Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A novel filter-wrapper hybrid gene selection approach for microarray data based on multi-objective forest optimization algorithm

Babak Nouri-Moghaddam[a], Mehdi Ghazanfari[a*] and Mohammad Fathian[a]

[a]Department of Industrial Engineering, Iran University of Science and Technology, Tehran, 1684613114, Iran

C H R O N I C L E

Article history:
Received April 5, 2020
Received in revised format:
May 9, 2020
Accepted May 29 2020
Available online
May 29, 2020

Keywords:
Gene Selection
Microarray Data
Multi-Objective Optimization
Metaheuristics Algorithm
Forest Optimization Algorithm
Hybrid Filter-Wrapper

A B S T R A C T

One of the most important solutions for dimensionality reduction in data preprocessing, and improving classification performance is gene selection in microarray data since they usually have several thousand genes with very few samples. Because of these characteristics, the complexity of classification models increases and their efficiency decreases. The gene selection problem inherently pursues two goals: reducing the number of genes and increasing the classification efficiency. Therefore, this paper presents a novel hybrid filter-wrapper solution based on the Fisher-score method and Multi-Objective Forest Optimization Algorithm (MOFOA). In the proposed method, as a preprocessing step, the Fisher-score method selects 500 discriminative genes by removing redundant/irrelevant genes. Then, MOFOA searches to find the subset of optimal genes using concepts such as repository, crowding-distance, and binary tournament selection. Moreover, the proposed method solves the gene selection problem and, at the same time, optimizes the kernel parameters in the SVM classification model. Six microarray datasets were used to evaluate the performance of the proposed method. Afterward, a comparison was made between its results and those of the four multi-objective hybrid methods presented in the literature in terms of classification performance, the number of selected genes, running time, and hypervolume criteria. According to the results, in addition to selecting fewer genes, the proposed solution has achieved greater classification accuracy in most cases and has been able to obtain a performance similar to or better than that of other multi-objective gene selection approaches.

1. Introduction

Increasing advances in computer and electronic technology have provided scientists with the opportunity to collect and study data from a variety of phenomena. These technologies include microarray technology, which allows collecting gene expression levels of cells and tissues. Microarray data can be used to identify diseases or differentiate between tumors. Data mining and machine learning are among the important techniques in analyzing and constructing disease diagnosis models based on microarray data (Bolón-Canedo et al., 2014). However, it is very difficult and challenging to use data mining methods in such data due to problems such as dimensionality curse and data complexity. This challenge raises because microarray data usually have a small number of samples (usually less than 200) and a large number of features (usually more than 2000) (Almugren & Alshamlan, 2019; Bolón-Canedo et al., 2014). One of the goals pursued by data mining in microarray data is to create classification models for disease diagnosis based on gene expression data. It is very time-consuming and expensive to build and develop classification models for high-dimensional data and therefore requires very complex models (Khalid et al., 2014). To overcome these problems, researchers employ dimensionality

* Corresponding author.
E-mail address: mehdi@iust.ac.ir (M. Ghazanfari)

doi: 10.5267/j.dsl.2020.5.006

reduction techniques such as feature selection and feature extraction that lead to reduced dimensions of problem space (Khalid et al., 2014). Features extraction methods such as Principal Component Analysis (PCA) (Jonnalagadda & Srinivasan, 2008) produce a new set of features with smaller dimensions compared to the original set by combining the initial features. These new features produced usually increase the resolution of the data and at the same time eliminate its true or physical meaning as a result of combining the features. Therefore, they are not suitable for the readability and interpretability required in medical diagnoses (Remeseiro & Bolon-Canedo, 2019). In contrast, feature selection methods select the quasi-optimal subset of primary features by removing redundant and irrelevant features. Feature selection has several advantages such as increasing the efficiency of learning models and identifying influential features while increasing the classification error (Nguyen et al., 2020; Remeseiro & Bolon-Canedo, 2019). Also, considering issues like a wide search space and complex relationships between features is a challenging task. Furthermore, reviews may consider a feature to be very effective alone, while it may become a redundant/irrelevant feature along with other features. In contrast, an irrelevant feature may also become an effective feature for the classification model along with other features (Chandrashekar & Sahin, 2014). Besides, to find the subset of the optimal features in a dataset with n features, search methods need to examine the 2^n possible subsets of the combination of features, which renders the search for large n impossible. Accordingly, the feature selection problem is considered as a combinatorial NP-hard problem (Amaldi & Kann, 1998).

The feature/gene selection methods can be divided into five general groups: filter, wrapper, embedded, ensemble, and hybrid. Filter methods (e.g., Mutual Information (Vergara & Estévez, 2014), Information Gain (Khalid et al., 2014), Symmetrical Uncertainty (Hall, 1999), and Laplacian Score (He et al., 2005)) attempt to identify prominent genes independent of a learning model only by relying on common inherent characteristics among genes via statistical techniques (Khalid et al., 2014). They have low computational overhead and high generalizability. In contrast, wrapper methods use a search method with a learning model to evaluate the subset of genes found in the search phase. Thanks to the use of a learning model, wrapper methods usually offer better classification performance than filter methods. In contrast, they have several disadvantages, such as high computational overhead and overfitting probability. Embedded methods prevent the consecutive training of the classification model to evaluate each of the subsets of the genes found by combining the gene selection process with the learning process. Compared to wrapper methods, embedded methods are less likely to overfit but, at the same time, they require higher knowledge and skills to design and build. The ensemble methods try to select the subset of highly stable genes by combining several different filter/wrapper methods. Another gene selection approach is the hybrid methods, which usually use the filter method as a preprocessing step to remove redundant/irrelevant features. They then employ the wrapper method to select the optimal gene subset among the remaining genes by considering classification performance (Khalid et al., 2014; Miao & Niu, 2016; Tyagi & Mishra, 2013).

In recent years, the use of hybrid filter-wrapper methods for solving gene selection problems has appealed the consideration of numerous researchers. The purpose of using these approaches is to improve the effectiveness of gene selection using a combination of the strengths of both wrapper and filter methods (Almugren & Alshamlan, 2019). Based on the search method used in the wrapper stage, hybrid methods can be divided into two categories; i.e., single-objective and multi-objective. In (Chuang et al., 2011), a method based on the correlation-based filter and single-objective GA-Taguchi wrapper approach is presented. Using the Fisher-score filter and ant colony-cellular learning automata (ACO-CLA) hybrid metaheuristic algorithm, Sharbaf et al. presented a hybrid single-objective gene selection method (Vafaee Sharbaf et al., 2016). Moreover, in (Shukla et al., 2020), a single-objective hybrid method is presented. In this method, the minimum Redundancy Maximum Relevance (mRMR) method (as a filter) is initially applied to the gene set to remove irrelevant and redundant genes. Then, the quasi-optimal subset of genes is selected using a single-objective wrapper approach based on Gravitational Search and Teaching-Learning-Based algorithms. Besides, multi-objective hybrid methods have been studied because they consider at least two objectives, namely classification performance maximization and gene number minimization. For example, in (Chakraborty & Chakraborty, 2013), a relevance-based filter and MOGA wrapper hybrid method is proposed considering two goals: maximization of classification accuracy and gene number minimization. Furthermore, in (Baliarsingh, Vipsita, Muhammad, & Bakshi, 2019), Baliarsingh et al. used the Fisher-score with multi-objective Emperor Penguin Optimization (MOEPO), which both solve the gene selection problem and try to improve the parameters of SVM classification model. Also, other multi-objective metaheuristic algorithms (e.g. MOGA (Chakraborty & Chakraborty, 2013), NSGA-II (Banerjee et al., 2007; Hasnat & Molla, 2017; Mohamad et al., 2008), MOPSO (Annavarapu et al., 2016; Lai, 2018; Shahbeig et al., 2018) ،MOACO (Ratnoo & Ahuja, 2017), MOBAT (Dashtban et al., 2018; Mishra et al., 2012), and MOBBA (Li & Yin, 2013)) have been used for this purpose.

Due to population-based search and producing various solutions, metaheuristic algorithms can be used to solve the gene selection problem as multi-objective problem. This is because the gene selection problem, thanks to its inherent characteristics, follows at least two objectives: 1) reducing the number of genes and 2) increasing classification performance. Therefore, it can be considered as a multi-objective optimization problem (MOP). In such a problem, the final solutions are usually presented as a set of non-dominated (ND) solutions; i.e., a trade-off between conflicting goals (Mukhopadhyay et al., 2014). On the other hand, according to the literature, more limited solutions have been proposed to solve the gene selection problem as a multi-objective hybrid method compared to single-objective hybrid mode. Most of the proposed hybrid solutions have solved the gene selection problem as a single-objective problem only by considering classification or clustering performance (Almugren & Alshamlan, 2019; Bolón-Canedo et al., 2014; Remeseiro & Bolon-Canedo, 2019).

