Analysis of high-dimensional biomedical data using an evolutionary multi-objective emperor penguin optimizer

Article *in* Swarm and Evolutionary Computation · May 2019

DOI: 10.1016/j.swevo.2019.04.010

CITATIONS
43

READS
322

4 authors, including:

Swati Vipsita
International Institute of Information Technology, Bhubaneswar
40 PUBLICATIONS **372** CITATIONS

SEE PROFILE

Khan Muhammad
Sungkyunkwan University
283 PUBLICATIONS **12,061** CITATIONS

SEE PROFILE

Analysis of high-dimensional biomedical data using an evolutionary multi-objective emperor penguin optimizer

Santos Kumar Baliarsingh[a], Swati Vipsita[a], Khan Muhammad[b,*], Sambit Bakshi[c]

[a]*Department of Computer Science and Engineering, International Institute of Information Technology, Bhubaneswar, India, baliarsingh.santosh@gmail.com, swati@iiit-bh.ac.in*
[b]*Assistant Professor, Department of Software, Sejong University, Seoul, 143-747, Republic of Korea, khanmuhammad@sju.ac.kr*
[c]*Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India*

Abstract

Over the last two decades, there has been an expeditious expansion in the generation and exploration of high-dimensional biomedical data. Identification of biomarkers from the genomics data poses a significant challenge in microarray data analysis. Therefore, for the methodical analysis of the genomics dataset, it is paramount to develop some effective algorithms. In this work, a multi-objective version of the emperor penguin optimization (EPO) algorithm with chaos, namely, multi-objective chaotic EPO (MOCEPO) is proposed. The suggested approach extends the original continuous single objective EPO to a competent binary multi-objective model. The objectives are to minimize the number of selected genes (NSG) and to maximize the classification accuracy (CA). In this work, Fisher score and minimum redundancy maximum relevance (mRMR) are independently used as initial filters. Further, the proposed MOCEPO is employed for the simultaneous optimal feature selection and cancer classification. The proposed algorithm is successfully experimented on seven well-known high-dimensional binary-class as well as multi-class datasets. To evaluate the effectiveness, the proposed method is compared with non-dominated sorting genetic algorithm (NSGA-II), multi-objective particle swarm optimization (MOPSO), chaotic version of GA for multi-objective optimization (CGAMO), and chaotic MOPSO methods. The experimental results show that the proposed framework achieves better CA with minimum NSG compared to the existing schemes. The presented approach exhibits its efficacy with regard to NSG, accuracy,

*Corresponding Author: Khan Muhammad; Email: khanmuhammad@sju.ac.kr

sensitivity, specificity, and F-measure.

Keywords:
Microarray, Fisher score, MOCEPO, mRMR, Kernel ridge regression

1. Introduction

The rapidly growing DNA microarray technology has enabled the researchers to measure the expression level of thousands of genes simultaneously in a single experiment [1]. The biomarker genes extracted from the microarray data helps in the clinical diagnosis, prognosis, and treatment of cancer. However, the high dimensionality of the microarray data increases the computational overhead and hence poses a significant challenge in the biomedical data analysis. To overcome this issue, the irrelevant and redundant genes need to be discarded using some feature selection [2, 3] techniques. In fact, it is expected that selecting the relevant genes reduces the size of the gene expression data and enhances the CA.

Several gene selection (GS) techniques have been suggested in the literature to identify the useful genes present in the microarray data. These GS techniques can be broadly classified into three categories, namely, filter methods, wrapper methods, and hybrid methods [4, 5]. Filter-based methods select the relevant genes from the original gene set based on some statistical characteristics. Despite the simplicity and computationally efficiency, filter techniques are incapable of exploiting the relationship among the genes, thereby reducing the overall accuracy. On the other hand, the wrapper-based techniques employ the knowledge of the classifiers, namely, kernel ridge regression (KRR) [6], support vector machine (SVM) [7], K-nearest neighbor (KNN) [8, 9], Naive Bayes (NB) [10], radial basis functions neural networks (RBFN) [11], and decision tree (DT) [12] to find the bio-markers. The wrapper models use bio-inspired algorithms to identify the optimal solutions by analyzing the search area from a set of solutions (population). The evolutionary algorithms such as GA [13–15], differential evolution (DE) [16, 17], artificial bee colony algorithm (ABC) [18], genetic bee colony optimization (GBC) [19], ant colony optimization (ACO) [20], salp swarm algorithm (SSA) [21], firefly algorithm (FA) [22], bidirectional elitist optimization [23], and PSO [24–28] have been successfully utilized for solving numerous feature selection problems. These methods are competent of learning the association among the genes and therefore, lead to better CA. The hybrid methods use the merits of the both by first employing a filter method to reduce the NSG and then

applying the wrapper method to explore the optimal gene subset.

In order to select the biomarker genes in a faster and efficient manner, multi-objective methods have been designed. In the last two decades, several multi-objective optimization methods, namely, MOPSO [29], CMOPSO [30], NSGA-II [31], CGAMO [32], multi-objective FA (MOFA) [33], multi-objective teaching-learning-based optimization (MOTLBO) [34], multi-objective gravitational search algorithm (MOGSA) [35], and multi-objective differential evolution (MODE) [36] algorithms have been proposed. These methods prove their effectiveness in solving multi-objective problems. Though all the above mentioned algorithms are competent enough in solving a specific task, they can not fix all optimization problems with dissimilar characteristics [37]. Hence, there always remain a room for a novel method which can solve a problem that can not be addressed by the present methods.

The two important phases of any metaheuristic algorithm are diversification and intensification [38, 39]. Diversification makes sure that the algorithm searches the various promising areas in a certain search space, whereas intensification investigates the optimal solutions around the promising areas which is resulted by the diversification phase [40]. The proper balancing between the above two phases is important for any optimization problem, which motivates us to employ the EPO algorithm. The second motivation is the 'no free lunch theorem', which says that none of the existing metaheuristic is capable of solving all optimization problems [37].

In this paper, a novel multi-objective version of the EPO algorithm, namely, MOCEPO is proposed. EPO [41] is a newly developed meta-heuristic method, originally designed for single objective optimization problems. In this work, we have extended the single objective EPO to multi-objective binary EPO by utilizing the multi-objective operators, namely, non-dominated sorting, and crowding distance. The two objectives of our problem are to minimize the NSG, and to maximize the CA. The CA is computed by the KRR classifier. In order to reduce the redundant genes, Fisher score and mRMR filters are employed independently.

The five major contributions of the suggested work are highlighted as:

- For the first time, a multi-objective version of the EPO algorithm is proposed.
- Chaos theory is introduced in the MOEPO for faster convergence.
- Multi-objective operator like non-dominated sorting is incorporated to rank the pareto optimal solutions.
- Selection of the fittest solution is carried out by the crowding distance operator.
- The proposed method is applied for simultaneous GS and cancer classification.

The proposed approach is implemented on seven standard datasets. The performance of the proposed framework is evaluated in terms of CA, NSG, F-measure, specificity, Matthews correlation coefficient (MCC), and sensitivity. The results show that our method can not only achieve higher CA, but also reduces the NSG effectively.

The remainder of the paper is organized as follows. Section 2 explains the methods used along with the proposed work. Experimental setup and performance metrics are presented in Section 3. The results of the work are presented and discussed in Section 4. Finally, we conclude the work in Section 5.

2. Method

2.1. Pre-selection of genes

To effectively filter out the highly redundant and irrelevant genes, usually, filter-based gene ranking algorithms are used. In this paper, we have employed Fisher score [42] and mRMR [43] filters separately for gene pre-selection, which have reliable performance in segregating the relevant genes [44, 45]. As compared to the methods like T-test, Z-score, and information gain, Fisher score and mRMR produce superior results [45, 46]. Nonetheless, every technique has its advantages that influence the stability of final results.

