

Hybrid Feature Selection Approach Based on GRASP for Cancer Microarray Data

Arpita Nagpal and Deepti Gaur

Department of Computer Science and Engineering, School of Engineering and Technology, The NorthCap University, Gurugram, Haryana, India

Microarray data usually contain a large number of genes, but a small number of samples. Feature subset selection for microarray data aims at reducing the number of genes so that useful information can be extracted from the samples. Reducing the dimension of data sets further helps in improving the computational efficiency of the learning model. In this paper, we propose a modified algorithm based on the tabu search as local search procedures to a Greedy Randomized Adaptive Search Procedure (GRASP) for high dimensional microarray data sets. The proposed Tabu based Greedy Randomized Adaptive Search Procedure algorithm is named as TGRASP. In TGRASP, a new parameter has been introduced named as Tabu Tenure and the existing parameters, NumIter and size have been modified. We observed that different parameter settings affect the quality of the optimum. The second proposed algorithm known as FFGRASP (Firefly Greedy Randomized Adaptive Search Procedure) uses a firefly optimization algorithm in the local search optimization phase of the greedy randomized adaptive search procedure (GRASP). Firefly algorithm is one of the powerful algorithms for optimization of multimodal applications. Experimental results show that the proposed TGRASP and FFGRASP algorithms are much better than existing algorithm with respect to three performance parameters viz. accuracy, run time, number of a selected subset of features. We have also compared both the approaches with a unified metric (Extended Adjusted Ratio of Ratios) which has shown that TGRASP approach outperforms existing approach for six out of nine cancer microarray datasets and FFGRASP performs better on seven out of nine datasets.

ACM CCS (2012) Classification: Theory of computation → Design and analysis of algorithms → Mathematical optimization → Discrete optimization → Optimization with randomized search heuristics

Keywords: feature selection, microarray, classification, GRASP, hill climbing, firefly algorithm, tabu search

1. Introduction

In recent years, there has been a lot of research in the medical community on microarray data. Many data analysis techniques have been applied to the classification of cancer microarray data [1], [2]. The greatest challenge when dealing with microarray data is its very high dimensionality of genes, but small number of samples compared with the large number of genes awakens the curse of dimensionality [3]. When such large number of genes are given as input to machine learning tasks such as clustering, or classification, it leads to the problem of overfitting. Also, it increases the classifiers, complexity and the time needed for its training and execution. The presence of irrelevant and redundant genes also affects the performance of the classifier. Hence, the task of feature subset selection becomes important so that the irrelevant and redundant features are removed. This will further lead to decrease in data acquisition cost and learning time to help in improving cancer diagnosis.

The problem of dimensionality can be solved by two techniques [4], feature extraction and feature selection. PCA (Principal Component Analysis) is a wellknown and efficient feature extraction technique in which the new attributes are created as a combination of old attributes. PCA based feature extraction with high dimensional spectral data and with different preprocessing steps has been investigated to improve machine learning accuracy [5]. In feature selec-

tion technique, some attributes whose information overlaps with other attributes called redundant attributes are removed from the dataset. It does not transform the features; instead, it helps in improving the accuracy of different classification tasks. From the dataset of a disease, one is interested in finding the specific features/genes which are responsible for its occurrence. The high dimensionality of the data makes it difficult to find the specific genes. In case of cancer microarray data, obtaining a transformed set of features using feature extraction does not help and hence dimension reduction is usually carried out using feature/gene selection.

A subset of the features is derived using the values of the features where the value of a feature can be calculated by some mathematical criteria. There are different evaluation criteria and according to those criteria, feature selection can be broadly classified into four categories which are filter, wrapper, embedded and hybrid approaches [6], [7], [8].

In filter feature selection, feature subset is selected as a preprocessing step before applying any learning and classification process. They become independent of the learning algorithm to be applied. They are usually faster and computationally more efficient than wrapper [7].

In the wrapper method [9], like simulated annealing or a genetic algorithm, features are selected in accordance with the learning algorithm which we are going to apply next. It gives better feature subset than filter approach as it is tuned according to the algorithm, but it is much slower than filter feature selection and has to be rerun when the algorithm changes.

When the number of features becomes very large, the filter model is usually chosen due to its computational efficiency and simplicity [8].

In literature, many feature subset selection algorithms based on filter and a wrapper approach for different applications have been proposed. All the algorithms aim to remove the irrelevant as well as redundant features from the original set of features and obtain a feature subset. Using the filter feature selection approach, many algorithms were proposed such as Relief [11], Relief-F [12], FOCUS, FOCUS-2 [13], Correlation based feature selection (CFS) [14], Fast Correlation based feature selection FCBS [8], FAST [6]. Algorithm by Yang *et al.* [15] re-

moves irrelevant features based on gene ranking methods of GS1 and GS2.

Kohavi and John [9] proposed a wrapper method which removes features from the dataset with the help of the learning algorithm. Its performance is always better than that of filter selection method, but it is computationally more expensive. For solving the gene selection and cancer classification problem, artificial bee colony (ABC) algorithm and minimum redundancy maximum relevance (mRMR) algorithm combined together have been proved to be an efficient approach [16]. This is a hybrid approach which offers a balance between filter and wrapper methods. Another hybrid approach which was recently proposed, uses a firefly algorithm in its wrapper phase of the algorithm for short term load forecasting application [17].

To take the advantage of the wrapper method, Bermejo, Gamez and Puerta [19] proposed a hybrid algorithm which can speed up the feature selection process. Their method is based on Greedy Randomized Adaptive Search Procedure (GRASP). GRASP is developed in two stages, filter evaluation method is used in the first stage and the solution found here is given as the starting point to a local search method in the next stage. To provide a good quality solution, these two phases are run many times. It tries to move from local optima provided by local search to a global optimum.

In this paper, we use a multi-start two stage algorithm named GRASP (Greedy Randomized Adoptive Search Procedure) [10]. The initial phase of constructing a reduced set of features in TGRASP and FFGRASP algorithms is taken from a fast hybrid algorithm for feature selection in the high dimensional dataset [19]. In the first phase, filter evaluation method of mutual information is used to take out irrelevant features. Then, a solution to the problem, i.e. a reduced set of features is constructed using the construction steps used by GRASP algorithm. The solution found is given as input to the local search method for its further improvement. The local search methods used are the tabu search and firefly optimization algorithm. They are computationally less expensive wrapper methods and provide better feature subsets in terms of accuracy as compared to other methods such as hill climbing and simulated annealing. Our aim of this study is to come up with an opti-

mization technique for microarray dataset that decreases the number of wrapper evaluations without affecting the accuracy. Tabu search and firefly algorithms have proved to have better accuracy than hill climbing as local search methods.

The rest of the paper is organized as follows: Section 2 gives a detailed description of the GRASP algorithm along with an existing fast hybrid algorithm. Section 3 discusses the proposed algorithms TGRASP and FFGRASP. Section 4 gives the description of the cancer microarray datasets along with the experimental results and comparisons found with the existing algorithm. Finally, in Section 5 we draw conclusions based on the experimental results.