One of the newest metaheuristic algorithms presented is the Forest Optimization Algorithm (FOA), which has been used successfully to solve optimization problems (Ghaemi & Feizi-Derakhshi, 2014, 2016; Mohapatra et al., 2018). FOA features include high speed, low number of function evaluation (NFE), and effective global/local search. In recent years, this algorithm has been used successfully as a single-objective to solve the feature selection problem (Ghaemi & Feizi-Derakhshi, 2014, 2016). According to the mentioned cases, the capabilities of this algorithm used as multi-objective in solving gene selection problems have not been studied so far.

1.1. Goals

The main purpose of this paper is to develop a gene selection multi-objective hybrid approach based on the Fisher-score filter method and FOA. Therefore, the proposed solution can reduce the number of genes and, as well as improving the classification accuracy and optimizing the parameters of the SVM kernel function. The resulting *ND* solutions set should outperform similar methods in terms of accuracy and number of selected genes. To this end, a multi-objective FOA-based algorithm was proposed based on concepts such as repository, binary tournament selection, dominance, and crowding-distance. A comparison has been made between the results of the proposed algorithm and those of four multi-objective hybrid methods on six benchmark microarray datasets. These datasets include several features, classes, and various examples. The objectives of this study are as follows:

- Providing a Fisher-score and multi-objective FOA hybrid algorithm;
- Performance analysis of the proposed method in finding optimal Pareto front compared to other hybrid multi-objective gene selection algorithms;
- Investigating the effectiveness of the proposed solution in increasing classification accuracy compared to other 'hybrid multi-objective gene selection algorithms'; and
- Comparing the proposed solution with single-objective and multi-objective hybrid methods in terms of performance and efficiency

The remainder of this paper is structured as follows. Section 3 presents the theoretical foundations of FOA and gene selection of previous approaches. Section 4 describes the proposed idea to solve the gene selection problem. Section 5 describes the design steps and details of the tests. Section 6 analyzes laboratory results. Finally, Section 7 provides conclusions and suggestions for future works.

2. Literature Review

This section first provides a summary of the concepts of FOA, multi-objective optimization, and SVM classifier followed by reviewing important research on hybrid gene selection methods.

2.1. FOA Expression

FOA is a state-of-the-art nature-inspired metaheuristic algorithm proposed by Ghaemi and Feizi-Derakhshi (Ghaemi & Feizi-Derakhshi, 2014). Here, FOA tries to simulate the pattern of growth and expansion of trees in the forest to solve optimization problems. In nature, like all creatures, trees compete with each other for resources such as sunlight, water, and suitable soil. During this competition, trees try a variety of methods such as seeding through water, wind, and animals to find new and more suitable sources for growth and expansion. Trees grow and die over time and are replaced by new trees as they age. Trees mainly use two seeding mechanisms. In the first mechanism, called local seeding, tree seeds settle around the parent tree and then begin to grow. In the second mechanism, called global seeding, tree seeds are transported to more remote areas using factors such as wind, water, or animals and placed in a completely new location. Using the idea of these two mechanisms, FOA seeks to create exploitation and exploration capabilities in the search process to find the best conditions for tree growth. The summary of the steps of this algorithm is as follows:

1) Initial Forest: Similar to other metaheuristic algorithms, FOA starts with a population of initial trees. Each tree is defined as a vector of $Tree = \{age, V_1, V_2, \ldots, V_n\}$, where V_1-V_n are the problem space variables and the initial forest is generated randomly. Furthermore, the age parameter of the tree is saved in the profile of each tree. When the algorithm starts working, the age of all the trees is set to zero. The tree lifetime is considered as one of the parameters of the algorithm to control the forest population, which is defined by the user according to the type of problem.

2) Local Seeding Operator: When trees are placed in the seeding stage, a part of their seeds usually falls to the ground around the parent tree. After a while, these seeds turn into young trees competing with each other for new resources. Trees located in places with sufficient resources will have a better chance of survival. Local seeding operator in FOA seeks to simulate the local search subprogram in nature. This operator runs in trees with age zero and adds several neighbors for each tree in the forest. The number of new trees generated by this operator for each tree is known as Local Seeding Change (LSC).

The age variable is used as one of the population control mechanisms in the forest. Therefore, in this algorithm, the new trees, and the best tree found will have age zero so that the algorithm can do more searching around them. As a result, trees that do not fit well gradually get older, through algorithm iterations. Eventually, they die after reaching the maximum allowable age.

3) Population Limitation: To control the population of forest trees, two parameters have been considered, namely maximum age and area limit. At this stage, first, trees older than the maximum allowable age are omitted from the forest and inserted to the candidate population. Then, if the number of remaining trees in the forest is greater than the designated *area limit*, the second stage of limitation of the forest will be implemented. At this stage, based on the fitness value, the trees are sorted in descending order and the optimal trees are transferred to the next generation as defined as the area limit. Other trees are removed and inserted to the candidate population.

4) Global Seeding Operator: In nature, some external factors such as wind, water, or animals spread trees seed in remote areas, which may help new trees, grow in neighborhoods away from their native areas. The global seeding operator simulates such a process. The operator selects several trees belonging to the candidate population according to the transfer rate parameter and subsequently changes several variables from each tree at random. Global seeding is used to create a divergence in the search process and improve the global search algorithm. The number of variables that need to be altered randomly is recognized by a factor called Global Seeding Change (GSC).

5) Stopping Condition: As with other evolutionary algorithms, one of the following conditions can be considered for stopping: 1) the number of pre-determined iterations, 2) the Number of Function evaluation (NFE), and 3) achieving a predefined accuracy.

2.2. Multi-Objective Optimization

Most real-world problems follow multiple often-conflicting goals that require to be optimized simultaneously. Solving such problems is usually presented as a set of solutions, indicating a tradeoff between different goals. The set of multi-objective problems solutions is known as Pareto optimal solutions. A multi-objective optimization problem (MOP) is defined as (Coello et al., 2007; Deb, 2001).

$$min\, \vec{F} \coloneqq [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})] \tag{1}$$

subject to

$$g_i(\vec{x}) \leq 0 \quad i = 1,2,3,\dots,m \tag{2}$$

$$h_i(\vec{x}) \leq 0 \quad i = 1,2,3,\dots,p \tag{3}$$

where $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $f_i: R^n \to R,\ i = 1,\dots,k$ is objective functions, and $g_i, h_j: R^n \to R, i = 1,\dots,m,\ j = 1,\dots,P$ is a constraint function. The following definitions must be considered for solving the problem at hand:

Definition 1: The vector $\vec{u} = (u_1, u_2, \dots, u_k)$ dominates the vector $\vec{v} = (v_1, v_2, ..., v_k)$ as Pareto, ($\vec{u} \preccurlyeq \vec{v}$), if and only if $\vec{u} \ \leq \ \vec{v}$ in all cases and at least one u_i for which $u_i < v_i$:

$$\forall i \in \{1,\dots,k\}: u_i \leq v_i \ \wedge \ \exists i \in \{1,\dots,k\}: u_i < v_i \tag{4}$$

Definition 2: The solution to $\vec{x} \in \mathcal{F}$ ($\mathcal{F}$ represents the space of the acceptable solutions) is a optimal Pareto solution, if and only if there is no solution $\vec{x}' \in \mathcal{F}$ for which $\vec{v} = \vec{F}(\vec{x}') = \left(\vec{f_1}(\vec{x}'), \dots., \vec{f_k}(\vec{x}')\right)$ dominates $\vec{u} = \vec{F}(\vec{x}) = (\vec{f_1}(\vec{x}), \dots., \vec{f_k}(\vec{x}))$.

Definition 3: For a given MOP, $\vec{F}(\vec{x})$, optimal Pareto set $\mathcal{P}^*$ is defined as follows:

$$\mathcal{P}^* := \{\vec{x} \in F \,|\, \exists \vec{x}' \ \in \ \mathcal{F} \ \vec{F}(\vec{x}') \ \preccurlyeq \ \vec{F}(\vec{x})\,\} \tag{5}$$

Definition 4: For a given MOP, $\vec{F}(\vec{x})$ and optimal Pareto set $\mathcal{P}^*$, Pareto optimal front $\mathcal{PF}^*$ are defined as follows:

$$\mathcal{PF}^* := \{\vec{u} = \vec{F}(x) \,|\, \vec{x} \ \in \ \mathcal{P}^*\}\,. \tag{6}$$

As a result, the optimal Pareto front of the $\mathcal{F}$ set of all decision variable vectors will include members who meet conditions (2) and (3).