2.1.1. Gene pre-selection by Fisher score

In Fisher score filter, statistical characteristic of every gene in different classes is used as a potential measure of discriminatory power. In this method, the relevant gene selection is carried out in a manner where the intra-class distance of the samples is kept minimum and the inter-class distance is kept maximum. Based on their Fisher score, a threshold of top 500 genes (as suggested by [44]), are selected for the next stage. The Fisher score filter selects a matrix of size $S \times F$ as input; F denotes the size of genes, and S denotes the size of instances. Further, the Fisher score of a gene g_i is computed as

$$FScore(g_i) = \frac{\sum_{k=1}^{c} N_k(\mu_k^i - \mu^i)}{\sum_{k=1}^{c} N_k(\sigma_k^i)^2} \tag{1}$$

In the above equation, N_k represents the sample size in class k, μ_k^i is the mean of the class k with respect to the gene i, and σ_k^i is the standard deviation of the class k with respect to the gene i. c represents the maximum number of target classes. The selection of relevant genes using the Fisher score method is shown in Algorithm 1.

Algorithm 1 Gene selection using Fisher score

Input: $M : S \times F$ size gene matrix; where S denotes size of samples and F denotes size of genes

Output: Identify top t genes

1: **Begin**
2: **for** every gene g_i **do**
3: ▷ i=1,2,...,F
4: Compute the Fisher score ($FScore_i$) value employing Eq. (1)
5: **end for**
6: Sort the genes in descending order based on their Fisher score ($FScore_i$)
7: Pick the top t genes
8: **End**

2.1.2. Gene pre-selection by mRMR

mRMR is a state-of-the-art filter technique that deals with not only the redundancy between the genes but also the relevance between the gene and class. The primary goal of employing the mRMR gene selection strategy is to select a subset of genes from the entire gene set, which either have combined the maximum relevance on the target class or have the minimum redundancy on the selected gene subset. Based on the two objectives, a threshold of top 500 genes are selected for the next stage. The relevance R_l of a set of selected genes F is defined as

$$R_l = max \frac{1}{|F|} \sum_{g_i \epsilon F} I(g_i, c) \tag{2}$$

where $I(g_i, c)$ denotes the mutual information value between an individual gene g_i that belongs to F and the class c. The redundancy R_d of a set of selected genes F is defined as

$$R_d = min \frac{1}{|F|^2} \sum_{g_i, g_j \epsilon F} I(g_i, g_j) \tag{3}$$

where $I(g_i, g_j)$ denotes the mutual information between the i^{th} and j^{th} genes that measures the mutual dependency between the two genes. The optimal combination of both R_l and R_d selects the gene g_i by the mRMR score.

$$mRMR\_Score(g_i) = R_l - R_d \tag{4}$$

Algorithm 2 Gene selection using mRMR

Input: $I : S \times F$ size gene matrix; where S denotes size of samples and F denotes size of genes

Output: Identify top t genes

Begin
for every gene g_i **do**
▷ i=1,2,...,F
relevance = Mutual_Information(g_i, class)
redundancy=0
for every gene g_j **do**
redundancy = redundancy + Mutual_Information(g_i, g_j)
end for
mRMR_Score(g_i)= relevance - redundancy
end for
Sort the genes in descending order based on their mRMR_Score
Pick the top t genes
End

The selection of relevant genes using the mRMR method is shown in Algorithm 2.

Based on their classification performance, the genes selected by either Fisher score filter or mRMR filter are then used by the proposed method to select the optimal gene subset.

2.2. Emperor penguin optimization (EPO)

EPO algorithm is a newly developed swarm intelligence technique proposed by Dhiman and Kumar [41] in 2018. The algorithm imitates the social huddling behavior of emperor penguins (solutions) in nature. During the huddling process, the solutions update their positions to obtain the effective mover (best solution) using the following mathematical expressions:

$$T' = \left(T - \frac{max\_iteration}{t - max\_iteration} \right) \tag{5}$$

$$T = \begin{cases} 0 & , if\ R \geq 0.5 \\ 1 & , if\ R < 0.5 \end{cases} \tag{6}$$

$$\overrightarrow{D_{ep}} = \left| S(\overrightarrow{A}) \cdot \overrightarrow{P}(t) - \overrightarrow{C} \cdot \overrightarrow{P_{ep}}(t) \right| \tag{7}$$

where T' is the temperature profile around the hurdle, t denotes the current iteration, R is a random number in the range of [0, 1], $\overrightarrow{A}$ and $\overrightarrow{C}$ are two vectors responsible for avoiding the collision between the neighbor EPs, $\overrightarrow{P}$ represents the position vector of the fittest solution, $\overrightarrow{P_{ep}}$ denotes the position vector of other solutions, $S()$ represents the social force of solutions which is responsible for converging towards the best solution, and $\overrightarrow{D_{ep}}$ denotes the distance between each candidate solution and the best solution. The vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ can be mathematically computed as

$$\overrightarrow{A} = (M \times (T' + P_{grid}(Accuracy)) \times r_1) - T' \tag{8}$$

$$P_{grid}(Accuracy) = \left| \overrightarrow{P} - \overrightarrow{P_{ep}} \right| \tag{9}$$

$$\overrightarrow{C} = r_2 \tag{10}$$

where M indicates the movement parameter which keeps a gap among the solutions to avoid collision. M is assigned a value as 2. $P_{grid}(Accuracy)$ represents the absolute difference between the candidate solutions and the best solution. r_1 and r_2 are two random numbers in the range of [0, 1]. Function $S(\overrightarrow{A})$ in Eq. (7) is computed as

$$S(\overrightarrow{A}) = \left(\sqrt{f \cdot e^{-\frac{t}{l}} - e^{-t}} \right)^2 \tag{11}$$

In the above equation, e represents the expression function. The two control parameters f and l are responsible for proper diversification and intensification whose values remain in the range of [2, 3] and [1.5, 2], respectively. The position vectors of solutions are updated by using Eq. (12).

$$\overrightarrow{P_{ep}}(t+1) = \overrightarrow{P}(t) - \overrightarrow{A} \cdot \overrightarrow{D_{ep}} \tag{12}$$

2.3. Proposed multi-objective chaotic EPO algorithm

Typically, in most of the established meta-heuristic methods, the initialization of parameters are done randomly with uniform or Gaussian distribution. However, in recent times, chaos theory has gained popularity to improve the parameters of population-based algorithms [47]. The properties of chaos theory are same as randomness, however, with superior statistical and dynamical nature. Chaos theory has three important characteristics, namely, sensitivity, stochasticity, and ergodicity to preliminary situations. Sensitivity to initial condition refers to any minor variation in the initial starting points, may cause different behavior. Stochasticity is the process of replacing the random variables with the values of chaotic maps. Lastly, ergodicity is defined as the capacity of chaotic variables to explore non-repeatedly all states within a particular range. The union of these properties guarantees the diversity of the resultant solutions and therefore boosts the performance of the population-based methods [48].

Amid the various chaotic maps [49], logistic function is a popularly used chaotic map, and is described as

$$z_{t+1} = 4 \cdot z_t \cdot (1 - z_t) \tag{13}$$

In the above equation, z_t denotes the chaotic map value obtained at iteration t.

In the proposed approach, the three random variables of EPO algorithm, such as, R, r_1, and r_2 are replaced with the logistic chaotic variables as they are responsible to retain an equilibrium between the exploitation and exploration process. So Eq. (6), (8), and (10) are updated as

$$T = \begin{cases} 0 & , if\ z_t \geq 0.5 \\ 1 & , if\ z_t < 0.5 \end{cases} \tag{14}$$

$$\overrightarrow{A} = (M \times (T' + P_{grid}(Accuracy)) \times z_t) - T' \tag{15}$$

$$\overrightarrow{C} = z_t \tag{16}$$

where t represents the current iteration, z_t denotes the t^{th} chaotic iteration value, and the starting value of z_0 is initialized randomly in the range of [0,1].

2.3.1. Multi-objective operators

• **Non-dominated sorting**

This technique is used to rank the pareto optimal solutions. Let the number of objective functions to be optimized is denoted by M, and the number of solutions is denoted by n.

i. **Domination**

A solution x_1 dominates another solution x_2, if two conditions are satisfied.

1. The solution x_1 is not worse than the solution x_2 in all objectives, denoted by $obj_m(x_1) \not< obj_m(x_2)$ for all $m = 1, 2, ..., M$.
2. The solution x_1 is strictly better than the solution x_2 in at least one objective, denoted as $obj_{\overline{m}}(x_1) > obj_{\overline{m}}(x_2)$ for at least $\overline{m} \in \{1, 2, ..., M\}$.

ii. **Non-domination**

A solution x_a in n is said to be non-dominated only if there does not exist a solution x_b in n which dominates x_a.

