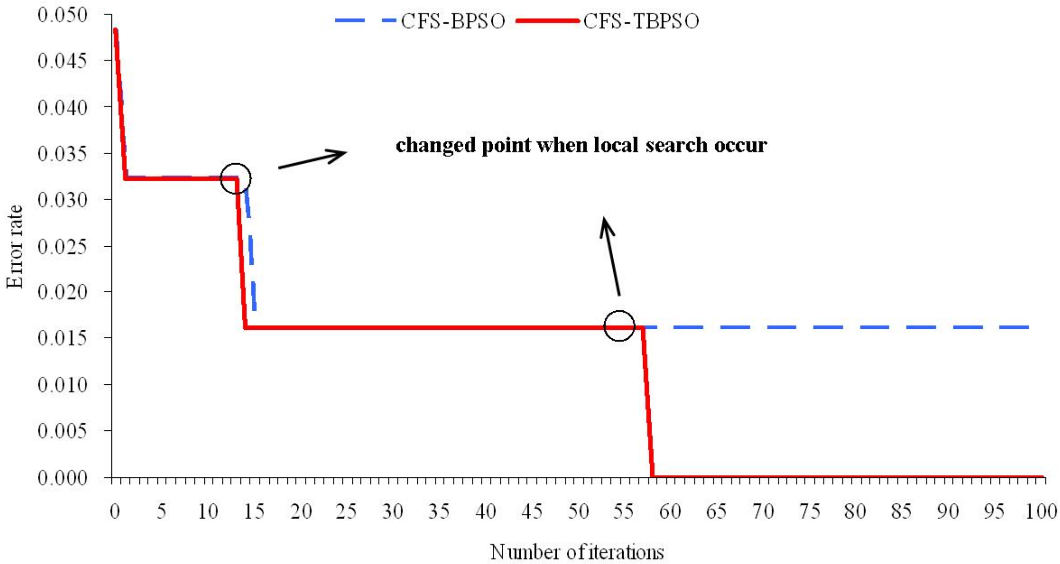


Srbct (b)

Figure 20 Number of iterations vs. Classification error rate (a) and features (b) in Srbct of microarray data

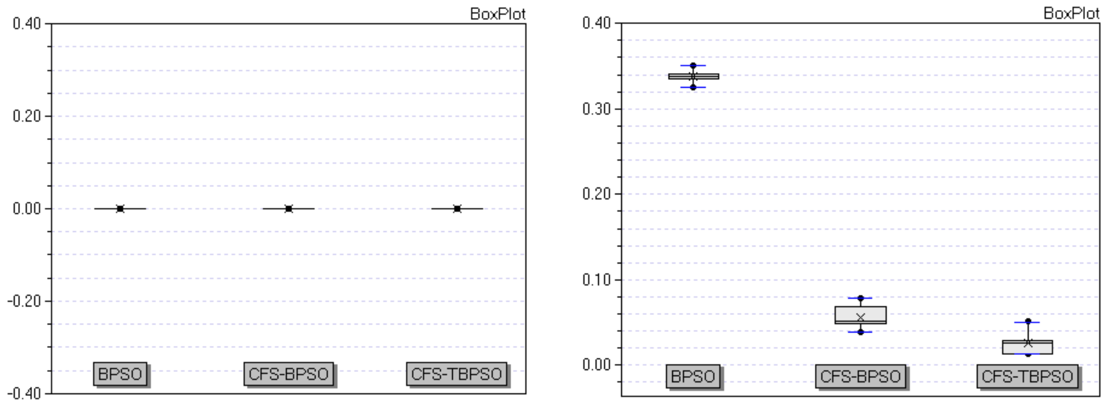


(a) NCI 60



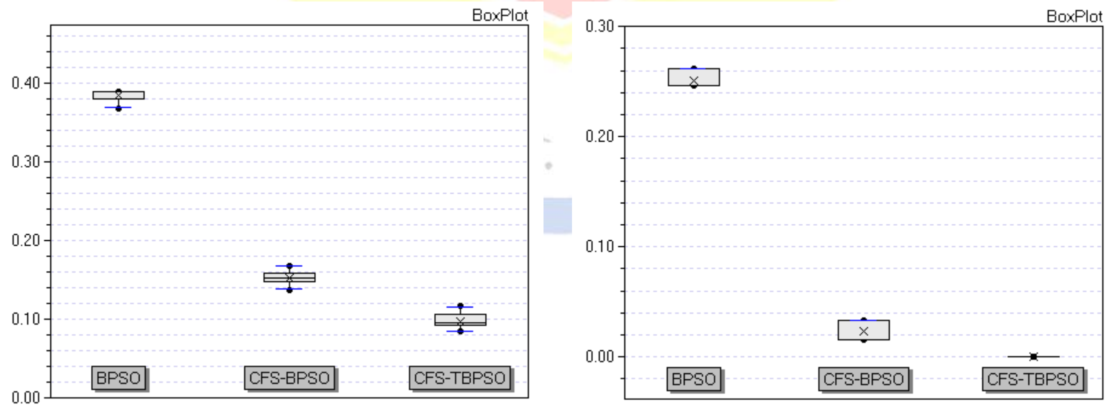
(b) Colon

Figure 21 Taguchi method effect in microarray data



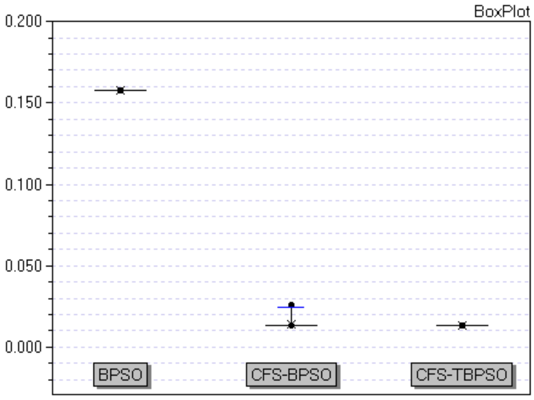
Leukemia　　Breast 2 class

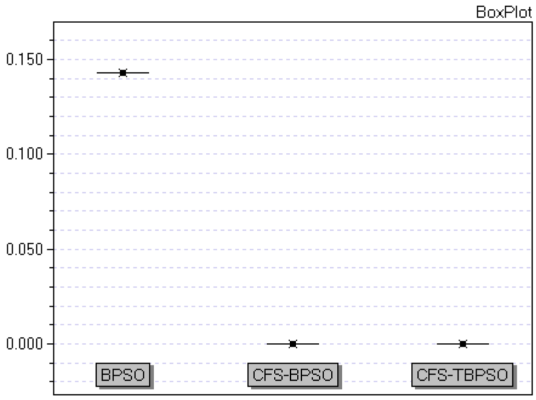
Figure 22 Statistical performances of the different data sets in BPSO, CFS-BPSO and CFS-TBPSO for ten independent runs in Leukemia and Breast 2 class



