Research Article

Gene Selection via a New Hybrid Ant Colony Optimization Algorithm for Cancer Classification in High-Dimensional Data

Ahmed Bir-Jmel, Sidi Mohamed Douiri, and Souad Elbernoussi

Laboratory of Mathematics, Computer Science & Applications-Security of Information, Department of Mathematics, Faculty of Sciences, Mohammed V University, Rabat, Morocco

Correspondence should be addressed to Ahmed Bir-Jmel; birjmel@gmail.com

Received 4 February 2019; Revised 14 August 2019; Accepted 9 September 2019; Published 13 October 2019

Academic Editor: Martti Juhola

Copyright © 2019 Ahmed Bir-Jmel et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The recent advance in the microarray data analysis makes it easy to simultaneously measure the expression levels of several thousand genes. These levels can be used to distinguish cancerous tissues from normal ones. In this work, we are interested in gene expression data dimension reduction for cancer classification, which is a common task in most microarray data analysis studies. This reduction has an essential role in enhancing the accuracy of the classification task and helping biologists accurately predict cancer in the body; this is carried out by selecting a small subset of relevant genes and eliminating the redundant or noisy genes. In this context, we propose a hybrid approach (MWIS-ACO-LS) for the gene selection problem, based on the combination of a new graph-based approach for gene selection (MWIS), in which we seek to minimize the redundancy between genes by considering the correlation between the latter and maximize gene-ranking (Fisher) scores, and a modified ACO coupled with a local search (LS) algorithm using the classifier 1NN for measuring the quality of the candidate subsets. In order to evaluate the proposed method, we tested MWIS-ACO-LS on ten well-replicated microarray datasets of high dimensions varying from 2308 to 12600 genes. The experimental results based on ten high-dimensional microarray classification problems demonstrated the effectiveness of our proposed method.

1. Introduction

In recent years, DNA microarray technology has grown tremendously, thanks to its unquestionable scientific merit. This technology developed in the early 1990s allowed researchers to simultaneously measure the expression levels of several thousand genes [1, 2]. These levels of expression are very important for the detection or classification of the specific tumor type. The microarray data is transformed into gene expression matrices, where a row represents an experimental condition and column represents a gene; each value of $x_{ij}$ is the measure of the level of expression of the $j^{th}$ gene in the $i^{th}$ sample (see Table 1).

For the cancer classification problem, each line contains information about the class of a sample (the type of cancer). Thus, DNA microarray analysis can be formulated as a supervised classification task [3].

In the cancer classification task, a small number of samples are available, while each sample is described by a very large number of genes. These characteristics of the microarray data make it very likely the presence of redundant or irrelevant genes, which limit the performance of classifiers. Thus, extracting a small subset of genes containing valuable information about a given cancer is one of the principal challenges in the microarray data analysis [4].

Gene selection has become more and more indispensable over the last few years. The main motivation of this selection is to identify and select the useful genes contained in a microarray dataset for distinguishing the sample classes. It also provides a better understanding and interpretation of the phenomena studied. Also, it surpasses the curse of dimensionality in order to improve the quality of classifiers. In general, gene selection methods are divided into two subclasses: wrapper approaches and filter approaches [5]. In
In the aforementioned method (wrapper), the classification algorithm is used several times at each evaluation. Generally, the accuracy according to the final subset of genes is high because of the bias of the process of generating the classifier used. Another advantage is their conceptual simplicity: just generate and test. However, they do not have any theoretical justification for the selection and do not allow us to understand the dependency relationships that may exist between genes. On the other hand, the selection procedure is specific to a particular classifier, and the found subsets are not necessarily valid if we change the classifier. Besides, they typically suffer from a possible overfitting and high computational cost [5, 6]. Also, these approaches become unfeasible because the evaluation of large gene subsets is computationally very expensive [7]. While in filter methods, the final subset is selected based on some gene score functions and significance measures. Unlike wrappers, the selection is independent of the classifier used. The operating principle of these methods is based on the evaluation of each gene individually to assign it a score. The gene selection is performed by selecting the best-ranked genes. Filters are typically less expensive in computing time, so they can be used in the case where the number of genes is very high because of their reasonable complexity. But, the main negative point of these methods is that they do not take into consideration the possible interactions between genes. In the literature, there are several individual gene-ranking methods (filter) such as t-test [8], Fisher score [9], signal-to-noise ratio [10], information gain [7], and ReliefF [11].

In wrapper methods, metaheuristics are commonly used to generate high-quality subsets of genes. Examples of classification algorithms used for measuring the quality of each candidate solution include support vector machines (SVMs) and K nearest neighbor (KNN) [12].

The first works on the DNA microarray classification were published at the end of the 1990s [13, 14]. In this context, several researchers have utilized metaheuristic methods and the ACO algorithm for solving the feature selection problem (particularly gene selection), in order to facilitate recognition of cancer cells: ACO [15–20] algorithm, PSO [4, 6, 21–25] genetic algorithm [4, 26, 27], incorporating imperialist competition algorithm (ICA) [28], and binary differential evolution (BDE) algorithm [29].

The ant colony optimization algorithm (abbreviated as ACO) is a population-based metaheuristic [30, 31]. Thanks to its efficiency, it has been used to solve several optimization problems in different fields. In the ACO algorithm, each ant presents a candidate solution to the problem, and the ants build approximate solutions iteratively (step-by-step). The process of constructing solutions can be regarded as a path (between home and food source of ants) on a graph. The choice of the best path by ants is influenced by the quantities of pheromone left in these pathways and a piece of heuristic information that indicates the goodness of the decision taken by an ant.

Thus, metaheuristics find application in solving the gene selection problem which is known to be NP-hard [32, 33]. In the last decade, several researchers have also adopted graph-based techniques to select near-optimal subset of a feature set [34–36].

In this study, we propose a hybrid approach for solving the gene selection problem. Our two-stage proposed approach starts with a first stage in which a new graph-based approach is proposed (MWIS) without using any learning model. In the second stage, a wrapper method based on a modified ACO and a new local search algorithm guided by the 1NN classifier is developed. In this step, the role of 1NN is to evaluate each candidate gene subset generated. The proposed approach has not been previously investigated by previous researchers.

This paper is organized as follows: in Section 2, we present the proposed gene selection method. Section 3 provides a detailed exposition of the experiments that we have put on ten microarray datasets to evaluate our approach. Finally, we conclude our paper.

## 2. Methods

### 2.1. Graph Theory Approach for Gene Selection

#### 2.1.1. Notations.

In this work, we use \( X \) to denote a dataset (Table 1) of \( M \) samples \( \{x_1, x_2, \ldots, x_M\} \), \( x_i \in \mathbb{R}^N \). We use \( \{g_1, g_2, \ldots, g_N\}, g_i \in \mathbb{R}^M \) to denote the \( N \) genes vectors. \( Y = \{y_1, y_2, \ldots, y_M\} \) are the class labels.

Graph theory gives an abstract model to represent the relationships between two or more elements (vertices) into a given system. Let \( G = (V, E) \) be an undirected graph where \( V \) is a nonempty finite set called the set of vertices and \( E \) is the set of edges. We define a vertex-weighted graph \( (G, W) \) as a graph \( G \) together with a function \( W \) (the vertex weighting function) such that \( W(u) \in \mathbb{R}^+ \) for all \( u \in V \) [37, 38]. The maximum weight independent set (MWIS) is one of the most important optimization problems, thanks to their several domains of application [39], particularly, in the gene selection problem, where we can transform the DNA microarray data into a vertex-weighted graph (gene-

### Table 1: A gene expression matrix composed with \( M \) samples and \( N \) genes.

| Gene id | Gene1 | Gene2 | ⋅⋅⋅ | GeneN | Class label |
|---------|-------|-------|-----|-------|-------------|
| Sample1 | \( x_{11} \) | \( x_{12} \) | ⋅⋅⋅ | \( x_{1N} \) | \( y_1 \) |
| Sample2 | \( x_{21} \) | \( x_{22} \) | ⋅⋅⋅ | \( x_{2N} \) | \( y_2 \) |
| Sample3 | \( x_{31} \) | \( x_{32} \) | ⋅⋅⋅ | \( x_{3N} \) | \( y_3 \) |
| ⋅⋅⋅     | ⋅⋅⋅   | ⋅⋅⋅   | ⋅⋅⋅ | ⋅⋅⋅   | ⋅⋅⋅         |
| SampleM | \( x_{M1} \) | \( x_{M2} \) | ⋅⋅⋅ | \( x_{MN} \) | \( y_M \) |
similarity graph). In this graph, each gene can be considered as a vertex and their Fisher score as weight of this vertex. The set of edges represents the existence of significant correlation (relationship) between these genes; this relation is nothing but the degree of linear association (Pearson correlation) between the latter. After transforming the DNA microarray data, we try to find the maximum weight independent set. This set of genes will be used in the second stage of our proposed method.

2.2. Construction of Gene-Similarity Graph. The construction of gene-similarity graph requires the definition of some statistical notions: starting with the Fisher score to calculate the weight of each vertex (gene).