In the same manner, every solution in n is compared with other solutions and the non-dominated solutions are extracted from n and assigned a level (rank) 1. The rest of the solutions in n are again sorted in the same process, then the non-dominated solutions are extracted and assigned level 2. This process continues until all the solutions in n are assigned a level. A set of solutions with the same level is termed as a pareto front (PF).

• **Crowding distance**

It is a measure of density of the solutions in the proximity of a particular solution. For a specific pareto front PF, let the total number of solutions on that front be Z ($Z = |PF|$). Now for every candidate solution in that PF, the crowding distance cd is computed as per the following procedure.

Step 1: For every solution i in the set, first assign $cd_i = 0$.

Step 2: For every objective function $m = 1, 2, ..., M$, sort the solutions in the PF in the worst order of the objective function obj_m.

Step 3: For $m = 1, 2, ..., M$, assign a larger crowding distance (∞) to the boundary solutions ($cd_1 = cd_2 = \infty$). For $i = 2$ to $Z - 1$, compute the cd as per the following equation.

$$cd_i^m = cd_i^m + \frac{obj_m^{i+1} - obj_m^{i-1}}{obj_m^{max} - obj_m^{min}} \tag{17}$$

where i denotes a solution in the sorted list. obj_m^i is the value of the m^{th} fitness function of i^{th} solution. A solution having the highest level (1 is highest) is selected as the fittest solution for the next generation. If multiple solutions lie on a same pareto front, then the solution with the maximum cd is chosen as the best solution.

2.3.2. Mathematical modelling

A multi-objective optimization problem can be mathematically formulated as

$$Optimize : F(x) = [f_1(x), f_2(x), ..., f_M(x), \;\; M > 1 \;\; and \;\; x \epsilon X] \tag{18}$$

Subject to:

$$G_i(x) \geq 0, \quad i = 1, 2, ..., A \tag{19}$$

$$H_i(x) = 0, \quad i = 1, 2, ..., B \tag{20}$$

$$L_i \leq x_i \leq U_i, \quad i = 1, 2, ..., D \tag{21}$$

where M denotes the number of objective functions, D represents the number of variables, A signifies the number of inequality constraints, B represents the number of equality constraints, G_i represents the i^{th} inequality constraint, H_i denotes the i^{th} equality constraint, and $[L_i, U_i]$ are the lower and upper bounds of i^{th} variable.

Here, we have two objective functions to be optimized, namely, f_1 and f_2.

$$f_1 = min(NSG) \tag{22}$$

$$f_2 = max(CA) \tag{23}$$

The MOCEPO algorithm is designed with binary representation which will be suitable for feature (gene) selection. To solve the multi-objective optimization problems, MOCEPO borrows the properties of multi-objective operators of NSGA-II, which is an established fast and efficient multi-objective method in the literature. The MOCEPO algorithm employs the $non\_dominated\_sort$ (nds) operator and the $crowding\_distance$ (cd) operator to select the fittest solution. The solution with highest level (level=1) and largest cd value is selected as the fittest solution. This selection criteria is used because the solution in the less crowded area of the objective space may guide the search process. Once the fittest solution is obtained, the rest of the solutions are modified as per the Eq. (12).

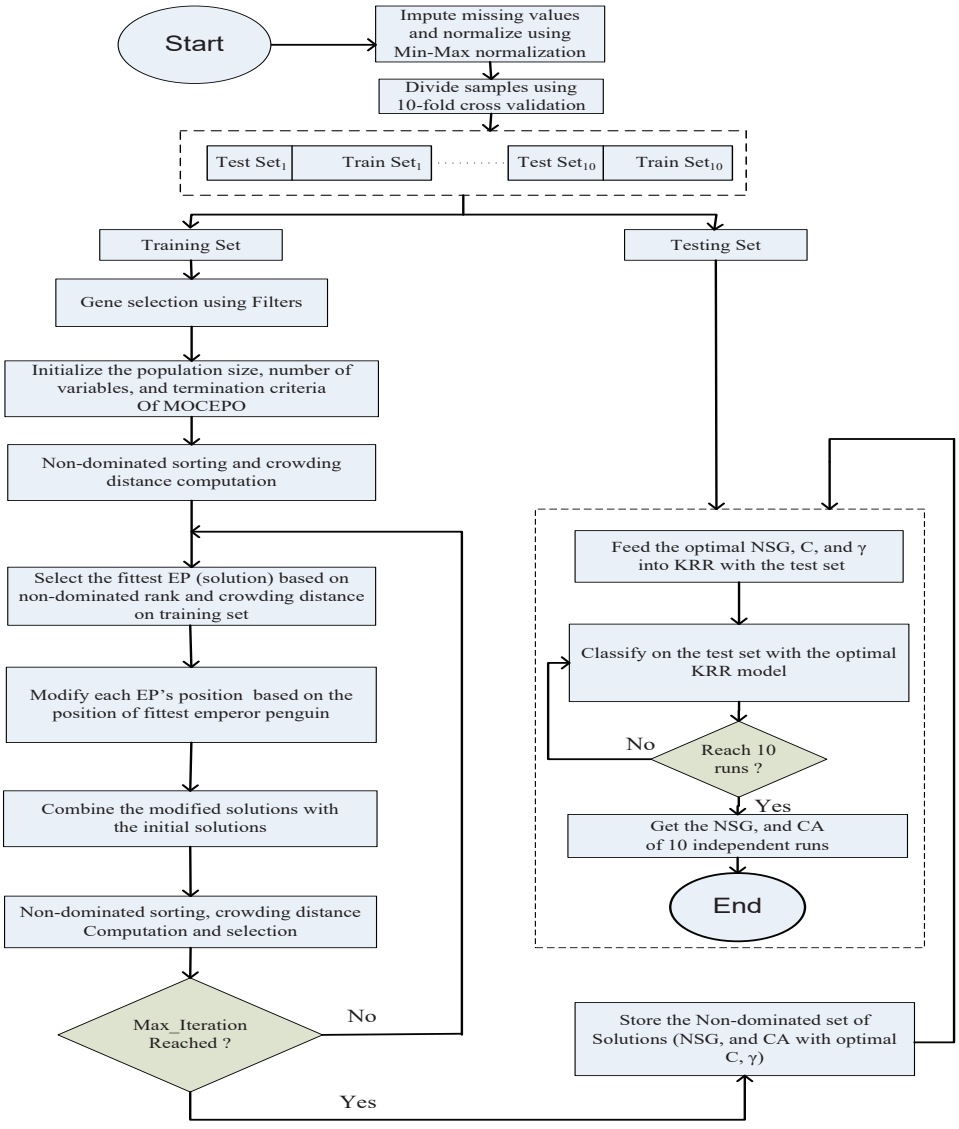


Figure 1 Flowchart of the proposed MOCEPO model

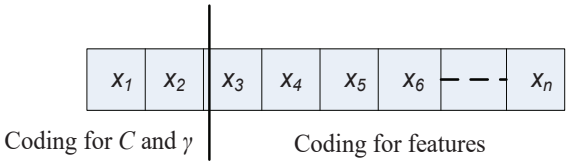


Figure 2 Coding scheme of a solution

After all the solutions are updated, the modified solutions (n) are combined with the initial population (n), resulting the total number of solutions to $2n$. These $2n$ solutions are again leveled based on their non-dominated pareto fronts and crowding distance operator. On the basis of the new leveling and cd value, top-n solutions are selected out of $2n$ solutions.

Figure 1 presents a detailed flowchart of MOCEPO. Suppose d represents the dimension of each solution of MOCEPO, from the d number of bits, initial 2 bits are used to code C and γ of KRR, and the last $d-2$ bits are utilized to code the features (see Figure 2). For these last $d-2$ bits, used for feature selection, the transformation from continuous value to binary value is done by a transfer function represented as

$$x_j^m = \begin{cases} 1 & \text{if } \ logsig(x_j^m) \geqslant 0.5 \\ 0 & otherwise \end{cases} \tag{24}$$

where

$$logsig(x_j^m) = \frac{1}{1+exp(-x_j^m)} \tag{25}$$

During search process, if any solution goes out of bound, that solution is amended within a range of [-1, 1] as

$$\overrightarrow{P_{ep}}(t+1) = \begin{cases} -1, & if \ \overrightarrow{P_{ep}}(t+1) < -1 \\ 1, & if \ \overrightarrow{P_{ep}}(t+1) > 1 \end{cases} \tag{26}$$

The pseudo-code of the proposed technique is demonstrated in Algorithm 3. The accuracy of each solution is evaluated using the KRR classifier with a 10-fold cross validation (CV). For a single run of the algorithm and for a single solution, the MOCEPO method computes the objective functions (obj_1, obj_2) only once in every iteration. Hence, the aggregate number of function evaluation = number of runs $\times$ number of candidate solutions $\times$ number of iterations.