2.3. SVM Expression

One of the well-known classification models is the SVM provided by Vanpik et al. (Cortes & Vapnik, 1995). The goal of SVM is to maximize the margin between separating hyperplane and training data points. The hyperplane is defined according to Eq. (7), where ω determines the direction of the hyperplane and b represents the bias of the hyperplane distance from the point of origin.

$$\omega \cdot x + b = 0 \tag{7}$$

To find the optimal hyperplane, the optimization problem must be solved according to Eq. (8):

$$\begin{aligned}&Min\ \frac{1}{2}\|\omega\|^2\\&s.t.\, y_i(w\cdot x_i+b)=1,\qquad i=1,2,\cdots,n\end{aligned}\tag{8}$$

The variable ξ_i is added to Eq. (8) such that it can solve nonlinear problems. In this case, the final equation will be as follows:

$$\begin{aligned}&min\ \frac{1}{2}\|\omega\|^2+\mathrm{C}\sum_{i=1}^{N}\xi_i\\&s.t.\, y_i(w\cdot x_i+b)=1,\qquad i=1,2,\cdots,n\end{aligned}\tag{9}$$

where C represents a tradeoff between the training error and the generalization of the model. To solve nonlinear problems, SVM maps the problem space dimensions to a space with higher dimensions. Therefore, the classification function is defined as follows:

$$f(x)=sign\left(\sum_{i=1}^{N}a_iy_iK(x_i,x_j)+b\right)\tag{10}$$

where a_i is Lagrange multiplier and $K(x_i,x_j)$ represents the kernel function. Regarding the high dimensionality of gene expression datasets and the ability of RBF kernel to solve high-dimensional classification problems, this article will use the RBF kernel. The RBF kernel is calculated by considering two samples (i.e., $\vec{x}_i$ and $\vec{x}_j$), as follows:

$$K(x_i,x_j)=\exp\left(-\gamma\|x_i-x_j\|^2\right)\tag{11}$$

where $\gamma > 0$ indicates the *Gaussian* width. There are two influential parameters in non-linear *SVM* classification problems, namely C and γ. C shows a tradeoff between training error and model generalization. If C is considered too large, the training error will decrease. At the same time, the model will lose its ability to generalize against unseen data. Moreover, if the value of C is considered too small, the training error of the SVM model will be very high. On the other hand, the parameter γ is also very effective in model training. If the value of γ is considered to be large, the probability of overfitting the SVM model will be very high. Therefore, both of these parameters will need to be optimized.

2.4. An Overview of the Hybrid Gene Selection Method

Feature selection methods or gene selection can be divided into five general groups, including filter, wrapper, embedded, ensemble, and hybrid. This research will focus on hybrid methods. From now on, this section will review previous work on gene selection using hybrid methods. Hybrid algorithms typically use a filter method to preprocess data and remove irrelevant/redundant genes to reduce search space to an acceptable level. In the following, a wrapper method will be used by considering the efficiency of a classification model to select a sub-optimal subset from the remaining genes. Depending on the type of search method, hybrid methods can be divided into two groups including single-objective and multi-objective. Single-objective hybrid algorithms usually have one of the goals of maximizing the classification efficiency or minimizing the number of selected genes, or a combination of the two. In (Shen et al., 2008), a hybrid solution is suggested to solve the gene selection problem by considering the classification accuracy. In this method, first, the number of genes is reduced using the t-test method and, subsequently, the subset of the optimal genes is selected using the hybrid PSO-Tabu Search algorithm. Yang et al. proposed a hybrid solution using the information gain filter criterion and GA (Yang et al., 2010). Another hybrid solution based on the correlation-based FS filter and the GA-Taguchi algorithm is presented in (Chuang et al., 2011). In the proposed solution, the Taguchi method is used as a local optimizer to improve GA solutions. The solutions obtained are evaluated based on the KNN classifier accuracy criterion. In (Lee & Leu, 2011), a different method is described to provide a hybrid solution. For this purpose, first, the GADP algorithm is used to select a sub-optimal subset of the genes followed by selecting an acceptable number of high-rank genes using the $\chi^2 - test$ method. Another method based on the $\chi - rank$ filter and differential evolutionary (DE) algorithm is provided by Apolloni et al. (Apolloni et al., 2016). One of the important metaheuristic algorithms is the Ant Colony algorithm (ACO), which is used along with the Fisher-Score filter, and cellular learning automata algorithms to present a hybrid gene selection method (Vafaee Sharbaf et al., 2016). In (Dashtban & Balafar, 2017), Dashtian et al. proposed a new hybrid method using a variable-length integer coding approach to express GA chromosomes. The proposed method examines the effects of using two filter methods, i.e., Laplacian and Fisher-Score filters, in combination with the GA algorithm. Other solutions based on adaptive GAs (AGAs) and different filter methods are presented in (Gangavarapu & Patil, 2019; Lu et al., 2017; Shukla et al., 2018). A nature-inspired method based on the Salp Swarm algorithm (SSA) in combination with Fisher-Score was proposed in (Baliarsingh, Vipsita, Muhammad, Dash, et al., 2019). In this solution, the random number generator (RNG) function is replaced in the algorithm by chaos-based function to guide the SSA algorithm purposefully. Furthermore, a hybrid solution based on the Gravitational search algorithm and Teaching-Learning-based optimization algorithm is presented in (Shukla et al., 2020). In the proposed idea, first, the number of dataset genes is reduced to an acceptable level using the mRMR method followed by applying the TLBOGSA hybrid algorithm to select the optimal gene subset from the remaining genes.

In recent years, solving the gene selection problem using multi-objective metaheuristic algorithms has drawn the consideration of many researchers. This is because the gene selection problem is inherently a multi-objective optimization problem that essentially pursues at least two conflicting goals; i.e., classification performance maximization and gene number minimization. Multi-objective metaheuristic algorithms can optimize multiple often conflicting goals simultaneously (Hancer et al., 2018). The solution to such algorithms is a set of *ND* solutions that provide a tradeoff between different objectives for users. The *ND* solution set is also called Pareto front solutions. In (Banerjee et al., 2007), a hybrid solution based on rough set theory and the NSGA-II algorithm is proposed to solve the feature selection problem that follows two goals, namely reducing the number of genes and increasing classification accuracy. Mohamad et al. proposed a new hybrid solution based on “between-group to within-group” filter and the NSGA-II algorithm (Mohamad et al., 2008). In the proposed method, conventional mutation and crossover operators were replaced by the estimation of the distribution (EDA) algorithm to better guide the NSGA-II search process. EDA aims to generate random numbers purposefully by learning the possible distribution of best solutions. In another research, a BAT and PCA-based algorithm are used in which the BAT algorithm also optimizes the weight of artificial neural networks in addition to feature selection (Mishra et al., 2012). In (Li & Yin, 2013), a hybrid solution based on Fisher-Markov and multi-objective binary biogeography-based optimization (MOBBBO) algorithm is presented. In this solution, redundant and irrelevant genes are removed from the dataset using the Fisher-Markov method. Then, the MOBBBO algorithm is used to select the optimal set of genes and also to optimize the SVM kernel function parameters. In another study, a combination of the NSGA-II algorithm and correlation coefficient filter was used to solve the gene selection problem (Hasnat & Molla, 2017). In (Shahbeig et al., 2018), Shahbeig et al. proposed a new solution to solve the gene selection problem using the popular MOPSO algorithm and a filter method. This method employs the basics of chaos theory to generate random numbers to improve MOPSO performance and guide the search as logical as possible. In (Dashtban et al., 2018), a combination of the MOBAT algorithm and the Fisher-score method is used for gene selection. The proposed method uses two strategies of random walk-based injected and extended local search to improve MOBAT solutions. Moreover, in (Baliarsingh, Vipsita, Muhammad, & Bakshi, 2019), Baliarsingh et al. used the Fisher-score method along with MOEPO to improve the parameters of the SVM classification model in addition to solving the gene selection problem.

Most studies have used single-objective metaheuristic algorithms to solve gene selection problems. Meanwhile, a limited number of studies have been performed on multi-objective hybrid gene selection methods compared to others. Given the promising FOA features such as operator simplicity, low NFE, and the low number of parameters, the algorithm’s ability to solve multi-objective gene selection problems has not yet been investigated. Therefore, this paper presents an FOA-based hybrid filter-wrapper multi-objective solution to solve the gene selection problem.

3. Proposed Method

Based on the issues raised in the previous sections, it can be concluded that the gene selection problem is inherently an optimization problem that pursues two conflicting goals: 1) reducing the number of genes and 2) increasing classification performance. FOA is a single-objective algorithm that has been used successfully to solve the feature selection problem as the wrapper method. However, no version of this algorithm has been proposed in the literature to solve the gene selection problem in the form of multi-objective. Therefore, this section presents a new method called Gene Selection Multi-objective Forest Optimization Algorithm (GSMOFOA) to solve the gene selection problem as a filter-wrapper.

Based on concepts such as dominance in multi-objective optimization, FOA requires a series of structural changes in calculating the probability of selection, maintaining and updating the Pareto front, generating new solutions, and managing candidate population. To make such changes, the concepts and ideas proposed in MOPSO (Coello Coello & Lechuga, 2002) and NSGA-II (Deb et al., 2002) algorithms have been utilized, which are the most popular metaheuristic algorithms for solving MOP. Based on this, the multi-objective FOA algorithm is presented using concepts such as repository, crowding-distance, binary tournament selection, and mapping continuous space to binary. Initially, to properly understand GSMOFOA, the flowchart and pseudo code of the proposed algorithm are shown in Fig. 1 and Algorithm 1, followed by the description of the main sections.