2.2.1. Fisher Score $F_i$ [9]. It is mainly applied in gene selection as a filter [40]. The Fisher score value of each gene represents its relevance to the dataset; a higher Fisher score means that the gene contributes more information. This information helps to measure the degree of separability of the classes through a given gene $g_i$. It is defined by

$$F_i = \frac{\sum_{k=1}^{c} n_k (\mu_k - \mu)^2}{\sum_{k=1}^{c} n_k (\sigma_k^2)},$$

where $c$, $n_k$, $\mu_k$, and $\sigma_k^2$ represent, respectively, the number of classes, the size of the $k$-th class, and mean and standard deviation of $k$-th class corresponding to the $i$-th gene. $\mu_i$ is the global mean of the $i$-th gene.

2.2.2. Pearson Correlation Coefficient. The Pearson correlation coefficient is a measure of the strength of the linear relationship between two variables (genes). Let $g_1$ and $g_2$ be two random variables, and the correlation coefficient between $g_1$ and $g_2$ is defined by

$$\rho_{g_1,g_2} = \frac{\text{cov}(g_1, g_2)}{\sigma_{g_1} \sigma_{g_2}},$$

where $\text{cov}(g_1, g_2)$ is the covariance between $g_1$ and $g_2$, $\sigma_{g_1}$ is the standard deviation of $g_1$, and $\sigma_{g_2}$ the standard deviation of $g_2$.

The correlation coefficient may take on a range of values from $-1$ to $1$. Let $(r_{ij} = |p_{g_ig_j}|)$ be the absolute value of the correlation between $g_i$ and $g_j$.

Now, we can define the adjacency matrix $A_G = (a_{ij})_{N \times N}$, with zeros on its diagonal to represent $(G, W)$. Where $a_{ij} = 1$ if $(i, j) \in E$ is an edge of $G$ and $a_{ij} = 0$ if $(i, j) \notin E$. More precisely, a value of 1 represents the existence of a relationship between $g_i$ (row $i$) and $g_j$ (column $j$), while a value of 0 means the nonexistence of this relationship. The creation of $A_G$ requires the definition of the absolute correlation matrix $R = (r_{ij})_{N \times N}$. Based on this matrix, we fill $A_G$; let $r_0$ be a fixed value in $[0, 1]$; we assume that if $r_{ij} > r_0$ then the mutual information between $g_i$ and $g_j$ is high (i.e., the two vertices are adjacent). More exactly, the matrix $A_G$ is filled based on the rule below: for $i \neq j$,

$$a_{ij} = \begin{cases} 1, & \text{if } r_{ij} > r_0, \\ 0, & \text{otherwise}, \end{cases}$$

where $r_0 \in [0, 1]$ is the minimum correlation value for which we consider two genes in relation. The experimental study carried out in our method proves that $r_0 = 0.35$ behaves well with the high-dimensional data. For example, if we have a dataset composed of 7 genes $\{g_1, g_2, \ldots, g_7\}$, Table 2 shows the corresponding absolute correlation matrix to these data.

For $r_0 = 0.35$, the adjacency matrix is given in Table 3. We define the weight of a vertex $i (g_i)$, by using the Fisher score: $W(i) = F_i$. A gene $i$ with a high score in the DNA microarray dataset corresponds to a vertex with a high weight in $(G, W)$. This weight gives important information about the gene relevancy to the data. Indeed, if there are two genes connected by an edge in $G$ we prefer the gene which has the best weight. On the basis of the steps defined before, we were able to transform a determined DNA data microarray into a vertex-weighted graph (Algorithm 1).

Figure 1 shows the gene-similarity graph equivalent to the adjacency matrix (Table 3); we associate to each gene (vertex) a weight by using the Fisher score.

In the context of gene selection for cancer classification, the microarray datasets are characterized by a very large number of genes. The application of an evolutionary algorithm such as ACO directly without passing by a preprocessing step is highly expansive. This is where filter methods become so useful in order to extract a subset of possibly informative genes, and then the evolutionary metaheuristic is applied to select the near-optimal subset of genes [19]. As examples, generalized Fisher score, ReliefF, and BPSO are combined in [6], an information gain filter and a memetic algorithm in [41], chi-square statistics and a GA are used in [26], information gain and improved simplified swarm optimization in [42], and ReliefF, mRMR (minimum redundancy maximum relevance), and GA in [11]. Zhao et al proposed a hybrid approach by combining the Fisher score with a GA and PSO [40]. In order to overcome the disadvantages of filter methods, we propose an efficient approach based on graph theory techniques to select the first subset. This method takes into account possible interactions between genes.

2.3. Gene Selection Based on the Maximum Weight Independent Set. Let $(G, W)$ be a vertex-weighted undirected graph, where $V$ is the set of its vertices, $E$ is the set of edges, and $W$ is the vertex weighting function. For each $v \in V$ we define $N_G(v)$ the neighborhood of $v$, i.e., $N_G(v) = \{u \in V : u \neq v, (u, v) \in E\}$. A subset $I \subseteq V$ is an independent set of $G$ if there are no two adjacent vertices in $I$ (i.e., connected by an edge). The MWIS is the independent set with the maximum weight (the weight of a subset of vertices in $V$ is defined as the sum of the weights of the vertices in this subset [43]).

We remark that in filter methods for gene selection based on the rank of genes, the correlation between the selected genes is not considered. This implies the selection of subsets with a high level of redundancy that penalizes the
classification performances; on the other hand, these methods eliminate the genes with a low individual score, ignoring the possibility that they can become highly relevant when combined with other genes [44]. This motivates us to propose a graph-based approach to overcome these problems. In the first stage of our method, we consider the gene selection problem as the search for the maximum weight independent set in the gene-similarity graph \((G, W)\).

The MWIS into a given graph is an NP-hard problem [45], and since in our case the gene-similarity graph is large (several thousands of vertices and edges), then it is impossible to find an exact solution to our problem in a reasonable time. For this, we propose a greedy algorithm (heuristic) to quickly obtain an approximate solution. The main lines of this algorithm are presented in Algorithm 2.

Table 2: Correlation (similarity) matrix.

|    | \(g_1\) | \(g_2\) | \(g_3\) | \(g_4\) | \(g_5\) | \(g_6\) | \(g_7\) |
|----|---------|---------|---------|---------|---------|---------|---------|
| \(g_1\) | 1       | 0.59    | 0.19    | 0.45    | 0.1     | 0.24    | 0.67    |
| \(g_2\) | 0.59    | 1       | 0.36    | 0.31    | 0.24    | 0.06    | 0.29    |
| \(g_3\) | 0.19    | 0.36    | 1       | 0.31    | 0.49    | 0.81    | 0.12    |
| \(g_4\) | 0.45    | 0.3     | 0.31    | 1       | 0.49    | 0.81    | 0.12    |
| \(g_5\) | 0.1     | 0.11    | 0.24    | 0.49    | 1       | 0.72    | 0.66    |
| \(g_6\) | 0.24    | 0.07    | 0.06    | 0.81    | 0.72    | 1       | 0.57    |
| \(g_7\) | 0.67    | 0.66    | 0.29    | 0.12    | 0.66    | 0.57    | 1       |

Table 3: Adjacency matrix.

|    | \(g_1\) | \(g_2\) | \(g_3\) | \(g_4\) | \(g_5\) | \(g_6\) | \(g_7\) |
|----|---------|---------|---------|---------|---------|---------|---------|
| \(g_1\) | 0       | 1       | 0       | 1       | 0       | 0       | 1       |
| \(g_2\) | 1       | 0       | 1       | 0       | 0       | 0       | 1       |
| \(g_3\) | 0       | 1       | 0       | 0       | 0       | 0       | 0       |
| \(g_4\) | 1       | 0       | 0       | 0       | 1       | 1       | 0       |
| \(g_5\) | 0       | 0       | 0       | 1       | 0       | 1       | 1       |
| \(g_6\) | 0       | 0       | 0       | 1       | 1       | 0       | 1       |
| \(g_7\) | 1       | 1       | 0       | 0       | 1       | 1       | 0       |

Algorithm 1: Construction of a gene-similarity graph.

**Input:** DNA microarray data, \(r_0\)  
**Output:** Gene-similarity graph \((G, W)\).  
**Begin**  
Calculate the weight of each gene \(W(g_i) = F_i\) by using the Fisher score (1).  
Calculate the absolute correlation matrix \(R = (r_{ij})_{N \times N}\) by using (2).  
Fill the adjacency matrix \(A_G = (a_{ij})_{N \times N}\) associated to \(G\), based on the rule (3).  
Create the gene-similarity graph \((G, W)\).  
**Return** \((G, W)\).