Algorithm 3 Pseudo code of the proposed MOCEPO algorithm

Input: M : Initial random population of EPs; $\overrightarrow{P_{ep}}(y);\ y = 1, 2, ..., n$

Output: Select the optimal solution $\overrightarrow{P}$

Begin
Initialize the random solutions, $max\_iteration$, f, and l
Calculate the values of the objective functions $< obj_1, obj_2 >$ of each solution
Extract the non-dominated solutions and sort the solutions into different non-domination levels
Assign a rank to each non-dominated level (1 is the best rank)
Select the best solution based on the non-domination level and crowding distance
$t = 1$
while ($t < max\_iteration$) **do**
    Compute the value of T' and T using Eq. (5) and (6), respectively
    for i=1 to $number\_of\_solutions$ **do**
        for j=1 to $number\_of\_solutions$ **do**
            Calculate the vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ using Eq. (8) and (10), respectively
            Calculate the value of $S(\overrightarrow{A})$ using Eq. (11)
            Update the position of each current solution with respect to the fittest non-dominated solution
        end for
    end for
    Update the values of T', $\overrightarrow{A}$, $\overrightarrow{C}$, $and\ S()$
    Limit the solutions that go out of bound using Eq. (26)
    Combine the updated solutions with the previous solutions (n and n counts to $2n$)
    Compute the objective function values $< obj_1, obj_2 >$ of all $2n$ number of solutions
    Find the non-dominated solutions and sort the solutions into different non-domination levels
    Assign a rank corresponding its non-domination level (rank 1 is given to the best level)
    Select the best n number of solutions out of $2n$ solutions based on their non-domination level and crowding distance
    $t = t + 1$
end while
Return the fittest solution $\overrightarrow{P}$
End

Table 1 Information about the microarray datasets

Dataset	No. of Genes	No. of Samples	No. of Classes
Leukemia [50]	7129	72	2
Colon tumor [52]	2000	62	2
Ovarian cancer [53]	15154	253	2
ALL-AML-3 [51]	7129	72	3
Lymphoma-3 [51]	4026	62	3
SRBCT [51]	2308	83	4
Lung cancer-5 [54]	12600	203	5

3. Experimental design

3.1. Datasets

The proposed method is applied on seven standard microarray cancer datasets [50, 51], listed in Table 1. Out of the seven datasets, three datasets belong to binary-class and four datasets belong to multi-class. Prior to feature selection by the Fisher score and mRMR filters, min-max normalization in the range of [-1,1] is applied on the whole dataset.

3.2. Experimental setup

MATLAB 2017b is used to carry out the experiments with 8GB of main memory and Core i5 processor (2.70 GHz). The simulation results are evaluated using 10-fold CV [55] to approximate the prediction accuracy of the solutions. Further, to overcome the randomness properties, the proposed method is executed for 10 times, and the mean of 10 independent trials is considered as the final result.

For a fair comparison, experiments have been conducted on four popular multi-objective meta-heuristics, namely, NSGA-II, MOPSO, CGAMO, and CMOPSO. For all the experiments, same parameter values have been considered, which is presented in Table 2. In this paper, radial basis function is used as the kernel for KRR. Lastly, to prove the competency of the proposed model, it is compared with nineteen benchmark methods.

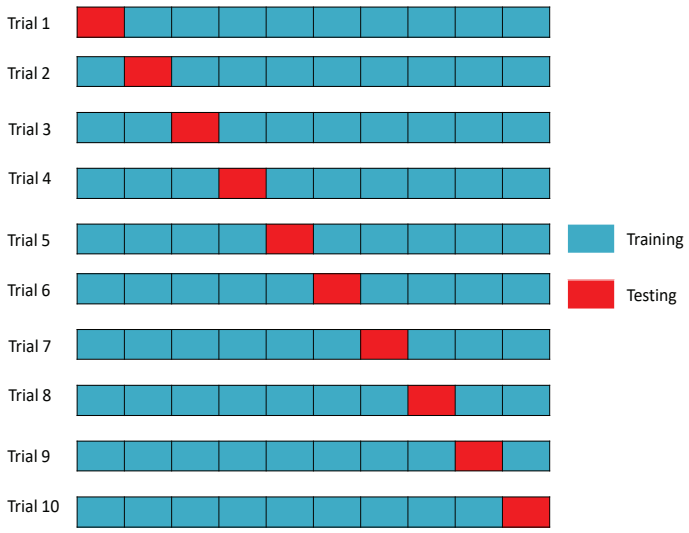


Figure 3 Illustration of 10-fold cross validation setting for a single run

Table 2 Control parameters

Parameters	Value(s)
Maximum iterations (MOCEPO, NSGA-II, CGAMO, MOPSO, CMOPSO)	100
Size of population (MOCEPO, NSGA-II, CGAMO, MOPSO, CMOPSO)	30
Probability of crossover (NSGA-II, CGAMO)	0.95
Probability of mutation (NSGA-II, CGAMO)	0.05
Maximum velocity (MOPSO, CMOPSO)	65%
Inertia weight (MOPSO, CMOPSO)	1
C_1 and C_2 (MOPSO, CMOPSO)	2.05

3.3. Performance metrics

The efficacy of the suggested approach is evaluated by six performance measures: sensitivity, specificity, F-measure, MCC, accuracy, and Kappa

- *Confusion matrix*: Sensitivity, specificity, and accuracy can be computed from the confusion matrix shown in Table 3.
- *F-measure*: It is a derived effectiveness measurement. The resultant value is interpreted as a weighted average of the precision and recall. It can be calculated from the confusion matrix as

$$F\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{27}$$

Table 3 Confusion matrix

	Target class		
	neg	**pos**	
Output class			
Classified as neg	tn	fn	$npv = \frac{tn}{tn + fn}$
Classified as pos	fp	tp	$Precision = \frac{tp}{tp + fp}$
	$Specificity = \frac{tn}{tn + fp}$	$Sensitivity(recall) = \frac{tp}{tp + fn}$	$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}$

- *Kappa*: The kappa statistic is used to test interrater reliability. The value of kappa can range from -1 to +1. A value of 1 implies perfect agreement and values less than 1 imply less than perfect agreement. It can be calculated as:

$$Kappa = (Observed\_accuracy - Expected\_accuracy)/(1 - Expected\_accuracy) \tag{28}$$

$$Observed\_accuracy = (tp + tn)/(tp + fp + tn + fn) \tag{29}$$

$$Expected\_accuracy = \frac{\frac{(tp+fn)\times(tp+fp)}{(tp+fp+tn+fn)} + \frac{(fp+tn)\times(fn+tn)}{(tp+fp+tn+fn)}}{(tp + fp + tn + fn)} \tag{30}$$

- *MCC*: The MCC is a correlation coefficient between the observed and predicted class. It returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates disagreement between prediction and observation. It can be calculated as:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \tag{31}$$

4. Results and discussion

4.1. Experimental results of feature selection using Fisher score and mRMR

In this experiment, two feature selection methods, namely, Fisher score and mRMR are used independently as initial filters to select top N statistically relevant biomarkers. N ranges from 1 to 500.