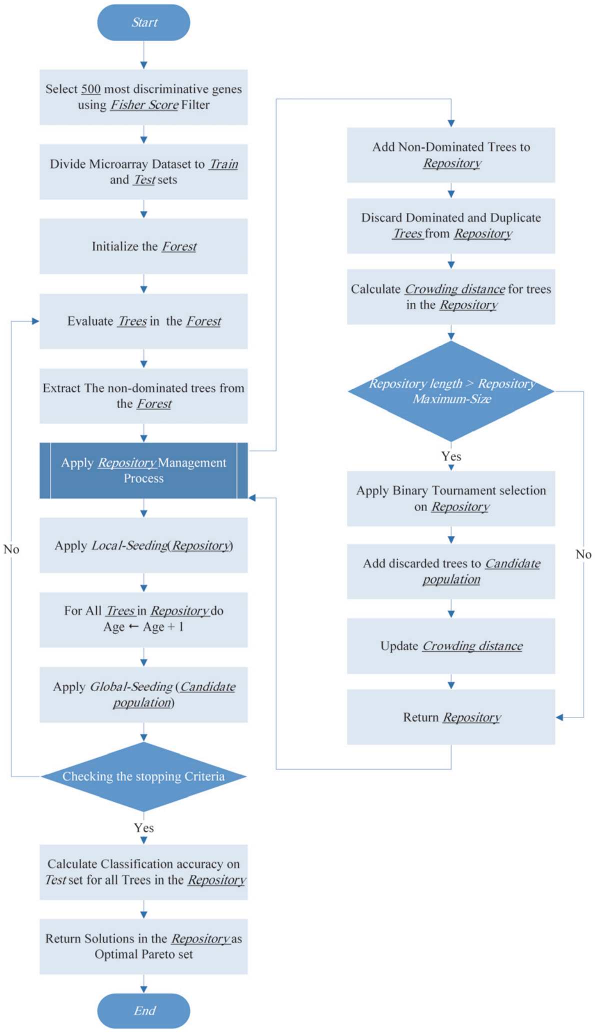
Start
Select 500 most discriminative genes using Fisher Score Filter
Divide Microarray Dataset to Train and Test sets
Initialize the Forest
Evaluate Trees in the Forest
Extract The non-dominated trees from the Forest
Apply Repository Management Process
Apply Local-Seeding(Repository)
For All Trees in Repository do Age ← Age + 1
Apply Global-Seeding (Candidate population)
Checking the stopping Criteria
No
Yes
Calculate Classification accuracy on Test set for all Trees in the Repository
Return Solutions in the Repository as Optimal Pareto set
End
Add Non-Dominated Trees to Repository
Discard Dominated and Duplicate Trees from Repository
Calculate Crowding distance for trees in the Repository
Repository length > Repository Maximum-Size
Yes
No
Apply Binary Tournament selection on Repository
Add discarded trees to Candidate population
Update Crowding distance
Return Repository

Fig. 1. Proposed GSMOFOA Flowchart

3.1. Selection of Effective Genes

In the first step of the proposed method, the Fisher-score filter method (Gu et al., 2012) selects 500 effective genes from the dataset (Fig. 1). The Fisher-score is used in many articles as an effective method in eliminating irrelevant and redundant genes (Baliarsingh, Vipsita, Muhammad, Dash, et al., 2019; Dashtban et al., 2018; Dashtban & Balafar, 2017; Gu et al., 2012).

3.2. Forest Initialization

Like most metaheuristic algorithms, initialization in GSMOFOA is random. In the proposed method, each tree is displayed as a vector that displays the tree *age*, the parameters C and γ of the kernel function, and the values of the genes. The length of each tree is denoted with $n + 3$, in which n is the number of dataset genes along with the three parameters of *age*, C, and γ. Fig. 2 represents a tree or solution in the proposed method. The *age* variable will be equal to zero in the initialization stage for new trees. In the proposed method, the parameters C and γ will be considered in the intervals $C \in [2^{-5}, 2^{15}]$ and $\gamma \in [2^{-15}, 2^{5}]$ according to the *SVM* structure and the studies performed.

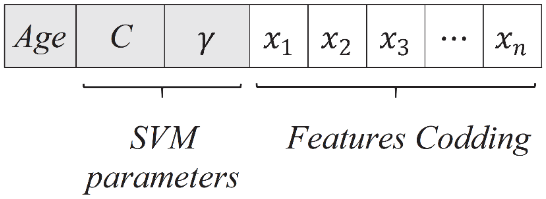

Fig. 2. Representation of a tree in the proposed method

The value of the variables $x_1, x_2, \cdots, x_n$ belongs to the continuous interval $[-2,2]$, which will be mapped to the binary space according to Eq. (12). In Eq. (12), if the output value of the $sigmoid(x_i)$ function is greater than 0.5, the binary value of the gene will be equal to 1; otherwise, it will be equal to 0. Here, 1 means gene selection and 0 means no selection.

$$\mathrm{f}(x_i) = \begin{cases} 1 & if\ sigmoid(x_i) \geq 0.5 \\ 0 & otherwise \end{cases} \tag{12}$$

$$sigmoid(x_i) = \frac{1}{1 + e^{-x_i}} \tag{13}$$

3.3. Repository Formation and Parent Selection Procedure

Many multi-objective algorithms such as MOPSO and PESA-II use a separate population called repository to hold the main Pareto front F_0. The proposed algorithm uses a repository to hold the Pareto front. In each iteration, *ND* solutions are extracted from the forest and added to the repository. Since adding new solutions to the Repository may cause some of the existing solutions to be dominated, the dominated members should be omitted from the Repository.

In single-objective algorithms, to select a parent using techniques such as the roulette-wheel, the population of the solutions is first sorted based on the value of the fitness function followed by calculating the selection probability according to the fitness value of each solution. However, in multi-objective algorithms, despite of numerous objectives that are not superior to each other, the usual solutions for parent selection do not work. To select the parent in such cases, two criteria (i.e., convergence to the main Pareto and diversity of the solutions) are considered. Diversity criteria show the main Pareto front cover and the variety of solutions obtained by the algorithm. The greater the variety and diversity of Pareto front solutions, the better the algorithm will perform. The NSGA-II algorithm uses the crowding-distance criterion to identify densely populated and sparsely populated areas in the Pareto front. In the proposed method, to select the parent, crowding-distance is calculated for the solution in the repository according to the Eq. (14).

$$cd_j^p = cd_j^p + \frac{obj_p^{j+1} - obj_p^{j-1}}{obj_p^{max} - obj_p^{min}}, \tag{14}$$

where $p = 1,2,\cdots,P$ is the number of objectives; j is the number of the solution on the list sorted based on the p-th objective; and obj_p^j represents the value of the p-th objective function for the j-th solution. The crowding-distance value for the border points is set to ∞. Then, in the parent selection stage, two solutions are selected from the repository randomly using the binary tournament selection method. Among the selected solutions, the solution with the higher crowding-distance value is selected as the parent for the local-seeding stage. The purpose of using this strategy is to direct the search to the less searched areas of the Pareto front. In this way, the algorithm achieves better solutions in terms of variety and diversity.

3.4. Limitation of the Forest

Limitation of the forest is one of the most important FOA operators, by which old trees and those with inappropriate fitness values are omitted from the main forest and inserted to the candidate population. The proposed MOFOA also considers two limitations for trees presenting in the repository, including repository size and tree age. Accordingly, after adding the *ND* solutions, if the number of repository members is higher than the maximum size specified, the redundant solutions should be deleted from the repository. In removing trees from the repository, solutions located in densely populated areas should be prioritized. Therefore, to omit a tree, two trees are selected from the repository randomly using binary tournament selection. Next, the tree with less crowding-distance is omitted from the repository and appended to the candidate population. When removing a tree, it is important to consider the *age* criterion. If the crowding-distances between the two randomly selected solutions at this stage are equal, the 'older solution' will be removed.

3.5. Generation of New Trees

To generate new trees in FOA, two operators are used: local seeding and global seeding, to create exploitation and exploration capabilities in the FOA search process, respectively. The local seeding operator will run on the trees in the repository and the parent tree selection process will be based on the descriptions provided in the previous sections. In basic FOA, the local seeding operator is run only on trees with *age* 0. However, in multi-objective mode, all new solutions generated may be dominated by the solutions in the repository and, thus, none of them will be added to the repository. Therefore, if only trees with *age* 0 are considered, to produce new trees, the algorithm may quickly stagnate and no longer be able to produce new solutions. Accordingly, the proposed algorithm has made it possible to select trees with $age \leq 2$. Based on the basic FOA, the local seeding operator generates the LSC number of new trees for each parent tree. To generate each new tree, one of the parent tree variables is selected randomly and its value is added to the random number r. The value of r will be selected from the range $[-\Delta x, \Delta x]$, where the Δx is determined based on the type of problem relative to the value of the allowable range. If the Δx is too large, the search will turn into a random search; besides, the small values of Δx will not make an effective change in the tree. Fig. 3 presents the example of the local seeding operator.