![Figure 1: An example of gene-similarity graph.](image-url)
2.4. Ant Colony Optimization for Gene Selection. ACO is one of the algorithms based on swarm intelligence. It was introduced as a method for solving optimization problems in the early 90s by Dorigo et al. [30, 31] and developed after in [46, 47]. Initially, ACO was designed to solve the traveling salesman problem by proposing the first ACO algorithm: “Ant System” (AS) [48]. Subsequently, other applications that were considered early in the history of ACO such as quadratic assignment [49], sum coloring [50], vehicle routing [51], constraint satisfaction [52], and gene selection [15–17, 19, 20].

The ACO algorithm is inspired by the social behavior of ants. The artificial ants used in the ACO can cooperate with each other (by exchanging information via pheromones) to solve difficult problems; this is performed by building approximate solutions iteratively (step-by-step). The feasible solutions can be regarded as a path between home and food source of ants. The method of choice of this last path is detailed in the next subsections.

2.4.1. ACO for Gene Selection. Denote the \( p \) genes as \( \{g_1, g_2, \ldots, g_p\} \) to adopt the ACO for gene selection problem, and a novel ACO is proposed; the path of each ant from the nest to food is coded as a \( p \)-dimensional binary string where each bit of the pathway is attached to a gene; the selection of the pathway “1” means that gene has been chosen. On the other hand, a pathway “0” indicates that the gene is not selected in the final subset. Suppose that \( p = 10 \), the coding of our modified ACO is presented and explained in Figure 3.

The ants seek to find the best path that maximizes the accuracy and minimizes the number of selected genes. Figure 4 describes the gene section procedure proposed on our ACO. Each ant starts from the nest to the food source with the aim to find the best path (best subset of genes). The building of this path is done step-by-step; in each step \( i \), the ant decides to add the gene \( i \) to the candidate subset of genes or not, based on the pheromone and heuristic information assigned to this gene (Figure 4). The ant terminates its tour in \( p \) steps and outputs a subset of selected genes as it reaches the food source.

As indicated previously, the task of each ant is to construct a candidate subset of genes using heuristic information and pheromone; this is performed via a probabilistic decision rule. We compute the probability of selecting a pathway as below:

\[
p_{ij} = \frac{(\alpha^p i_j)(\beta^p j_i)}{(\alpha^p i_j)(\beta^p j_i) + (\tau^0 i_j)(\eta^1 j_i)} \quad (i = 1, 2, \ldots, p; j = 1, 0),
\]

\[(4)\]
where \( i \) represents the \( i \)th gene, \( j \) takes the value 1 or 0 to denote whether the corresponding gene has been selected or not, \( \tau_{ij} \) is the pheromone intensity that indicates the importance of the selection of the \( i \)th gene, and \( \eta_{ij} \) represents the heuristic reflecting the desirability of the selection of this gene or not. \( \alpha \) and \( \beta \) are two parameters controlling the relative importance of the pheromone intensity versus visibility; with \( \alpha = 0 \), only the visibility (heuristic information) of the gene is taken into account, and the ants will decide to select or not a given gene based just on \( \eta_{ij} \). Since the previous research experience is lost, therefore there is no cooperation between ants in this case. On the contrary, with \( \beta = 0 \), only the trail pheromone trails play. To avoid too rapid convergence of the ACO algorithm, a compromise between these two parameters is necessary to ensure the diversification and intensification of the search space.

### 2.4.2. The Heuristic

The choice of a good heuristic, which will be used in combination with the pheromone information to build solutions, is an important task in the ACO implementation [53]. In our ACO, this heuristic is used to indicate the quality of a gene based on a scoring algorithm.

For a given ant, the heuristic information \( \eta_{i} \) is the desirability of adding the gene \( i \) to the subset of selected genes. We define this quantity based on the Fisher score \( F_i \) (1) which measures the quality of this gene and the number of genes selected by the ant before arriving at gene \( i \). \( \eta_{i} \) is calculated as follows:

\[
\eta_{i} = \frac{F_i}{1 + N_s}.
\]

For the value of \( \eta_{0} \), we combine the mean of the scores of Fisher of all genes and \( N_s \). This means that the ants tend to choose the small subsets of genes that have high relevance:

\[
\eta_{i0} = \frac{1}{p(1 + N_s)} \sum_{k=1}^{p} F_i.
\]

### 2.4.3. Updating the Pheromone Trail

The goal of the pheromone update is to increase the pheromone values associated with good solutions while reducing those associated with bad ones.

The updates of pheromones are made in two stages, a local update and a global update.

Once the ant \( k \) has finished the built of its path, the pheromone in all of the pathways will be updated. The updated formula is described below:

\[
\tau_{ij} \leftarrow (1 - \rho_{loc}) \tau_{ij} + \rho_{loc} \Delta \tau_{ij},
\]

where \( \rho_{loc} \) is the local pheromone evaporation coefficient parameter (0 < \( \rho_{loc} < 1 \)) which represents the evaporation of trail and \( \Delta \tau_{ij} \) is the amount of pheromone deposited by the ant \( k \); in our ACO, it is given by

\[
\Delta \tau_{ij} = \begin{cases} 
CA_{1NN}(S) \cdot \left( \frac{\lambda \cdot \# \text{genes}}{p} \right), & \text{if the ant } k \text{ uses the pathway} \\
0, & \text{otherwise},
\end{cases}
\]

where \( S \) is the candidate solution created by the ant, \( CA_{1NN} \) is the \( r \)-fold cross-validation classification accuracy of 1NN classifier (nearest neighbor) based on \( S \), \( \# \text{genes} \) is the number of selected genes in \( S \), and \( \lambda \) is a parameter that indicates the importance of the number of selected genes in \( S \) (1 ≤ \( \lambda \)).

At each iteration \( T \), after all ants finish their traverses, a global update of pheromone quantities is made for all pathways chosen by the best ant (the best candidate solution) during the iteration \( T \).

The global update is carried out as follows:

\[
\tau_{ij} \leftarrow (1 - \rho_{glob}) \tau_{ij} + \rho_{glob} \Delta \tau_{ij}(T),
\]

where \( \rho_{glob} \) is the global pheromone evaporation coefficient parameter and \( \Delta \tau_{ij}(T) \) is the amount of pheromone deposited by the best ant during the iteration \( T \) given by Chiang et al. [15].

To avoid stagnation of the search, the range of possible pheromone trails is limited to an interval \([\tau_{min} \tau_{max}]\).

### 2.5. Fitness Function

In order to guide our novel ACO towards a high-quality subset of genes, we need to define a “fitness function” \( f \). The quality of a candidate subset can be measured by combining the number of genes into this subset (size) and the classification accuracy using a specific classifier, and in gene selection the aim is to maximize the accuracy and minimize the number of genes used.

The estimation of the classification accuracy is measured by a given classifier using the cross-validation rule. In this study, we use the K-nearest neighbor classifier (KNN).

#### 2.5.1. K-Nearest Neighbor (KNN)

The KNN method is a supervised learning algorithm and was introduced by Fix and Hodges in 1951 [54]. It is based on the notion of proximity (neighbor) between samples for making a decision (classification) [55].

In order to determine the class of a new example, we calculate the distance between the new one and all testing data, and finally the classification is given by a majority vote of its \( K \) neighbors. The neighbors are determined by the Euclidean distance which is defined as follows:
In our proposed method, we use the 1NN classifier, which is a particular case of KNN (with \( K = 1 \)). Let \( X \) be a new sample to classify and \( T \) a sample from the training data, then the class of \( X \) is determined as follows:

\[
\text{Class}(X) = \text{Class}(\arg \min (D(X, T))).
\] (11)

Note that, the genes into gene expression data had different scales, and the KNN classifier is influenced by the measure of distances between samples. Therefore, we modify our 1NN by normalizing the training data to transform them to a common scale. This transformation is carried out based on the mean and the standard deviation of each gene, and the latter values are used for the scaling of the test data.

### 2.5.2. Objective Function.

The fitness value of a candidate solution \( S \) in our ACO is calculated as follows:

\[
f(S) = \omega_1 \cdot \text{CA}_{1\text{NN}}(S) + (1 - \omega_1) \cdot \left(1 - \frac{\#\text{genes}}{p}\right),
\] (12)

where \( \omega_1 \) is a weight coefficient in \([0, 1]\) that controls the aggregation of both objectives (maximizing the predictive accuracy and minimizing the number of genes), \( \#\text{genes} \) is the number of selected genes in \( S \), and \( p \) is the total number of genes.

Mention that "\( \text{CA}_{1\text{NN}} \)" is nothing but the average cross-validation classification accuracy calculated by the KNN classifier, using leave-one-out cross-validation (LOOCV) [56], in which we divide our dataset into \( M \) nonoverlapping subsets (\( M \) tissue samples). At each iteration, we train our KNN classifier on \((M-1)\) samples based on the selected genes, and we test it on the remaining sample. The "\( \text{CA}_{1\text{NN}} \)" associated to LOOCV is calculated based on the rule below:

\[
\text{CA}_{1\text{NN}} = \frac{\text{the number of correctly predicted samples}}{M}
\] (13)

### 2.6. Local Search.

The local search algorithm is used to improve the solutions given by ants and provide good solutions within a reasonable time. With this aim, we are inspired by the framework proposed in [57], in which a local search based on the filter ranking method is used to solve the feature selection problem.