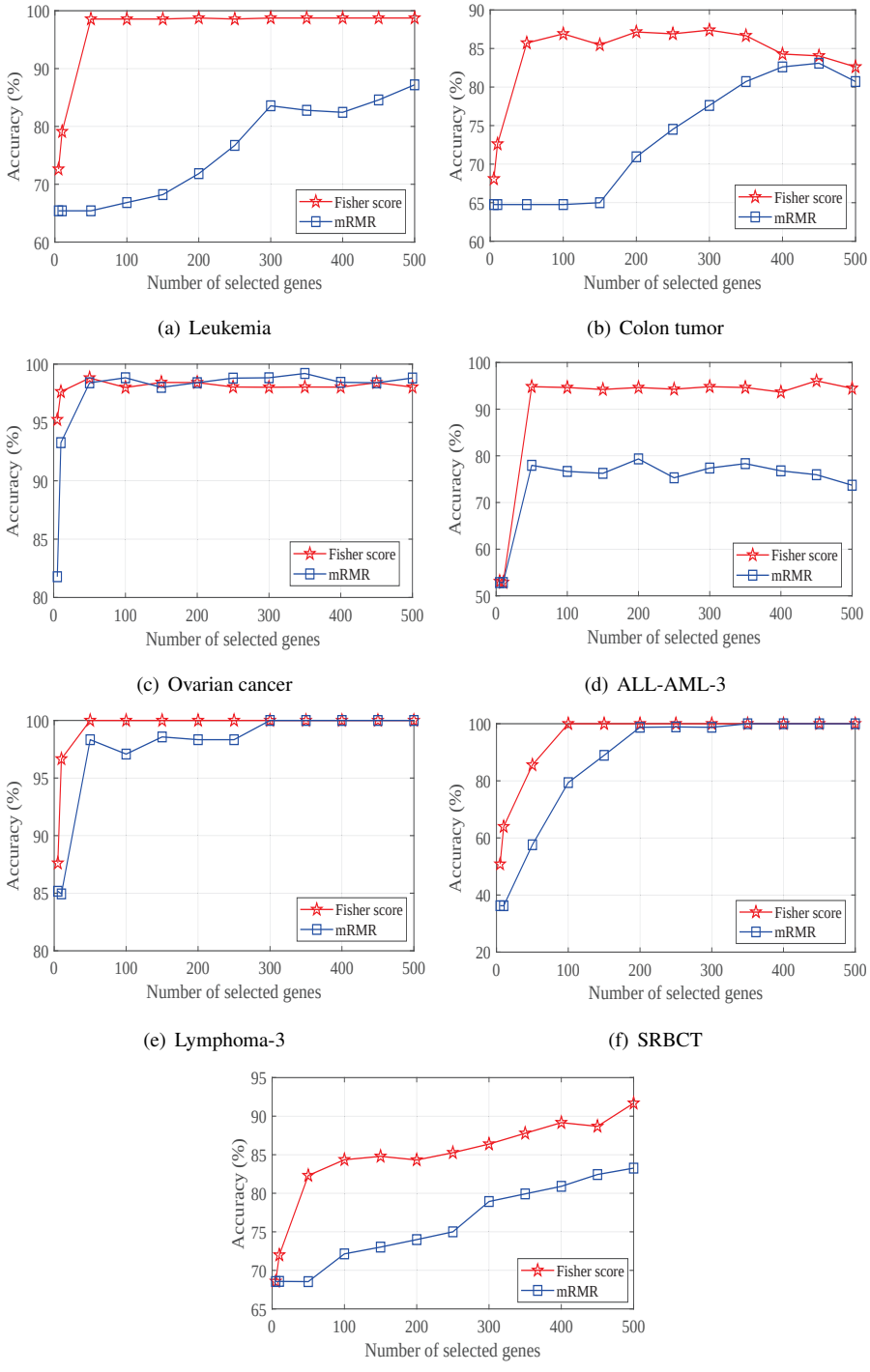
Further, these selected features are sent to the KRR model with default C and γ. In this work, the default values of C and γ are taken as 1 and 100, respectively. Figure 4 shows the change in CA with the increment in NSG on various datasets by Fisher score and mRMR filters. It is observed from the figure that there is no monotonic increase in accuracy with the increase in number of genes. This tells that the top-N best genes are not always the best genes.

For instance, in Leukemia dataset, highest accuracy of 98.57% is achieved with only 50 genes by Fisher score filter, however, mRMR filter results a maximum accuracy of 87.20% with 500 genes. In case of Colon cancer, an accuracy of 87.38% is obtained with top 300 genes, whereas mRMR filter results 83.09% accuracy with 450 genes. In Ovarian cancer dataset, both Fisher score and mRMR filter produce 98.8% accuracy with top 50 and 100 genes, respectively. Further, in Lymphoma-3 dataset, an accuracy of 100% is achieved by both Fisher score and mRMR filters with 50 and 300 genes, respectively. In case of Lung cancer-5 dataset, with top 500 genes, Fisher score produces 91.68% and mRMR produces 83.26% accuracy. Similarly, in SRBCT dataset, a maximum accuracy of 100% is reached by Fisher score filter with 100 genes and by mRMR with 350 genes.

In all the six above mentioned datasets, Fisher score filter gives better accuracy than mRMR with less or equal number of genes. However, in case of ALL-AML-3 dataset, Fisher score yields an accuracy of 96.07% with top 450 features, whereas, mRMR gives an accuracy of 79.34% with only 200 genes. From this experiment, it can be inferred that for majority of the datasets, Fisher score yields better result compared to mRMR filter. Therefore, the genes pre-selected by Fisher score filter are only used in the next multi-objective model.

4.2. Binary-class results

In order to evaluate the performance of the proposed algorithm, three publicly available binary-class microarray gene expression datasets are used: Ovarian cancer, Colon tumor, and Leukemia. To study the influence of the parameter 'population size', an experiment is conducted varying the population size from 30 to 50 with an interval



(a) Leukemia

(b) Colon tumor

(c) Ovarian cancer

(d) ALL-AML-3

(e) Lymphoma-3

(f) SRBCT

(g) Lung cancer-5

Figure 4 The results of top$N$ selected genes on 7 datasets by Fisher score and mRMR filters using KRR classifier with default C and γ

Table 4 Influence of population size on accuracy(%) for binary-class datasets

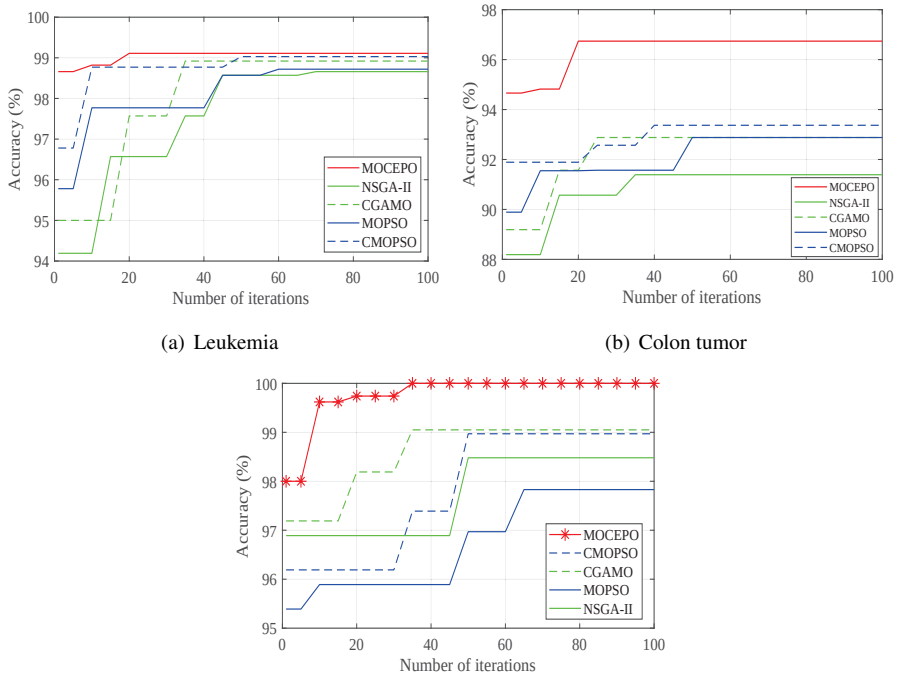
Dataset	Population Size		
	30	40	50
Leukemia	99.11	98.67	99.05
Colon tumor	96.74	96.74	95.92
Ovarian cancer	100	99.23	100

of 10 and keeping the other parameters as previously defined values. The results are reported in Table 4. From the table, it is observed that no significant improvement in accuracy is achieved with the increase in population size. Hence, the rest of the experiments are conducted with the initial population size of 30. The average results by the proposed method using 10-fold CV over 10-trials are noted in Table 5. From the table, it is noticed that MOCEPO method provides highest result for the ovarian dataset with 100% accuracy. For Colon cancer and Leukemia datasets, our method results an accuracy of 96.74% and 99.11% respectively.

