This article can be cited as A. Naseri and S. M. H. Hasheminejad, An Unsupervised Gene Selection Method Based on Multiobjective Ant Colony Optimization, International Journal of Artificial Intelligence, vol. 17, no. 2, pp. 1-22, 2019.

An Unsupervised Gene Selection Method Based on Multiobjective Ant Colony Optimization

Atefeh Naseri[1] **and Seyed Mohammad Hossein Hasheminejad**[2]

[1]Department of Computer Engineering
Alzahra University
Tehran, Iran
a.naseri@student.alzahra.ac.ir

[2]Department of Computer Engineering
Alzahra University
Tehran, Iran
smh.hasheminejad@alzahra.ac.ir

ABSTRACT

The feature selection process can be defined as an optimization problem aimed to find all the relevant and informative features from the whole set of features. Most feature selection methods use class labels and are considered as supervised methods. But access to these labels is not possible in many real world problems. The identification of a subset of effective genes from the microarray data is one of these problems, which plays a key role in discovery and treatment of diseases. An unsupervised gene selection method based on the multiobjective ant colony optimization has been proposed in this study in which both univariate and multivariate techniques are used for evaluation of the relationship between genes to calculate the fitness function, so that with considering the correlation between genes it will have a better performance in addition to speed. This method was used to find the genes with the highest discriminative power and the minimum level of similarity and redundancy. According to the results, the accuracy of the proposed method has improved in most cases than other methods. On the other hand, it has reported that the proposed method has a low computational complexity, so it can be used for large-scale datasets.

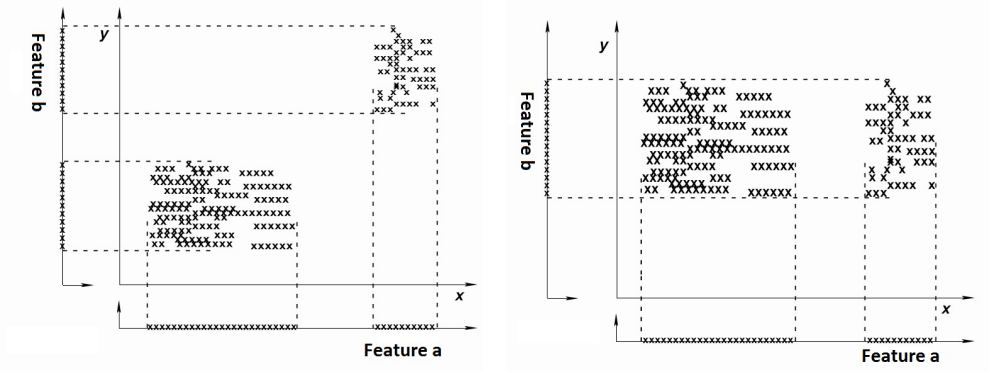
Keywords: Gene Selection, Feature Selection, Multi-Objective Ant Colony Optimization (MOACO), Microarray Data.

Computing Classification System (CCS): G.1.6, I.5.2, G.3, I.2.8, I.2.1

1 Introduction

Microarray-based assay technology provides investigators with the ability to measure the expression profile of thousands of genes in a single experiment(Kumar, Shaik, Abdul Rahim and Sravan Kumar, 2016). This technology can accelerate the identification of potential drugs for treating diseases and provide fruitful results in the drug discovery. But, classification or clustering of microarray data face with the known problem of curse of dimensionality due to the

nature of dimensions of this type of data and low number of samples (Hira and Gillies, 2015). Gene selection is considered as a common technique for microarray data preprocessing. This method refers to a process for identifying a subset of effective genes from the major gene set that improves classification performance and reduces computational costs. Features relevance and redundancy always are concerned in determining efficiency or usefulness of that feature or subset of features (Xue, Zhang, Browne and Yao, 2016). By eliminating irrelevant and redundant features, feature selection can reduce data dimension and increase performance by accelerating the process of learning and simplifying the model. The examples of such features are shown in Figure 1 (Ang, Mirzal, Haron and Hamed, 2016). As shown in this figure, the features with redundancy and irrelevant features reduce the discriminative power.



(a) (b)

Figure 1: (a) Feature a and b are redundant because b provides the same information with regard to discriminating the two clusters. (b) Feature b is irrelevant because it does not contribute to cluster discrimination. On the other hand, if feature a is omitted, then only one cluster will be recognized (Ang, Mirzal, Haron and Hamed, 2016).

Feature selection methods can be divided into four groups (Xue et al., 2016): filter, wrapper, hybrid and embedded. The filter approach identifies the features that maximize the evaluation criterion and do not rely on any particular learning model. The wrapper approach uses a learning algorithm for evaluation of the selected feature subset. This is associated with high computational costs in a large-scale dataset. On the other hand, the speed of the filter-based methods is higher than wrapper methods, but the quality of the final result in these methods is less than that of the wrapper due to the lack of a learning model in the search process. The computational efficiency of the filter model and the proper performance of the wrapper model are used in hybrid methods. But because filter and wrapper models are considered as two separate steps, they don't have proper performance in terms of accuracy. Finally, methods integrating feature selection and classifier learning into a single process are called embedded approaches.

In feature selection approaches, two univariate and multivariate strategies are used for evaluation of the relationship between genes (Saeys, Inza and Larrañaga, 2007). In a univariate strategy, evaluation and ranking of each gene is considered separately. Univariate strategy can

effectively identify and remove irrelevant genes independently of any learning algorithms, but they are unable of removing redundant genes (Moradi and Rostami, 2015). Univariate methods are fast and efficient, but may have less accuracy due to the neglect of the dependencies between genes. The multivariate strategy concerns the correlation between genes and can handle both irrelevant and redundant genes. Therefore, the performance of multivariate-based methods is better than univariate-based methods, but they can be trapped into the local optimum (Lai, Reinders, van't Veer and Wessels, 2006).

Metaheuristic methods have been successfully applied for solving optimization problem in many fields such as image processing(Kheirinejad, Hasheminejad and Riahi, 2018), software design(Tawosi, Jalili and Hasheminejad, 2015), modeling of dynamical systems(Saadat, Moallem and Koofigar, 2017), Knowledge-based systems(Vaščák, 2012), control of aerodynamic systems(Roman, Precup and David, 2018) and computational biology(Niu, Fan, Wang, Li and Wang, 2011) (Gao, Song, Cheng, Todo and Zhou, 2018), (Penas, Banga, González and Doallo, 2015). Recently, swarm intelligence based methods such as ACO and PSO have attracted much attention due to their great function in feature selection area of research (Wan, Wang, Ye and Lai, 2016)(Varma, Kumari and Kumar, 2016)(Gu, Cheng and Jin, 2018)(Tran, Xue and Zhang, 2018). Most of the proposed gene selection methods use class labels of the microarray data in their search processes. However, there are some data whose samples are incorrectly labeled or misslabeled (Niijima and Okuno, 2009). Therefore, the importance of using the unsupervised gene selection methods that have been neglected in the DNA microarray research area is considerable.