Given a candidate solution generated by an ant, we define \( X \) and \( Y \) as the subset of selected and eliminated genes, and \( X \) and \( Y \) both are ranked using Fisher score, respectively. We further define two basic operators of the local search algorithm:

(i) **Add**: select gene from \( Y \) based on its ranking and add it to \( S \)

(ii) **Del**: select gene from \( X \) based on its ranking and remove it from \( S \)

The selection of the gene \( i \) from \( Y \) to move it to \( S \) by **Add** operator in our proposed method is based on the Roulette wheel developed by Holland [58]. Let \( Y = \{g_1, g_2, \ldots, g_n\} \) and \( \{F_1, F_2, \ldots, F_n\} \) be its Fisher score values. Then the selection probability \( P_i \) for gene \( g_i \) is defined as follows:

\[
P_i = \frac{F_i}{\sum_{j=1}^{n} F_j}
\] (14)

Similarly, for the operator **Del**, we define the probability of selecting a gene \( g_j \) of \( X = \{g_1, g_2, \ldots, g_n\} \) to remove it from \( S \) with a probability defined by:

\[
P_i = \frac{\bar{F}_i}{\sum_{j=1}^{n} \bar{F}_j}
\] (15)

where \( \bar{F}_j = \max(F_1, \ldots, F_n) - F_j \), for \( j = 1, \ldots, n_2 \), and \( \{F_1; F_2, \ldots, F_n\} \) are the Fisher score values of \( \{g_1, g_2, \ldots, g_n\} \).

Based on the probabilities defined before, we can remark that **Add** operator prefers the genes with the high score to add to \( S \), on the other hand, **Del** operator prefers the genes with the low score to remove from \( S \).

Our local search algorithm (Algorithm 3) is characterized by the number maximal of **Add** \( n_{\text{add}} \) and **Del** \( n_{\text{del}} \) operations, and \( h_{\text{max}} \) the maximal number of consecutive iterations without improvement in the best solution. In addition, this local search algorithm is general and efficient, for example, if we fix \( n_{\text{add}} \) at 0, the local search algorithm becomes a **backward** generation, in which we try to remove the not relevant genes at each iteration.

### 2.7. Proposed Method for Gene Selection (MWIS-ACO-LS).

Our hybrid method for solving the gene selection problem is based on combining filter and wrapper approaches. This is carried out taking advantage of the low computing time in filters (MWIS) and the high quality of the subsets provided by the wrapper methods (ACO and LS). The overall process of MWIS-ACO-LS can be seen in Figure 5.

The process begins by transforming the initial dataset into a vertex-weighted graph (Algorithm 1), where we search the MWIS, which is well-known as an NP-hard problem, so we have proposed a greedy algorithm (Algorithm 2) to find a near-optimal set of vertices (representing genes in our problem). The subset of genes selected in the later stage is taken as input into the second stage of selection, which used an evolutionary algorithm (ACO), combined with a local search algorithm to select the minimum number of genes that gives the maximum classification accuracy for the INN classifier. In this stage, artificial ants cooperate to build a high-quality subset of genes based on the transition rules already presented in Section 2. Also, a local search (Algorithm 3) is proposed to help the ants to achieve good results in a reasonable time. The pseudocode of our proposed method is presented as follows.

### 2.8. Complexity Analysis of MWIS-ACO-LS.

Suppose that \( N \) is the number of the original genes and \( M \) is the number of samples. Our method is divided into three principal stages:
Stage 1: In the first step (Algorithm 1), the weight values of the genes are evaluated using the Fisher score, thus the time complexity is $O(NM)$. Moreover, the absolute correlation values between each pair of genes are computed, so the time complexity is $O(N^2M)$. And finally, for the filling of the adjacency matrix $A_G$ (implicitly the construction of gene-similarity graph), the time complexity is $O(N^2)$. In the second step of this stage (Algorithm 2), the weight of each vertex is already defined, and then we

**Algorithm 3: Local search algorithm for gene selection.**

**Stage 1: MWIS**

- Load the dataset
- Score each gene based on a Fisher score
- Select a first subset of genes based on a greedy algorithm (Algorithm 2)
- Create a new dataset using the selected genes

**Stage 2: ACO and LS**

- Setup the ACO and the LS parameters
- Initialize pheromone matrix
- Each ant constructs it solution based on (4)
- Store the local and the global best solution
- Update solution based on the new pheromone matrix
- Local search
- Update pheromone matrix

- Termination criteria?

**Flowchart**

*Figure 5: Flowchart of our proposed approach for gene subset selection in DNA microarray data.*
can conservatively assume that in each iteration we remove only the vertex itself to get a time complexity of $O(N)$. Therefore, the overall time complexity of this stage (MWIS) is $O(NM + N^2M + N^2 + N) = O(N^2M)$.

Stage 2: First, we mention that $p$ represents the number of selected genes in the first stage; generally $(p \ll N)$.

In this stage, the fitness of each candidate subset of genes is calculated using LOOCV (leave-one-out-cross-validation) and the 1NN as a classifier equation (13). Let us analyze now the complexity of fitness calculation using 1NN (LOOCV); we compute the distance between the single sample of the testing set and each training set sample, requiring $O(p(M−1))$, this process is repeated $M$ times, so the fitness calculation need $O(p(M−1)M)$.

In each iteration of our ACO, each ant from the $m$ ants starts from $g_0$ to $g_p$ passing by all $p$ genes; then, the construction of a path by an ant as $O(p)$, and each path is evaluated by LOOCV. This process is repeated $n_{\text{max}}$ times by the $m$ ants, in addition, updating the pheromone values has $O(pn_{\text{max}})$; therefore, the overall computational complexity of ACO without local search is $O(pn_{\text{max}} + n_{\text{max}}mp(M − 1)M) = O(n_{\text{max}}mp^2(M − 1)M)$.

Concerning the local search algorithm, for the LS used in the second stage (line 20 Algorithm 4) repeated $n_{\text{max}}$ times, the complexity time is $O((n_{\text{add}} + n_{\text{del}})n_{\text{max}}p(M − 1)M)$.

Generally, $(n_{\text{add}} + n_{\text{del}})n_{\text{max}} = m$, so the total complexity time of the second stage is $O(n_{\text{max}}mp^2(M − 1)M)$.

Stage 3: For the last stage (line 24 Algorithm 4), the complexity of the backward generation is $O(it_{\text{max}}n_{\text{del}}p(M − 1)M)$.

Consequently, the total time complexity of the proposed method MWIS-ACO-LS is $O(it_{\text{max}}n_{\text{del}}p(M − 1)M + n_{\text{max}}mp^2(M − 1)M + N^2M) = O(n_{\text{max}}mp^2(M − 1)M + N^2M)$.

3. Experimental Studies

This section presents the performance of our proposed approach (MWIS-ACO-LS) on ten well-known gene expression classification datasets, and we compare our results with those of the state-of-the-art. Furthermore, the characteristics of the used datasets, the parameter settings, and the numerical results will be described in the following sections.

The implementation of the proposed approach (MWIS-ACO-LS) is performed using Matlab R2017a.

As far as the KNN classifier is concerned, we have chosen a predefined function in Matlab. Similarly for the SVM classifier[59, 60] used in the comparison a predefined binary linear classifier was chosen. In addition, we have developed a multiclass SVM classifier based on the one-against-all strategy.

Concerning the logistic regression (LR) classifier we have regularized the cost function by two penalties, the first is lasso ($L_1$) and the second is the elastic net regularization. The minimizing the cost functions used on LR – $L_1$ and LR-Elasticnet is assured by the stochastic gradient descent (SGD) algorithm implemented in the Scikit-learn package [61]. Experimental initial parameters are given in Table 4.

Additionally, in this study, we use leave-one-out-cross-validation (LOOCV) to measure the quality of the candidate subsets of genes and for comparing our results with the other works.

3.1. Environment. To evaluate our approach, we have chosen ten datasets (DNA microarray) concerning the recognition of cancers [62], which are publicly available and easily accessible. In addition, these datasets are used in several supervised classification works, particularly in the papers using in the “Comparison with state-of-the-art algorithms” section.

All datasets used are described in Table 5. The latter datasets have a multitude of distinguishing characteristics (number of genes, number of samples, and binary classes or multiclass). The number of samples in some datasets is small (Brain_Tumor2, 9_Tumors, etc.), while others have a higher number (Lung_Cancer, 11_Tumors, etc.). Also, some of them have binary classes (Prostate_Tumor, DLBCL) while others have multiclass (Leukemia1, Lung_Cancer, etc.). And as our proposed method is designed for the high-dimensional microarrays, all these datasets are characterized by thousands of genes ranging from 2308 to 12600.

3.2. Parameters. We note that our approaches have been run on an Acer Aspire 7750g laptop with Intel Core I5 2.30 GHz processor and 8 GB RAM, under system running Windows 7 (64 bit).