2. GRASP Algorithm for Feature Selection

GRASP is a meta heuristic algorithm developed in two stages. It was developed by Feo and Resende [10], [20]. Further, GRASP has been used to develop many applications [21], [22]. The two stages of the GRASP algorithm are the construction phase and the improvement phase.

1. Construction phase: In this stage, a solution with reduced features is constructed using some heuristic approaches. It starts from an empty set and keeps on adding elements till a solution is obtained.
2. Improvement Phase: The result obtained in the construction phase is modified using some local search algorithm.

Esseghir and Casado-Yusta have proposed a feature subset selection (FSS) algorithm based on the technique of GRASP [23], [24]. In their algorithm, for the construction phase, they have used filter algorithm such as Relief [11] and FCBF [8]. For the improvement phase, some wrapper techniques were applied. The drawback of these two proposed algorithms is that they do not work well if the data is high dimensional.

For high dimensional datasets, a fast hybrid algorithm based on GRASP was introduced by Bermejo, Gamez and Puerta [19]. It alternately switches between the filter and wrapper

approaches. The first phase of this algorithm is taken from a standard hybrid algorithm: Incremental Wrapper Subset Selection (IWSS) algorithm [25], [26]. The second phase, which is the improvement phase, has been modified so that it takes n wrapper evaluation for each step. In order to have reduced computation time for high dimensional data, it maintains a set of non-dominated solutions (NDS) in the construction phase. Non-dominated set of solution is the set in which every solution in the set is different, in terms of number of features and accuracy. Suppose there are two solutions S_1 and S_2 , S_2 is non-dominated by S_1 if $|S_1| \leq |S_2|$ and accuracy (S_1) $>$ accuracy (S_2). According to their observations, they have recommended hill climbing as a local search procedure to be used in the improvement phase. We have named their algorithm as FCGRASP that uses a hill climbing in its improvement phase.

The FCGRASP algorithm is depicted in Figure 1. The input given to the algorithm is Size: which gives the number of variables to be considered at each iteration and *NumIter* which gives the number of iterations in both phases. The output of the algorithm is the final selected subset.

In the initialization phase of the algorithm, NDS is initialized to be empty. The relevance of each feature with its class label or target feature is calculated and stored in the array named "relevance". It is calculated by any filter based approach. Then this relevance value is used to calculate the probability of each feature in the dataset. The probability of occurrence of each feature is stored in the ProbSel array.

Now the construction and improvement phases are performed till the number of iterations specified by the user is over. The construction phase is depicted in lines 7 to 16. In each iteration, construction phase adds a new solution to the NDS. Line 7 selects the subset of features from X determined by the parameter Size and stores it in subset variable. A feature is selected based on its value in "ProbSel" array. The features in subset variable are sorted and stored in array R . The function evaluate (C, S, D) calculates the classification accuracy of the dataset D over classifier C with S variables using 5-cross validation. Then, using the loop from second feature to last, it is checked whether a feature im-

proves the classification accuracy or not. If it improves the accuracy, it is added to S .

Lines 17 to 20 perform the improving step. Here the value of solution S returned by construction step is added to NDS if and only if the solution is not present in the non-dominated set. Update function modifies NDS and improving method is executed with S as starting point in line 19. This improving method is one of the local-search based FSS algorithm. There are different choices for local-search based algorithms such as hill climbing [19], genetic algorithms and simulated annealing. We have found that hill climbing has some drawbacks.

- Possibility of being trapped at a local optimum as it performs only a single run of the iterative improvement.

In D : Data set; F : filter measure; C : class label/target feature;

Size: number of variables to consider at each iteration;

numIt: number of iterations in algorithm

Out S : The selected subset

// **initialization**

1. NDS = NULL
2. for each $X_i \in X$
3. Relevance [i] = $F(X_i, C) + \epsilon$
4. for each $X_i \in X$
5. ProbSel [i] = Relevance [i] / $\sum_{j=1}^n$ Relevance [j]

// **GRASP**

6. for $i = 1$ to numIt

// **constructive step**

7. subset = some Size features selected using ProbSel [] from X without replacement
8. R [] = rank the features based on scores value
9. $S = R$ [1]
10. BestValue = evaluate(C, S, D)
11. for $i_r = 2$ to $R.size()$
12. $S_{aux} = S \cup R_{i_r}$
13. NewValue = evaluate (C, S_{aux}, D)
14. **if** (NewValue > BestValue) **then**
15. $S = S_{aux}$
16. BestValue = NewValue

// **improving step**

17. **if** (update (NDS, S)) **then**
 18. $X_{nds} = \cup_{s_i \in NDS} S_{i_r}$
 19. $S' = \text{ExecuteImprovingMethod}(X_{nds}, S, C, D)$
 20. update (NDS, S')
 21. return all or best solution(s) in NDS
-

Figure 1. FCGRASP algorithm for FSS.

- There is no chance of an escape from local minima.
- It may cycle over the same solutions again and again and thus become redundant

The drawbacks found in case of using simulated annealing as a local search algorithm are:

- The result found using simulated annealing can vary on each run of the algorithm, in other words, every run of simulated annealing gives a different solution.
- Only under certain circumstances, the optimum solution is found. So, you may have to run the algorithm many times to obtain the optimum solution.

3. The Proposed Method

To overcome the drawbacks of hill climbing and simulated annealing used in the improvement phase of GRASP, we have adopted tabu search optimization and firefly optimization approaches in the improvement phase of the GRASP algorithm.

3.1. Tabu Search Based GRASP Approach (TGRASP)

The tabu search uses neighborhood search like simulated annealing and hill climbing, but also uses memory to record the history. So, the major advantage of using this technique is that the number of solutions to be tested keeps reducing. This is because it uses memory called tabu list, that records the recent history of the search and cycling back to previously visited solutions is avoided. In many practical problems, it is seen that a reasonable size tabu list improves the performance of hill climbing. For high dimensional datasets like microarray data, the solution obtained by tabu search rivals and often surpasses the best solutions previously found by simulated annealing and hill climbing. Using tabu search we can avoid repeating the search. By using tabu list as used in tabu search approach, the problem of redundancy can be removed from the hill climbing approach. In general, tabu list has a fixed size to memorize, and it follows FIFO in maintaining the list.

The pseudo code of the algorithm is demonstrated in Figure 2. Initially the first solution

Input: S : Start State
 NumSol: List of all possible solutions
 Output: Best: an improved solution

```

begin
  Best = S
  Tabulist = [ ]
  Size = size(NumSol)
  Repeat until Size
   $S_{\text{best}}$  = Neighborhood solution ( $s$ )
    if (accuracy( $S_{\text{best}}$ ) > accuracy(best) &&
       $S_{\text{best}}$  not in Tabulist)
      Best =  $S_{\text{best}}$ 
      Tabulist.add (Best)
    end
    if (Tabulist.size > maxsize)
  Tabulist.removefirst()
    end
  Return Best
end

```

Figure 2. A new version of tabu search in improvement phase.

found in the list of all possible solutions is taken as the best solution. Tabu list is initialized to be empty.