The convergence rate of MOCEPO, NSGA-II, CGAMO, MOPSO, and CMOPSO models are studied and shown in Figures 5(a)-5(c). It is noticed from the figure that, for almost all the datasets, the accuracy gradually improves from 1^{st} iteration to 100^{th} iteration. For the dataset Leukemia, no notable rise in classification accuracy is observed after 19^{th}, 51^{st}, 58^{th}, 34^{th}, and 71^{st} iterations using MOCEPO, CMOPSO, MOPSO, CGAMO, and NSGA-II methods respectively. For Colon cancer, no improve in accuracy is noticed after 20^{th}, 40^{th}, 48^{th}, 25^{th}, and 34^{th} iterations using MOCEPO, CMOPSO, MOPSO, CGAMO, and NSGA-II methods respectively. Likewise, for Ovarian cancer, no rise in accuracy is observed after 35^{th}, 48^{th}, 64^{th}, 36^{th}, and 49^{th} iterations using MOCEPO, CMOPSO, MOPSO, CGAMO, and NSGA-II methods respectively. From the above study, it is observed that the rate of convergence is adequately faster in case of chaotic versions in comparison with non-chaotic versions. It is inferred from the results that, the chaotic theory could remarkably improve the rate of convergence.

4.3. Multi-class results

On the filtered results of four multi-class datasets, MOCEPO model is repeated 10 runs with 10-fold CV. To examine the influence of the parameter 'population size', an experiment is carried out changing the population size from 30 to 50 with an interval



(a) Leukemia

(b) Colon tumor

(c) Ovarian cancer

Figure 5 Number of iterations versus accuracy on binary class datasets

Table 5 Average classification performance in 10-fold CV by MOCEPO on binary-class datasets

Dataset	**Accuracy** (%)	**Specificity**	**Sensitivity**	**F-measure**	**Kappa**	**MCC**
Leukemia	99.11	0.975	1	0.993	0.980	0.981
Colon tumor	96.74	0.952	0.974	0.974	0.929	0.932
Ovarian cancer	100	1	1	1	1	1

Table 6 Influence of population size on accuracy(%) for multi-class datasets

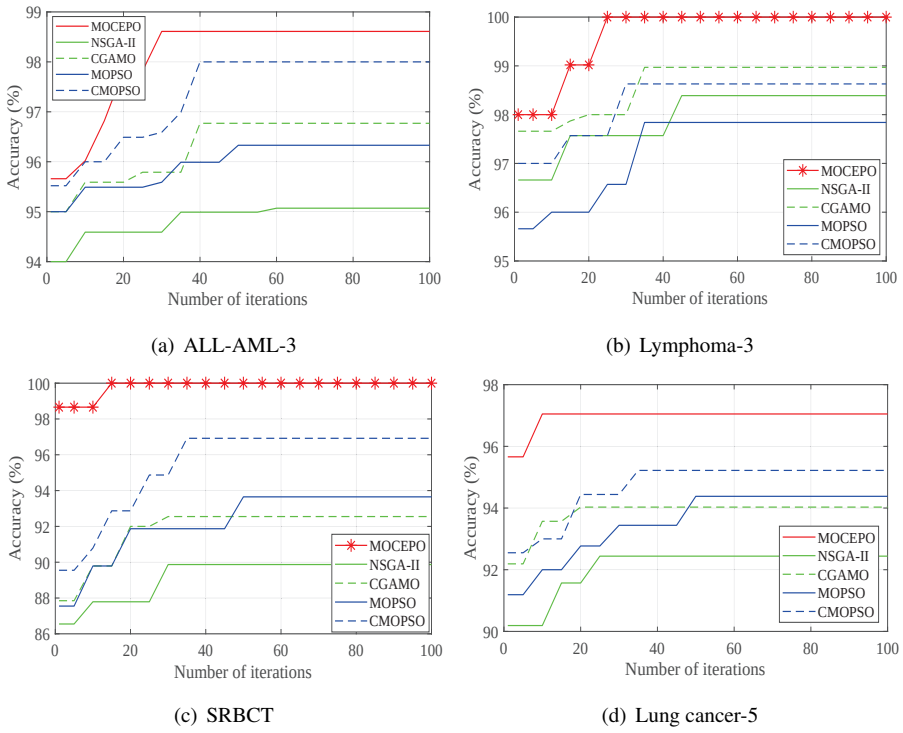
Dataset	Population Size		
	30	40	50
ALL-AML-3	98.61	98.55	98.21
Lymphoma-3	100	100	99.94
SRBCT	100	100	100
Lung cancer-5	97.05	96.88	97.05

Table 7 Average classification performance in 10-fold CV by CEPO-KRR on multi-class datasets

Dataset	**Accuracy** (%)	**Specificity**	**Sensitivity**	**F-measure**	**Kappa**	**MCC**
ALL-AML-3	98.61	0.992	0.990	0.989	0.968	0.983
Lymphoma-3	100	1	1	1	1	1
SRBCT	100	1	1	1	1	1
Lung cancer-5	97.05	0.984	0.948	0.960	0.908	0.949

of 10 and keeping the other parameters constant. The obtained results are reported in Table 6. From the table, it is observed for almost all datasets, the accuracy either remains constant or decreases with the increase in population size. Hence, the rest of the experiments are conducted with population size 30. The average results by the proposed method using 10-fold CV over 10-runs are reported in Table 7. From the table, it is noticed that MOCEPO provides highest result for the Lymphoma-3 and SRBCT (small round blue cell tumors) datasets with 100% accuracy. For the datasets ALL-AML-3 and Lung cancer-5, our method results an accuracy of 98.61% and 97.05% respectively.

The convergence rate of MOCEPO, NSGA-II, CGAMO, MOPSO, and CMOPSO models are shown in Figures 6(a)-6(d). It is noticed from the figure that, for almost all the datasets, the accuracy gradually improves from 1^{st} iteration to 100^{th} iteration. For ALL-AML-3, no significant rise in classification accuracy is observed after 30^{th}, 39^{th}, 51^{st}, 41^{st}, and 69^{th} iterations using MOCEPO, CMOPSO, MOPSO, CGAMO, and NSGA-II methods respectively. For Lymphoma-3, no rise in classification accuracy is noticed after 25^{th}, 29^{th}, 34^{th}, 35^{th}, and 46^{th} iterations using MOCEPO, CMOPSO, MOPSO, CGAMO, and NSGA-II methods respectively. Likewise, for SRBCT, no rise in accuracy is observed after 14^{th}, 36^{th}, 50^{th}, 30^{th}, and 31^{st} iterations using



(a) ALL-AML-3 (b) Lymphoma-3

(c) SRBCT (d) Lung cancer-5

Figure 6 Number of iterations versus accuracy on multi class datasets

MOCEPO, CMOPSO, MOPSO, CGAMO, and NSGA-II models respectively. Lastly, for Lung cancer-5 dataset, the proposed MOCEPO algorithm, CMOPSO, MOPSO, CGAMO, and NSGA-II converge after 10^{th}, 36^{th}, 48^{th}, 20^{th}, and 24^{th} iterations respectively. From the above study, it is observed that the rate of convergence is adequately faster in case of chaotic versions in comparison with non-chaotic versions. It is inferred from the results that, the chaotic theory could remarkably improve the rate of convergence.

4.4. Average execution time(in seconds) on benchmark datasets

The complexity of the proposed MOCEPO algorithm depends on two operations, namely, non-dominated sorting and crowding distance calculation. For M number of objective functions with n number of solutions, the maximum number of computations required for non-dominated sorting is $O(M \times n^2)$. The crowding distance operation needs $O(M \times nlogn)$ computations. Therefore, the total complexity of the algorithm is $O(M \times n^2)$. The average elapsed time by the Fisher score method and the proposed

Table 8 Average execution time(in seconds) by the proposed approach

Class	Dataset	Filter time	MOCEPO time (training + testing)	Total time
Binary-class	Ovarian cancer	0.808	43.281	44.089
	Colon tumor	0.301	148.270	148.571
	Leukemia	0.787	155.813	156.6
Multi-class	Lymphoma-3	0.554	57.305	57.859
	ALL-AML-3	0.809	172.537	173.346
	Lung cancer-5	2.352	212.817	215.169
	SRBCT	0.319	151.038	151.357

MOCEPO method for ten independent runs is reported in Table 8. It can be observed from the table that the total time taken by the suggested approach is 44.089, 148.571, 156.6, 57.859, 173.346, 215.169, and 151.357 seconds for Ovarian cancer, Colon tumor, Leukemia, Lymphoma-3, ALL-AML-3, Lung cancer-5, and SRBCT datasets, respectively.