In this paper, we present novel multiobjective unsupervised filter based gene selection method using ACO. The efficiency of the filter approach and the suitable performance of the ant colony search strategy are combined without using any learning model. A new proposed multiobjective fitness function evaluates the performance of the found subsets of genes without using any learning model. Both univariate and multivariate strategies are applied in calculating the fitness function simultaneously.

The remainder of this study is organized as follows: the studies related to the feature selection and the main method of the ant colony has been investigated in Section 2. Then, the proposed method is presented in Section 3 in detail. In Section 4, the proposed algorithm is compared with other available feature selection methods. The conclusions of this study are presented in Section 5.

2 Background and related works

Evaluation of all possible subsets is necessary to find the optimal features subset. This means the evaluation of the search space, n^2, which n indicates the number of features. Evaluation of such a space requires time-consuming computations, and can't be used even for a medium-sized dataset. Therefore, the final solution must be provided by the feature selection

methods at an acceptable computational time by establishing an appropriate balance between the solution quality and the computational cost.

2.1 Related works

Nowadays, the ant colony algorithm has been considered for solving the feature selection problem. In (Fallahzadeh, Dehghani-Bidgoli and Assarian, 2018), breast cancer diagnosis by finding the best Raman features using the ACO algorithm is proposed. In this study, error of classification is considered as cost function and by reducing the number of features, model complexity and consequently, its construction time decreases. The related features selection for categorizing the text is developed using an ant colony in (Aghdam, Ghasem-Aghaee and Basiri, 2008)(Aghdam, Ghasem-Aghaee and Basiri, 2009). These studies have modeled the problem states as a graph and used a specific classifier to evaluate the quality of subset of selected features. Kashef and NezamAbadi (Kashef and Nezamabadi-pour, 2015) have used Binary Ant Colony to select a subset of related features for solving the classification problem. They rebuild a graph model, so that each node includes two sub-nodes that are used to select or not to select features. In (Rashno, Nazari, Sadri and Saraee, 2017), a feature selection method based on ant colony has been proposed to classify astronomical images. In their proposed method, both a feature subset is selected for all classes, as well as a feature subset is selected for each of the pixel classes. In (Chen, Chen and Chen, 2013), an ant colony-based feature selection method is proposed for image categorization. (Shunmugapriya and Kanmani, 2017) has proposed a framework for selecting a set of proper features based on a combination of ant colony and bees colony algorithms.
Shekofteh et al. have proposed a hybrid method for feature selection for predicting the soil state using an ant colony algorithm and a fuzzy system in order to minimize the number of features and classification errors (Shekofteh, Ramazani and Shirani, 2017). In (Jameel and Rehman, 2018) a feature selection method using a modified wrapper-based ant colony optimization is proposed. In the proposed approach, the complete graph is used as a search space. The search space also consists of a terminal node. The terminal node is utilized to end the search and is associated with every node in the graph. The classification accuracy is calculated as a fitness function.

Several methods are proposed based on ant colony algorithm to select genes from a microarray dataset. Li et al. have proposed a two-step dimension reduction method based on the Ant colony algorithm that irrelevant genes are eliminated using a modified ant colony system in the first stage, and the final gene set was determined using an improved ant colony system in the second stage (Li, Wang, Chen, Shi and Qin, 2013). Yu et al. have introduced a modified ant colony algorithm to select tumor marker genes and used the SVM classification in the ant colony search process to evaluate the performance of selected genes (Yu, Gu, Liu, Shen and Zhao, 2009). Nemati et al. have proposed a hybrid method for features selection in predicting the pseudoprotective activity of proteins using ant colony and genetic algorithms (Nemati, Basiri, Ghasem-Aghaee and Aghdam, 2009). Tabakhi et al. (Tabakhi, Najafi, Ranjbar and Moradi, 2015) have proposed an unsupervised feature selection method using an ant

colony algorithm. The combination of an ant colony algorithm and the neural network classifier is presented by Kabir et al. to select a subset of prominent features (Kabir, Shahjahan and Murase, 2012). In (Ahuja and Ratnoo, 2017) an ant colony algorithm is proposed with multi-objective supervisory using the classifier function.

2.2 Ant colony optimization

The ACO algorithm is a probabilistic technique for solving computational problems that can be reduced to finding good paths through graphs. This algorithm aims to search for an optimal path in a graph based on the behaviour of ants seeking a path between their colony and source of food. Ants are able to indirectly communicate through *pheromone* (a chemical substance which every ant deposits while operating in its environment) trails to find the shortest path between a food source and their nest.
Initially, the ants start their search through random movements and drop pheromone at a uniform rate. However, the ants that follow the shortest path to the food source and back to the nest will return faster than the ones following a longer path. As a result, more pheromone accumulates on the shorter path making this path attractive for other ants to follow. In this way, increasing amount of pheromone becomes a positive feedback. More and more ants prefer to choose the trail with larger amount of pheromone deposited and eventually all the ants converge to the shortest path (Dorigo, 1992).

3 The proposed method

Most related works have focused on supervised algorithms, while providing labels is difficult in microarray data. On the other hand, the use of the univariate technique in the evaluation of genes has less accuracy, because the dependence between genes is not considered, although this problem isn't observed in multivariate techniques, but they can be trapped into the local optimum. Therefore, this study has proposed a new multiobjective unsupervised gene selection method based on ant colony optimization for microarray data, in which both univariate and multivariate strategies are used to calculate the fitness function simultaneously. Then, search space, transition state, the pheromone update rules, and the multiobjective fitness function are addressed.

Search Space Display: The search space for genes selection is connected as a fully connected weighted graph in which the nodes represent the major set of genes, and the edges of the graph indicates the relationship between the genes (Figure 2) (Basiri and Nemati, 2009).

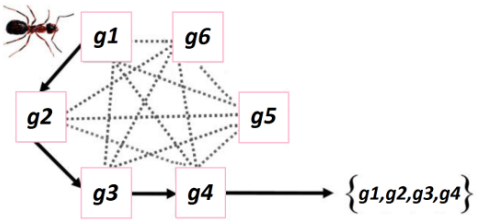


Figure 2: Representation of the search space (Basiri and Nemati, 2009)

In addition, the edge weight is determined between the g_i and g_j genes according to the similarity between them, which is calculated using the following formula:

$$sim\,(g_i, g_j) = \left| \frac{g_i g_j}{\|g_i\| \, \|g_j\|} \right| = \left| \frac{\sum_{s=1}^{p} g_{is} g_{js}}{\sqrt{\sum_{s=1}^{p} {g_{is}}^2} \sqrt{\sum_{s=1}^{p} {g_{js}}^2}} \right| \tag{3.1}$$

Where p is the number of samples and g_{is} represents the amount of the gene i for the sample s. According to the equation, it can be seen that the similarity value is between 0 and 1, Where the value of 0 means there are two completely non-identical genes and the value of 1 means that the two genes are completely similar.