Tabu search stops when all the solutions found in non-dominated set (NDS) have been evaluated. If the accuracy of the new solution S_{best} is better than the accuracy of the best solution last found, then the best solution is replaced with the new solution (S_{best}). We also check that the new solution (S_{best}) is not presented in the existing tabu list before making it a best solution. This new solution is then added in the tabu list.

Here, we have used short term tabu list, which keeps record of last 20 solutions. The advantage of using tabu list is that the repeated solutions are not checked again. When the solution reaches the maximum size of the list, then the first added solution from the tabu list is deleted. The algorithm returns the best solution out of the solutions found in the construction of GRASP.

3.2. Novel Firefly Algorithm

In this section, we present a discrete firefly algorithm (FFGRASP) for the improvement phase of the GRASP algorithm. Yang and He

[27] have surveyed and shown that firefly algorithm has been used in many applications and produces better performance in terms of time and optimality than other algorithms.

It obtains better global results than simulated annealing, PSO and artificial bee colony (ABC) algorithm. Firefly algorithm is a bio-inspired optimization algorithm proposed by Xin-She Yang at Cambridge University [28], [29]. It is based on the pattern of flashing light of fireflies. This algorithm focuses on two important issues obtained from the behaviour of fireflies: the variation of light intensity and formulation of the attractiveness. The attractiveness of two fireflies is proportional to the brightness of the fireflies. The lesser bright fireflies move towards the brighter ones. If there is no brighter one than a particular firefly, it will move randomly. Further, brightness is determined by the light intensity of the fireflies which affects the objective function f . Brightness, I_i of a firefly i at location y is given by $I(y) \propto f(y)$. Attractiveness β , is determined by the adjacent fireflies and it varies with distance r_{ij} between firefly i and firefly j .

Exploration is the acquisition of new information through searching. Exploration is a main concern for all optimizers because it might lead to new search regions that might contain better solutions. Exploitation is defined as the application of known information. The good sites are exploited via the application of a local search [30]. Variation in the light intensity controls the tradeoff between exploration and exploitation in the firefly algorithm. The firefly i is attracted to another brighter firefly j , its position at iteration $t + 1$ is determined by:

$$x_i^{(t+1)} = x_i^t + \beta_o e^{(-\gamma r_{ij}^2)} (x_j^t - x_i^t) + \alpha_t \epsilon_i^t \quad (1)$$

The second component of the equation 1 is the exploration and it is used for attraction between the two fireflies. The parameters used in this part are β and γ . β is the attractiveness which is proportional to the light intensity seen by other fireflies. β_o is the attractiveness at $r = 0$. γ controls the average distance of a group of fireflies that can be seen by adjacent groups. The distance between two fireflies is the Cartesian distance which is given as:

$$\sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2)$$

The third component of the equation 1 is the exploitation. Exploitation is controlled by the randomization parameter α_t , which is tuned during each iteration so that it can vary with iteration counter t .

In our implementation, the objective function is the average accuracy obtained through the classifier, naïve Bayes and SVM. We have taken $\beta_o = 1$ as used by most of the applications. The parameters γ and α have been used in the algorithm such that they are dependent on the number of features selected at each iteration.

Parameters γ and α are updated at each iteration as follows:

1. Gamma parameter (γ) is given by, $\gamma = 1/\sqrt{z}$ where $Z = (|D| - |d_i|)/|D|$ where $|D|$ is the total number of features. $|d_i|$ is the number of features selected at each iteration. Iteration (i) varies from 1 to Numiter, Numiter is the number of iterations.
2. Alpha parameter (α) is given by, $\alpha = (1 - \Delta)Z$, Δ is the cooling factor for randomness. We have used $\Delta = 0.98$ in the experiments performed here.

Formally, the modified discrete firefly algorithm as used in the improvement phase of the GRASP is given in Figure 3.

4. Empirical Study

An existing hybrid algorithm, which uses hill climbing as a local search procedure in the improvement phase of GRASP named FCGRASP has been used to compare the proposed algorithms. In this section, we have compared our proposed algorithms named TGRASP and FFGRASP with FCGRASP on different publically available cancer datasets.

4.1. Experimental Setup and Dataset

The experiments have been performed on nine cancer microarray datasets of high dimensions. The data set is described in Table 1. Some

Input:

- Initial Values to parameters: α the randomness parameter, β the attraction coefficient, γ the parameter to control randomness, $\Delta = 0.98$
- Objective function $f(y)$, $y_i = (y_1, y_2, \dots, y_d)^T$
- All solutions in NDS, $S_i = (S_{i1}, S_{i2}, \dots, S_{in})$

Output: Best solution set S_{best}

begin

Calculate light intensity l_i at y_i by $f(y_i)$

$S_{best} = \text{NULL}$

While ($t < \text{numiter}$)

for $i = 1:n$ // all n solutions at this step

for $j = 1:i$

Determine the position of i using equation

if ($l_i > l_j$)

Move firefly i towards j in all d dimensions

$S_{best} = S_i$

else

Change position of i randomly

endif

Determine new solutions from NDS and revise light intensity

end j

end i

Return S_{best}

Calculate new α and γ values

end while

Figure 3. Discrete firefly algorithm in the improvement phase.

datasets have two classes, while some have more than two classes. Datasets are obtained from different sources. Breast, colon, leukemia and prostate datasets were obtained from Kent Ridge Biomedical Dataset data repository [31]. For SRBCT, Khan dataset has been used [2]. Table 1 gives the further details about these datasets.

Before actually using the data in the experiments, a preprocessing procedure has been applied to them. Datasets of breast and endometrium contained null values. The attributes containing more than 30% missing values have been left out. Other null values were replaced with the class wise mean of their respective attributes. Thus, 3000 attributes were left for endometrium and 5000 for breast cancer data [32]. Data for colon, SRBCT, leukemia and melanoma were used as they were. For other datasets, we adopted the technique suggested by Yang *et al.* [15] and Ramaswamy *et al.* [33].

Table 1. Dataset description.

Dataset	Instances	Original genes	Preprocessed genes	Classes	Description
Colon_I	37	22883	8826	2	8 Normal samples and 29 tumour samples
Prostate	102	12600	5966	2	52 prostate tumour samples and 50 normal samples
Breast	97	24482	5000	2	46 benign samples, 51 malignant samples
Colon	62	2000	2000	2	40 cancer samples and 22 normal samples
SRBCT	83	4773	2308	4	29 EWS samples, 18 NB samples, 11 BL samples, and 25 RMS samples
Endometrium	42	2308	3000	4	13 serous papillary, 3 clear cell, 19 endometrial cancer, and 7 age-matched normal endometrial
Leukemia	72	8872	7129	3	28 AML samples, 24 ALL samples and 20 MLL samples
Melanoma	38	7129	8076	3	12 Lentigo samples, 19 Acral samples, 7 Nodular samples
CNS-v1	34	8076	2277	2	9 Normal samples, 25 brain tumour samples

For prostate dataset, floor value of 100 and a ceiling value of 16000 with a variation of the Max / Min ratio as 5 and Max-Min difference of 50 were used to filter the values. CNS-v1, Colon-I used the intensity threshold value as floor and ceiling between 20 – 16000 with Max/Min ratio as 5, 3 and Max-Min difference of 500, 100 respectively. All datasets were normalized using z-score normalization before using them in experiments.