Several tests were carried out in order to obtain an appropriate parameterization; indeed, a set of initial values for the parameters were fixed, and then we change the value of one parameter for different runs until the solutions could not be ameliorated. The process of adjustment was repeated for each parameter until the solutions could not be improved. This process is carried out based on one dataset of cancer classification. Table 5 represents the parameters of the proposed approach.

3.3. Results and Comparisons. Firstly, in order to limit the search space and accelerate the speed of convergence of our proposed approach, the first subset of genes was selected based on a graph-theory algorithm for gene selection (MWIS), and then a modified ACO-1NN coupled with a local search algorithm was applied to find more excellent subset of genes. The quality of a candidate subset is measured by the performance of the KNN classifier obtained using LOOCV and the size of this subset.
Input: DNA microarray data; $n_{\text{max}}$; $m$; $\alpha$; $\beta$; $\rho_{\text{loc}}$; $\rho_{\text{glob}}$; $\lambda$; $\tau_{\text{ini}}$; $\tau_{\text{min}}$; $\tau_{\text{max}}$; $\omega_1$.
Output: The global best candidate solution $S_{\text{gbest}}$.

Begin

Stage 1: The selection of the first subset of gene
3:  Step 1: Use the Algorithm 1 to construct the gene-similarity graph.
   Step 2: Apply the greedy algorithm (Algorithm 1) to select an initial subset of genes.

Stage 2: The application of ACO to the subset of gene selected in the first stage
6:  Step 1: ACO combined with the local search
   Initialize the pheromone matrix by ones.
   for $T = 1; T < n_{\text{max}}; T + +$ do
      9:   for $i = 1; i < m; i + +$ do
            build the path (candidate solution $S$) of the ant based on the probabilistic decision rule defined by (4), (5) and (6).
            Calculate the fitness of the candidate solution using LOOCV in (11).
         12:   if $i == 1$ then
                $S_{\text{best}} = S$
         end if
      15:   if $f(S_{\text{best}}) \leq f(S)$ then
                $S_{\text{best}} = S$
         end if
   18:   Do a local update of pheromones based on $S$.
   end for
   Apply the Local search (Algorithm 3) to $S_{\text{best}}$.
21:   Do a global update of pheromones based on $S_{\text{best}}$.
   end for
24:  Step 2: Apply a backward generation to $S_{\text{gbest}}$.
   Return $S_{\text{gbest}}$.

Algorithm 4: Proposed approach (MWIS-ACO-LS).

| Table 4: Parameters used for experiments (common parameters for MWIS-ACO-LS). |
|-----------------------------------------------|------------------------------|
| Common parameters for MWIS                  | Value                        |
| $r_0$                                        | The correlation for crating a vertex | 0.35 |
| Common parameters for ACO                   |                              |
| $m$                                          | Population size | 30 |
| $n_{\text{max}}$                             | The number of iteration | 100 |
| $\alpha$                                    | Influence of the pheromone | 1.2 |
| $\beta$                                     | Heuristic information | 0.2 |
| $\rho_{\text{loc}}$                          | Local evaporation of pheromone | 0.002 |
| $\rho_{\text{glob}}$                         | Local evaporation of pheromone | 0.06 |
| $\lambda$                                   | Factor of updating pheromone | 1.6 |
| $\tau_{\text{ini}}$                          | The initial pheromone of pathway 0 | 1.0 |
| $\tau_{\text{ini}}$                          | The initial pheromone of pathway 1 | 1.0 |
| $\tau_{\text{min}}$                          | The lower pheromone | 0.05 |
| $\tau_{\text{max}}$                          | The upper pheromone | 1.4 |
| $\omega_1$                                   | The weight coefficient in the fitness function | 0.99 |
| Common parameter for the local search algorithm |                               |
| $n_{\text{max}}$                             | The maximal number of iteration without improvement | 5 |
| $n_{\text{add}}$                             | The maximal number of Add operations | 3 |
| $n_{\text{del}}$                             | The maximal number of Del operations | 5 |
| Common parameter for the backward generation algorithm |                          |
| $n_{\text{max}}$                             | The maximal number of iteration without improvement | 20 |
| $n_{\text{add}}$                             | The maximal number of Add operations | 0 |
| $n_{\text{del}}$                             | The maximal number of Del operations | 2 |
| Common parameter for the KNN classifier      |                               |
| $K$                                          | The number of neighbors | 1 |
| The distance used                            | The Euclidean distance |
| Common parameters for the logistic regression model (LR) |                        |
| $\alpha_{\text{LR}}$                         | The regularization term constant | 0.0001 |
| $\eta_{\text{ratio}}$                        | The weight given to $L_1$ into the Elasticnet | 0.5 |
The objectives of the experiments carried out on the ten datasets of (DNA microarray) are as follows: to test the effect of gene selection on the improvement of the classification accuracy and to validate the proposed method and verify its effectiveness.

Given the nondeterministic nature of our approach and the SGDlogistic classifier, ten independent runs were performed for each dataset to obtain a more reliable result.

Table 6 shows the results obtained using a new graph-based approach (MWIS) for gene selection, and then, using the MWIS-ACO method where we apply to the subset selected by the MWIS and ACO algorithm, and finally using our new improved method MWIS-ACO-LS, where the ACO is coupled with the local search (LS) method. The classification accuracy in MWIS, MWIS-ACO, and MWIS-ACO-LS is calculated using the 1NN classifier, on the other hand, these methods are compared with SVM, 1NN, and SGDlogistic penalized classifiers without selection to demonstrate the usefulness of our selection approach. We analyze our results in three ways:

(i) The classification accuracy
(ii) The number of genes used in the classification
(iii) The execution time

4. Discussion

First of all, we start by the execution time analysis of our proposed methods. We can remark that the execution time is appropriate to the complexity analysis; in the filter-based approach MWIS, the execution time is low, but the accuracy is not good since the selection is independent to the classifier. While in MWIS-ACO and MWIS-ACO-LS the execution time is important because of the nature of the wrappers method used and the use of the 1NN classifier, on the other hand, these methods are compared with SVM, 1NN, and SGDlogistic penalized classifiers without selection to demonstrate the usefulness of our selection approach. We analyze our results in three ways:

(i) The classification accuracy
(ii) The number of genes used in the classification
(iii) The execution time

4. Discussion

First of all, we start by the execution time analysis of our proposed methods. We can remark that the execution time is appropriate to the complexity analysis; in the filter-based approach MWIS, the execution time is low, but the accuracy is not good since the selection is independent to the classifier. While in MWIS-ACO and MWIS-ACO-LS the execution time is important because of the nature of the wrappers method used and the use of the 1NN classifier at each evaluation, but the classification accuracy is high. Now passing to the analysis of the different stages of our proposed method MWIS-ACO-LS (from Figures 6 and 7 and Table 6), we can remark that the role of the ACO is to improve the classification accuracy and reduce the number of genes used. In addition, the local search has a primordial role in the refinement of the candidate solutions provided by the ants by reducing the number of genes, while retaining the classification accuracy proved by ants.

The proposed approach (MWIS-ACO-LS) derives its effectiveness from the remarkable improvement in the classification accuracy and the reduction of the number of the genes used in the classification (shown in bold in Table 6), in all datasets (Figure 8).

The “MWIS,” “MWIS-ACO,” and “MWIS-ACO-LS” methods select a reduced subset of informative genes compared to the original subset of genes in the datasets.

From Table 6 and Figure 8, it can be observed that MWIS overcomes the results obtained by the 1NN classifier for “9_Tumors,” “Lung_Cancer,” “SRBCT,” and “DLBCL” datasets which is amazing because the role of the MWIS algorithm is just to find a candidate subset of genes to apply our modified ACO. That subset can contain weak genes and the process of the selection in this method is independent of the classifier used.

Based on the experiments and the application of our approach on ten dataset concern the cancer recognition, we can observe that the proposed method (MWIS-ACO-LS) outperforms all four algorithms in terms of classification accuracy and the number of genes used in the classification. The improvement in performance is more significant for the 9_Tumors; we are passed from a classification accuracy less than 60% to a perfect classification using just 40. So, we can conclude (from Table 6 and Figure 8) that MWIS-ACO-LS can successfully select a small subset of genes which can obtain a high classification accuracy. For all datasets, the “MWIS-ACO-LS” approach has reached a great classification accuracy, more exactly, a classification greater than 99.42% using only a small subset of genes from the original genes. In addition, MWIS-ACO-LS gave a perfect accuracy of 100% for the majority of datasets: (9_Tumors, Brain_Tumor1, Brain_Tumor2, Leukemia1, Leukemia2, SRBCT, Prostate_Tumor, and DLBCL) using just 5 genes for Leukemia1, and 6 genes for SRBCT and DLBCL dataset.

However, with regard to the MWIS approach based on some graph theory principles, we remark that the subset of genes selected by this method gives acceptable classification accuracy according to the number of genes used. This goes...
In this second stage, our algorithm tries to find the smallest subset of genes that gives the highest classification accuracy. Note: the best results are shown in bold. Remark: as the SVM, INN, and MWIS are of deterministic nature, the classification is calculated just in one run.