Table 9 Comparison of MOCEPO with some other methods on binary-class datasets

Methods	Datasets		
	Leukemia	Colon tumor	Ovarian cancer
BCGS [1]	94.1(35)	83.8(23)	98.8(26)
BDE-XRankf [17]	82.4(6)	75(4)	95(3)
DRFO-CFS [5]	91.18(13)	90(10)	**100(16)**
GEM [14]	91.5(3)	91.2(8)	-
IRLDA [56]	97(72)	-	-
IWSS [10]	94.4(7.9)	-	-
IWSS-MB-NB [10]	97.1(6.4)	86(5.2)	-
8-S PMSO [26]	98.1(20)	94.2(20)	-
AEN-CMI [57]	91.05(26.85)	89.30(25.20)	-
SLR [58]	95.51(7)	94.61(5)	-
DFS [59]	98.61	87.09	-
NSGA-II	98.66(8)	91.39(11)	98.97(14)
CGAMO	98.92(9)	92.88(9)	99.05(11)
MOPSO	98.72(10)	92.88(6)	97.83(12)
CMOPSO	99.03(12)	93.37(5)	98.48(9)
MOCEPO	**99.11(5)**	**96.74(4)**	**100(4)**

4.5. Comparative performance with existing competitive methods:

In order to compare the performance of MOCEPO, a comparative study is carried out with NSGA-II, CGAMO, MOPSO, CMOPSO and nineteen other existing schemes. A comparison presents a tentative measurement of the performance of the suggested MOCEPO model, and any technically sound work must include, as done in this paper, a comparison with the benchmark techniques present in that area.

The performance of MOCEPO, NSGA-II, CGAMO, MOPSO, CMOPSO along with other existing methods is presented in terms of CA and NSG for binary-class datasets in Table 9. From the table, it is observed that for Leukemia and Colon tumor datasets, our method performs better than existing techniques. In case of Ovarian cancer dataset, the proposed approach is at par with one of the existing techniques, namely, DRFO-CFS. For Leukemia, the highest accuracy of 99.11% is obtained with MOCEPO with only 5 genes. A similar performance results have been achieved for Colon cancer and Ovarian cancer.

In case of the multi-class datasets, the state-of-the-art techniques are considered for comparison with MOCEPO as reported in Table 10. We can observe from the table that on three multi-class datasets, our proposed approach performs better than the existing algorithms in terms of both CA and NSG. In case of SRBCT dataset, the proposed method is at par with an existing method, namely, DFS. In Lymphoma-3 and SRBCT datasets, the MOCEPO technique achieves as high as 100% CA. In the Lung cancer-5 datasets, MOCEPO also shows excellent performance compared to other methods. However, in case of ALL-AML-3 dataset, our method performs slightly better than CMOPSO.

5. Conclusion

Evolutionary algorithms play an important role in finding the relevant genes from high-dimensional microarray data and hence help the system biologist in cancer diagnosis. Identification of biomarkers with smaller numbers and higher CA substantially improves the quality of the expert systems used in the hospitals.

In the present work, a multi-objective model based on the principle of chaotic EPO algorithm has been proposed for microarray cancer classification. There are two major merits of the MOCEPO algorithm. Firstly, the gene subset selected by the proposed method results in high CA. Secondly, with a promising CA, the NSG are always very small.

Table 10 Comparison of MOCEPO with some other methods on multi-class datasets

Methods	Datasets			
	ALL-AML-3	Lymphoma-3	SRBCT	Lung cancer-5
mRMR-ABC [43]	96.12(20)	96.96(5)	96.30(10)	-
GBC [19]	95.83(8)	98.48(5)	96.38(6)	-
CC-PSO [27]	-	96.8(306)	93.7(63)	-
PSO-AKNN [28]	90.66(3.3)	-	94(8.5)	-
GALA [15]	93.96(3)	-	99.34(6)	-
MCSO [4]	-	-	71.04(100)	-
D-ECOC [60]	79.79	-	98.70	-
MGSACO [20]	-	-	74.49(20)	85.72(20)
DFS [59]	97.22	98.48	**100**	-
NSGA-II	95.07(9)	98.39(11)	89.87(8)	92.44(15)
CGAMO	96.77(11)	98.97(7)	92.55(8)	94.03(11)
MOPSO	96.33(7)	97.84(10)	93.65(8)	94.38(10)
CMOPSO	98(9)	98.63(10)	96.92(10)	95.22(8)
MOCEPO	**98.61(3)**	**100(4)**	**100(5)**	**97.05(5)**

The proposed model has been experimented on seven high-dimensional datasets. The performance of the proposed model is compared with nineteen competitive techniques, and the experimental results reveal that the proposed model achieves improved results over the competent schemes.

In future, the performance of the proposed method can be further improved by implementing the algorithm in MapReduce framework [61, 62] using a cluster of computers. More recent techniques like convolutional neural networks [63–66] can be applied to enhance the classification performance. Furthermore, it will be interesting to see how the proposed algorithm is performing on other multi-objective optimization problems.

References

[1] S. Pang, I. Havukkala, Y. Hu, N. Kasabov, Classification consistency analysis for bootstrapping gene selection, Neural Computing and Applications 16 (6) (2007) 527–539.

[2] E. K. Tang, P. N. Suganthan, X. Yao, Feature selection for microarray data using least squares SVM and particle swarm optimization, in: 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, IEEE, 2005, pp. 1–8.

[3] W. Ding, C.-T. Lin, W. Pedrycz, Multiple relevant feature ensemble selection based on multilayer co-evolutionary consensus mapreduce, IEEE Transactions on Cybernetics (99) (2018) 1–1.

[4] P. Mohapatra, S. Chakravarty, P. Dash, Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system, Swarm and Evolutionary Computation 28 (2016) 144–160.

[5] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Distributed feature selection: An application to microarray data classification, Applied Soft Computing 30 (2015) 136–150.

[6] L. Zhang, P. N. Suganthan, Benchmarking ensemble classifiers with novel co-trained kernel ridge regression and random vector functional link ensembles [research frontier], IEEE Computational Intelligence Magazine 12 (4) (2017) 61–72.

[7] L. Zhang, P. N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, IEEE transactions on cybernetics 45 (10) (2015) 2165–2176.

[8] Y. Ren, P. Suganthan, Empirical mode decomposition-k nearest neighbor models for wind speed forecasting, Journal of Power and Energy Engineering 2 (04) (2014) 176.

[9] Z.-H. Cao, L.-W. Ko, K.-L. Lai, S.-B. Huang, S.-J. Wang, C.-T. Lin, Classification of migraine stages based on resting-state eeg power, in: Neural Networks (IJCNN), 2015 International Joint Conference on, IEEE, 2015, pp. 1–5.

[10] A. Wang, N. An, G. Chen, J. Yang, L. Li, G. Alterovitz, Incremental wrapper based gene selection with Markov blanket, in: Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on, IEEE, 2014, pp. 74–79.

[11] C.-T. Lin, Y.-T. Liu, S.-L. Wu, Z. Cao, Y.-K. Wang, C.-S. Huang, J.-T. King, S.-A. Chen, S.-W. Lu, C.-H. Chuang, Eeg-based brain-computer interfaces: A novel neurotechnology and computational intelligence method, IEEE Systems, Man, and Cybernetics Magazine 3 (4) (2017) 16–26.

[12] R. Katuwal, P. Suganthan, L. Zhang, An ensemble of decision trees with random vector functional link networks for multi-class classification, Applied Soft Computing 70 (2018) 1146–1153.

[13] M. Z. Ali, N. H. Awad, P. N. Suganthan, A. M. Shatnawi, R. G. Reynolds, An improved class of real-coded genetic algorithms for numerical optimization, Neurocomputing 275 (2018) 155–166.