Breast 3 class　　NCI 60

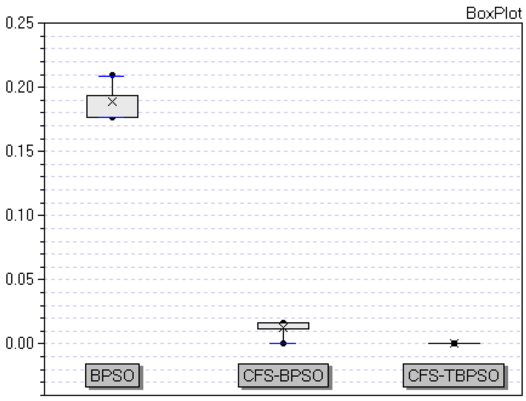
Figure 23 Statistical performances of the different data sets in BPSO, CFS-BPSO and CFS-TBPSO for ten independent runs in Breast 3 class and NCI 60

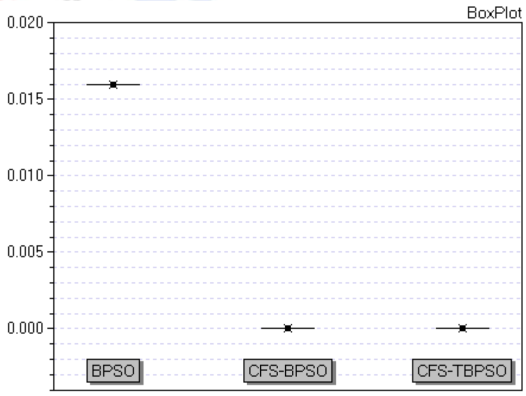




(e) Adenocarcinoma (f) Brain

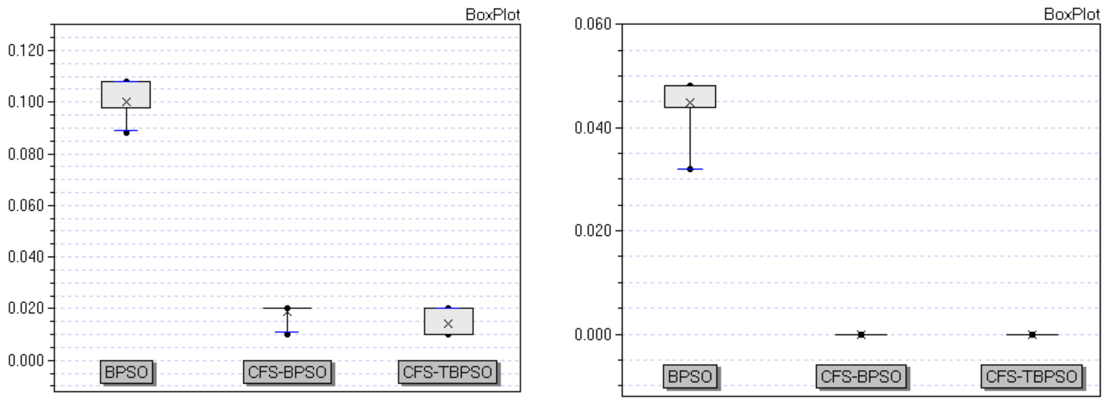
Figure 24 Statistical performances of the different data sets in BPSO, CFS-BPSO and CFS-TBPSO for ten independent runs in Adenocarcinoma and Brain





(g) Colon (h) Lymphoma

Figure 25 Statistical performances of the different data sets in BPSO, CFS-BPSO and CFS-TBPSO for ten independent runs Colon and Lymphoma



(i) Prostate (j) Srbct

Figure 26 Statistical performances of the different data sets in BPSO, CFS-BPSO and CFS-TBPSO for ten independent runs in Prostate and Srbct

4.3.1.1 Statistical analysis

To investigate the statistical robustness of CFS-TBPSO, its classification accuracies were compared to the average classification accuracy of classification methods in two statistical tests, the Friedman test and the multiple comparison approach [67]. The Friedman test was used to test whether the accuracies of the different classification methods were equal. The multiple comparison approach was used to determine which method had significantly different accuracies if the Friedman test was rejected.

Friedman Test

The Friedman test is a nonparametric counterpart of the parametric two-way analysis of variance test and was used to compare the classification accuracy of the classification methods when the distribution of the underlying population was not specified. The hypothesis being tested was that all the methods had equal classification accuracy, and the alternative hypothesis was that all methods did not have equal classification accuracy. Let R_{ij} be the rank (from 1 to k) assigned to method j on problem i. It will equal 1 if it is the lowest value among the methods. In the case of ties, average ranks are used. The test statistic is defined by the following equations:

where $$T_f = \frac{(n-1)\ \{B_f - \frac{nk(k+1)^2}{4}\}}{A_f - B_f} \tag{13}$$

$$R_j = \sum_{i=1}^{n} R_{ij}\ , \text{ for } \ j = 1,2,\ldots,k \tag{14}$$

$$A_f = \sum_{i=1}^{n}\sum_{j=1}^{k} R_{ij}^2 \tag{15}$$

$$B_f = \tfrac{1}{n}\sum_{j=1}^{k} R_j^2 \tag{16}$$

The null hypothesis is rejected at the α significance level if the value of the test statistic exceeds the 1-α quantile of the F-distribution with k-1 and (n-1)(k-1) degrees of freedom.