Table 6: Comparison of SVM, INN, MWIS-1NN, and MWIS-ACO-LS (LOOCV).

| Datasets       | Performance | SVM | LR-L_1 | Avg | Best | LR-Elasticnet | INN | MWIS | MWIS-ACO | MWIS-ACO-LS | Best |
|---------------|-------------|-----|--------|-----|------|---------------|-----|------|----------|-------------|------|
| 11_Tumors     | Accuracy (%)| 85.63 | 88.22 | 93.1 | 86.38 | 88.22 | 74.14 | 67.24 | 94.90 | 96 | 99.14 | 99.42 |
| Genes         | 12533       | —    | —      | —    | —    | —    | 12533 | 1308 | 463.00 | 460 | 166.9 | 101 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.82  | 91.33 | —    | —    | 123.2 | —    |
| 9_Tumors      | Accuracy (%)| 33.38 | 35.00 | 50.00 | 29.50 | 38.38 | 40.00 | 60.00 | 98.83 | 100 | 100.00 | 100.00 |
| Genes         | 5726        | —    | —      | —    | —    | —    | 5726  | 263  | 90.10 | 83 | 51 | 40 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.34  | 21.3 | —    | —    | 34.48 | —    |
| Brain_Tumor1  | Accuracy (%)| 88.89 | 85.67 | 88.89 | 85.44 | 88.89 | 85.56 | 80.00 | 96.56 | 100 | 99.22 | 100.00 |
| Genes         | 5920        | —    | —      | —    | —    | —    | 5920  | 246  | 55.90 | 46 | 22.9 | 19 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.22  | 29.14 | —    | —    | 45.81 | —    |
| Brain_Tumor2  | Accuracy (%)| 70.00 | 27.20 | 32.00 | 29.20 | 36.00 | 60.00 | 48.00 | 95.40 | 100 | 99.40 | 100.00 |
| Genes         | 10367       | —    | —      | —    | —    | —    | 10367 | 110  | 22.40 | 18 | 11.1 | 11 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.55  | 17.89 | —    | —    | 27.13 | —    |
| Leukemia1     | Accuracy (%)| 97.22 | 91.81 | 94.44 | 92.64 | 95.83 | 83.33 | 66.67 | 100.00 | 100.00 | 100.00 | 100.00 |
| Genes         | 5327        | —    | —      | —    | —    | —    | 5327  | 297  | 63.00 | 56 | 9.4 | 5 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.26  | 25.94 | —    | —    | 43.77 | —    |
| Leukemia2     | Accuracy (%)| 97.22 | 91.81 | 95.83 | 91.67 | 95.83 | 83.61 | 73.61 | 100.00 | 100.00 | 100.00 | 100.00 |
| Genes         | 11225       | —    | —      | —    | —    | —    | 11225 | 203  | 45.80 | 42 | 13.9 | 11 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.44  | 23.85 | —    | —    | 37.67 | —    |
| Lung_Cancer   | Accuracy (%)| 95.07 | 91.67 | 94.10 | 91.03 | 93.60 | 87.68 | 90.15 | 98.42 | 99.01 | 98.92 | 99.51 |
| Genes         | 12600       | —    | —      | —    | —    | —    | 12600 | 602  | 180.00 | 183 | 34.8 | 36 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.82  | 74.61 | —    | —    | 107.3 | —    |
| SRBCT         | Accuracy (%)| 100.00 | 97.59 | 100.00 | 97.10 | 98.79 | 85.54 | 91.57 | 100.00 | 100.00 | 100.00 | 100.00 |
| Genes         | 2308        | —    | —      | —    | —    | —    | 2308  | 109  | 15.60 | 15 | 7.6 | 6 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.14  | 23.48 | —    | —    | 38.24 | —    |
| Prostate_Tumor| Accuracy (%)| 92.16 | 81.47 | 88.23 | 80.69 | 83.33 | 82.35 | 79.41 | 98.24 | 99.04 | 99.12 | 100.00 |
| Genes         | 10509       | —    | —      | —    | —    | —    | 10509 | 193  | 47.30 | 43 | 20.3 | 21 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.39  | 29.2 | —    | —    | 46.59 | —    |
| DLBCL         | Accuracy (%)| 97.40 | 91.95 | 97.40 | 91.43 | 94.80 | 84.42 | 87.01 | 100.00 | 100.00 | 100.00 | 100.00 |
| Genes         | 5469        | —    | —      | —    | —    | —    | 5469  | 147  | 25.7 | 22 | 7.2 | 6 |
| Time (min)    | —           | —    | —      | —    | —    | —    | 0.17  | 27.18 | —    | —    | 43.63 | —    |

Note: the best results are shown in bold. Remark: as the SVM, INN, and MWIS are of deterministic nature, the classification is calculated just in one run.

In Figures 9–12, the abscissa axis expresses the number of generations in the second stage of MWIS-ACO-LS, and the ordinate axis expresses the classification accuracy of the best candidate solution during each iteration. This is done for the average of all solutions and the best solution found for the datasets “9_Tumors, Brain_Tumor1, Brain_Tumor2, and Leukemia1.” These figures clearly show that the use of our modified ACO and the local search algorithm play a crucial role in the amelioration of the classification accuracy. As we can remark in these figures the difference between the best solution and the average solution is not great. Therefore, MWIS-ACO-LS possesses a faster convergence speed and achieves the optimal solution rapidly.

In Figures 13–16, we show the evolution of the number of genes selected on the ordinate axis relative to the number of generations (the abscissa axis) for the “9_Tumors, Brain_Tumor1, Brain_Tumor2, and Leukemia1” datasets. These figures illustrate the role of our wrapper algorithm based on ACO in reducing the number of genes. Moreover, the second stage of our proposed approach based on the modified ACO and the local search algorithm plays a key role in increasing the classification accuracy and minimizing the number of genes used. Indeed, the ACO aims to identify the near-optimal subset of candidate genes, called the best ant, during each
iteration that maximizes the objective function, and once
the subset in question is found, our local search algorithm
is called to ameliorate the accuracy or reduce the number
of genes used while retaining the classification accuracy
found previously. After 100 generations of the ACO
algorithm, we apply a backward local search algorithm to
reduce the number of genes used in the last found best
solution (Figures 13–16). Thereafter, statistical analysis
has been performed using the Kruskal–Wallis statistical
test to evaluate our results and test the significance of the
difference in the results (accuracy) obtained by our
approach.

The Kruskal–Wallis statistical test presented in Figures
17 and 18 shows a comparison of the results obtained by
MWIS, MWIS-ACO, MWIS-ACO-LS, and 1NN classifier.
According to these figures, the performance of MWIS-ACO
and MWIS-ACO-LS approaches exceeds that of the MWIS
method and the 1NN classifier. In terms of the statistical
significance of the results (classification accuracy), the said
test proves that the difference between the classification
accuracy in ("1NN," "MWIS") and "MWIS-ACO-LS" is
statistically significant (remarkable).

Table 7 lists the best subset of genes selected by our
proposed approach (MWIS-ACO-LS) for the datasets in
which MWIS-ACO-LS gives the best performances com-
pared to the other works (9_Tumors, Brain_Tumor1,
Brain_Tumor2, and Prostate_Tumor). These genes are po-
tential biomarkers in cancer identification.

Based on the experiments we carried out, we can con-
clude that our approach of gene selection (MWIS-ACO-LS)
is well-founded. Indeed, of the ten datasets used, our method
has achieved a high classification accuracy. More exactly,
the proposed method yielded a classification accuracy equal to
or greater than 99.42% for all datasets, with a perfect
classification (100%) for 9_Tumors, Brain_Tumor1,
Brain_Tumor2, Leukemia1, Leukemia2, SRBCT, Prosta-
te_Tumor, and DLBCL using less than 40 genes. The high
classification accuracy found by our proposed methodology
returns to two elements: the first is the combination of
a method of the graph theory (MWIS) and the ACO met-
aheuristic, and the second is the use of a modified 1NN
classifier where we normalize the training data in order to
transform it to a common scale.

In the following, we do a comparison between our
proposed method and some recent optimization algorithms
using several classification datasets.

4.1. Comparison with State-of-the-Art Algorithms. In this
section, we compare our method with eight recently referred
algorithms in the literature [6, 21–25, 28]. And to make sense
of this comparison, the experiments are performed under
the same conditions in each algorithm. Specifically, our
approach is executed ten times on each dataset, and then we
choose the average and the best subset of genes found. We
indicate that the papers [22, 23] report just the best results
found.

Table 8 summarizes the classification accuracy and the
number of selected genes (taken from the original papers)
for the different approaches. The (−) symbol indicates that
the result is not reported in the related work. We remark
that the results obtained by our approach are very en-
couraging compared to previous work. Indeed, for most of
the datasets examined, the classification accuracies ob-
tained by the proposed gene selection method matched or
outperformed those obtained using other methods [6, 21–25, 28].