[14] J. C. H. Hernandez, B. Duval, J.-K. Hao, A genetic embedded approach for gene selection and classification of microarray data, in: European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer, 2007, pp. 90–101.

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

[16] S. Das, S. S. Mullick, P. N. Suganthan, Recent advances in differential evolution–an updated survey, Swarm and Evolutionary Computation 27 (2016) 1–30.

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

[18] N. Lynn, P. N. Suganthan, Modified artificial bee colony algorithm with comprehensive learning re-initialization strategy, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2015, pp. 2129–2134.

[19] H. M. Alshamlan, G. H. Badr, Y. A. Alohali, Genetic bee colony (GBC) algorithm: A new gene selection method for microarray cancer classification, Computational Biology and Chemistry 56 (2015) 49–60.

[20] S. Tabakhi, A. Najafi, R. Ranjbar, P. Moradi, Gene selection for microarray data classification using a novel ant colony optimization, Neurocomputing 168 (2015) 1024–1036.

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

[22] I. Fister, I. Fister Jr, X.-S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm and Evolutionary Computation 13 (2013) 34–46.

[23] W. Ding, J. Wang, Y. Li, X. Cheng, A cascaded co-evolutionary model for attribute reduction and classification based on coordinating architecture with bidirectional elitist optimization, Chinese Journal of Electronics 26 (1) (2017) 13–21.

[24] W. Ding, C. Lin, Z. Cao, Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping PSO with nearest-neighbor memeplexes, IEEE Transactions on Cybernetics (2018) 1–14.

[25] W. Ding, C.-T. Lin, M. Prasad, Z. Cao, J. Wang, A layered-coevolution-based attribute-boosted reduction using adaptive quantum-behavior PSO and its consistent segmentation for neonates brain tissue, IEEE Transactions on Fuzzy Systems 26 (3) (2018) 1177–1191.

[26] J. García-Nieto, E. Alba, Parallel multi-swarm optimizer for gene selection in DNA microarrays, Applied Intelligence 37 (2) (2012) 255–266.

[27] A. Chinnaswamy, R. Srinivasan, Hybrid feature selection using correlation coefficient and particle swarm optimization on microarray gene expression data, in: Innovations in Bio-Inspired Computing and Applications, Springer, 2016, pp. 229–239.

[28] S. Kar, K. D. Sharma, M. Maitra, Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique, Expert Systems with Applications 42 (1) (2015) 612–627.

[29] S.-Z. Zhao, P. Suganthan, Two-lbests based multi-objective particle swarm optimizer, Engineering Optimization 43 (1) (2011) 1–17.

[30] Y. Sun, Y. Gao, X. Shi, Chaotic multi-objective particle swarm optimization algorithm incorporating clone immunity, Mathematics 7 (2) (2019) 146.

[31] V. Ravi, D. Pradeepkumar, K. Deb, Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms, Swarm and Evolutionary Computation 36 (2017) 136–149.

[32] R. Qi, F. Qian, S. Li, Z. Wang, Chaos-genetic algorithm for multiobjective optimization, in: 2006 6th World Congress on Intelligent Control and Automation, Vol. 1, IEEE, 2006, pp. 1563–1566.

[33] M. K. Marichelvam, T. Prabaharan, X. S. Yang, A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems, IEEE Transactions on Evolutionary Computation 18 (2) (2014) 301–305.

[34] V. K. Patel, V. J. Savsani, A multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO), Information Sciences 357 (2016) 182–200.

[35] E. Rashedi, E. Rashedi, H. Nezamabadi-pour, A comprehensive survey on gravitational search algorithm, Swarm and Evolutionary Computation 41 (2018) 141 – 158.

[36] J. Cheng, G. G. Yen, G. Zhang, A grid-based adaptive multi-objective differential evolution algorithm, Information Sciences 367-368 (2016) 890 – 908.

[37] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 67–82.

[38] E. Alba, B. Dorronsoro, The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, IEEE Transactions on Evolutionary Computation 9 (2) (2005) 126–142.

[39] O. Olorunda, A. P. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, IEEE, 2008, pp. 1128–1134.

[40] M. Lozano, C. García-Martínez, Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report, Computers & Operations Research 37 (3) (2010) 481–497.

[41] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowledge-Based Systems 159 (2018) 20 – 50.

[42] Q. Song, H. Jiang, J. Liu, Feature selection based on FDA and F-score for multi-class classification, Expert Systems with Applications 81 (2017) 22–27.

[43] H. Alshamlan, G. Badr, Y. Alohali, mRMR-ABC: A hybrid gene selection algorithm for cancer classification using microarray gene expression profiling., BioMed Research International 2015 (2015) 604910–604910.

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

[45] J. Lv, Q. Peng, X. Chen, Z. Sun, A multi-objective heuristic algorithm for gene expression microarray data classification, Expert Systems With Applications 59 (2016) 13–19.

[46] A. Yang, X. Jiang, L. Shu, J. Lin, Bayesian variable selection with sparse and correlation priors for high-dimensional data analysis, Computational Statistics 32 (1) (2017) 127–143.

[47] A. H. Gandomi, X.-S. Yang, Chaotic bat algorithm, Journal of Computational Science 5 (2) (2014) 224–232.

[48] Q. Zhang, Z. Li, C. Zhou, X. Wei, Bayesian network structure learning based on the chaotic particle swarm optimization algorithm, Genetics and Molecular Research 12 (4) (2013) 4468–4479.

[49] G. I. Sayed, G. Khoriba, M. H. Haggag, A novel chaotic salp swarm algorithm for global optimization and feature selection, Applied Intelligence (2018) 1–20.

[50] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, science 286 (5439) (1999) 531–537.

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

[52] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, A. J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proceedings of the National Academy of Sciences 96 (12) (1999) 6745–6750.

[53] E. F. Petricoin III, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, L. LA, Use of proteomic patterns in serum to identify ovarian cancer, The lancet 359 (9306) (2002) 572–577.

[54] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, M. Meyerson, Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses, Proceedings of the National Academy of Sciences 98 (24) (2001) 13790–13795.

[55] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: International Joint Conference on Artificial Intelligence (IJCAI), Vol. 14, Montreal, Canada, 1995, pp. 1137–1145.

[56] A. Sharma, K. K. Paliwal, S. Imoto, S. Miyano, A feature selection method using improved regularized linear discriminant analysis, Machine Vision and Applications 25 (3) (2014) 775–786.

[57] Y. Wang, X.-G. Yang, Y. Lu, Informative gene selection for microarray classification via adaptive elastic net with conditional mutual information, Applied Mathematical Modelling 71 (2019) 286–297.

[58] Z. Y. Algamal, M. H. Lee, A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification, Advances in Data Analysis and Classification (2018) 1–19.

[59] S. P. Potharaju, M. Sreedevi, Distributed feature selection (DFS) strategy for microarray gene expression data to improve the classification performance, Clinical Epidemiology and Global Healthdoi:https://doi.org/10.1016/j.cegh.2018.04.001.
URL http://www.sciencedirect.com/science/article/pii/S2213398418300721

[60] K.-H. Liu, Z.-H. Zeng, V. T. Y. Ng, A hierarchical ensemble of ecoc for cancer classification based on multi-class microarray data, Information Sciences 349 (2016) 102–118.

[61] W. Ding, J. Wang, J. Wang, A hierarchical-coevolutionary-mapreduce-based knowledge reduction algorithm with robust ensemble pareto equilibrium, Information Sciences 342 (2016) 153–175.

[62] W. Ding, C.-T. Lin, S. Chen, X. Zhang, B. Hu, Multiagent-consensus-mapreduce-based attribute reduction using

co-evolutionary quantum PSO for big data applications, Neurocomputing 272 (2018) 136–153.

[63] K. Muhammad, J. Ahmad, S. W. Baik, Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing 288 (2018) 30–42.

[64] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, S.-H. Wang, Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation, Multimedia Tools and Applications (2017) 1–20.

[65] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, S. W. Baik, Efficient Deep CNN-based fire detection and localization in video surveillance applications, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2018) 1–16.

[66] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, S. W. Baik, Convolutional neural networks based fire detection in surveillance videos, IEEE Access 6 (2018) 18174–18183.