Furthermore, to use the ACO algorithm in the gene selection problem, "desirability" and "heuristic information" must be defined as the basic components. The desirability, $\tau_i, \forall i = 1.....,n$ called pheromone, is associated with the graph nodes (i.e.,genes) and shows the information collected by ants during the search process. Moreover, the heuristic information represents the prior knowledge about the problem. In the proposed method, the heuristic information is simply defined as the inverse of the similarity between genes which is assigned to the graph's edges.

State Transition Rule: The "state transition rule" is designed based on a combination of the heuristic information and the node pheromone values as follows:
when the ant k is located on the gene i, the next gene j can be selected in a greedy way or in a probabilistic way. In the greedy way, the next gene is selected according to the following formula:

$$j = \underset{u \in J_i^k}{\arg\max} \{[\tau_{iu}][\eta(g_i, g_u)]^\beta\}, \qquad if\ q \leq q_0 \tag{3.2}$$

where J_i^k is the unvisited gene set, τ_{iu} is the pheromone value on the edge (g_i, g_u) , $\eta(g_i, g_u) = 1/sim(g_i, g_u)$ is heuristic information which was chosen to be the inverse of the similarity value between genes, parameter β is a parameter which is used to control the importance of pheromone versus heuristic information $(\beta > 0)$, q is a random number uniformly distributed in $[0...1]$, and q_0 is a predefined constant parameter $(0 \leq q_0 \leq 1)$.
In the probabilistic way, the next feature j will be selected based on the probability $P_k(i,j)$, which is defined as follows:

$$P_k(i,j) = \begin{cases} \frac{[\tau_{ij}][\eta(g_i,g_j)]^\beta}{\sum_{u \in J_i^k} [\tau_{iu}][\eta(g_i,g_u)]^\beta}, & if\ j \in J_i^k \\ 0, & otherwise \end{cases} \quad if q > q_0 \tag{3.3}$$

The probabilistic rule (Equation 3.3) allows the ants to build a variety of different solutions in order to explore a larger solution space, while the greedy rule (Equation 3.2) has the strong local search ability.
As we see, state transition rule depends on the parameters q and q_0, which is a tradeoff between $Exploitation$ and $Exploration$. If $q \leq q_0$, then ants select the best gene in the greedy way $(Exploitation)$; otherwise, each gene has a chance of being selected corresponding to its probability value $(Exploration)$. The advantage of the probabilistic way is to avoid being trapped into a local optimum.

The pheromone update rule: The pheromone update rule is applied on all edges after passing each ant. so future ants can utilise this information. Pheromone values are updated using the following formulae:

$$\tau_{ij}\left(t+1\right) = \left(1-\rho\right)\ \tau_{ij}\left(t\right) + \frac{EC\left[i,j\right]}{\sum_{u,v=1,\ldots,n} EC\left[u,v\right]} + \sum_{k=1}^{A} \Delta\tau_{ij}^{k}\left(t\right) \tag{3.4}$$

$where$

$$\Delta\tau_{ij}^{k}\left(t\right) = \begin{cases} fitness\left(k\right), & if\left(i,j\right) \in subset\left(k\right) \\ 0, & otherwise \end{cases} \tag{3.5}$$

Where, ρ is pheromone evaporation rate $(0 < \rho < 1)$ and $\tau_{ij}(t)$ and $\tau_{ij}(t+1)$ show the pheromone values on edge (g_i, g_j) at time t and $t+1$, respectively, n is the number of main genes, $EC[i,j]$ is the number of times which the edge (gi, gj) is viewed by ants and A is the number of ants. $Fitness(k)$ indicates a fitness function that determines the quality of the k^{th} ant solution, and the $subset(k)$ represents the subset of genes selected by k^{th}-ant.

Multi-objective fitness function: The proposed method was aimed to use the analysis of relevance and redundancy in the gene selection process. In other words, on the one hand, a subset of genes with the highest discriminative power should have a greater fitness function and, on the other hand, subsets with the least redundancy and similarity between the genes should be selected by ants. For achieving this purpose, both univariate and multivariate strategies are used for calculating the fitness function.
In this paper, the *difference* between *Term variance* (Theodoridis and Koutroumbas, 2008) as a univariate criterion and *mutual correlation* (Haindl, Somol, Ververidis and Kotropoulos, 2006) as a multivariate criterion is used for calculating the fitness function for each gene. The maximum term of Term variance and minimum mutual correlation will be more appropriate and result in a higher fitness function value.
Term Variance is the simplest univariate evaluation of relevance of genes, and shows that genes with larger variance contain more valuable information that is computed as follows:

$$TV\left(g_i\right) = \frac{1}{p}\sum_{s=1}^{p}\left(g_{is} - \overline{g_i}\right)^2 \tag{3.6}$$

where p is the number of samples, g_{is} denotes the value of gene i for sample s, and $\bar{g}_i$ is the average value of all the samples corresponding to gene i. Pearson correlation coefficient to

measure correlation between different genes as follows:

$$Cor(g_i, g_j) = \frac{\sum_{s=1}^{p} (g_{is} - \overline{g_i})(g_{js} - \overline{g_j})}{\sqrt{\sum_{s=1}^{p} (g_{is} - \overline{g_i})^2} \sqrt{\sum_{s=1}^{p} (g_{js} - \overline{g_j})^2}} \tag{3.7}$$

where $Cor(g_i, g_j)$ is the correlation coefficient between two gene i and j, and p is the number of the samples, g_{is} and g_{js} denote the values of the genes i and j for the s^{th} sample, respectively, and $\bar{g_i}$ and $\bar{g_j}$ represent the mean values of g_i and g_j over all of the p samples.

According to equation 3.7, the correlation coefficient between two genes computes the similarity between the genes. After computing the correlation coefficient for all possible combinations of genes, the correlation value for gene i is calculated as follows (Moradi and Gholampour, 2016):

$$Cor\,(g_i) = \frac{\sum_{j=1}^{f} |Cor(g_i, g_j)|}{f - 1} \quad if \quad i \neq j \tag{3.8}$$

where f is the number of all genes and $Cor(g_i, g_j)$ denotes the Pearson correlation value between genes i and j.

Whatever correlation value for a gene is higher, that means this gene is highly similar to other genes and causes redundancy, while the lower value means that this gene is more distinct from others and can provide a higher discriminative power.
The fitness value of solution k is computed as follows:

$$fitness(k) = \frac{1}{|subset(k)|} \sum_{i=1}^{|subset(k)|} \left(TV(g_i^k) - Cor(g_i^k) \right) \tag{3.9}$$

where $subset(k)$ is the subset of genes selected by ant k, $|subset(k)|$ is the size of $subset(k)$, g_i^k is the i-th gene in $subset(k)$. This particular type of fitness function is independent of any learning model.
This is due to the fact that the filter-based methods dose not employ a learning model in their search operations and thus in each iteration of these algorithms, the classifier accuracy is not needed to compute the fitness of each solution. Training a given classifier with the full gene set is time-consuming especially due to the high-dimensionality of microarray datasets (Figure 3).