First, for the dataset (9_Tumors) we achieve a perfect accuracy classification with only 40 genes. We find that the best performance for this dataset is attained by our approach (MWIS-ACO-LS), exceeding the best-known result by 5% in the accuracy [6]. We note that the number of genes reported in the FBPSO-SVM [6] is 71 genes to have a good accuracy.

Similarly, for the datasets (11_Tumors, Brain_Tumor1, Brain_Tumor2, and Prostate_Tumor), we get the best performance. In addition, we have a perfect classification (100%) for (Brain_Tumor1, Brain_Tumor2, and Prostate_Tumor) with less than 21 genes.

Table 9 reports the rank of the proposed method compared to other existing methods according to the average accuracy. The results mentioned in the table show that the proposed method has achieved the best average accuracy in most datasets. Indeed, we clearly see that our method is more suitable for gene selection. As shown in Tables 8 and 9, we match or exceed the
Figure 11: Comparison of the evolution of the classification accuracy for "Brain_Tumor2."

Figure 12: Comparison of the evolution of the classification accuracy for "Leukimia1."

Figure 13: Comparison of the evolution of the number of genes used for "9_Tumors."

Figure 14: Comparison of the evolution of the number of genes used for "Brain_Tumor1."

Figure 15: Comparison of the evolution of the number of genes used for "Brain_Tumor2."

Figure 16: Comparison of the evolution of the number of genes used for "Leukimia1."
performance of all comparison methods; except for the Brain_Tumor2 and Prostate_Tumor datasets, in which our approach comes in the second rank after the FBPSO-SVM approach.

The results of this comparative analysis with previous methods for the gene selection in the context of cancer classification have enabled us to conclude that our nature-inspired optimization method is useful in the gene selection problem.

5. Conclusion

In this work, we have presented a hybrid approach (MWIS-ACO-LS) for the gene selection in DNA microarray data. The two-stage proposed approach consists of a preselection phase carried out using a new graph-theoretic approach to select first a small subset of genes; in this stage, we model the gene selection problem as an MWIS problem, and we
Table 8: A comparison between our method (MWIS-ACO-LS) and methods of state-of-the-art.

| Datasets         | Method          | MWIS-ACO-LS | RBPSO-1NN | FBPSO-SVM | FRBPSO | HICATS | EPSO | TS-BPSO | IBPSO | IBPSO |
|------------------|-----------------|-------------|-----------|-----------|---------|--------|------|---------|-------|-------|
|                  | Performances    | This work   | [6]       | [6]       | [21]    | [28]   | [25] | [23]    | [22]  | [24]  |
| 11_Tumors        | Best #Acc (%)   | 99.42       | —         | —         | 97.70   | 96.55  | 97.35| 93.10   | 95.4  |       |
|                  | Best #Genes     | 101         | —         | —         | 287     | 243    | 3206 | 2948    | 228   |       |
|                  | Average #Acc (%)| 99.14 <1>  | —         | —         | 95.86   | 95.40  | —    | 95.06   |       |       |
|                  | Average #Genes  | 166.9       | —         | —         | 307.5   | 237.70 | —    | 240.9   |       |       |
| 9_Tumors         | Best #Acc (%)   | 100.00      | 83.33     | 95.00     | —       | 83.33  | 76.67| 81.63   | 78.33 | 78.33 |
|                  | Best #Genes     | 40          | 20        | 71        | —       | 259    | 251  | 2948    | 248   |       |
|                  | Average #Acc (%)| 100.00 <1> | 81.83     | 92.222    | —       | 78.33  | 75.00| —       | 75.5  |       |
|                  | Average #Genes  | 51          | 29.1      | 45        | —       | 248.5  | 247.10| —       | 240.6 |       |
| Brain_Tumor1     | Best #Acc (%)   | 100.00      | 94.44     | 97.78     | —       | 94.44  | 93.33| 95.86   | 94.44 | 93.33 |
|                  | Best #Genes     | 19          | 11        | 21        | —       | 6      | 8    | 2913    | 754   | 5     |
|                  | Average #Acc (%)| 99.22 <1>  | 94.00     | 97.22     | 90.67   | 93.10  | 92.11| —       | 92.56 |       |
|                  | Average #Genes  | 22.9        | 24.7      | 22.4      | 803     | 8.9    | 7.5  | —       | 11.2  |       |
| Brain_Tumor2     | Best #Acc (%)   | 100.00      | 96.00     | 100.00    | —       | 94.00  | 94.00| 92.65   | 94.00 |       |
|                  | Best #Genes     | 11          | 15        | 12        | —       | 3      | 4    | 5086    | 1197  |       |
|                  | Average #Acc (%)| 99.40 <2>  | 92.80     | 100.00    | 87.6    | 92.60  | 92.4  | —       | 91.00 |       |
|                  | Average #Genes  | 11.1        | 24.5      | 14.3      | 662     | 8.9    | 6.0  | —       | 6.4   |       |
| Leukemia1        | Best #Acc (%)   | 100.00      | 100.00    | 100.00    | —       | 100.00 | 100.00| 100.00  | 100.00| 100.00 |
|                  | Best #Genes     | 5           | 8         | 6         | —       | 3      | 2    | 2577    | 1034  | 2     |
|                  | Average #Acc (%)| 100.00 <1> | 99.72     | 100.00    | 98.89   | 100.00 | 100.00| —       | 100.00|       |
|                  | Average #Genes  | 9.4         | 11.7      | 8.4       | 825     | 3      | 3.2  | —       | 3.2   |       |
| Leukemia2        | Best #Acc (%)   | 100.00      | 100.00    | 100.00    | —       | 100.00 | 100.00| 100.00  | 100.00| 100.00 |
|                  | Best #Genes     | 11          | 5         | 6         | —       | 5      | 4    | 5609    | 1292  | 4     |
|                  | Average #Acc (%)| 100.00 <1> | 100.00    | 100.00    | 97.50   | 100.00 | 100.00| —       | 100.00|       |
|                  | Average #Genes  | 13.9        | 13.1      | 8.6       | 1028    | 6.8   | 6.8  | —       | 6.7   |       |
| Lung_Cancer      | Best #Acc (%)   | 99.51       | —         | —         | —       | 97.04  | 96.86| 99.52   | 96.55 | 96.55 |
|                  | Best #Genes     | 36          | —         | —         | —       | 7      | 7    | 6958    | 1897  | 10    |
|                  | Average #Acc (%)| 98.92 <1>  | —         | —         | —       | 96.16  | 95.67| —       | 95.86 |       |
|                  | Average #Genes  | 34.8        | —         | —         | —       | 7.8    | 8.3  | —       | 14.9  |       |
| SRBCT            | Best #Acc (%)   | 100.00      | 100.00    | 100.00    | —       | 100.00 | 100.00| 100.00  | 100.00| 100.00 |
|                  | Best #Genes     | 6           | 7         | 10        | —       | 9      | 7    | 1084    | 431   | 6     |
|                  | Average #Acc (%)| 100.00 <1> | 99.02     | 100.00    | —       | 98.04  | 99.02| 95.45   | 92.61 | 98.04 |
|                  | Average #Genes  | 7.6         | 11.7      | 12.4      | 213     | 11.7  | 14.90| —       | 17.5  |       |
| Prostate_Tumor   | Best #Acc (%)   | 100.00      | 99.02     | 100.00    | —       | 98.04  | 99.02| 95.45   | 92.61 | 98.04 |
|                  | Best #Genes     | 20          | 9         | 6         | —       | 5      | 5    | 5320    | 1294  | 7     |
|                  | Average #Acc (%)| 99.12 <2>  | 98.24     | 100.00    | 92.43   | 97.75  | 97.84| —       | 97.94 |       |
|                  | Average #Genes  | 20.3        | 11.2      | 8.3       | 418     | 7.2   | 6.6  | —       | 13.6  |       |
| DLBCL            | Best #Acc (%)   | 100.00      | 100.00    | 100.00    | —       | 100.00 | 100.00| 100.00  | 100.00| 100.00 |
|                  | Best #Genes     | 6           | 6         | 4         | —       | 3      | 3    | 2671    | 1042  | 4     |
|                  | Average #Acc (%)| 100.00 <1> | 100.00    | 100.00    | 96.49   | 100.00 | 100.00| —       | 100.00|       |
|                  | Average #Genes  | 7.2         | 12.5      | 6.7       | 105     | 4.10  | 4.70 | —       | 6     |       |

< >: the rank of our method in a specific average accuracy. RBPSO-1NN = a gene selection method based on the combination of ReliefF and BPSO and 1NN as a classifier. FBPSO-SVM = a gene selection method based on the combination of Fisher score and BPSO and the SVM as a classifier; FRBPSO = a fuzzy rule based binary PSO; HICATS= Hybrid Binary Imperialist Competition Algorithm and Tabu Search; EPSO = an enhancement of binary particle swarm optimization; TS-BPSO = A combination of tabu search and BPSO; IBPSO = an improved binary PSO.
present a greedy algorithm to approximate the MWIS of genes and a search phase that determines a near-optimal subset of genes for the cancer classification. The latter is based on a modified ACO and a LS algorithm.