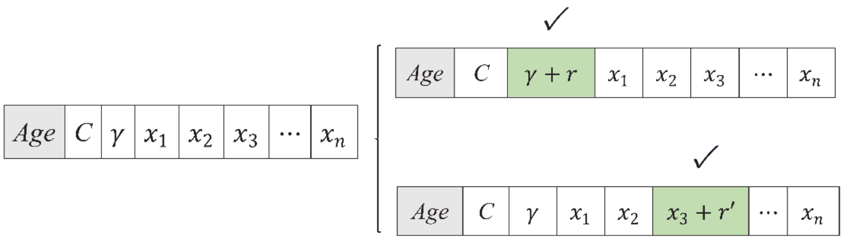
Age | C | γ | x1 | x2 | x3 | ⋯ | xn
✓
Age | C | γ + r | x1 | x2 | x3 | ⋯ | xn
✓
Age | C | γ | x1 | x2 | x3 + r′ | ⋯ | xn

Fig. 3. Example of local seeding operator in GSMOFOA with $LSC = 2$.
$r \in [-\Delta x, \Delta x]$ and $r' \in [-\Delta x', \Delta x']$

As already mentioned, the global seeding operator is another tool for generating new trees in FOA. First, it selects as many as transfer rate from the trees in the candidate population and then changes the GSC number of variables in each tree. To change the variables, a random value is generated in the allowed range of the variable to replace the previous value. Fig. 4 shows the example of this operator.

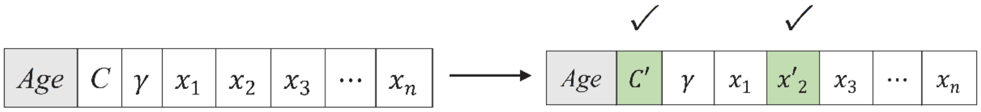

Fig. 4. Example of Global seeding operator in GSMOFOA with $GSC = 2$; the new value of x', C' will be randomly selected from the ranges of $C' \in [2^{-5}, 2^{15}]$ and $x' \in [-2,2]$

3.6. Objective Functions

To solve the gene selection problem in multi-objective mode, two objectives have been considered, namely minimization of the number of genes and maximization of classification accuracy. Classification accuracy is calculated using Eq. (15):

$$Acc = \frac{TP + TN}{FP + FN + TP + TN} \tag{15}$$

Based on the confusion matrix, true positive, true negative, false positive, and false negative, are given by TP, TN, FP, and FN, respectively.

Algorithm 1: GSMOFOA pseudo code

```
begin
Apply Fisher-score Filter
Select 500 most discriminative features
Divide Microarray Dataset into a Training set and a Test set
Randomly initialize the Forest
Evaluate the initial Trees in the Forest (i.e., classification accuracy and selected features ratio)
while ($x$ < Max-Iteration)
    Extract non-dominated trees from the forest
    Add non-dominated trees to Repository
    Discard dominated Trees from Repository
    Discard duplicate Trees from Repository
    Calculate crowding-distance for all solutions in the Repository
    if Repository size > Repository Maximum-Size
        Apply Binary Tournament-Selection  to eliminate Trees from Repository with age > 2
          /* the solution with less crowding-distance wins the tournament */
        Add removed Trees to the Candidate population
        Update crowding-distance for all solutions in the Repository
    end if
    for j ← 1 to AreaLimit
        Apply Binary Tournament-Selection  to select Parent Tree with age <=2 from Repository
        /* the solution with more crowding-distance wins the tournament */
        Apply Local-Seeding(selected Parent, LSC)
        Add new Trees to the forest
    end for
    For All Trees in Repository do Age ← Age + 1
    Randomly select some trees of the Candidate population according to the "transfer rate"
    for each selected tree from Candidate population
        Apply Global-Seeding(selected tree, GSC)
        Add new Trees to the forest
    end for
    $x \leftarrow x + 1$
end while
For the solutions in Repository, calculate the classification accuracy on the Test set
Return the solutions in Repository as final Pareto front with their Training and Test classification accuracy
End
```

5. Experiment Design

To perform the experiments and comparisons, six microarray datasets (Zhu et al., 2007) were selected with a variety of samples, genes, and classes, with their details presented in Table 1. Each dataset was divided into two sets: 80% train and 20% test while maintaining the class ratio. The SVM classification model with RBF kernel was utilized to assess the subset of the selected genes. The kernel parameters (i.e., C and γ) of SVM are also optimized by the proposed method. In many gene classification studies, the SVM classifier yields better results than other classifiers such as DT and KNN (Chuang et al., 2009, 2011; Li & Yin, 2013; Mohamad et al., 2008; Motieghader et al., 2017; Sharma & Rani, 2019).

Table 1
Outline of the datasets

Datasets	# of features	# of Classes	# of Samples
Colon	200	2	62
SRBCT	2308	4	83
9-Tumors	5726	9	60
Leukemia1	7129	2	72
Lung-Cancer	12600	5	203
Prostate	12600	2	101

K-fold cross-validation with $k = 10$ was applied to evaluate the subset of genes selected by each tree/individual (Kohavi & John, 1997). In this case, the training subset is divided to 10 parts in stratified way and the classification accuracy in a loop is calculated using the SVM algorithm on each of these parts. The average classification accuracy obtained on each of the 10 sections is considered as the objective function value. After completing the training steps, a subset of the selected genes will be evaluated using the test data set.

To evaluate and compare the proposed GSMOFOA results, four hybrid multi-objective methods were selected: MOBBBO (Li & Yin, 2013), MOPSO (Shahbeig et al., 2018), NSGA-II (Hasnat & Molla, 2017), and MOBAT (Dashtban et al., 2018). All metaheuristic methods are implemented using the Python 3.7 programming language. Furthermore, for the classification model, the SVC function implemented in the Scikit-learn library using the default settings. To perform the experiments, a computer with the following hardware was used: Intel Core i7 6700HQ and 16GB RAM.

Table 2
Summary of the parameters and settings of the compared algorithms

Algorithms	Representation	Operators	Parameters
NSGA-II	Binary	Single Point Crossover, One point mutation	*Crossover rate*=0.8, *Mutation rate*=0.05
MOBAT	Binary	Standard Binary Bat update	$\alpha, \gamma \in [0, 1]$
MOBBBO	Binary	Basic Habitat Migration and Mutation Strategy	*Habitat modification probability = 1, mutation rate = 0.5*
MOPSO	Binary	Standard PSO particle update	$c_1, c_2 = 1.73$, *Inertial weight* $w = 0.76$, *MOPSO Repository size=unlimited*
GSMOFOA	Continuous	Local Seeding, Global seeding	*transfer-rate*=20%, *lifetime*=25, *LSC*=10, *GSC* = 15, *Repository Max-Size*=75

Then, to evaluate and compare the results fairly, the parameters of the selected algorithms were used, as stated in the reference article; otherwise, the optimal parameters were selected for each of them based on the Taguchi experiment design method (Taguchi et al., 2005). In all algorithms, the number of individuals (i.e., tree, chromosome, habitat, bat, and particle) is set to 50 and the number of function evaluations is set to 10000 for the stopping condition. Moreover, the overall results are examined in 70 independent runs. For the NSGA-II algorithm, the roulette-wheel method was used to select the parents. In continuous expression algorithms, the threshold is set to 0.5 to select or not to select a gene. Table 2 provides a summary of the parameters and settings of the compared algorithms.

In Section 6, the comparisons presented are based on the following criteria: the number of selected genes, classification accuracy, and running time of the algorithm. To form the final Pareto front in each of the methods, the resulting solution sets of 70 independent runs are placed in a union set. Then, *ND* solutions are extracted from union set and reported as the final Pareto set. Furthermore, in the box-plot charts presented in Section 6, the data collected in the union set are used.

6. Experiments and Results

This section indicates the experiment results in three parts: 1) comparing the proposed GSMOFOA with multi-objective algorithms, 2) evaluating the performance of the proposed GSMOFOA using the Wilcoxon-rank sum statistical test, and 3) comparing multi-objective methods in terms of CPU run-time, and Space and time complexity analysis. In the following, each of the above sections will be described.