Multiple Comparison Approach

The multiple comparison approach was used to determine which method had significantly different classification accuracy. Methods i and j are considered different if the following inequality is satisfied:

$$\left|R_j - R_i\right| > t(\alpha/2)\sqrt{2n(A_f - B_f)/(n-1)(k-1)} \tag{17}$$

where R_i, R_j, A_f, and B_f are given previously, and $t(\alpha/2)$ is a critical value on the t-table using (n-1)(k-1) degrees of freedom ($\alpha/2 = \text{P}(\text{t} > t(\alpha/2))$).

After the Friedman test, the calculated value of T_f = 19.65 is greater than the critical value of $F_{0.05}$ (7, 63) = 2.507. We rejected the null hypothesis that all the methods had the same classification accuracy at a significance level of α = 0.05. After multiple comparisons were executed, the classification accuracies of the nine methods were ordered in an array, and a rank was assigned to each corresponding value according to its order. The rank sums of CFS-TBPSO, CFS-BPSO, CFS, SC.s, NN.vs, BPSO, RF (s.e.=1), and RF (s.e.=0) were 76.5, 70.5, 56.5, 39.0, 32.5, 31.5, 27.0, and 26.5, respectively; if the rank sums of any two methods are more than 12.79 units apart (with α = 0.05), they might be regarded as having unequal accuracy of prediction. Therefore, it can be concluded that the CFS-TBPSO, CFS-BPSO method is statistically superior to CFS, SC.s, NN.vs, BPSO, RF (s.e.=1), and RF (s.e.=0) methods for the data sets tested.

4.3.2 Experiment of SNP data

4.3.2.1 Accuracy estimation

This chapter presents the common approach to appraise as medical diagnostic, namely, positive hit rate (i.e. Sensitivity), negative hit rate (i.e. Specificity) and accuracy rate. The accuracy using the binary class dataset can be demonstrated. As Table 11 shown, "+" represents some cases with the 'positive' class (with disease) be classified as positive correctly (i.e. True Positive, TP). Contrariously, some case with the 'positive' class be classified as negative (False Negative, FN). In contrast, if correctly predict some cases with the ‘negative class as negative (True Negative, TN). Contrariously, some case with the negative class be classified as positive (False Positive, FP). Sensitivity is the proportion of cases with positive class that are classified as positive. On the other hand, the specificity is the proportion of cases with negative class that are classified as negative. The sensitivity and specificity are computed as formula (18) and (19). The accuracy rate is calculated as formula (20).

$$Sensitivity = \frac{TP}{TP + FN} \tag{18}$$

$$Specificity = \frac{TN}{TN + FP} \tag{19}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{20}$$

Table 11 A prediction contingency table

		Class (or disease)	
		+	-
Predicted (or test)	+	True Positive (*TP*)	False Positive (*FP*)
	-	False Negative (*FN*)	True Negative (*TN*)

4.3.2.2 Results

In all the experiments including (non-feature selection) K nearest-neighbor, C4.5, random forest, support vector machine, naïve bayes and (feature selection) correlation-based feature selection with 5 nearest-neighbor classifier (CFS-5NN), and genetic algorithm with 5 nearest-neighbor classifier (GA-5NN) were constructed as implemented in Weka [38]. Those experiment results were estimated using the holdout cross validation in ten runs. To compare the accuracy of the proposed BPSO-KNN and TBPSO-KNN approach with these methods, the results were shown as Table 12 and Table 13. In Table 12 and Table 13, the average classification accuracy is 74.39±3.21 for the TBPSO-KNN that is better than others. There is an interested result in Table 13, after feature selection (CFS and GA) instead of increasing, the accuracy had declined. Table 13 shows the number of feature selected, the TBPSO-KNN selected subset is 8.5±1.5 highger than CFS-5NN, GA-5NN. It might indicate CFS-5NN and GA-5NN to ignore some important attribute for classifier. The results show the K of Nearest-Neighbor, when K equals 5 the accuracy better than K equals 1 and 3. Therefore, we used 5NN classfier to estimate for CFS and GA.

Table 12 Classification results on non-feature selection approach

Classifier	Sensitivity			Specificity			Accuracy		
1NN	43.49	±	9.49	74.34	±	9.49	62.55	±	4.40
3NN	45.33	±	10.72	81.72	±	10.72	67.86	±	3.68
5NN	47.46	±	5.82	**85.67**	±	**5.82**	71.12	±	2.79
C4.5	60.28	±	6.27	74.81	±	6.27	69.08	±	4.23
RF	53.11	±	6.11	79.43	±	6.11	69.29	±	2.78
SVM	55.95	±	6.30	82.29	±	6.30	72.14	±	3.77
NB	56.02	±	5.82	83.16	±	5.82	72.86	±	3.62
TBPSO-KNN	**60.94**	±	**5.80**	82.30	±	3.57	**74.39**	±	**3.21**

All sensitivities, specificities and accuracies are estimated using the holdout cross validation (i.e. train:test = 2:1). The results were shown as mean±standard deviation (ten runs). Boldfaced values highlight the best results. Each classifiers represent (1) NN: Nearest-Neighbor; (2) SVM: Support Vector Machine; (3) RF: Random Forest; (4) NB: Naïve Bayes; (5) BPSO-KNN: Our propose approach.

Table 13 Classification results on feature selection approach

Classifier	Feature selected	Sensitivity	Specificity	Accuracy
CFS-5NN	2.2 ± 0.40	54.34 ± 10.08	79.15 ± 6.66	69.69 ± 3.57
GA-5NN	5.9 ± 1.14	55.06 ± 5.13	79.55 ± 7.09	70.10 ± 5.04
BPSO-KNN	6.7 ± 1.55	**61.53** ± **4.68**	81.27 ± 5.16	73.78 ± 3.10
TBPSO-KNN	8.5 ± 1.5	60.94 ± 5.80	**82.30** ± **3.57**	**74.39** ± **3.21**

All sensitivities, specificities and accuracies are estimated using the holdout CV (i.e. train:test = 2:1), and The GA and BPSO as wrapper approach during traing step were used 10-fold CV. The results were shown as mean±standard deviation (ten runs). Boldfaced values highlight the best results. Each method represents (1) CFS-5NN: Correlation-based Feature Selection with 5 Nearest-Neighbor classifier; (2) GA-5NN: Genetic Algorithm with 5 Nearest-Neighbor classifier; (3) BPSO-KNN: Our propose approach.