The algorithms FCGRASP and TGRASP are implemented in matlab on the same PC. Classification algorithms embedded in wrapper evaluation are created as functions in matlab and are called wherever needed.

The parameters used to compare both algorithms are the number of features, runtime and classification accuracy. Runtime is machine dependent, so we have implemented and compared both algorithms on the same machine. The classification accuracy is calculated using 10 fold cross validation strategy for the training and testing sets. The training set consists of 90% of the values and the test set consist of 10% of values. For each classification algorithm, we obtain the average classification accuracy, number of selected features, runtime found under each algorithm and each dataset.

One parameter named EARR (Extended Adjusted Ratio of Ratios) proposed by Wang [34] has also been used to compare both algorithms.

It is a multicriteria metric, where the classification accuracy, runtime and number of features selected are integrated. EARR evaluates the performance by taking the ratio of the metric values. Let $D = \{D_1, D_2, \dots, D_n\}$ be a set of n datasets, and $A = \{A_1, A_2, \dots, A_n\}$ be a set of M FSS algorithms. Then, the EARR of A_i to A_j over D_k can be defined as:

$$EARR_{A_i, A_j}^{D_k} = \frac{acc_i^k / acc_j^k}{1 + \alpha \cdot \log(t_i^k / t_j^k) + \beta \cdot \log(n_i^k / n_j^k)}, \quad (1 \leq i \neq j \leq M, 1 \leq k < N) \quad (3)$$

α and β are user defined parameters which tell us how much the runtime and number of features selected should respectively dominate accuracy. acc_i^k is the accuracy of i^{th} algorithm of k^{th} dataset. t_i^k and n_i^k are the runtime and number of selected features of dataset k on i^{th} algorithm respectively. The greater the value of EARR, the better the corresponding algorithm on a given dataset D [34].

As discussed in the algorithm, we have used Symmetric Uncertainty (SU) for the filter evaluation [6], [8]. For the wrapper phase, Naïve Bayes and Support Vector Machine (SVM) classifiers are used. Support vector machine performs very well on most of the problems in high dimensional space. It is difficult to find a linear classifier to separate different classes

in the dataset. This problem can be solved using SVM. It is proved to be relatively new and promising classifier over other classifiers [18]. Naïve Bayes is quite sensitive to the presence of redundant and irrelevant predicted attributes [19].

4.2. Experimental Results

The proposed algorithms TGRASP and FFGRASP have been compared with an existing algorithm FCGRASP. They have been compared in terms of three performance measures, i.e., classification accuracy, runtime and number of features. The classifiers used are Naïve Bayes classifier and SVM.

In all of the three algorithms, Size and NumIter are the parameters used to carry out multiple wrapper evaluations. Tabu tenure is a parameter that is used only for TGRASP algorithm. Tabu tenure is the length of the tabu list kept at the time of performing the experiments. This parameter is a static tabu list with short term memory for storing a maximum of 20 values. In our experiments, we have varied the size of the tabu tenure from 5 to 15. For most of the datasets, we observed that after the size of Tabu tenure reaches 10, the solutions repeat themselves. So, to compare all datasets on a common value of Tabu tenure, we have shown the results when its value was 8.

Table 2 and Table 3 give the comparison of TGRASP, FFGRASP and FCGRASP by keeping the value of tabu tenure as 8. NumIter parameter is fixed to 50 and then 100 for all datasets and for both classifiers. The value of parameter "Size" varies based on the total number of features in each dataset.

Table 2 gives the average number of features and average runtime in seconds obtained using 10 fold cross validation. From the results we observe that

- As the number of iterations (NumIter) increased from 50 to 100, it showed a significant change in the number of wrapper evaluation and hence we could observe that there was approximately 25 – 30% increase in runtime in most of the datasets.
- In case of two class dataset, when NumIter = 50, and Naïve classifier was considered, the average runtime of FFGRASP

decreased by 13.2, 27.7 percent over TGRASP and FCGRASP respectively. The runtime of TGRASP in case of SVM Classifier has been decreased by 84.5, 67.8 percent of that of FCGRASP and FFGRASP respectively.

- For four class dataset, when Naïve Bayes classifier was used, TGRASP ranks 1. Its average runtime over both the datasets decreased by 95.6, 86.9 percent of that of FCGRASP and FFGRASP respectively. When SVM classifier was considered, FFGRASP has a decreased average runtime by 65.2, 21.38 percent by TGRASP and FCGRASP respectively.
- In all the datasets, FFGRASP algorithm selects less number of features in the range of 2 to 227 numbers of features, as compared to FCGRASP.
- In the majority of the datasets, on average, approximately 80 percent of the features selected by FFGRASP and TGRASP are common with the features selected by FCGRASP.

Table 3 shows the average accuracy of the three algorithms found using 10 fold cross validation of two classifiers on nine cancer datasets. The results observed using Naïve Bayes classifier are as follows:

- For two class datasets, when NumIter = 50, as compared to FCGRASP, the classification accuracy of Naïve Bayes has been improved by TGRASP algorithm by 12, 15, 14 percent in case of colon, prostate and breast datasets respectively. TGRASP shows an improvement over FFGRASP in classification accuracy by 40, 9.4 and 1 percent for colon_I, colon and breast datasets respectively. FFGRASP has increased the classification accuracy in case of prostate dataset by 18 and 2 percent from FCGRASP and TGRASP respectively. However, CNS-v1 is the only two class dataset which has decreased the classification accuracy of TGRASP and FFGRASP by 20 and 50.01 percent respectively from FCGRASP.
- For two class datasets, when NumIter = 100, the classification accuracy of Naïve Bayes has been improved by TGRASP algorithm by 7 and 9 percent in case of

Table 2. Comparison of FCGRASP, TGRASP and FFGRASP in terms of runtime and number of features selected when NumIter = 50 and 100.