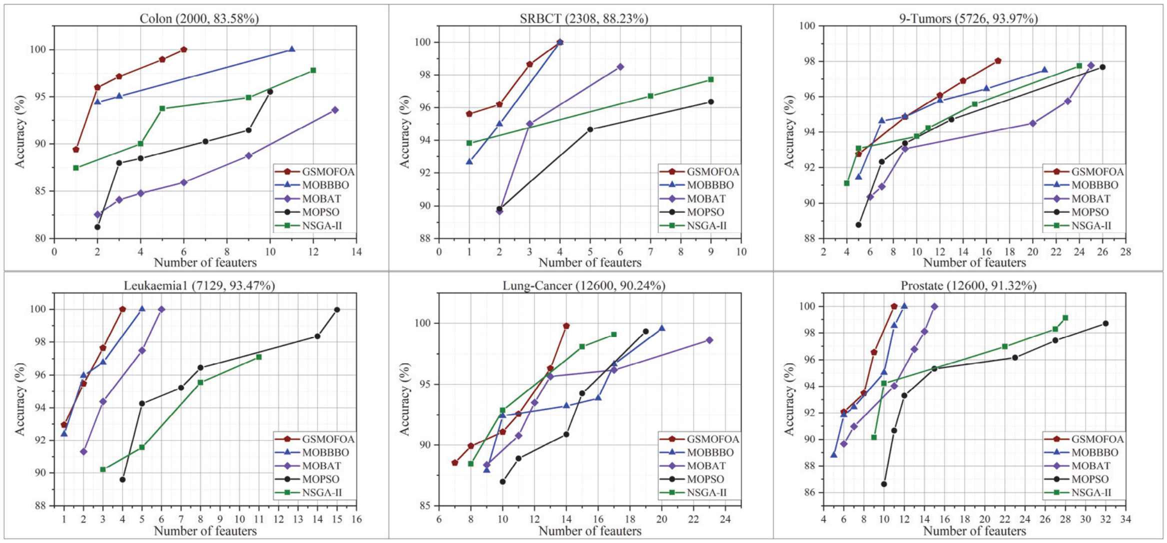

Fig. 5. Comparison of GSMOFOA *ND* solutions with other multi-objective algorithms on the *test* set

6.1. Comparison of Multi-Objective GSMOFOA with other Multi-Objective Methods

Figs. 5-8 present the results of comparisons between GSMOFOA and four multi-objective hybrid methods, which are MOBBBO, MOBAT, NSGA-II, and MOPSO. These figures are drawn based on the *ND* solutions and the accuracy distribution of the *ND* solutions in the union set (found by each algorithm in 70 independent runs). According to Fig. 5, the proposed method was able to obtain better solutions in 5 of the 6 datasets, with regards to number of selected genes and accuracy, compared to the other four methods. For example, in the case of dataset Prostate, the GSMOFOA was able to achieve 100% accuracy by selecting 11 genes out of 12,600 existing genes. Meanwhile, the MOBBBO and MOBAT methods achieved this accuracy by selecting more genes. However, in the SRBCT dataset, both GSMOFOA and MOBBBO were able to achieve 100% accuracy by selecting 4 genes from 2308 genes.

Table 3
Results of SCC measure on the *test* set

Datasets	GSMOFOA	NSGA-II	MOBBBO	MOBAT	MOPSO
Colon	5	0	0	0	0
SRBCT	3	0	1	0	0
9-Tumors	4	2	2	0	0
Leukemia1	3	0	1	0	0
Lung-Cancer	4	1	0	0	0
Prostate	4	0	1	0	0

The Success Counting (SCC) criterion (Sierra & Coello Coello, 2005) has been used for a making a better-quantified comparison between GSMOFOA and four other methods. To calculate this criterion, the final Pareto front obtained by each of the methods is placed in the union set. Next, the optimal Pareto front obtained by the five methods is extracted from them. Finally, the SCC criterion calculates the number of optimal Pareto front solutions obtained by each of the methods based on the Eq. (16).

$$SCC = \sum_{i=1}^{n} S_i \tag{16}$$

in which, the total number of optimal Pareto front members is denoted by n. Here, S_i will be equal to 1 if the i-th solution belongs to the method under consideration; otherwise, its value will be equal to 0. According to this criterion, the method with higher SCC values has a greater effect on the formation of the Pareto front. Table 3 shows the results of the SCC criterion obtained on the test dataset. According to this table, it can be seen that GSMOFOA was able to obtain a higher SCC value compared to other methods in most datasets, i.e., its greater participation in identifying the optimal Pareto front.

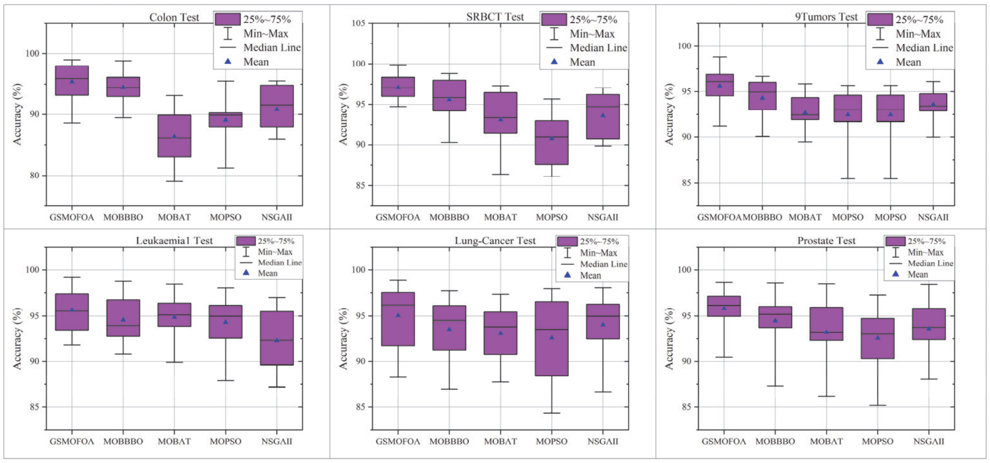

Fig. 6. Comparison between GSMOFOA solutions and other multi-objective algorithms on the *test* set in terms of accuracy distribution

Fig. 6 compares the classification accuracy distribution in the solutions obtained by GSMOFOA and other methods in the test data. Accordingly, in most datasets, GSMOFOA was able to obtain the best solution with respect to accuracy in 70 independent runs. Furthermore, the comparison between GSMOFOA method and other methods in terms of mean and median of classification accuracy of the obtained solutions indicates its better performance in the test dataset.

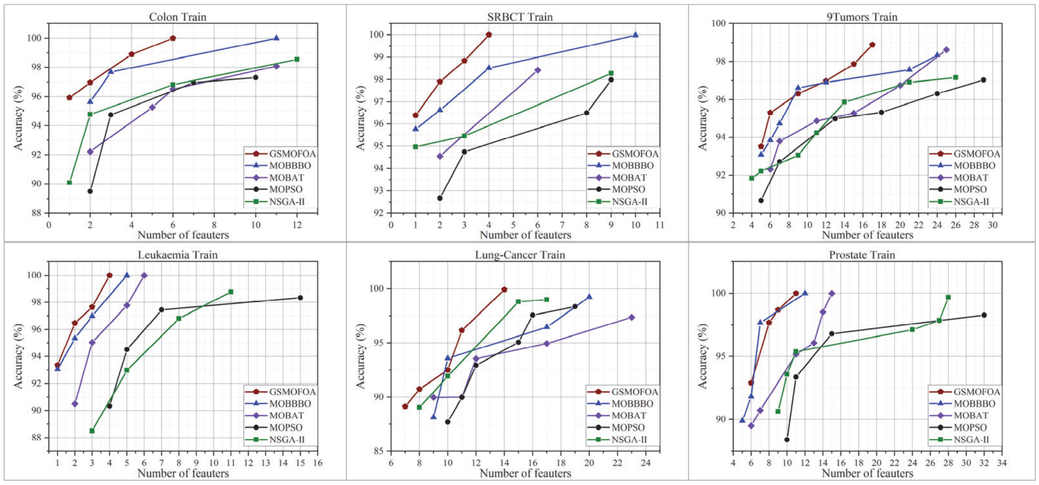

Fig. 7. A comparison between GSMOFOA *ND* solutions and other multi-objective algorithms on the *train* set

Fig. 7 shows that GSMOFOA *ND* solutions on the train dataset dominate the solutions obtained by other methods in respect of two criteria, namely the number of genes selected and classification accuracy, in most datasets. The only exception is the dataset Prostate, in which the MOBBBO method obtained more *ND* solutions than the proposed method. However, in this dataset, GSMOFOA was able to achieve 100% accuracy by selecting a smaller number of genes. The results of the SCC criterion presented in Table 4 confirm that GSMOFOA is more involved in the formation of the optimal Pareto front. In most cases, more than half of the optimal Pareto front is obtained by the solutions provided by this algorithm.

Table 4
Results of SCC measure on the *train* set

Datasets	MOFOA	NSGA-II	MOBBA	MOBAT	MOPSO
Colon	4	0	0	0	0
SRBCT	4	0	0	0	0
9-Tumors	5	0	2	0	0
Leukemia1	4	0	0	0	0
Lung-Cancer	4	0	1	0	0
Prostate	3	0	2	0	0

By studying the results of classification accuracy distribution of GSMOFOA solutions and other methods, it can be concluded that the GSMOFOA method obtains a solution with higher classification accuracy or similar to other methods in most datasets. Moreover, in cases where the maximum accuracy obtained by GSMOFOA and other methods is similar, the proposed method could obtain better solutions in terms of mean and median of classification accuracy.