4.4 Discussion

Many classifiers (e.g., K-NN, linear and quadratic discriminant analysis, support vector machine etc.) show good performance on microarray data. Each approach has its strong and weak points, so no single one can be considered ideal. As a classifier, K-NN performs well for cancer classification, compared to the more sophisticated classifiers. It is an easily implemented method that has a simple parameter (the number of nearest neighbors) to be pre-defined, given that the distance metric is Euclidean [68].

During the last decade, the advent of microarray data sets stimulated a new line of research in bioinformatics. To deal the challenges microarray data pose for computational techniques, feasible feature reduction techniques are needed. A general overall feature selection approach can be found in [40]. Feature selection methods using a wrapper approach are very much dependent on the classifier or the pattern recognition approach used to assign the feature (gene) subset. On the other hand, filter approaches take only intrinsic features of the data into account. Finally, an embedded approaches similar to a wrapper approach has the advantage that it includes interactions with the classification model, while at the same time it is far less computationally intensive than a wrapper method [40]. However, Wang *et al*., [69] indicate that filter approaches can select more relevant feature subsets faster than wrapper approaches. On the other hand, wrapper approaches tend to obtain better classification accuracies in general. Inza, *et al*. [70] and Xiong *et al*. [71] used a wrapper approach to implement feature selection, and selected better feature subsets to boost classification accuracy. Nevertheless, optimal solutions are difficult to find due to the large size of search space if only a wrapper approach is used. In this section, we combined a filter and wrapper approach instead of other methods. CFS is a filter method that searches the entire feature space efficiently, and TBPSO is a wrapper method that uses an induction algorithm to evaluate the feature subsets directly. As stated above, wrapper methods generally outperform filter methods in terms of prediction accuracy rate. Since the individual advantages of wrapper and filter methods complement each other well [72]. We used a hybrid two-stage strategy to increase the classification accuracy. The Taguchi method implemented under the BPSO procedure is responsible for the local search. Taguchi method is a robust design approach, which used many ideas from statistical experimental design to improve optimize in products, processes and equipment [49]. The Taguchi principle is used to improve the quality of a product by minimizing the effect of the causes of variation without eliminating these causes [49]. The two-level orthogonal array and the SNR of the Taguchi method are used for exploitation. The optimum particles can easily be found by using both experimental runs and SNRs instead of executing combinations all of factor levels.

Consequently, a superior candidate feature subset with high classification performance for the classification task at hand, can be obtained in a subsequent iteration. In illustrative example is given in the example section. Since feature subsets b_1 and b_2 have *seven* different features, 2^7=128 possible experimental trials have to be considered in a full factorial experimental design. OAs are used to decrease the number of experimental trials associated with these *seven* different features to eight (see Table 5). Prior to the classification process, feature subset evaluation efforts can thus be significantly reduced based on the two-dimensional, fractional factorial experimental design matrix. Features important and relevant for pattern classification can easily be identified. In this chapter, in order to avoid overfitting problem, the microarray data characteristically have a high dimension and small sample size, which is subsequently reduced by a filter feature selection method. After feature reduction, the LOOCV technique enhances the training data for classification in a wrapper-based feature selection method.

In the K-NN parameter K, the best choice of the number of neighbors depends upon the data. Generally, larger values of K reduce the effect of noise on the classification, but make boundaries between classes less distinct. Also, Ghosh [75] indicated the suitable K depends on the specific data set and is to be computed using the available training sample observations. On the other hand, since the time complexity of K-NN is

O(Kn log n), the parameter K directly influences the performance efficiency. However, this thesis utilizes the BPSO to optimize parameter K that enhances classification and excludes manual setting or trial and error.

Overfitting appears when computationally intensive search algorithms are used. Estimates may be overfitted and yield biased predictions under these circumstances [41]. If the training data lies too closely together, the classifier predictions are of poor quality. This occurs when there is insufficient data to train the classifier and the data does not fully cover the concept being learned. This problem is common in many real world samples where the available data may be rather noisy [42]. In order to avoid overfitting, some additional techniques have been discussed, such as cross-validation, regularization, and early termination or resampling [43, 44]. However the best way to avoid overfitting is to use an abundant amount of training data. The Figure 27 was shown the BPSO-KNN during search period, the training accuracy and testing accuracy were calculated each iteration. The result shows we can avoid the overfitting problem.

Table 13 shows that the number of feature subset selected of three feature selection method in 10 runs. We can see clearly, the mainly frequency distribution on age, menopausal and TNFα-857 (SNP_1). We also implement a filter approach – information gain to caluculate each feature score. In the 10 runs, there are only three score of feature highger than 0 which also age, menopausal and TNFα-857 (SNP_1). However, our propsed approach not only these feature can seleted high frequency, but also selected other feature and improved accuracy.