Dataset	Parameters		Classifier	Runtime (s)			No. of features		
	Size	NumIter		FCGRASP	TGRASP	FFGRASP	FCGRASP	TGRASP	FFGRASP
Colon_I	176	50	Naïve Bayes	358.39	177.88	185.13	105	101	98
			SVM	707.25	402.89	336.56	201	201	200
	88	100	Naïve Bayes	453.86	253.20	191.79	216	206	200
			SVM	855.04	534.66	340.46	365	351	340
Prostate	110	50	Naïve Bayes	285.9	281.7	226.72	245	219	119
			SVM	350.41	177.01	640.99	225	222	198
	59	100	Naïve Bayes	330.25	316.69	340.91	395	154	135
			SVM	531.25	180.97	654.84	399	369	355
Breast	40	50	Naïve Bayes	70.77	76.69	51.91	397	172	170
			SVM	108.56	51.52	106.55	230	222	200
	20	100	Naïve Bayes	149.25	83.221	61.23	275	274	200
			SVM	1117.92	66.65	217.16	355	348	300
Colon	100	50	Naïve Bayes	254.95	239.35	187.07	217	211	100
			SVM	244.74	131.56	218.48	259	247	212
	50	100	Naïve Bayes	276.68	334	197.66	384	370	360
			SVM	222.50	291.6	279.95	490	458	359
SRBCT	45	50	Naïve Bayes	155.35	84.75	149.54	234	231	190
			SVM	1343.17	615.38	251.65	492	492	350
	23	100	Naïve Bayes	190.39	140.20	156.20	405	405	390
			SVM	1549.4	797.05	325.51	661	661	521
Endometrium	60	50	Naïve Bayes	301.76	148.89	287.11	195	166	155
			SVM	495.22	353.56	334.56	661	561	480
	30	100	Naïve Bayes	401.55	199.57	290.86	322	312	218
			SVM	13.88	455.94	536.1	877	877	775
Leukemia	140	50	Naïve Bayes	660.50	326.47	307.77	217	216	198
			SVM	1933.9	957.76	930.91	629	629	550
	70	100	Naïve Bayes	985.71	682.4	345.28	392	370	297
			SVM	1960.0	986.44	1042.21	981	980	880
Melanoma	160	50	Naïve Bayes	742.27	297.22	491.99	174	163	163
			SVM	1017.4	865.27	610.25	634	634	556
	80	100	Naïve Bayes	987.41	386.84	376.75	392	219	200
			SVM	1109.0	918.6	956.35	1065	1065	9456
CNS-v1	45	50	Naïve Bayes	64.13	34.17	52.22	103	105	99
			SVM	92.65	51.64	64.74	136	137	100
	22	100	Naïve Bayes	85.53	91.751	59.39	182	184	180
			SVM	117.68	58.15	67.23	236	222	212

prostate and colon datasets respectively. FFGRASP shows an improvement over FCGRASP in the range of 1 to 11 percent. Unfortunately, in case of dataset CNS-v1 the accuracy of TGRASP and FFGRASP has been decreased by 20 percent from FCGRASP.

- In case of endometrium dataset, FFGRASP classification accuracy has been improved

by 21.3 and 4 percent from FCGRASP and TGRASP respectively when NumIter = 50. The accuracy was improved by 22.5 and 5.5 percent from FCGRASP and TGRASP respectively when NumIter = 100.

- For SRBCT dataset, when NumIter = 50, maximum classification accuracy of 100% was achieved by FCGRASP and TGRASP algorithms, and the accuracy

Table 3. Comparison of FCGRASP, TGRASP and FFGRASP in terms of accuracy when numIter = 50 and 100.

Dataset	Parameters		Classifier	Accuracy		
	Size	NumIter		FCGRASP	TGRASP	FFGRASP
Colon_I	176	50	Naïve Bayes SVM	100 100	100 100	71.42 77.77
	88	100	Naïve Bayes SVM	85.71 98	85.71 100	71.42 94.44
Prostate	110	50	Naïve Bayes SVM	77.87 80	90 85	92 88.23
	59	100	Naïve Bayes SVM	85 80	91.6 90	94 84.31
Breast	40	50	Naïve Bayes SVM	81.13 0.580	91.66 0.806	83.66 0.709
	20	100	Naïve Bayes SVM	83.33 80.65	91.66 87.10	92.56 83.87
Colon	100	50	Naïve Bayes SVM	68.42 77.08	78.42 79.17	77.89 77.08
	50	100	Naïve Bayes SVM	73.68 72.92	73.68 75	74.5 73.83
SRBCT	45	50	Naïve Bayes SVM	100 100	100 100	88.75 98.5
	23	100	Naïve Bayes SVM	81.25 100	75 100	89.25 98.5
Endometrium	60	50	Naïve Bayes SVM	75 94.07	87.5 95.47	91 96.5
	30	100	Naïve Bayes SVM	77.5 90.04	85 90.04	95 95
Leukemia	140	50	Naïve Bayes SVM	92.85 95.46	98.5 95.46	100 96.00
	70	100	Naïve Bayes SVM	78.57 97.72	85.71 98.72	86.42 98.71
Melanoma	160	50	Naïve Bayes SVM	57.14 38.21	57.14 38.21	71.42 61.21
	80	100	Naïve Bayes SVM	42.85 13.07	42.85 13.07	57.14 50.12
CNS-v1	45	50	Naïve Bayes SVM	100 87	83.33 81	66.66 68.75
	22	100	Naïve Bayes SVM	100 93	83.33 81	83.33 87.5

of FFGRASP has been decreased by 12.67 percent. When NumIter = 100, the classification accuracy of FFGRASP has been increased by 9.84 and 19 percent from FCGRASP and TGRASP respectively.

- In case of three class dataset, when NumIter = 50, the classification accuracy of FFGRASP has shown an improvement

over FCGRASP and TGRASP by 7.7 and 1.5 percent respectively in leukemia dataset and by 24.9 percent over both the algorithms in melanoma dataset.

- For three class dataset, when NumIter = 100, FFGRASP takes over by 9.9 and 0.8 percent over FCGRASP and TGRASP respectively for leukemia dataset and by

Table 4. EARR Values calculated to compare both the algorithms on all datasets.

Dataset	EARR values	FCGRASP	TGRASP	FFGRASP
Colon_I	FCGRASP		0.9997	1.1995
	TGRASP	1.0002		1.2010
	FFGRASP	0.8336	0.8333	
Prostate	FCGRASP		0.9275	0.9038
	TGRASP	1.0781		0.9680
	FFGRASP	1.1314	0.9999	
Colon	FCGRASP		0.90886	0.8998
	TGRASP	1.1001		0.9899
	FFGRASP	1.1113	1.01009	
Breast	FCGRASP		1.00006	0.9889
	TGRASP	0.9999		0.9889
	FFGRASP	1.0112	1.0111	
SRBCT	FCGRASP		1.0831	0.9102
	TGRASP	0.9231		0.8403
	FFGRASP	1.0985	1.1942	
Endometrium	FCGRASP		0.9114	0.8156
	TGRASP	1.0970		0.8933
	FFGRASP	1.2346	1.1178	
Leukemia	FCGRASP		0.91652	0.9085
	TGRASP	1.09107		0.9913
	FFGRASP	1.1006	1.0087	
Melanoma	FCGRASP		0.9993	0.7493
	TGRASP	1.00015412		0.7498
	FFGRASP	1.334437	1.3335	
CNS-v1	FCGRASP		1.2000	1.1998
	TGRASP	0.8332		0.9998
	FFGRASP	0.8344	1.0001	

33.3 percent over FCGRASP and TGRASP for melanoma dataset.