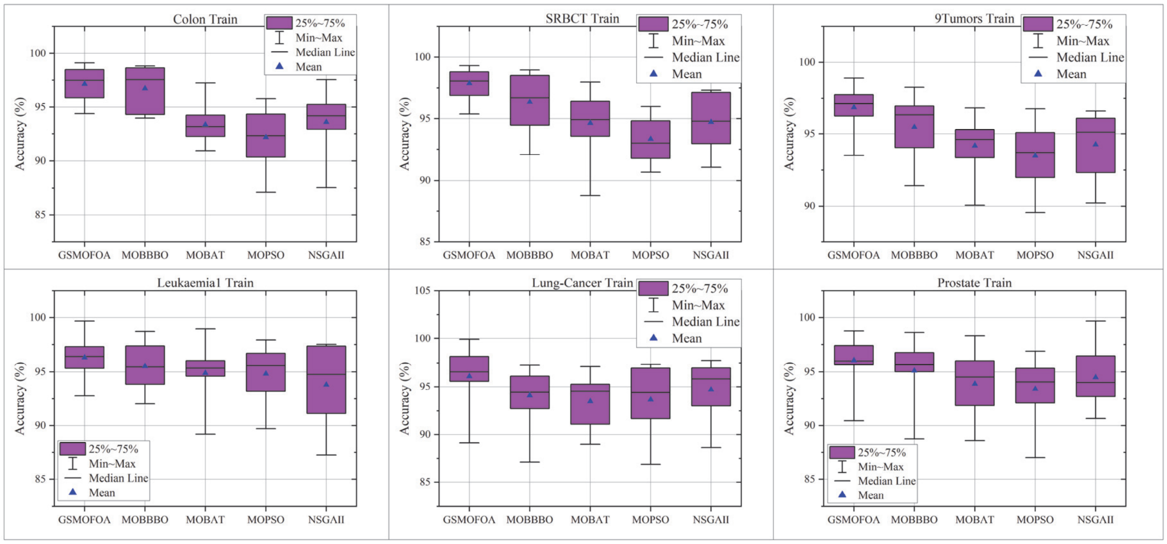

Fig. 8. Comparison of the accuracy of GSMOFOA solutions with those provided by other multi-objective algorithms on the *train* set

6.2. Further Investigation

To further investigate the effectiveness of GSMOFOA compared to other multi-objective algorithms, the Wilcoxon-rank sum statistical test was performed on hypervolume values. The hypervolume indicator is one of the criteria used to evaluate the performance of multi-objective algorithms, which can evaluate the diversity and convergence of the found Pareto front simultaneously (Auger et al., 2009; Brockhoff et al., 2008). According to Eq. (17), hypervolume values are calculated on each run on the two Pareto fronts of the train and test sets. In 70 independent runs, hypervolume values are calculated for the GSMOFOA, MOBBBO, NSGA-II, MOBAT, and MOPSO algorithms. Then, the Wilcoxon-rank sum test was performed, considering the significance level of 0.05 on the normalized Hypervolume values. Tables 5 and 6 present the Wilcoxon-rank sum results in both test and training modes, respectively. The comparisons are made from the top (i.e., GSMOFOA) to the left. The signs '+', '=', and '-' denote superior performance, similar performance, and worse performance of the GSMOFOA algorithm compared to the corresponding algorithm, respectively.

$$HV = volume\left(\cup_{i=1}^{|P|} v_i\right) \tag{17}$$

Based on Table 5, GSMOFOA provides significantly superior performance in most cases over test data, compared to other algorithms. Furthermore, MOBBBO and GSMOFOA algorithms have performed similarly in three datasets, namely 9-tumors, Leukemia1, and Prostate. Besides, there was no significant difference between the NSGA-II and GSMOFOA in the Lung-Cancer dataset as well as the MOBAT and GSMOFOA in the Prostate dataset.

Table 5
Wilcoxon rank-sum test of hypervolume ratios in the test data

Datasets	Colon MOFOA	SRBCT MOFOA	9-Tumors MOFOA	Leukemia1 MOFOA	Lung-Cancer MOFOA	Prostate MOFOA
NSGA-II	+	+	+	+	=	+
MOBBA	+	+	=	=	+	=
MOBAT	+	+	+	+	+	=
MOPSO	+	+	+	+	+	+

Table 6 presents the results of the Wilcoxon rank-sum test based on train data. As can be seen, the GSMOFOA outperforms other multi-objective methods in most datasets. Among the compared algorithms, only the MOBBBO algorithm could obtain similar results with GSMOFOA in three cases. In contrast, MOPSO presented worse results compared to the proposed method in all datasets.

Table 6
Wilcoxon rank-sum test of hypervolume ratios in the train data

Datasets	Colon MOFOA	SRBCT MOFOA	9-Tumors MOFOA	Leukemia1 MOFOA	Lung-Cancer MOFOA	Prostate MOFOA
NSGA-II	+	+	+	+	=	+
MOBBA	=	+	+	=	+	=
MOBAT	+	+	+	=	+	+
MOPSO	+	+	+	+	+	+

6.3. Space and Time Complexity Analysis

One of the most important criteria used to evaluate the efficiency of an algorithm is to analyze its space and time complexity. This section examines the complexity analysis of the proposed method.

6.3.1. Time Complexity Analysis

According to the algorithm 1, GSMOFOA has high time complexity in steps 8, 10, and 12, which affects the overall performance of the algorithm in terms of running time. In the eighth step, *ND* solutions are extracted from the forest and added to the repository. This step can be done with time order $O(N \times log^{M-1}N)$ (Jensen, 2003). The 10th step is to delete the dominated solutions from the Repository, which will be of a time order $O(M \times N \times R)$ in the worst case, where R determines the maximum repository size. Moreover, in the 12th step, crowding-distance is calculated for the *ND* solutions in the Repository. These calculations will be of order $O(M \times R \times logR)$. Therefore, it can be concluded that the time complexity of the GSMOFOA will be of order $O(M \times N \times R)$ in the worst-case scenario. Due to the size of the repository, this time complexity is much lower compared to non-dominated sorting-based methods (e.g. NSGA-II) of order $O(M \times N^2)$. Table 7 shows the results of the average computational time based on 70 independent runs for each dataset. The GSMOFOA has a lower average computational time compared to other algorithms. This shorter time can be attributed to the use of simple local and global seeding operators with lower computational overhead and the use of repository structure for *ND* solutions maintenance. Regarding GSMOFOA, it should be noted that FOA is single-parent in new solution generation operators and requires only one change to be applied to tree in an operator such as local seeding. Besides, since it uses a repository and single-parent structure, MOPSO has an average running time close to that of GSMOFOA. In contrast, methods such as NSGA-II, which are based on the non-dominated sorting method, require more running time.

Table 7
Comparison of results based on computational time (in minutes)

Datasets	MOFOA	NSGA-II	MOBBA	MOBAT	MOPSO
Colon	1.32	2.31	2.18	2.34	1.87
SRBCT	2.39	3.87	3.43	3.64	2.59
9-Tumors	2.81	5.44	5.37	5.49	2.94
Leukemia1	2.48	3.18	3.12	3.99	2.66
Lung-Cancer	4.25	7.61	7.37	7.52	4.47
Prostate	3.07	3.79	3.36	4.26	3.24

6.3.2. Space Complexity Analysis

The GSMOFOA has three sections for storing solutions: repository, candidate population, and forest. The space required to store the forest with space complexity $O(N \times d)$, where d represents the dimensions of each tree. The storage space required for the repository is of order $O(R \times d)$. Besides, candidate population storage requires the complexity $O(N_c \times d)$, where N_c indicates the number of solutions stored in the candidate population. Therefore, it can be concluded that the total space complexity of this algorithm will be of order $O(\max{(N, R, N_c)} \times d)$.

6.4. Discussion

According to the results of the present study, GSMOFOA outperformed other multi-objective methods in respect of the number of selected genes, classification accuracy, and running time. Owing to the use of a local seeding operator, GSMOFOA offers faster and better exploitation. Furthermore, thanks to the use of a global seeding operator, GSMOFOA can have better exploration compared to other multi-objective methods. GSMOFOA also uses a repository to maintain *ND* solutions, which is simpler and has less computational overhead compared to non-dominated sorting-based methods. Moreover, the use of crowding-distance and binary tournament selection leads the search to sparsely populated areas of Pareto front. As a result, the algorithm's ability to fully detect the optimal Pareto front will increase. In addition, the simultaneous optimization of kernel parameters of the SVM classifier led to an increase in accuracy and classification performance for data with fewer genes. Accordingly, the proposed method has also been able to increase classification accuracy by selecting fewer genes.

7. Conclusion

This paper aimed to develop a hybrid method based on the Fisher-Score method and multi-objective FOA to solve the gene selection problem in microarray data, which was successful by presenting the GSMOFOA method. GSMOFOA is based on concepts such as repository, crowding-distance, binary tournament selection, and simultaneous optimization of kernel parameters of the SVM classifier. The proposed method employs a repository to maintain *ND* solutions, with a much lower computational overhead compared to non-dominated sorting-based methods. Furthermore, GSMOFOA's ability to navigate and find different areas on the optimal Pareto front increased, due to the use of crowding-distance and binary tournament selection. To assess the effectiveness and efficiency of the proposed algorithm, a comparison was made between its results and those of four methods (i.e., a multi-objective hybrid of MOBBBO, MOBAT, NSGA-II, and MOPSO) on six high-dimensional microarray datasets. The results showed that GSMOFOA outperformed the other four methods on both the test and train sets, in most cases both in terms of dimensionality reduction and classification accuracy. Furthermore, the computational time study showed that GSMOFOA spent less computational time than other methods, thanks to the use of simple computational operators to generate new solutions and use the repository structure to maintain *ND* solutions. GSMOFOA was applied successfully to solve the gene selection problem in microarray data. However, the MOFOA has not yet been provided as an ensemble method to solve the gene selection problem; therefore, its ability to provide such solutions has not yet been investigated. Hence, future studies will examine the effect of using ensemble filters and ensemble classifiers in solving gene selection problem using MOFOA. Also, it seems that applying changes such as adding self-adaptation to local and global seeding operators of the MOFOA can improve the capabilities of this algorithm in solving problems with different dimensions. So, future studies will seek to improve the performance of seeding operators of MOFOA.