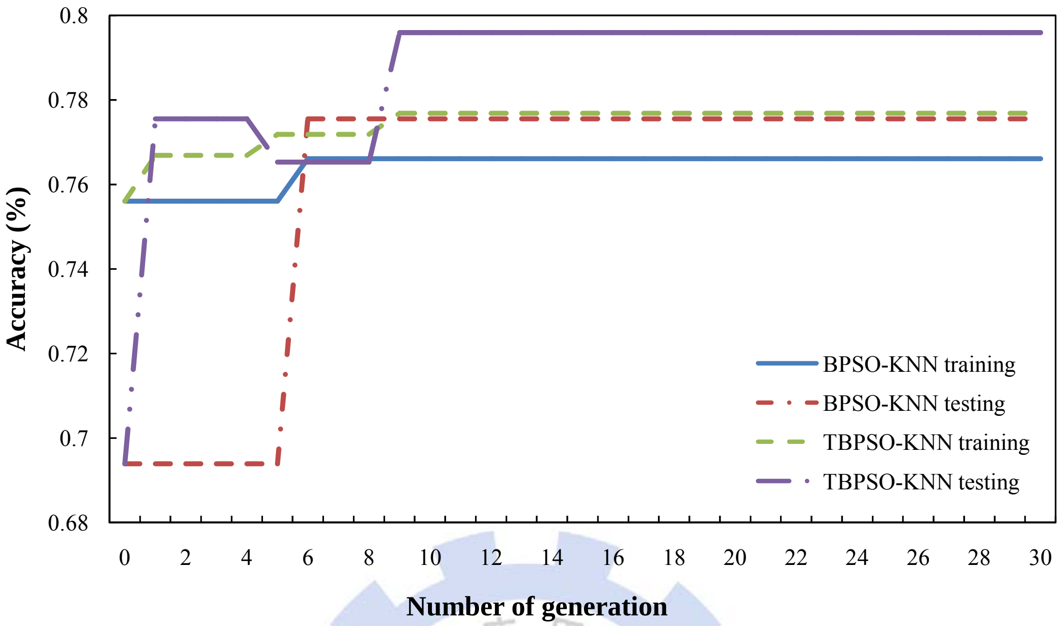


Figure 27 The classification accuracies estimated during training and the accuracies for testing

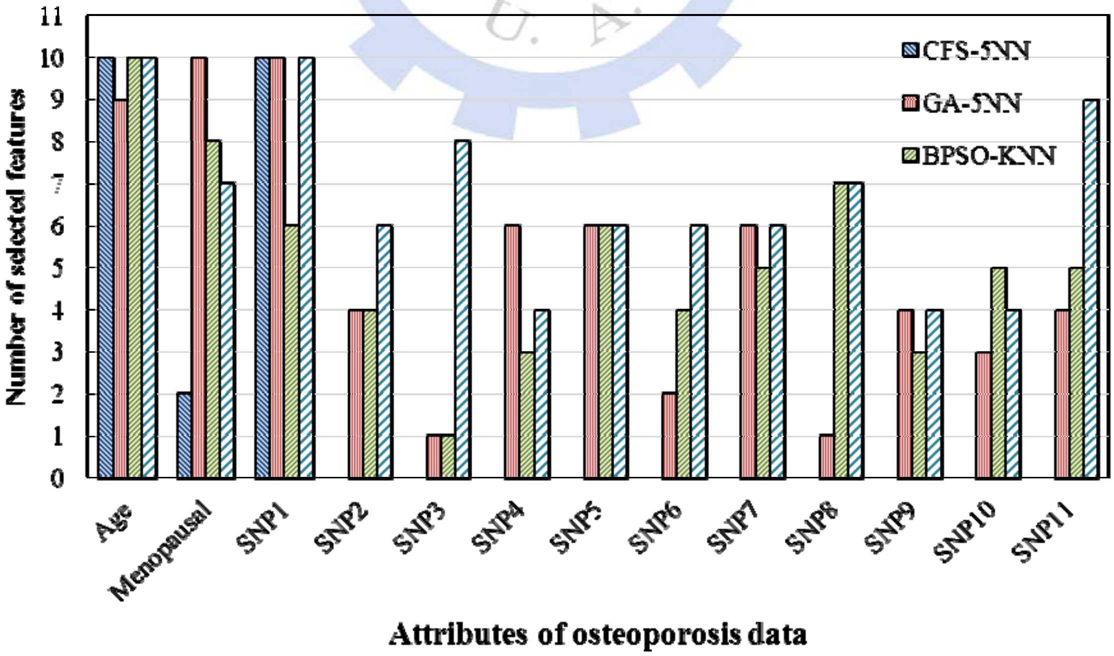


Figure 28 Number of feature subset selected in 10 runs

5. CONCLUSION AND FUTURE WORKS

In this thesis, the disease prediction using machine learning is proposed, binary particle swarm optimization with K nearest-neighbor served as classifier for microarray and SNP profiles. In microarray application, this thesis is compared with the proposed approache that is against random forest, shrunken centroids and nearest neighbor methods with variable selection that have been used for classification and feature selection of large-dimensional microarray data sets. In addition, within the SNP application, we proposed a method to against K nearest-neighbor, C4.5, random forest, support vector machine, naïve bayes, correlation-based feature selection with 5 nearest-neighbor and genetic algorithm with 5 nearest-neighbor were constructed from Weka. The experimental results for both of the classification accuracy and the selected numbers of features show that the proposed method has the most important features and the highest accuracy.It represents a superior role of feature selection (gene/SNPs selection) and classifier. The method can conceivably use in other research projects that implementing the feature selection. The outcome is successfully available to provide the medical disease prediction or feature selection of microarray/SNPs in the near future.

This thesis proposed a binary particle swarm optimization for the feature selection and parameter optimization. The binary particle swarm optimization is a population based stochastic optimization technique. However, the generating random sequences with a long period and good uniformity are very important for a heuristic algorithm. Since chaos is non-repetitive, a heuristic algorithm can be embedded. Chaos can be described as the complex behavior of a nonlinear deterministic system that has ergodic and stochastic properties [76]. Therefore, the stochastic optimization algorithm can be improved by using the chaotic theory. The classification problem, the K nearest-neighbor served as classifier. Recently, there are many supiror classifiers proposed such as: the support vector machine [77] and nearest shrunken centroids [78]. In order to enrich the classification accuracy, those classifiers can be used according to their data characteristics. Taguchi method is a statistical method devised for robust design of complex systems. It has been successfully applied in many manufacturing problems. The use of Taguchi method for a local search method in algorithm is illustrated in this thesis. It is use for parameter optimization [79], classifier optimization [26] or algorithm improvement design [80].

6. REFERENCES

[1] J. Watson, "The human genome project: past, present, and future," *Science,* vol. 248, pp. 44-49, 1990.

[2] F. S. Collins, A. Patrinos, E. Jordan, A. Chakravarti, R. Gesteland, L. Walters, t. m. o. t. DOE, and NIH planning groups, "New Goals for the U.S. Human Genome Project: 1998-2003," *Science,* vol. 282, pp. 682-689, 1998.

[3] R. Jiang, W. Tang, X. Wu, and W. Fu, "A random forest approach to the detection of epistatic interactions in case-control studies," *BMC bioinformatics,* vol. 10, p. S65, 2009.

[4] A. V. Kulkarni, N. S. Williams, Y. Lian, J. D. Wren, D. Mittelman, A. Pertsemlidis, and H. R. Garner, "ARROGANT: an application to manipulate large gene collections," *Bioinformatics,* vol. 18, pp. 1410-1417, 2002.