This approach aims to select a small subset of relevant genes from an original dataset which contains redundant, noisy, or irrelevant data.

The experimental results show that our approach compares very favorably with the reference methods in terms of the classification accuracy and the number of selected genes. Although the results obtained are interesting and encouraging, many points are likely to be studied in future works, such as

(i) Modifying the MWIS method in order to improve the quality of the first subset of genes

(ii) Combining the MWIS filter with other meta-heuristics such as VNS

This field of research will always remain active as long as it is motivated by the advances of data collection and storage systems on one hand, and by the oncology requirements on the other hand. The best approach for judging this selection of genes is to collaborate with experts (biologists) for a good interpretation of the results.

Data Availability
The datasets (DNA microarray) used in our paper are publicly available and easily accessible [62] (http://web.archive.org/web/20180625075744/http://www.gems-system.org/) (visited on 2018).

Conflicts of Interest
The authors declare that there are no conflicts of interest regarding the publication of this paper.

References
[1] A. C. Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, and S. P. Fodor, "Light-generated oligonucleotide arrays for rapid DNA sequence analysis," Proceedings of the National Academy of Sciences, vol. 91, no. 11, pp. 5022–5026, 1994.

[2] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," Science, vol. 270, no. 5235, pp. 467–470, 1995.

[3] B. Kumari and T. Swarnkar, "Filter versus wrapper feature subset selection in large dimensionality micro array: a review," International Journal of Computer Science and Information Technologies, vol. 2, no. 3, pp. 1048–1053, 2011.

[4] S. Li, X. Wu, and M. Tan, "Gene selection using hybrid particle swarm optimization and genetic algorithm," Soft Computing, vol. 12, no. 11, pp. 1039–1048, 2008.

[5] R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artificial Intelligence, vol. 97, no. 1-2, pp. 273–324, 1997.

[6] A. Bir-Jmel, S. M. Douiri, and S. Elbernoussi, "Gene selection via BPSO and Backward generation for cancer classification," RAIRO: Recherche Opérationnelle, vol. 53, no. 1, 2019.

[7] B. Bonev, F. Escolano, and M. Cazorla, “Feature selection, mutual information, and the classification of high-dimensional patterns,” Pattern Analysis and Applications, vol. 11, no. 3-4, pp. 309–319, 2008.

[8] P. Jafari and F. Azuaje, "An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors," BMC Medical Informatics and Decision Making, vol. 6, no. 1, p. 27, 2006.

[9] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection,” February 2012, https://arxiv.org/abs/1202.3725.

[10] D. Mishra and B. Sahu, “Feature selection for cancer classification: a signal-to-noise ratio approach,” International Journal of Scientific Engineering Research, vol. 2, no. 4, pp. 1–7, 2011.

[11] S. S. Shreem, S. Abdullah, M. Z. A. Nazi, and M. Alzaqebah, "Hybridizing ReliefF, MRMR filters and GA wrapper approaches for gene selection," Journal of Theoretical and Applied Information Technology, vol. 46, no. 2, pp. 1034–1039, 2012.

[12] X. Wu, V. Kumar, J. Ross Quinlan et al., "Top 10 algorithms in data mining," Knowledge and Information Systems, vol. 14, no. 1, pp. 1–37, 2008.

[13] A. A. Alizadeh, M. B. Eisen, R. E. Davis et al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," Nature, vol. 403, no. 6769, pp. 503–511, 2000.

[14] T. R. Golub, D. K. Slonim, P. Tamayo et al., “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” Science, vol. 286, no. 5439, pp. 531–537, 1999.

[15] Y. M. Chiang, H. M. Chiang, and S. Y. Lin, “The application of ant colony optimization for gene selection in microarray-based
computational and mathematical methods in medicine 19
cancer classification,” in Proceedings of the 2008 International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4001–4006, IEEE, Kunming, China, July 2008.

[16] Y. Li, G. Wang, H. Chen, L. Shi, and L. Qin, “An ant colony optimization based dimension reduction method for high-dimensional datasets,” Journal of Bionic Engineering, vol. 10, no. 2, pp. 231–241, 2013.

[17] F. V. Sharbaf, S. Mosafer, and M. H. Moattar, “A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization,” Genomics, vol. 107, no. 6, pp. 231–238, 2016.

[18] S. Tabakhi, P. Moradi, and F. Akhlaghian, “An unsupervised feature selection algorithm based on ant colony optimization,” Engineering Applications of Artificial Intelligence, vol. 32, pp. 112–123, 2014.

[19] S. Tabakhi, A. Najafi, R. Ranjbar, and P. Moradi, “Gene selection for microarray data classification using a novel ant colony optimization,” Neurocomputing, vol. 168, pp. 1024–1036, 2015.

[20] H. Yu, G. Gu, H. Liu, J. Shen, and J. Zhao, “A modified ant colony optimization algorithm for tumor marker gene selection,” Genomics, Proteomics & Bioinformatics, vol. 7, no. 4, pp. 200–208, 2009.

[21] S. Agarwal, R. Rajesh, and P. Ranjan, “FRBPSO: a Fuzzy rule based binary PSO for feature selection,” Proceedings of the National Academy of Sciences, India Section A: Physical Sciences, vol. 87, no. 2, pp. 221–233, 2017.

[22] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, “Improved binary PSO for feature selection using gene expression data,” Computational Biology and Chemistry, vol. 32, no. 1, pp. 29–38, 2008.

[23] L.-Y. Chuang, C.-H. Yang, and C.-H. Yang, “Tabu search and binary particle swarm optimization for feature selection using microarray data,” Journal of Computational Biology, vol. 16, no. 12, pp. 1689–1703, 2009.

[24] M. S. Mohamad, S. Omatu, S. Deris, and M. Yoshioka, “A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data,” IEEE Transactions on Information Technology in Biomedicine, vol. 15, no. 6, pp. 813–822, 2011.

[25] M. S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah, and Z. Ibrahim, “An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes,” Algorithms for Molecular Biology, vol. 8, no. 1, p. 15, 2013.

[26] C.-P. Lee and Y. Leu, “A novel hybrid feature selection method for microarray data analysis,” Applied Soft Computing, vol. 11, no. 1, pp. 208–213, 2011.

[27] S. K. Pati, A. K. Das, and A. Ghosh, “Gene selection using multi-objective genetic algorithm integrating cellular automata and rough set theory,” in International Conference on Swarm, Evolutionary, and Memetic Computing, pp. 144–155, Springer, 2013.

[28] S. Wang, W. Kong, W. Zeng, and X. Hong, “Hybrid binary imperialist competition algorithm and tabu search approach for feature selection using gene expression data,” BioMed Research International, vol. 2016, Article ID 9721713, 12 pages, 2016.

[29] J. Apolloni, G. Leguizamón, and E. Alba, “Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments,” Applied Soft Computing, vol. 38, pp. 922–932, 2016.
problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[49] L. M. Gambardella, É. D. Taillard, and M. Dorigo, “Ant colonies for the quadratic assignment problem,” *Journal of the Operational Research Society*, vol. 50, no. 2, pp. 167–176, 1999.

[50] S. M. Douiri and S. Elberoussi, “A new ant colony optimization algorithm for the lower bound of sum coloring problem,” *Journal of Mathematical Modelling and Algorithms*, vol. 11, no. 2, pp. 181–192, 2012.

[51] B. Bullnheimer, R. F. Hartl, and C. Strauss, “An improved ant System algorithm for the vehicle Routing Problem,” *Annals of Operations Research*, vol. 89, pp. 319–328, 1999.

[52] C. Solnon, “Ants can solve constraint satisfaction problems,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 347–357, 2002.

[53] J. Levine and F. Ducatelle, “Ant colony optimization and local search for bin packing and cutting stock problems,” *Journal of the Operational Research Society*, vol. 55, no. 7, pp. 705–716, 2004.

[54] E. Fix and J. L. Hodges Jr., *Discriminatory Analysis-Non-parametric Discrimination: Consistency Properties*, University of California, Berkeley, CA, USA, 1951.

[55] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[56] C. Ambroise and G. J. McLachlan, “Selection bias in gene extraction on the basis of microarray gene-expression data,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pp. 6562–6566, 2002.

[57] 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 (Cybernetics)*, vol. 37, no. 1, pp. 70–76, 2007.

[58] J. Holland, *Adaptation in Artificial and Natural Systems*, The University of Michigan Press, Ann Arbor, MI, USA, 1975.

[59] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[60] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, “Large margin DAGs for multiclass classification,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 547–553, Denver, CO, USA, 2000.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[62] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, “A comprehensive evaluation of multiclassification methods for microarray gene expression cancer diagnosis,” *Bioinformatics*, vol. 21, no. 5, pp. 631–643, 2004.