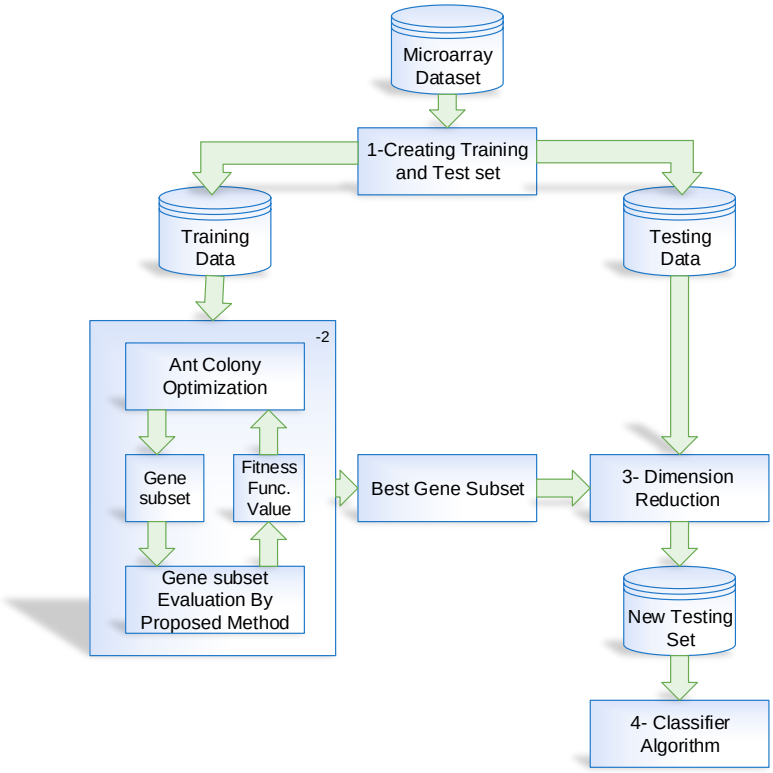


Figure 3: The overall flowchart of the proposed method

3.1 The framework of the proposed gene selection method based on ACO

The proposed method has two main parts including the initialization part and the gene selection part. In the initialization part, the similarity values between genes are calculated and assigned to the edges in the graph. Then, the initial intensity of pheromone values on the edges is set to a constant value. Finally, the relevance of each gene is simply evaluated using the new fitness function.

The gene selection part is an iterative process. At each iteration, the edge counter (EC) is defined to count the number of times that a specific edge between two genes is visited by ants, and their initial values are set to zero. Additionally, ants are randomly placed on the graph nodes as their starting nodes. Each ant constructs a candidate solution by iteratively adding a gene to the current selected gene subset according to a "state transition rule" which is a combination of the heuristic information and the pheromone values. An ant prefers to visit an gene with low similarity to its previously selected gene as well as high intensity of pheromone values. When a given edge is visited by the ant, its corresponding edge counter $(i.e., EC[i, j])$ is increased. This step continues until a given number of genes are selected by each ant.

The candidate subsets of genes are evaluated using a new proposed fitness function. Then, the subset of genes with the better fitness value is kept as the best result in the current iteration. The intensity of pheromone values on the edges are updated according to a "pheromone updating rule". In other words, a fraction of pheromone values on each edge is evaporated; the edges with greater EC values get the greater amount of pheromone; and all ants deposit an

amount of pheromone on the edges which belong to the fitness of their selected gene subsets. This process continues until the maximum number of iterations I is reached. Finally, the global best subset of genes in all iterations is chosen as the final subset of genes. The pseudo code of the proposed algorithm is shown in Figure 4.

Algorithm 1.

Input

D: $p \times n$ matrix, p patterns of a n dimensional training set.
$m(\leq n)$: the number of genes for the final reduced gene set.
I: the maximum number of allowed iterations.
A: the number of ants.
NG: the number of genes selected by ants in each iteration.

Output

$\widetilde{D}$: $p \times m$ matrix, reduced dimensional training set.

Begin

Compute the similarity $sim(g_i, g_j)$ between pairs of genes, $\forall\, i,j = 1 \ldots n$.
Compute the relevance r_i of genes, $\forall\, i = 1 \ldots n$.
Set the initial intensity of pheromone $\tau_{ij}(1)$ on the edges to constant value c, $\forall\, i,j = 1 \ldots n$.
for $t = 1$ to I **do**
    Initialize the edge counter $EC[i,j]$ to zero, $\forall\, i,j = 1 \ldots n$.
    Place the ants randomly on the nodes in the graph.
    for $i = 1$ to NG **do**
        for $k = 1$ to A **do**
            Choose the next gene among unvisited genes according to the state transition rule.
            Move the k-th ant to the new selected gene.
            Increment edge counter corresponding to the visited edge.
        end for
    end for
    Evaluate the candidate subsets of selected genes using fitness function.
    Find global best solution.
    Update pheromone values by applying the pheromone updating rule.
end for
Keep the global best solution in all iterations.
Build $\widetilde{D}$ from D based on the global best solution.

End

Figure 4: Pseudo code of the proposed gene selection method

4 Results

In this section, we empirically evaluate the performance of the proposed method upon five wellknown microarray datasets. Seven frequently used gene/feature selection methods were selected to be compared with the proposed method. Term Variance(TV) (Theodoridis and Koutroumbas, 2008) and Laplacian Score(LS) (Liao, Jiang, Liang, Zhu, Cai and Cao, 2014) are

univariate methods that can eliminate unrelated genes in an effective manner. The relevance-redundancy feature selection(RRFS) (Ferreira and Figueiredo, 2012), the random subspace method(RSM) (Lai, Reinders and Wessels, 2006) and mutual correlation(MC) (Haindl et al., 2006), are considered as multivariate methods that eliminate unrelated genes with redundancy. Since the proposed method is an ant colony-based gene selection method, MGSACO(Tabakhi et al., 2015) and UFSACO(Tabakhi, Moradi and Akhlaghian, 2014) have been selected as the benchmark approaches which are the latest ACO-based gene selection methods.
As the proposed method is a filter-based gene selection method without using any classifiers in the gene selection process, it should have good performance on different types of classifiers. Therefore, three frequently used classifiers including support vector machine(SVM)(Guyon, Weston, Barnhill and Vapnik, 2002), naïve Bayes(NB)(Wang, Tetko, Hall, Frank, Facius, Mayer and Mewes, 2005), and decision tree(DT)(Chen, Wang, Tsai, Wang, Adrian, Cheng, Yang, Teng, Tan and Chang, 2014) were considered to evaluate the classification prediction capability of the selected genes over the datasets.
The true positive (TP), false positive (FP), true negative (TN) and false negative (FN) criteria are used to evaluate the results of the classifiers. The formula for the performance criteria used is as follows:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} \tag{4.1}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.2}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.3}$$

And the classification error is calculated as follows:

$$\mathrm{ClassificationErrorRate} = 1 - \mathrm{ClassificationAccuracy} \tag{4.4}$$

The description of the datasets used in the experiment, the parameter settings, and the experimental results are presented in the following subsections.

