Fire Fly Based Feature Selection Approach

Hema Banati[1] and Monika Bajaj[2]

[1] University of Delhi, Department of Computer Science, Dyal Singh College
Delhi, India

[2] University of Delhi, Department of Computer Science
Delhi, India

Abstract

Irrelevant, noisy and high dimensional data, containing large number of features, degrades the performance of data mining and machine learning tasks. One of the methods used in data mining to reduce the dimensionality of data is feature selection. Feature selection methods select a subset of features that represents original features in problem domain with high accuracy. Various methods have been proposed that utilize heuristic or nature inspired strategies along with Rough Set Theory (RST) to find these subsets. However these methods either consume more time to find subset or compromise with optimality. The paper presents a new feature selection approach that combines the RST with nature inspired *'firefly'* algorithm. The algorithm simulates the attraction system of real fireflies that guides the feature selection procedure. The experimental result proves that the proposed algorithm scores over other feature selection method in terms of time and optimality.

Keywords: *Feature Selection, Rough Set, Firefly Algorithm.*

1. Introduction

Applications such as machine learning, pattern recognition, signal processing, text processing, and web content classification involve huge dataset, containing large number of features that are difficult to process. Furthermore presence of irrelevant, noisy and redundant data degrades the quality of output from these applications. One of the pre processing steps that remove the redundant and irrelevant data and maps the original high dimensional space data onto a new reduced dimensionality space data is dimensionality reduction. It helps in visualizing and representing the data that improves the performance of the application. The most common method used for dimensionality reduction is feature selection. Feature selection is a technique to select optimal subset of features that represent the original features in problem domain with high accuracy [4]. The feature selection process can proceed in either forward or backward direction known as forward selection and backward selection method respectively. The forward selection method initiates with blank set and progress by adding one feature at each step that decreases the error till any further addition does not significantly decrease the error. The backward selection method as the name suggests begins with set of all features and eliminate one by one at each step that decreases the error the most until any further removal increases the error significantly.

One of the important mathematical tools to find a subset (termed as reduct) of the original features that characterizes the basic properties of original features in problem domain is Rough Set Theory (RST) [15,16,17,18]. It discovers data dependencies and reduce the number of attributes contained in data set by purely structural methods [3,6]. A '*reduct'* is a minimal subset of attributes that enables the same classification of elements of the universe as a whole set of Universe. Rough Set Based Attribute Reduction (RSAR) [3] approach provides a theoretical background to feature selection problem. However RSAR based methods need exhaustive search to identify optimal subset of features which is quite impractical for most datasets. Various heuristic and random search strategies based methods such as Quick Reduct [3] and Entropy Based Reduct method [4] are capable to avoid such complexity. These strategies ensure the success in less time at a cost of degree of optimality. They produce the same result every time which is close to minimal reduct though which is still useful in greatly reducing data set dimensionality. Nature inspired algorithm are more powerful for such complex optimization problems. Various efforts have been made that combine the RSAR approach with nature inspired algorithm to improve the performance such as GenRSAR [14], AntRSAR [7], PSO-RSAR [1] and BeeRSAR [20] based on Genetic algorithm, Ant colony Optimization, Particle Swarm Optimization and Bee Colony optimization respectively. These algorithms are able to increase the degree of optimality but suffer from some limitations. The performances of GenRSAR, AntRSAR and PSO-RSAR are not consistent as it varies with the parameter values which are application dependent. Though BeeRSAR algorithm does not require any random parameter assumption but it consumes more time to find the reduct [20]. In this paper we present a novel approach for feature selection based on nature inspired *"Firefly"* algorithm (FA). Firefly algorithm simulates the attraction system of real fireflies. Real fireflies produce luminescent flashes as a signal system to communicate with other fireflies, especially to prey attraction [22]. Firefly algorithm (FA) [21] formulates this flashing characteristic of real firefly with the objective function of the problem to be optimized. The proposed algorithm (FA_RSAR) is an effort that combines FA together with RST to ensure the success in less time without compromising the degree of optimality in terms of size of subset and corresponding dependency degree. Moreover the algorithm does not require any random parameter assumption and produce the same result every time. The rest of this paper is structured as follows. The RSAR approach is described in followed section. Section 3 provides brief review of various RSAR based feature selection methods. Section 4 explains the proposed algorithm. Experimental results are shown in section 5 followed by section 6 that concludes the paper.