[5] R. Diaz-Uriarte and S. Alvarez de Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics,* vol. 7, p. 3, 2006.

[6] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," *Bioinformatics,* vol. 21, pp. 631-643, 2005.

[7] P. Yang, B. Zhou, Z. Zhang, and A. Zomaya, "A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data," *BMC bioinformatics,* vol. 11, p. S5, 2010.

[8] S. Buch, C. Schafmayer, H. Völzke, C. Becker, A. Franke, H. v. Eller-Eberstein,

C. Kluck, I. Bässmann, M. Brosch, F. Lammert, J. F. Miquel, F. Nervi, M. Wittig, D. Rosskopf, B. Timm, C. Höll, M. Seeger, A. ElSharawy, T. Lu, J. Egberts, F. Fändrich, U. R. Fölsch, M. Krawczak, S. Schreiber, P. Nürnberg, J. Tepel, and J. Hampe, "A genome-wide association scan identifies the hepatic cholesterol transporter ABCG8 as a susceptibility factor for human gallstone disease," *Nature genetics,* vol. 39, pp. 995-999, 2007.

[9] B. W. Zanke, C. M. Greenwood, J. Rangrej, R. Kustra, A. Tenesa, S. M. Farrington, J. Prendergast, S. Olschwang, T. Chiang, E. Crowdy, V. Ferretti, P. Laflamme, S. Sundararajan, S. Roumy, J.-F. Olivier, F. Robidoux, R. Sladek, A. Montpetit, P. Campbell, S. Bezieau, A. M. O'Shea, G. Zogopoulos, M. Cotterchio, P. Newcomb, J. McLaughlin, B. Younghusband, R. Green, J. Green, M. E. M. Porteous, H. Campbell, H. Blanche, M. Sahbatou, E. Tubacher, C. Bonaiti-Pellié, B. Buecher, E. Riboli, S. Kury, S. J. Chanock, J. Potter, G. Thomas, S. Gallinger, T. J. Hudson, and M. G. Dunlop, "Genome-wide association scan identifies a colorectal cancer susceptibility locus on chromosome 8q24," *Nature genetics,* vol. 39, pp. 989-994, 2007.

[10] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters,* vol. 28, pp. 459-471, 2007.

[11] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[12] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, pp. 273-324, 1997.

[13] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Boston, MA: Kluwer Academic Publishers, 1998.

[14] X. Liu, A. Krishnan, and A. Mondry, "An Entropy-based gene selection method for cancer classification using microarray data," *BMC Bioinformatics,* vol. 6, p. 76, 2005.

[15] H. Kodaz, S. Ozsen, A. Arslan, and S. Gunes, "Medical application of information gain based artificial immune recognition system (AIRS): Diagnosis of thyroid disease," *Expert Systems with Applications,* vol. 36, pp. 3086-3092, 2009.

[16] S. Verron, T. Tiplica, and A. Kobi, "Fault detection and identification with a new feature selection based on mutual information," *Journal of Process Control,* vol. 18, pp. 479-490, 2008.

[17] M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning," PhD thesis, Department of Computer Science, University of Waikato, 1999.

[18] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 26, pp. 1424-1437, 2004.

[19] M. A. Tahir, A. Bouridane, and F. Kurugollu, "Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier," *Pattern Recognition Letters,* vol. 28, pp. 438-446, 2007.

[20] J. R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.

[21] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan

Kaufmann Publishers, 1993.

[22] L. Breiman, "Random Forests," *Machine learning,* vol. 45, pp. 5-32, 2001.

[23] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Information Processing Letters,* vol. 93, pp. 255-261, 2005.

[24] B. R. Secrest and G. B. Lamont, "Visualizing particle swarm optimization - Gaussian particle swarm optimization," in *IEEE Swarm Intelligence Symposium* Indianapolis, IN, 2003, pp. 198-204.

[25] T. C. Chang, F. C. Tsai, and J. H. Ke, "Data mining and Taguchi method combination applied to the selection of discharge factors and the best interactive factor combination under multiple quality properties," *The International Journal of Advanced Manufacturing Technology,* vol. 31, pp. 164-174, 2006.

[26] S. Y. Sohn and H. W. Shin, "Experimental study for the comparison of classifier combination methods," *Pattern Recognition,* vol. 40, pp. 33-40, 2007.

[27] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks,* vol. 13, pp. 143-159, 2002.

[28] W.-C. Chen, P.-H. Tai, M.-W. Wang, W.-J. Deng, and C.-T. Chen, "A neural network-based approach for dynamic quality prediction in a plastic injection molding process," *Expert Systems with Applications,* vol. 35, pp. 843-849, 2008.

[29] R. Herbrich, *Learning Kernel Classifiers*. Cambridge, MA: The MIT Press, 2002.

[30] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks,* vol. 5, pp. 537-550,

1994.

[31] S. Zhang, C. Zhang, and Q. Yang, "Data Preparation for Data Mining," *Applied Artificial Intelligence,* vol. 17, pp. 375-381, 2003.

[32] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," *Informatica,* vol. 31, pp. 249–268, 2007.

[33] S. Ali and K. A. Smith, "On learning algorithm selection for classification," *Applied Soft Computing,* vol. 6, pp. 119-138, 2006.

[34] H. Demuth and M. Beale, *Neural Network Toolbox For Use with Matlab* 4ed. Natick, MA: The MathWorks, Inc., 2002.

[35] E. F. Tjong Kim Sang, "Machine Learning of Phonotactics," in *Linguistics*. vol. PhD. Groningen, Netherlands: University of Groningen, 1996.

[36] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, 2 ed. New York. NY: Springer-Verlag, 2001.

[37] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *14th international joint conference on Artificial intelligence*. vol. 2 Montreal, Canada Morgan Kaufmann Publishers Inc., 1995, pp. 1137-1143.

[38] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2 ed. San Francisco, CA: Morgan Kaufmann, 2005.

[39] M. Dash and H. Liu, "Feature Selection for Classification " *Intelligent Data Analysis,* vol. 1, pp. 131-156, 1997.

[40] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, pp. 2507-2517, 2007.