When we consider SVM Classifier in Table 3, we observe that:

- For two class datasets, when NumIter = 50, TGRASP algorithm classification accuracy is better than FCGRASP and FFGRASP in the range of 2 to 29 percent in case of prostate, colon and breast datasets. For colon_I dataset, TGRASP and FC-
- GRASP have maximum accuracy of 100%, which has been increased from FFGRASP by 28.58 percent. For dataset, CNS-v1 the classification accuracy of TGRASP and FFGRASP has been decreased by 7.4 and 26.5 percent respectively from FCGRASP.
- For two class datasets, when NumIter= 100, Classification accuracy has been improved by TGRASP as compared to FCGRASP by 2, 12, 8, 4 percent for colon_I, prostate,

colon and breast datasets respectively. FFGRASP has improved the classification accuracy over FCGRASP in the range of 2 to 4 percent. However, in the case of dataset CNS-v1 the accuracy of TGRASP and FFGRASP has been decreased by 14.8 and 6.2 percent respectively from FCGRASP.

- In case of endometrium dataset, the classification accuracy of FFGRASP has been improved by 2.58 and 1.07 percent from FCGRASP and TGRASP respectively at NumIter = 50. When NumIter = 100, an improvement in accuracy of 5 percent has been obtained in FFGRASP from both algorithms.
- For SRBCT dataset, maximum classification accuracy of 100% is achieved by FCGRASP and TGRASP algorithms, but the accuracy of FFGRASP has been decreased by 1.52 percent.
- In case of three class dataset, when NumIter = 50, FFGRASP wins over FCGRASP and TGRASP by 0.5 percent in leukemia dataset and by 60.19 percent over melanoma dataset.
- For three class datasets, when NumIter = 100, FFGRASP as compared to FCGRASP and TGRASP has been showing an improvement of 1.01 and 28.3 percent over leukemia and melanoma datasets respectively.

From the above results, it was concluded that TGRASP performs better than the other two algorithms for two class datasets. For multiclass datasets FFGRASP performs more efficiently.

In order to further explore which feature selection algorithm is significantly better, we performed a test on a unified metric, EARR [34]. To do calculations in EARR equation, we have taken Naïve Bayes classification accuracies, runtime, number of features selected values at NumIter = 100 from Table 2 and Table 3 and parameters $\alpha = \beta = 0.001$. Using equation 3, we calculate the EARR values between each of the two algorithms on all datasets. Table 4 depicts the different EARR values found between each of the two different algorithms.

According to Wang *et al.* [34], the value of EARR A_i, A_j is greater than (or equal to, or smaller than) that of EARR A_j, A_j indicates that A_i is better than (or equal to, or worse than) A_j .

For example, in colon_I dataset, it is found that $EARR(TGRASP, \overline{FCGRASP}) = 1.0002$ is greater than $EARR(\overline{FCGRASP}, TGRASP) = 0.9997$. So, we can say that TGRASP is better than FCGRASP for colon_I dataset. Similarly, values of EARR for other datasets are found and depicted in Table 4. We observe that for six out of nine datasets TGRASP has greater EARR values than FCGRASP.

Comparing FFGRASP algorithm with FCGRASP on colon_I dataset, the EARR (FCGRASP, FFGRASP) value is 1.1995 which is greater than EARR (FFGRASP, FCGRASP) in value of 0.8336. Here we can say that FCGRASP is better than FFGRASP approach for this dataset. For all other datasets, on comparing FFGRASP with FCGRASP, the results indicate that FFGRASP is performing better than FCGRASP in case of seven out of nine datasets.

On comparing TGRASP with FFGRASP, according to the EARR values found in Table 4, we can conclude that FFGRASP algorithm is statistically better than TGRASP approach in all the datasets except in the case of colon_I dataset.

NumIter and tabu tenure are the parameters which can control the tradeoff between accuracy and runtime. In our experiments, we have varied the size of tabu tenure from 5 to 15 to varying the size enables us to determine how this parameter affects the performance based on runtime and accuracy. For six out of nine datasets there is a high positive correlation between tabu tenure with runtime and accuracy. This effect has been shown in Figure 4 and Figure 5 for leukemia dataset.

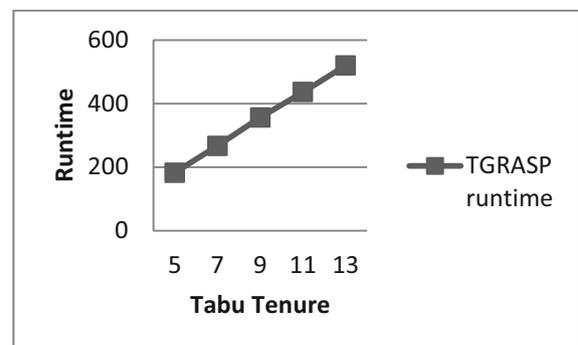


Figure 4. Depicting the effect of runtime and tabu tenure on Leukemia dataset.

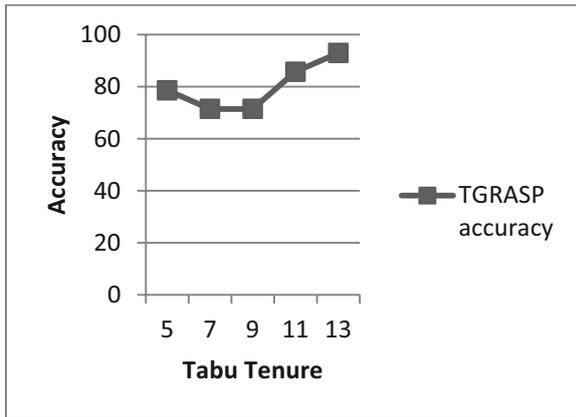


Figure 5. Depicting the effect of accuracy and tabu tenure on leukemia dataset.

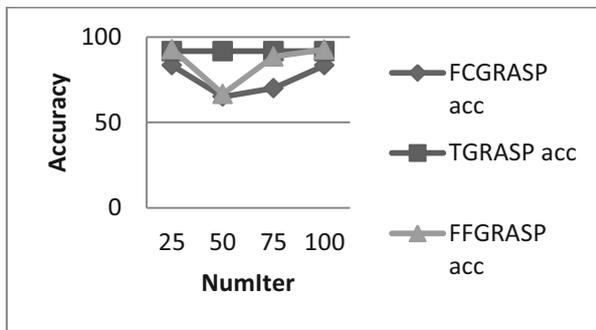


Figure 6. The effect of NumIter parameter on accuracy on dataset colon.

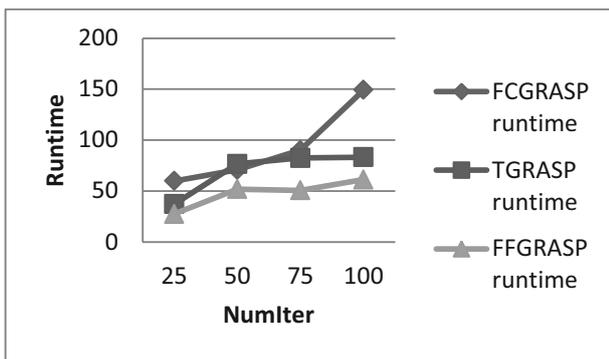


Figure 7. The effect of NumIter parameter on runtime for dataset Colon.