4.1 Datasets

The proposed method was evaluated using five microarray datasets related to different types of cancer. These sets are available in the UCI datasets. Table1 shows some characteristics of datasets.

Dataset	Genes	Classes	Patterns
Colon	2000	2	62
SRBCT	2308	4	83
Leukemia	7129	2	72
Prostate Tumor	10509	2	102
Lung Cancer	12600	5	203

Table 1: Characteristics of the datasets used in the experiments.

4.2 Parameters setting

The proposed method includes a different number of adjustable parameters. The proper values of these parameters were chosen after a number of preliminary runs, and not meant to be optimal. The maximum number of cycles is set to 50, the number of ants is set to 100, parameter q_0 in equation 3.2 and 3.3 is set to 0.7, the initial pheromone values on each edge are set to 0.2 $(\tau_{ij} = 0.2, \forall i, j = 1...n)$, importance of the pheromone and heuristic information is assumed equal $(\beta = 1)$, and the evaporation rate parameter is set to 0.2 $(\rho = 0.2)$.
For the rest of the methods, there are parameters to be set. To make a fair comparison, the parameters of MGSACO and UFSACO are set to $NC_{max} = 50$, $N_{Ant} = 100$, as reported in (Tabakhi et al., 2014) and (Tabakhi et al., 2015). Moreover, for the RRFS method, the maximum allowed similarity between pairs of features is set in the range of $[0.5, 1)$. Finally, for the RSM method, the number of iterations is set to 50, and the size of the subspace in each iteration is set to 200.
Three frequently used classifiers including support vector machine(SVM)(Guyon et al., 2002), naïve Bayes(NB)(Wang et al., 2005), and decision tree(DT)(Chen et al., 2014) were used to evaluate the classification prediction capability of the selected features over the datasets.
The WEKA machine learning software library (Hall, Frank, Holmes, Pfahringer, Reutemann and Witten, 2009) is used to implement classifiers. SMO with the polykernel was selected as the SVM classifier and it was applied with the one-against-rest strategy for the multiclass problems. Also, in SMO classifier, the complexity parameter c was set to 1 and the tolerance parameter was set to 0.001. Additionally, NaïveBayes was used as the NB classifier. Moreover, $J48$ was adopted as the DT classifier, in which the post-pruning technique was used in the pruning phase, its confidence factor was set to 0.25, and the minimum number of samples per leaf was set to 2. The dataset is randomly divided into the training dataset ($\frac{2}{3}$ dataset) and the test dataset in each run. The tests are performed on a Core $i5$ processor with 2.13 GHz processor with 4 GB of RAM using Java.

4.3 Experimental results

The results of comparison between the proposed method and unsupervised methods including MGSACO, UFSACO, RSM, MC, RRFS, TV, and LS in terms of the average error of classification (in percent), using the support vector machine and Bayesian and the decision tree classifiers when the number of selected genes is 20, over 5 independent runs are presented in tables 2 to 4.

Table 2: Average classification error rate over 5 independent runs of different feature selection methods using SVM classifier.

Dataset	Our Method	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
Colon	20.36	21.81	21.81	24.54	38.18	24.54	21.81	33.63
SRBCT	22.069	25.51	28.27	37.93	45.51	31.72	39.31	36.55
Leukemia	23.7	17.94	41.02	37.64	38.23	23.52	20.58	35.29
Prostate	19.143	26.85	40.57	22.85	34.28	30.85	28	48
Lung Cancer	11.429	14.28	17.14	35.71	28.57	19.14	27.71	18

Table 3: Average classification error rate over 5 independent runs of different feature selection methods using Bayesian classifier.

Dataset	Our Method	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
Colon	18.182	20	28.18	26.36	31.81	32.72	41.81	47.27
SRBCT	11.034	15.86	20	37.92	37.93	28.27	38.62	32.41
Leukemia	26.77	7.69	41.02	42.35	29.41	35.29	32.35	8.82
Prostate	34.286	37.14	39.42	30.28	33.71	31.42	33.14	32.57
Lung Cancer	17.143	20	35.71	23.57	59.04	21.71	31.99	29.99

Table 4: Average classification error rate over 5 independent runs of different feature selection methods using decision tree classifier.

Dataset	Our Method	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
Colon	20.909	23.63	24.54	28.18	33.63	34.54	31.81	39.09
SRBCT	21.149	22.75	27.58	58.62	44.13	28.96	22.75	45.51
Leukemia	23.529	23.07	30.76	38.82	32.35	20.58	20.58	29.41
Prostate	25.714	29.71	33.71	33.71	36	37.71	38.85	43.99
Lung Cancer	18.571	20	28.57	30.71	31.42	20.28	24.28	21.43

It can be seen from Table 2 and 4 that the proposed method obtains the lowest classification error rate compared to the other methods on all of the datasets, expect for the Luekemia Tumor dataset, that gets the second lowest error rates.

The performance of the proposed method on the SRBCT dataset for different numbers of genes was evaluated using different types of classifiers in the second series of experiments and the results are presented in Tables 5 to 7. In these tables, the average classification error rates, over 5 independent runs of the proposed method and those of unsupervised methods is reported. In most cases, the performance of the proposed method is better than other methods.

Table 5: Average classification error rates (in percent) over 5 runs of the gene selection methods on SRBCT dataset using SVM classifier. The best result is marked in boldface and underlined and second best is in boldface.

♯ selected genes	Our Method	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
10	44.828	39.3	51.72	59.99	64.13	45.51	46.89	44.13
20	22.069	25.51	28.27	37.93	45.51	31.72	39.31	36.55
30	13.793	14.48	14.48	35.86	41.38	15.86	28.96	28.96
40	6.897	7.58	9.65	39.3	34.48	13.79	25.51	13.79
50	2.759	7.58	4.14	18.62	17.23	4.14	19.31	17.24
60	4.138	4.14	4.14	18.62	17.93	4.14	6.9	13.1
70	3.448	3.45	4.14	16.55	14.48	8.96	3.45	18.62
80	2.06	2.07	3.45	15.86	15.17	2.76	5.52	10.34
90	0.2	0.69	5.52	13.79	11.03	4.14	3.45	9.65
100	1.379	2.07	0.69	8.96	9.65	2.07	3.45	6.89

Table 6: Average classification error rates (in percent) over 5 runs of the gene selection methods on SRBCT dataset using Bayesian classifier.