[41] J. Reunanen, I. Guyon, and A. Elisseeff, "Overfitting in Making Comparisons Between Variable Selection Methods " *Journal of Machine Learning Research,* vol. 3, 2003.

[42] J. Loughrey and P. Cunningham, "Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets," in *Research and Development in Intelligent Systems XXI*, 2005, pp. 33-43.

[43] C. Schaffer, "Overfitting avoidance as bias," *Machine learning,* vol. 10, pp. 153-178, 1993.

[44] D. H. Wolpert, "On overfitting avoidance as bias," Technical Report SFI-TR-92-03-5001, Santa Fe Institute1993.

[45] E. Rich and K. Knight, *Artificial Intelligence*, 2nd ed. New York, NY: McGraw-Hill, 1991.

[46] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks* Perth, WA, 1995, pp. 1942-1948.

[47] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics* Orlando, FL, 1997, pp. 4104-4108.

[48] G. Taguchi, S. Chowdhury, and S. Taguchi, *Robust Engineering*. New Yourk, NY: McGraw-Hill, 2000.

[49] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, "Hybrid Taguchi-genetic algorithm for global numerical optimization," *IEEE Transactions on Evolutionary Computation,* vol. 8, pp. 365-377, 2004.

[50] Y. Wu and A. Wu, *Taguchi Methods for Robust Design*. New York, NY: ASME

Press 2000.

[51] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory,* vol. 13, pp. 21-27, 1967.

[52] E. Fix and J. Hodges, "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," Technical Report. USAF School of Aviation Medicine, Randolph Field, TX.1951.

[53] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[54] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science,* vol. 286, pp. 531 - 537, 1999.

[55] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature genetics,* vol. 415, pp. 530 - 536, 2002.

[56] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J. C. F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown, "Systematic variation in gene expression patterns in human cancer cell lines,"

Nature Genetics, vol. 24, pp. 227 - 235, 2000.

[57] S. Ramaswamy, K. N. Ross, E. S. Lander, and T. R. Golub, "A molecular signature of metastasis in primary solid tumors," *Nature Genetics,* vol. 33, pp. 49 - 54, 2003.

[58] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature,* vol. 415, pp. 436 - 442, 2002.

[59] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 96, pp. 6745 - 6750, 1999.

[60] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature,* vol. 403, pp. 503 - 511, 2000.

[61] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell,* vol. 1, pp. 203 - 209, 2002.

[62] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine,* vol. 7, pp. 673 - 679, 2001.

[63] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation* Anchorage, AK, 1998, pp. 69-73.

[64] H.-L. Huang, C.-C. Lee, and S.-Y. Ho, "Selecting a minimal number of relevant genes from microarray data to design accurate tissue classifiers," *Biosystems,* vol. 90, pp. 78-86, 2007.

[65] E. B. Huerta, B. Duval, and J. Hao, "A hybrid ga/svm approach for gene selection and classification of microarray data," *Lecture Notes in Computer Science,* vol. 3907, pp. 34-44, 2006.

[66] K. Deb and A. Raji Reddy, "Reliable classification of two-class cancer data using evolutionary algorithms," *Biosystems,* vol. 72, pp. 111-129, 2003.

[67] W. J. Conover, *Practical nonparametric statistics*, 3 ed. New York, NY: John Wiley & Sons Inc., 1980.

[68] O. Okun and H. Priisalu, "Dataset complexity in gene expression based cancer

classification using ensembles of k-nearest neighbors," *Artificial Intelligence in Medicine,* vol. 45, pp. 151-162, 2009.

[69] Y. Wang, I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. X. Mayer, and H. W. Mewes, "Gene selection from microarray data for cancer classification--a machine learning approach," *Computational Biology and Chemistry,* vol. 29, pp. 37-46, 2005.

[70] I. Inza, P. Larranaga, R. Blanco, and A. J. Cerrolaza, "Filter versus wrapper gene selection approaches in DNA microarray domains," *Artificial Intelligence in Medicine,* vol. 31, pp. 91-103, 2004.

[71] M. Xiong, X. Fang, and J. Zhao, "Biomarker Identification by Feature Wrappers," *Genome Research,* vol. 11, pp. 1878-1887, 2001.

[72] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-Filter Feature Selection Algorithm Using a Memetic Framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 37, pp. 70-76, 2007.

[73] G.-T. Lin, H.-F. Tseng, C.-K. Chang, L.-Y. Chuang, C.-S. Liu, C.-H. Yang, C.-J. Tu, E.-C. Wang, H.-F. Tan, C.-C. Chang, C.-H. Wen, H.-C. Chen, and H.-W. Chang, "SNP combinations in chromosome-wide genes are associated with bone mineral density in Taiwanese women," *Chinese Journal of Physiology,* vol. 51, pp. 32-41, 2008.

[74] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," in *IEEE Swarm Intelligence Symposium* Honolulu, HI, 2007, pp. 120-127.

[75] A. K. Ghosh, "On optimum choice of k in nearest neighbor classification,"

Computational Statistics & Data Analysis, vol. 50, pp. 3113-3123, 2006.

[76] H. G. Schuster and W. Just, *Deterministic chaos: an introduction* 4th ed. Weinheim: Wiley-VCH, 2005.

[77] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.

[78] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 99, pp. 6567-6572, 2002.

[79] B.-W. Cheng and C.-L. Chang, "A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm," *Expert Systems with Applications,* vol. 32, pp. 415-421, 2007.

[80] A. R. YildIz, "A new design optimization framework based on immune algorithm and Taguchi's method," *Computers in Industry,* vol. 60, pp. 613-620, 2009.

[81] M. C. Brandon, M. T. Lott, K. C. Nguyen, S. Spolim, S. B. Navathe, P. Baldi, and D. C. Wallace, "MITOMAP: a human mitochondrial genome database--2004 update," *Nucleic Acids Research,* vol. 33, pp. D611-D613, 2005.

[82] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins, "Clustal W and Clustal X version 2.0," *Bioinformatics,* vol. 23, pp. 2947-2948, 2007.

[83] S. B. Robert and J. S. Moore, "A fast string searching algorithm,"

Communications of the ACM vol. 20, pp. 762-772, 1977.