Figures 6 and Figure 7 depict the classification accuracy and runtime of one of the two class datasets, colon with different NumIter values. It was observed that by changing the value of NumIter, there is a change in the value of accuracy and runtime. Proposed TGRASP algorithm shows better results than FCGRASP in

this dataset. Runtime of FFGRASP is less than that of TGRASP and FCGRASP in colon dataset.

Figure 8 and Figure 9 depict the classification accuracy and runtime of a multiclass dataset, leukemia with different NumIter values. It depicts that by changing the value of the number of iterations, there is a change in the value of accuracy and runtime. So, it can be said that NumIter parameter affects the quality of optimum.

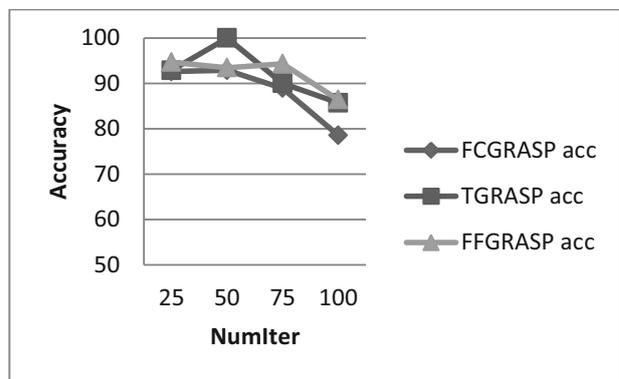


Figure 8. The effect of NumIter parameter on accuracy on leukemia dataset.

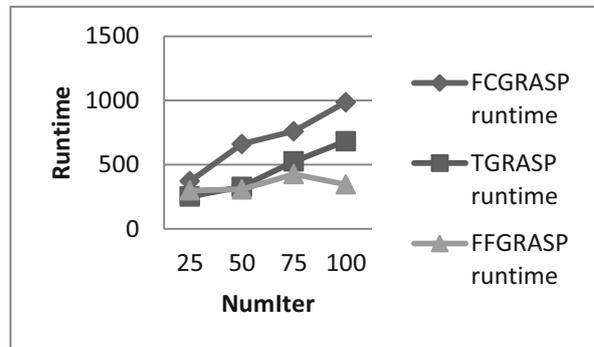


Figure 9. The effect of NumIter parameter on runtime on leukemia dataset.

Table 5 lists the gene accession numbers for some of the features obtained as the best features in regard to maximum accuracy and minimum number of genes. Due to limited space in the table, we are not able to mention all the genes selected. We have mentioned only the top few genes which came out using the naïve Bayes classifier.

Table 5. Accession number of microarray dataset.

Dataset	Gene List
Prostate	33299_at, 41455_at, 39578_at, 38203_at, 32447_at, 31749_f_at, 38537_at, 40307_at
Colon	Hsa.43684 H72110, Hsa.2777 Z23064, Hsa.2986 X61587, Hsa.43279 H64489, Hsa.41323U32519, Hsa.3201 U26710, Hsa.9174 D28137, Hsa.1768 M63509, Hsa.2555 X63432, Hsa.41338D31716
Breast	Contig42973_RC AI275405, Contig54295_RC, Contig41613_RC, Contig36517_RC, Contig54974_RC, Contig17316_RC, Contig31572_RC, Contig17159_RC, Contig56093_RC, Contig48760_RC, Contig34895_RC
SRBCT	814773, 841334, 246035, 769890, 70332, 1461737, 759173, 52996, 627273, 758266, 203469, 43021
Endometrium	169412, 167364, 165682, 162534, 165784, 162381, 160543
Leukemia	D21241_xpt1_s, D26579_at, X16609_s_at, M60556_ma2_at, L11370_at, D87071_at
Melanoma	141562, 724112, 842906, 151418, 898258, 232628, 795598, 730149, 841263, 28469, 950430, 503097, 294881, 814526, 44975, 773236, 122822, 307532, 782259, 299388

5. Conclusion

In this paper, TGRASP and FFGRASP algorithms are proposed based on GRASP method applied to cancer microarray data. The tabu search and firefly optimization algorithms are applied in the improvement phase of the GRASP method and are compared with the existing hybrid algorithm that uses hill climbing in the improvement phase (FCGRASP). According to the results obtained we can say that, GRASP with tabu search algorithm TGRASP is better or comparable for two class datasets in terms of all three performance parameters viz. accuracy, runtime and number of a selected subsets of features. Firefly optimization approach with GRASP (FFGRASP) has been performing more efficiently for multiclass cancer microarray datasets than FCGRASP. We have also compared both proposed algorithms on a

unified metric which has proved that TGRASP and FFGRASP algorithms are performing better than FCGRASP on six out of nine and on seven out of nine datasets respectively. The other advantage of the proposed TGRASP over previous algorithms is that it not only improves the accuracy, but can also control the tradeoff between accuracy and runtime. It is possible to increase or decrease the accuracy and number of selected features by changing the value of the parameters NumIter and Tabu tenure. Comparison of FFGRASP with TGRASP on a unified metric, lets us decide that FFGRASP is performing better on eight out of nine datasets. In the future, we would like to explore which specific values of NumIter and Tabu tenure would be set in order to obtain an optimal number of features from a dataset.

References

- [1] T. R. Golub *et al.*, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring", *Science*, vol. 286 no. 5439, pp. 531–537, 1999. <http://dx.doi.org/10.1126/science.286.5439.531>
- [2] J. Khan *et al.*, "Classification and Diagnosis Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks", *Nature Medicine*, vol. 7, no. 6, pp. 673–679, 2001. <http://dx.doi.org/10.1038/89044>
- [3] Z. Bellman, "Adaptive Control Processes: A Guided Tour", Princeton University Press, 1961. <http://dx.doi.org/10.1515/9781400874668>
- [4] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection", *Journal of Machine Learning Research*, vol.3, pp.1157–1182, 2003.
- [5] T. Howley *et al.*, "The Effect of Principal Component Analysis on Machine Learning Accuracy with High-Dimensional Spectral Data", *Knowledge-Based Systems*, vol.19, no.5, pp. 363–370, 2006. <http://dx.doi.org/10.1016/j.knosys.2005.11.014>
- [6] Q. Song *et al.*, "A Fast Clustering Based Feature Subset Selection Algorithm for High Dimensional Data", *IEEE Transaction on Knowledge and Data Engineering*, vol.25, no.1, pp.1–14, 2013. <http://dx.doi.org/10.1109/TKDE.2011.181>
- [7] J. Huang *et al.*, "A Filter Approach to Feature Selection Based on Mutual Information", *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on*, vol. 1, 2006. <http://dx.doi.org/10.1109/COGINF.2006.365681>