♯ selected genes	Our Method	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
10	10.345	18.62	31.03	46.2	35.17	37.24	47.58	42.06
20	11.034	15.86	20	37.92	37.93	28.27	38.62	32.41
30	9.31	11.72	10.34	39.31	20.69	22.06	27.58	27.58
40	6.897	9.65	11.72	25.51	26.89	22.75	24.13	27.58
50	6.552	15.17	12.41	22.75	23.44	23.45	18.62	26.2
60	12.414	7.58	6.21	22.75	21.37	15.86	17.93	27.58
70	3.44	3.45	11.03	20.68	17.24	11.72	19.31	12.41
80	3.448	4.14	13.1	25.51	23.44	5.52	8.96	22.76
90	6.207	8.27	3.45	20.69	23.44	10.34	8.96	26.89
100	6.897	5.52	13.1	22.06	16.55	22.75	17.24	24.13

Table 7: Average classification error rates (in percent) over 5 runs of the gene selection methods on SRBCT dataset using decision tree classifier.

♯ selected genes	Our Method	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
10	26.207	21.37	31.03	53.1	55.17	36.55	41.37	50.34
20	21.149	22.75	27.58	58.62	44.13	28.96	22.75	45.51
30	22.89	19.3	26.89	41.37	39.31	24.82	32.41	24.83
40	12.414	22.07	14.48	45.51	35.16	22.07	27.58	20
50	18.621	16.56	25.51	34.48	41.37	26.2	24.13	25.51
60	19.31	17.93	21.38	37.24	29.65	22.75	23.44	22.75
70	17.241	13.79	26.2	40	28.96	20.69	24.13	21.37
80	22.069	17.24	20	37.93	36.55	17.93	17.24	21.37
90	20	22.06	18.62	37.93	40.68	22.06	20.68	26.2
100	18.379	15.86	22.75	31.72	38.62	23.44	31.03	28.96

The results of 30 independent runs of the proposed method, UFSACO, MGSACO methods

and the best, worst, mean and standard deviation of the classification performance on different dataset for selecting 30 genes are presented in Tables 8 - 16. According to the results, the proposed method has better accuracy in most cases.

Table 8: Calculating the best, worst, mean and standard deviation of classification performance of proposed method over 30 independent runs on Colon dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	95.45	68.18	82.12	7.05	95.45	68.18	82.57	6.86	90.9	68.18	79.84	7.32
Precision	100	61.11	81.16	7.91	100	66.66	85.42	8.3	100	66.66	82.65	8.01
Recall	100	78.57	94.51	5.84	100	73.33	89.02	6.8	100	60	85.69	12.13

Table 9: Calculating the best, worst, mean and standard deviation of classification performance of proposed method over 30 independent runs on SRBCT dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	93.1	55.17	77.93	8.8	96.55	79.31	89.54	5.83	100	58.62	78.03	10.69
Precision	100	46.15	85.67	16.58	100	66.66	92.2	9.44	100	44.44	84.27	16.61
Recall	100	53.84	89.21	12.99	100	77.77	94.82	7.23	100	41.66	77.95	15.32

Table 10: Calculating the best, worst, mean and standard deviation of classification performance of proposed method over 30 independent runs on Luekemia dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	92	60	76.01	9.44	91.17	58.82	73.23	9.37	91.17	60	74.1	7.99
Precision	88.88	62.5	75.33	8.48	94.73	60	72.01	9.21	100	66.66	76.03	8.26
Recall	100	73.33	93.32	7.33	100	80	92.16	5.52	95	60	83.83	8.83

Table 11: Calculating the best, worst, mean and standard deviation of classification performance of UFSACO method over 30 independent runs on Colon dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	90.90	59.09	77.12	8.13	90.90	59.09	77.12	8.13	90.90	54.54	75.9	7.74
Precision	100.0	56.25	75.86	9.75	100	56.25	75.86	9.75	100.0	61.11	76.73	8.33
Recall	100.0	81.25	95.87	5.68	100	81.25	95.87	5.68	100.0	63.63	88.81	9.95

Table 12: Calculating the best, worst, mean and standard deviation of classification performance of UFSACO method over 30 independent runs on SRBCT dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	89.65	48.27	69.88	9.32	93.1	68.96	83.44	7.09	89.65	55.17	73.44	9.92
Precision	100	35.29	74.18	22.3	100	58.33	89.49	12.27	100	44.44	80.87	14.01
Recall	100	46.15	83.93	16.48	100	54.54	83.9	13.09	100	40	74.46	15.81

Table 13: Calculating the best, worst, mean and standard deviation of classification performance of UFSACO method over 30 independent runs on Luekemia dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	88	48	71.6	11.08	92	56	74.26	9.43	88	48	69.2	10.02
Precision	93	45.83	75.46	12.61	100	61.11	83.56	11.44	91.66	52.94	75.59	10.47
Recall	100	66.66	90.59	10.62	100	42.85	78.28	14.29	94.11	55	79.41	9.87

Table 14: Calculating the best, worst, mean and standard deviation of classification performance of MGSACO method over 30 independent runs on Colon dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	100	63.63	80.45	9.99	100	45.45	76.81	10.96	95.45	59.09	79.54	9.07
Precision	100	61.9	80.39	11.46	100	54.54	82.91	10.57	93.33	58.82	80.46	8.86
Recall	100	82.35	95.1	5.82	100	33.33	81.22	17.44	100	70.58	90.66	9.17

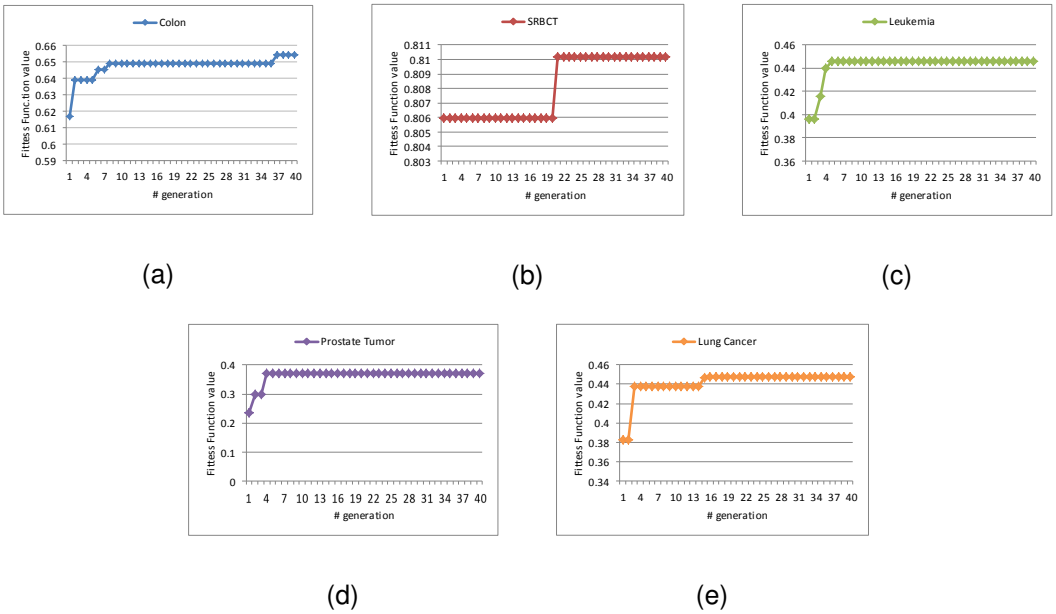
Table 15: Calculating the best, worst, mean and standard deviation of classification performance of MGSACO method over 30 independent runs on SRBCT dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	100	48.27	73.9	11.4	96.55	75.86	89.77	5.08	93.1	51.72	75.51	9.59
Precision	100	34.78	81.06	20.28	100	71.42	95.06	7.45	100	37.5	83.6	15.51
Recall	100	37.5	87.8	15.88	100	66.66	92.72	9.53	100	50	78.73	13.06