[84] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali, "DrugBank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research,* vol. 36, pp. D901-D906, 2008.

[85] D. Cotter, P. Guda, E. Fahy, and S. Subramaniam, "MitoProteome: mitochondrial protein sequence database and annotation system," *Nucleic Acids Research,* vol. 32, pp. D463-D467, 2004.

[86] A. C. Smith and A. J. Robinson, "MitoMiner, an Integrated Database for the Storage and Analysis of Mitochondrial Proteomics Data," *Molecular & Cellular Proteomics,* vol. 8, pp. 1324-1337, 2009.

[87] H. Prokisch, C. Andreoli, U. Ahting, K. Heiss, A. Ruepp, C. Scharfe, and T. Meitinger, "MitoP2: the mitochondrial proteome database--now including mouse data," *Nucleic Acids Research,* vol. 34, pp. D705-D711, 2006.

PUBLICATION

Journal papers

1. Chuang, Li-Yeh, Yang, Cheng-San, Wu, Kuo-Chuan, and Yang, Cheng-Hong, Correlation-based Gene Selection and Classification Using Taguchi-BPSO, *Methods of information in medicine,* vol. 49, 2010, pp. 254-268.

Conference papers

1. Chuang, Li-Yeh, Wu, Kuo-Chuan, Chang, Hsueh-Wei, and Yang, Cheng-Hong, 機器學習用於疾病預測, in 27th Workshoop on Combinatorial Mathematics and Computation Theory Taichung, Taiwan, Apr. 30-May. 1, 2010, pp. 50-55.
2. Chuang, Li-Yeh, Wu, Kuo-Chuan, and Yang, Cheng-Hong, Gene Selection and Classification Using CFS-TCBPSO, in *18h Symposium on Recent Advances in Cellular and Molecular Biology* Pingtung, Taiwan, Jan. 20-22, 2010, p. 107.
3. Chuang, Li-Yeh, Wu, Kuo-Chuan, Chang, Hsueh-Wei, and Yang, Cheng-Hong, Prediction Osteoporosis using BPSO-KNN, in *National Computer Symposium* Taipei, Taiwan, Nov. 27-28, 2009, pp. 112-121.
4. Chuang, Li-Yeh, Wu, Kuo-Chuan, Chang, Hsueh-Wei, and Yang, Cheng-Hong, SNP-based prediction for disease susceptibility using Weka, in *Symposium of Bioinformatics and Systems Biology in Taiwan* Taipei, Taiwan, Oct. 8-9, 2009, pp. 77-78.
5. Chuang, Li-Yeh, Wu, Kuo-Chuan, Chang, Hsueh-Wei, and Yang, Cheng-Hong, Risk Prediction for Breast Cancer Apply Single Nucleotide Polymorphism using

BPSO-SVM, in *Joint conference on Medical Informatics in Taiwan* Taipei Taiwan, Oct. 3-5, 2009, pp. 82-88.

6. Chuang, Li-Yeh, Wu, Kuo-Chuan, and Yang, Cheng-Hong, A Hybrid Feature Selection Method Using Gene Expression Data, in *9th IEEE International Conference on Bioinformatics and BioEngineering*, Taichung, Taiwan, Jun 22-24, 2009, pp. 100-106.
7. Yang, Cheng-Hong, Huang, Chi-Chun, Wu, Kuo-Chuan, and Chang, Hsin-Yun, A Novel GA-Taguchi-Based Feature Selection Method, in *9th International Conference on Intelligent Data Engineering and Automated Learning*, Daejeon, South Korea, Nov. 2-5, 2008, pp. 112 - 119.
8. Chuang, Li-Yeh, Wu, Kuo-Chuan, and Yang, Cheng-Hong, Hybrid feature selection method using gene expression data, in *2008 IEEE Conference on Soft Computing in Industrial Applications*, Muroran Hokkaido, Jun. 25-27, Japan, 2008, pp. 199-204.
9. Yang, Cheng-Hong, Wu, Kuo-Chuan, and Chuang, Li-Yeh, Meta-PSO 最佳化參數用於分類問題, in *2008 International Conference on Advanced Information Technologies* Taichung, Taiwan, Apr. 25-26, 2008, p. 141.
10. Yang, Cheng-Hong, Li, Chien-Ting, and Wu, Kuo-Chuan, Feature Selection and Parameter Optimization Using TMGA in Gene Expression, in *16th Symposium on Recent Advances in Cellular and Molecular Biology* Pingtung, Taiwan, Jan. 23-25, 2008, p. 238.

11. C.-S. Yang, Wu, Kuo-Chuan, Chuang, Li-Yeh, Horng, Ji-Hwei, and Yang, Cheng-Hong, 運用 Tabu-BPSO 分析生物資訊資料集之分類問題, in *18th International Conference on Information Management* Taipei, Taiwan, May 26, 2007, p. 85.

12. Yang, Cheng-Hong, Wu, Kuo-Chuan, and Chuang, Li-Yeh, Feature Selection and SVM Parameter Optimization for Microarray with PSO, in *15th Symposium on Recent Advances in Cellular and Molecular Biology* Pingtung, Taiwan, Feb. 1-3, 2007, p. 257.

13. Yang, Cheng-Hong, Wu, Kuo-Chuan, and Horng, Ji-Hwei, 粒子族群最佳化用於特徵選取及支持向量機參數最佳化, in *11th Artificial Intelligence and Applications* Kaohsiung, Taiwan, Dec. 15-16, 2006, p. 48.

14. Yang, Cheng-Hong, Tu, Chung-Jui, Wu, Kuo-Chuan, Chang, Chun-Yang, and Liu, Hsiu-Hsien, Tabu-PSO 用於特徵選, in *5th Information Technology and Applications in Outlying Islands* Chinmen, Taiwan, Jun. 2, 2006, p. 3.

15. Chuang, Li-Yeh, Yang, Cheng-Hong, Wu, Kuo-Chuan, Tu, Chung-Jui, and Lin, Hsien-Lung, Tabu-PSO 用於基因資料表現的特徵選取, in *9th Engineering Technology and Applications of Chinese and Western Medicine* Taichung, Taiwan, May 28, 2006, pp. 80-85.