- [8] Yu Lei and Huan Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution", *Proceedings of the Twentieth International Conference on Machine Learning*, Washington DC, vol. 20 no. 2, 2003, pp. 856–863.
- [9] R. Kohavi and G. H. John, "Wrapper for Feature Subset Selection", *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.
[http://dx.doi.org/10.1016/S0004-3702\(97\)00043-X](http://dx.doi.org/10.1016/S0004-3702(97)00043-X)
- [10] T. A. Feo and M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures", *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.
<http://dx.doi.org/10.1007/BF01096763>
- [11] K. Kira and L. A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm", *Proc. of 10th National Conference on Artificial Intelligence*, 1992, pp. 129–134.
- [12] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF", *Proceedings of European Conference Machine Learning*, 1994, pp. 171–182.
http://dx.doi.org/10.1007/3-540-57868-4_57
- [13] H. Almuallim and T. G. Dietterich, "Algorithms for Identifying Relevant Features", *Proc. of Ninth Canadian Conf. on Artificial Intelligence*, 1991, pp. 38–45.
- [14] M. A. Hall, "Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning", in *Proceedings of 17th International Conference on Machine Learning*, 2000, pp. 359–366.
- [15] K. Yang *et al.*, "A Stable Gene Selection in Microarray Data Analysis", *BMC Bioinformatics*, vol. 7, pp. 228, 2006.
<http://dx.doi.org/10.1186/1471-2105-7-228>
- [16] H. Alshamlan *et al.*, "mRMR-ABC: A Hybrid Gene Selection Algorithm for Cancer Classification Using Microarray Gene Expression Profiling", *Biomed Research International*, 2015.
<http://dx.doi.org/10.1155/2015/604910>
- [17] Z. Hu *et al.*, "Hybrid Filter-Wrapper Feature Selection for Short-Term Load Forecasting", *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 17–27, 2015.
<http://dx.doi.org/10.1016/j.engappai.2014.12.014>
- [18] H. Alshamlan *et al.*, "A Comparative Study of Cancer Classification Methods Using Microarray Gene Expression Profile", in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Singapore, Springer, vol. 285, 2014, pp. 389–398.
http://dx.doi.org/10.1007/978-981-4585-18-7_44
- [19] P. Bermejo *et al.*, "A GRASP Algorithm for Fast Hybrid (Filter-Wrapper) Feature Subset Selection in High-Dimensional Datasets", *Pattern Recognition Letters*, vol. 32, no. 5, pp. 701–711, 2011.
<http://dx.doi.org/10.1016/j.patrec.2010.12.016>
- [20] T. A. Feo and M. G. C. Resende, "A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem", *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, 1989.
[http://dx.doi.org/10.1016/0167-6377\(89\)90002-3](http://dx.doi.org/10.1016/0167-6377(89)90002-3)
- [21] M. G. C. Resende and C. C. Ribeiro, "Greedy Randomized Adaptive Search Procedures: Advances, Hybridizations and Applications", in *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Berlin, Heidelberg: Springer, vol. 146, 2010, pp. 283–319.
http://dx.doi.org/10.1007/978-1-4419-1665-5_10
- [22] P. Festa and M. G. C. Resende, "Effective Application of GRASP", in *Wiley Encyclopedia of Operations Research and Management Sciences*, vol. 3, 2011, pp. 1609–1617.
<http://dx.doi.org/10.1002/9780470400531.eorms0281>
- [23] M. A. Esseghir, "Effective Wrapper-Filter Hybridization Through Grasp Schemata", *Feature Selection in Data Mining*, vol. 10, pp. 45–54, 2010.
- [24] S. C. Yusta, "Different Metaheuristic Strategies to Solve the Feature Selection Problem", *Pattern Recognition Letters*, vol. 30, no. 5, pp. 525–534, 2009.
<http://dx.doi.org/10.1016/j.patrec.2008.11.012>
- [25] R. Ruiz *et al.*, "Incremental Wrapper-Based Gene Selection from Microarray Data for Cancer Classification", *Pattern Recognition*, vol. 39, no. 12, pp. 2383–2392, 2006.
<http://dx.doi.org/10.1016/j.patcog.2005.11.001>
- [26] M. J. Flores *et al.*, "Mining the ESROM: A Study of Breeding Value Classification in Manchego Sheep by Means of Attribute Selection and Construction", *Computers and Electronics in Agriculture*, vol. 60, no. 2, pp. 167–177, 2008.
<http://dx.doi.org/10.1016/j.compag.2007.08.004>
- [27] X. S. Yang and X. He, "Firefly Algorithm: Recent Advances and Applications", *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.
<http://dx.doi.org/10.1504/IJSI.2013.055801>
- [28] X. S. Yang, "Nature-Inspired Metaheuristic Algorithm", Luniver Press, 2008.
- [29] X. S. Yang, "Firefly Algorithms for Multimodal Optimization", *Proc. of 5th Symposium on Stochastic Algorithms, Foundations and Applications*, (Eds. O. Watanabe and T. Zeugmann) Lecture Notes in Computer Science, 5792, 2009, pp. 169–178.
http://dx.doi.org/10.1007/978-3-642-04944-6_14
- [30] H. M. Zawbaa *et al.*, "Feature Selection via Chaotic Antlion Optimization", *PLoS ONE*, vol. 11, no. 3, 2016.
<http://dx.doi.org/10.1371/journal.pone.0150652>
- [31] <http://datam.i2r.a-star.edu.sg/datasets/krbd/>

- [32] M. Sardana *et al.*, "An Incremental Feature Selection Approach Based on Scatter Matrices for Classification of Cancer Microarray data" *International Journal of Computer Mathematics*, vol.92, no.2, 2015, pp. 277–295.
<http://dx.doi.org/10.1080/00207160.2014.905680>
- [33] S. Ramaswamy *et al.*, "Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures", *Proc. of Nat. Acad. Sci. USA*, vol. 98, no. 26, 2001, pp.15149–15154.
<http://dx.doi.org/10.1073/pnas.211566398>
- [34] G. Wang *et al.*, "A Feature Subset Selection Algorithm Automatic Recommendation Method", *Journal of Artificial Intelligence Research*, vol. 47, pp. 1–34, 2013.

Contact addresses:

Arpita Nagpal
 Department of Computer Science and Engineering
 School of Engineering and Technology
 The NorthCap University
 Gurugram, 122017 Haryana
 India
 e-mail: er.arpitanagpal@gmail.com

Deepti Gaur
 Department of Computer Science and Engineering
 School of Engineering and Technology
 The NorthCap University
 Gurugram, 122017 Haryana
 India
 e-mail: deeptigaur@ncuindia.edu

Received: May 2016
Revised: June 2017
Accepted: June 2017

ARPITA NAGPAL received the bachelor's degree from Punjab Technical University, Punjab, India and Master's in Technology from Jaypee Institute of information and Technology, Noida in 2011. She is currently a PhD student in Department of computer science at NorthCap University, Gurugram, India. She has teaching, and industrial experience as data scientist. Her research interest includes data mining, machine learning.

DEEPTI GAUR received the M.Tech in CSE degree from BIT Mesra Ranch, India and PhD from Banashali University, Banasthali India. Dr. Deepti Gaur is working as an Associate Professor in NorthCap University Gurugram, Haryana, India. She has 20 years of teaching and research experience in the field of computer science and information technology. She had successfully completed a project of AICTE govt. of India, based on data mining and machine learning. She has published more than 25 research papers in international journals and international conference of repute. She was an organizing chair of an IEEE International Conference (IACC 2014) in India.