Table 16: Calculating the best, worst, mean and standard deviation of classification performance of MGSACO method over 30 independent runs on Luekimia dataset for selecting 30 genes using different classifiers.

Classifier	Support Vector Machine				Naïve Bayesian				Decision Tree			
-	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std
Accuracy	96	60	76.66	9.1	96	56	76.13	9.54	92	52	74.13	10.69
Precision	100	61.9	77.9	10.57	100	68.18	83.43	9.26	100	64.7	80.26	9.39
Recall	100	66.66	92.26	8.81	100	47.05	80.75	14.23	100	47.05	81.68	11.86

An example of changes in the total fitness function of proposed method for different datasets is shown in Figure 5. These changes are reported for each 20 replicates from 800 replicates set for a run.

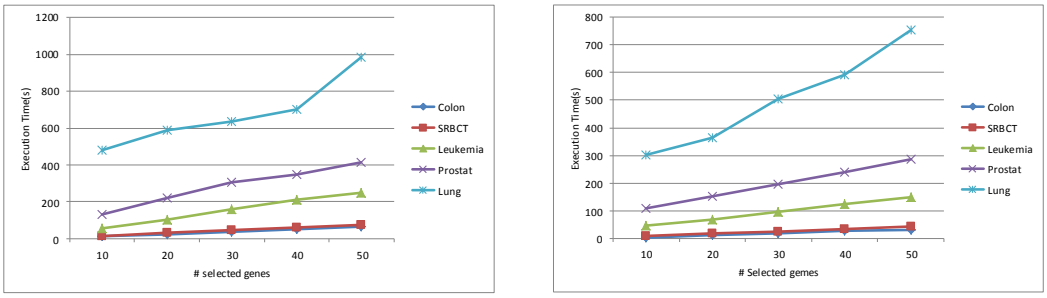


(a) (b) (c)

(d) (e)

Figure 5: changes of the total fitness function for different datasets: a)Colon b)SRBCT c)Luekemia d)Prostate e)Lung

Evaluation of the execution time of the proposed method over different numbers of genes on all of the datasets was done. The average execution time of the proposed method and MGSACO (in seconds) is shown in Figure 6. It can be concluded that, when the number of main genes in the datasets increases, more time is required to find the subset of genes.



(a) (b)

Figure 6: Evaluation of the execution time: a)proposed method b)MGSACO(Tabakhi et al., 2015)

An example of genes that are selected by the proposed method is shown in table 17. The numbers listed in this table are the indexes of the genes in the dataset.

Table 17: genes selected by the proposed method

Dataset	subset gene selected
Colon	260-261-262-263-317-408-415-765-878-1208-1312-14-17-1423-1617-1810-1895
SRBCT	129-187-198-220-566-1227-1319-1764-1916-2046
Leukemia	1734-1735-1893-2913-3486-5987-6201-6794-6954
Prostate	160-454-1098-1249-3982-4692-6144-7003-9354-10193
Lung Cancer	205-1182-3526-4943-6240-6641-9405-11316-11655-12515

5 Conclusion

In the present study, an unsupervised gen selection method in microarray data is proposed based on ant colony algorithm. In order to improve the efficiency of the proposed method, the computational efficiency of filter selection method and the good efficiency of the Ant colony search strategy is combined. In addition, a new and multi-objective fitness function has been used to evaluate the subset of selected genes without using a learning model in order to increase the efficiency of the proposed method. The performance of the proposed method was examined on five datasets using three different classifiers. According to the results, the proposed method has better performance.

References

Aghdam, M. H., Ghasem-Aghaee, N. and Basiri, M. E. 2008. Application of ant colony optimization for feature selection in text categorization, *IEEE World Congress on Computational Intelligence*, IEEE, pp. 2867–2873.

Aghdam, M. H., Ghasem-Aghaee, N. and Basiri, M. E. 2009. Text feature selection using ant colony optimization, *Expert Systems with Applications* **36**(3): 6843–6853.

Ahuja, J. and Ratnoo, S. 2017. Dimension reduction for microarray data using multi-objective ant colony optimisation, *International Journal of Computational Systems Engineering* **3**(1-2): 58–73.

Ang, J. C., Mirzal, A., Haron, H. and Hamed, H. N. A. 2016. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **13**(5): 971–989.

Basiri, M. E. and Nemati, S. 2009. A novel hybrid aco-ga algorithm for text feature selection, *IEEE World Congress on Computational Intelligence*, IEEE, pp. 2561–2568.

Chen, B., Chen, L. and Chen, Y. 2013. Efficient ant colony optimization for image feature selection, *Signal Processing* **93**(6): 1566–1576.

Chen, K.-H., Wang, K.-J., Tsai, M.-L., Wang, K.-M., Adrian, A. M., Cheng, W.-C., Yang, T.-S., Teng, N.-C., Tan, K.-P. and Chang, K.-S. 2014. Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm, *BMC Bioinformatics* **15**(1): 49.

Dorigo, M. 1992. *Optimization, learning and natural algorithms*, PhD thesis, Politecnico di Milano.

Fallahzadeh, O., Dehghani-Bidgoli, Z. and Assarian, M. 2018. Raman spectral feature selection using ant colony optimization for breast cancer diagnosis, *Lasers in Medical Science* pp. 1–8.

Ferreira, A. J. and Figueiredo, M. A. 2012. An unsupervised approach to feature discretization and selection, *Pattern Recognition* **45**(9): 3048–3060.

Gao, S., Song, S., Cheng, J., Todo, Y. and Zhou, M. 2018. Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **15**(4): 1365–1378.

Gu, S., Cheng, R. and Jin, Y. 2018. Feature selection for high-dimensional classification using a competitive swarm optimizer, *Soft Computing* **22**(3): 811–822.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. 2002. Gene selection for cancer classification using support vector machines, *Machine Learning* **46**(1-3): 389–422.