References

Almugren, N., & Alshamlan, H. (2019). A Survey on Hybrid Feature Selection Methods in Microarray Gene Expression Data for Cancer Classification. *IEEE Access*, *7*, 78533–78548.

Amaldi, E., & Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, *209*(1–2), 237–260.

Annavarapu, C. S. R., Dara, S., & Banka, H. (2016). Cancer microarray data feature selection using multi-objective binary particle swarm optimization algorithm. *EXCLI Journal*, *15*, 460–473.

Apolloni, J., Leguizamón, G., & Alba, E. (2016). Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, *38*, 922–932.

Auger, A., Bader, J., Brockhoff, D., & Zitzler, E. (2009). Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In *Proceedings of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms, FOGA'09* (pp. 87–102). ACM Press.

Baliarsingh, S. K., Vipsita, S., Muhammad, K., & Bakshi, S. (2019). Analysis of high-dimensional biomedical data using an evolutionary multi-objective emperor penguin optimizer. *Swarm and Evolutionary Computation*, *48*(May), 262–273.

Baliarsingh, S. K., Vipsita, S., Muhammad, K., Dash, B., & Bakshi, S. (2019). Analysis of high-dimensional genomic data employing a novel bio-inspired algorithm. *Applied Soft Computing Journal*, *77*, 520–532.

Banerjee, M., Mitra, S., & Banka, H. (2007). Evolutionary rough feature selection in gene expression data. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *37*(4), 622–632.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., Benítez, J. M., & Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. *Information Sciences*, *282*, 111–135.

Brockhoff, D., Friedrich, T., & Neumann, F. (2008). Analyzing hypervolume indicator based algorithms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5199 LNCS, pp. 651–660).

Chakraborty, G., & Chakraborty, B. (2013). Multi-objective optimization using pareto GA for gene-selection from microarray data for disease classification. *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, 2629–2634.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, *40*(1), 16–28.

Chuang, L. Y., Yang, C. H., & Yang, C. H. (2009). Tabu search and binary particle swarm optimization for feature selection using microarray data. *Journal of Computational Biology*, *16*(12), 1689–1703.

Chuang, L. Y., Yang, C. H., Wu, K. C., & Yang, C. H. (2011). A hybrid feature selection method for DNA microarray data. *Computers in Biology and Medicine*, *41*(4), 228–237.

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (Vol. 5). Boston, MA: Springer US.

Coello Coello, C. A., & Lechuga, M. S. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* (Vol. 2, pp. 1051–1056). IEEE.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Dashtban, M., & Balafar, M. (2017). Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts. *Genomics*, *109*(2), 91–107.

Dashtban, M., Balafar, M., & Suravajhala, P. (2018). Gene selection for tumor classification using a novel bio-inspired multi-objective approach. *Genomics*, *110*(1), 10–17.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (Vol. 16). John Wiley & Sons. Retrieved from https://www.wiley.com/en-us/Multi+Objective+Optimization+using+Evolutionary+Algorithms-p-9780471873396

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Gangavarapu, T., & Patil, N. (2019). A novel filter–wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets. *Applied Soft Computing Journal*, *81*, 105538.

Ghaemi, M., & Feizi-Derakhshi, M.-R. (2014). Forest Optimization Algorithm. *Expert Systems with Applications*, *41*(15), 6676–6687.

Ghaemi, M., & Feizi-Derakhshi, M.-R. (2016). Feature selection using Forest Optimization Algorithm. *Pattern Recognition*, *60*, 121–129.

Gu, Q., Li, Z., & Han, J. (2012). Generalized Fisher Score for Feature Selection. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 266–273. Retrieved from http://arxiv.org/abs/1202.3725

Hall, M. A. (1999). Correlation-based Feature Selection for Machine Learning. The University of Waikato.

Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization. *Information Sciences*, *422*, 462–479.

Hasnat, A., & Molla, A. U. (2017). Feature selection in cancer microarray data using multi-objective genetic algorithm combined with correlation coefficient. *Proceedings of IEEE International Conference on Emerging Technological Trends in Computing, Communications and Electrical Engineering, ICETT 2016*.

He, X., Cai, D., & Niyogi, P. (2005). Laplacian Score for feature selection. *Advances in Neural Information Processing Systems*, 507–514.

Jensen, M. T. (2003). Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(5), 503–515.

Jonnalagadda, S., & Srinivasan, R. (2008). Principal components analysis based methodology to identify differentially expressed genes in time-course microarray data. *BMC Bioinformatics*, *9*(1), 267.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *Proceedings of 2014 Science and Information Conference, SAI 2014*, (August 2014), 372–378.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324.

Lai, C. M. (2018). Multi-objective simplified swarm optimization with weighting scheme for gene selection. *Applied Soft Computing Journal*, *65*, 58–68.

Lee, C. P., & Leu, Y. (2011). A novel hybrid feature selection method for microarray data analysis. *Applied Soft Computing Journal*, *11*(1), 208–213.

Li, X., & Yin, M. (2013). Multiobjective binary biogeography based optimization for feature selection using gene expression data. *IEEE Transactions on Nanobioscience*, *12*(4), 343–353.

Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, *256*, 56–62.

Miao, J., & Niu, L. (2016). A Survey on Feature Selection. *Procedia Computer Science*, *91*(Itqm), 919–926.

Mishra, S., Shaw, K., & Mishra, D. (2012). A New Meta-heuristic Bat Inspired Classification Approach for Microarray Data. *Procedia Technology*, *4*, 802–806.

Mohamad, M. S., Omatu, S., Deris, S., & Yoshioka, M. (2008). Multi-objective optimization using genetic algorithm for gene selection from microarray data. *Proceedings of the International Conference on Computer and Communication Engineering 2008, ICCCE08: Global Links for Human Development*, 1331–1334.

Mohapatra, S., Aryendu, I., Panda, A., & Padhi, A. K. (2018). A Modern Approach for Load Balancing Using Forest Optimization Algorithm. In *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 85–90). IEEE.

Motieghader, H., Najafi, A., Sadeghi, B., & Masoudi-Nejad, A. (2017). A hybrid gene selection algorithm for microarray cancer classification using genetic algorithm and learning automata. *Informatics in Medicine Unlocked*, *9*(August), 246–254.

Mukhopadhyay, A., Member, S., Maulik, U., & Member, S. (2014). A Survey of Multiobjective Evolutionary Algorithms for Data Mining : Part I, *18*(1), 4–19.

Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation*, *54*(February), 100663.

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

Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. *Computers in Biology and Medicine*, *112*(February), 103375.

Shahbeig, S., Rahideh, A., Helfroush, M. S., & Kazemi, K. (2018). Gene selection from large-scale gene expression data based on fuzzy interactive multi-objective binary optimization for medical diagnosis. *Biocybernetics and Biomedical Engineering*, *38*(2), 313–328.

Sharma, A., & Rani, R. (2019). C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. *Computer Methods and Programs in Biomedicine*, *178*, 219–235.

Shen, Q., Shi, W. M., & Kong, W. (2008). Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. *Computational Biology and Chemistry*, *32*(1), 53–60.

Shukla, A. K., Singh, P., & Vardhan, M. (2018). A hybrid gene selection method for microarray recognition. *Biocybernetics and Biomedical Engineering*, *38*(4), 975–991.

Shukla, A. K., Singh, P., & Vardhan, M. (2020). Gene selection for cancer types classification using novel hybrid metaheuristics approach. *Swarm and Evolutionary Computation*, *54*(December 2019).

Sierra, M. R., & Coello Coello, C. A. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ∈-Dominance. In *International conference on evolutionary multi-criterion optimization* (pp. 505–519). Springer.

Taguchi, G., Chowdhury, S., & Wu, Y. (2005). *Taguchi's quality engineering handbook*. Wiley.

Tyagi, V., & Mishra, A. (2013). A Survey on Different Feature Selection Methods for Microarray Data Analysis. *International Journal of Computer Applications*, *67*(16), 36–40.

Vafaee Sharbaf, F., Mosafer, S., & Moattar, M. H. (2016). A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization. *Genomics*, *107*(6), 231–238.

Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural Computing and Applications*, *24*(1), 175–186.

Yang, C. H., Chuang, L. Y., & Yang, C. H. (2010). IG-GA: A hybrid filter/wrapper method for feature selection of microarray data. *Journal of Medical and Biological Engineering*, *30*(1), 23–28.

Zhu, Z., Ong, Y. S., & Dash, M. (2007). Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, *40*(11), 3236–3248.