Haindl, M., Somol, P., Ververidis, D. and Kotropoulos, C. 2006. Feature selection based on mutual correlation, *Iberoamerican Congress on Pattern Recognition*, Springer, pp. 569–577.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. 2009. The weka data mining software: an update, *ACM SIGKDD Explorations Newsletter* **11**(1): 10–18.

Hira, Z. M. and Gillies, D. F. 2015. A review of feature selection and feature extraction methods applied on microarray data, *Advances in Bioinformatics* **2015**.

Jameel, S. and Rehman, S. U. 2018. An optimal feature selection method using a modified wrapper-based ant colony optimisation, *Journal of the National Science Foundation of Sri Lanka* **46**(2).

Kabir, M. M., Shahjahan, M. and Murase, K. 2012. A new hybrid ant colony optimization algorithm for feature selection, *Expert Systems with Applications* **39**(3): 3747–3763.

Kashef, S. and Nezamabadi-pour, H. 2015. An advanced aco algorithm for feature subset selection, *Neurocomputing* **147**: 271–279.

Kheirinejad, S., Hasheminejad, S. M. H. and Riahi, N. 2018. Max-min ant colony optimization method for edge detection exploiting a new heuristic information function, *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 12–15.

Kumar, A., Shaik, F., Abdul Rahim, B. and Sravan Kumar, D. 2016. *DNA Micro Array Analysis*, Springer Singapore, Singapore, pp. 39–47.

Lai, C., Reinders, M. J., van't Veer, L. J. and Wessels, L. F. 2006. A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets, *BMC Bioinformatics* **7**(1): 235.

Lai, C., Reinders, M. J. and Wessels, L. 2006. Random subspace method for multivariate feature selection, *Pattern Recognition Letters* **27**(10): 1067–1076.

Li, Y., Wang, G., Chen, H., Shi, L. and Qin, L. 2013. An ant colony optimization based dimension reduction method for high-dimensional datasets, *Journal of Bionic Engineering* **10**(2): 231–241.

Liao, B., Jiang, Y., Liang, W., Zhu, W., Cai, L. and Cao, Z. 2014. Gene selection using locality sensitive laplacian score, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **11**(6): 1146–1156.

Moradi, P. and Gholampour, M. 2016. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, *Applied Soft Computing* **43**: 117–130.

Moradi, P. and Rostami, M. 2015. A graph theoretic approach for unsupervised feature selection, *Engineering Applications of Artificial Intelligence* **44**: 33–45.

Nemati, S., Basiri, M. E., Ghasem-Aghaee, N. and Aghdam, M. H. 2009. A novel aco–ga hybrid algorithm for feature selection in protein function prediction, *Expert Systems with Applications* **36**(10): 12086–12094.

Niijima, S. and Okuno, Y. 2009. Laplacian linear discriminant analysis approach to unsupervised feature selection, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **6**(4): 605–614.

Niu, B., Fan, Y., Wang, H., Li, L. and Wang, X. 2011. Novel bacterial foraging optimization with time-varying chemotaxis step, *International Journal of Artificial Intelligence*^{TM} **7**(A11): 257–273.

Penas, D. R., Banga, J. R., González, P. and Doallo, R. 2015. Enhanced parallel differential evolution algorithm for problems in computational systems biology, *Applied Soft Computing* **33**: 86–99.

Rashno, A., Nazari, B., Sadri, S. and Saraee, M. 2017. Effective pixel classification of mars images based on ant colony optimization feature selection and extreme learning machine, *Neurocomputing* **226**: 66–79.

Roman, R.-C., Precup, R.-E. and David, R.-C. 2018. Second order intelligent proportional-integral fuzzy control of twin rotor aerodynamic systems, *Procedia Computer Science* **139**: 372–380.

Saadat, J., Moallem, P. and Koofigar, H. 2017. Training echo state neural network using harmony search algorithm, *International Journal of Artificial Intelligence*^{TM} **15**(1): 163–179.

Saeys, Y., Inza, I. and Larrañaga, P. 2007. A review of feature selection techniques in bioinformatics, *Bioinformatics* **23**(19): 2507–2517.

Shekofteh, H., Ramazani, F. and Shirani, H. 2017. Optimal feature selection for predicting soil cec: Comparing the hybrid of ant colony organization algorithm and adaptive network-based fuzzy system with multiple linear regression, *Geoderma* **298**: 27–34.

Shunmugapriya, P. and Kanmani, S. 2017. A hybrid algorithm using ant and bee colony optimization for feature selection and classification (ac-abc hybrid), *Swarm and Evolutionary Computation* **36**: 27–36.

Tabakhi, S., Moradi, P. and Akhlaghian, F. 2014. An unsupervised feature selection algorithm based on ant colony optimization, *Engineering Applications of Artificial Intelligence* **32**: 112–123.

Tabakhi, S., Najafi, A., Ranjbar, R. and Moradi, P. 2015. Gene selection for microarray data classification using a novel ant colony optimization, *Neurocomputing* **168**: 1024–1036.

Tawosi, V., Jalili, S. and Hasheminejad, S. M. H. 2015. Automated software design using ant colony optimization with semantic network support, *Journal of Systems and Software* **109**: 1–17.

Theodoridis, S. and Koutroumbas, K. 2008. Pattern recognition, *IEEE Transactions on Neural Networks* **19**(2): 376.

Tran, B., Xue, B. and Zhang, M. 2018. A new representation in pso for discretization-based feature selection, *IEEE Transactions on Cybernetics* **48**(6): 1733–1746.

Varma, P. R. K., Kumari, V. V. and Kumar, S. S. 2016. Feature selection using relative fuzzy entropy and ant colony optimization applied to real-time intrusion detection system, *Procedia Computer Science* **85**: 503–510.

Vaščák, J. 2012. Adaptation of fuzzy cognitive maps by migration algorithms, *Kybernetes* **41**(3/4): 429–443.

Wan, Y., Wang, M., Ye, Z. and Lai, X. 2016. A feature selection method based on modified binary coded ant colony optimization algorithm, *Applied Soft Computing* **49**: 248–258.

Wang, Y., Tetko, I. V., Hall, M. A., Frank, E., Facius, A., Mayer, K. F. and Mewes, H. W. 2005. Gene selection from microarray data for cancer classification—a machine learning approach, *Computational Biology and Chemistry* **29**(1): 37–46.

Xue, B., Zhang, M., Browne, W. N. and Yao, X. 2016. A survey on evolutionary computation approaches to feature selection, *IEEE Transactions on Evolutionary Computation* **20**(4): 606–626.

Yu, H., Gu, G., Liu, H., Shen, J. and Zhao, J. 2009. A modified ant colony optimization algorithm for tumor marker gene selection, *Genomics, proteomics & bioinformatics* **7**(4): 200–208.