2. Fundamentals of Rough Sets

The rough set [16] is the approximation of a vague concept by a pair of precise concepts, called lower and upper approximations, which are informally a classification of the domain of interest into disjoint categories. Thus objects belonging to the same category characterized by the same attributes are not distinguishable. Rough Set Attribute Reduction (RSAR) [3] employs a rough set to remove redundant conditional attributes from discrete-valued datasets, while retaining their information content. Central to RSAR is the concept of indiscernibility. Let I = (U,A) be an information system, where U is a non-empty set of finite objects (the universe); A is a non-empty finite set of attributes such that $a : U \to V_a$ for every $a \subset A$. With any $P \subseteq A$ there is an associated equivalence relation *IND*(*P*):

$$INP(P) = \{(x,y) \subset U^2 \,|\forall a \subset P a(x) = a(y)\} \quad (1)$$

The partition of U, generated by *IND (P)* is denoted U/*P* and can be calculated as follows:

$$U / P = \otimes\{a \subset P : U / IND(\{a\})\} \quad (2)$$

where

$$A \otimes B = \{X \cap Y : \forall X \subset A, \forall\, Y \subset B, X \cap Y \neq \emptyset \quad (3)$$

If $(x, y) \subset IND(P)$, then x and y are indiscernible by attributes from P. The equivalence classes of the P-indiscernibility relation are denoted $[x]_p$.

Let $X \subseteq$ U, the P-*lower* approximation $\underline{P}X$ and upper approximation PX of set X can now be defined as:

$$\underline{P}X = \{x|\, [x]_P \subseteq X\} \quad (4)$$

$$\overline{P}X = \{x|\, [x]_P \cap X \neq \emptyset\} \quad (5)$$

Let P and Q be equivalence relations over U, then the positive, negative and boundary regions can be defined as:

$$POS_P(Q) = \bigcup_{x \in U/Q} \underline{P}X \quad (6)$$

$$NEG_P(Q) = \mathbb{U} - \bigcup_{x \in U/Q} \overline{P}X \quad (7)$$

$$BND_P(Q) = \bigcup_{x \in U/Q} \overline{P}X \quad \bigcup_{x \in U/Q} \underline{P}X \quad (8)$$

The positive region contains all objects of U that can be classified to classes of U/Q using the knowledge in attributes P.

An important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes Q depends totally on a set of attributes P, denoted $P \Rightarrow Q$, if all attribute values from Q are uniquely determined by values of attributes from P. If there exists a functional dependency between values of Q and P, then Q depends totally on P. Dependency can be defined in the following way:

For $P,Q \subset$ A, it is said that Q depends on P in a degree k ($0 \leq k \leq 1$), denoted P $\Rightarrow_K Q$, *if*

$$K = \gamma p(Q) = \frac{|POSp(Q)|}{|U|} \quad (9)$$

If k = 1, Q depends totally on P, if $0 < k < 1$ Q depends partially (in a degree k) on P, and if $k = 0$ then Q does not depend on P. headings are to be column centered in a bold font without underline. They need be numbered. "2. Headings and Footnotes" at the top of this paragraph is a major heading.

3. Reduction Methods

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same quality of classification as the original. A *reduct* is defined as a subset R of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$ A given dataset may have many attribute reduct sets, so the set R of all reduct is defined as:

$$R = \{X : X \subseteq C, \gamma_X(D) = \gamma_C(D)\} \quad (10)$$

The intersection of all the sets in R is called the *core*, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the dataset. In RSAR, a reduct with minimum cardinality is searched for; in other words an attempt is made to locate a single element of the minimal reduct set $R_{min} \subseteq R$:

$$R_{min} = \{X : X \subset R, \forall Y \subset R, |X| \leq |Y|\} \quad (11)$$

The most basic solution to obtain minimal reduct is to simply generate all possible reducts and choose any one with minimal cardinality. Exploring all the possible reducts and select the best one is an expensive task as well as not practical possible for large data set. Many real applications need only one reduct with minimal cardinality. So all the calculations involved in discovering the rest are pointless. To improve the RSAR method a better approach is need. This section explains how different methods have been improved the performance of RSAR.

3.1 Quick Reduct

The Quick Reduct Algorithm given in fig1 applies a forward selection approach to identify minimal reduct without exhaustively generating all possible subsets. It starts with an empty set and adds one feature that results in the greatest increase in dependency at a t ime, until the maximum possible value for the dataset is achieved.

```
QUICKREDUCT (C,D)
C, the set of all conditional features;
D, the set of decision features.
R ← { }
do
    T← R
    $\forall x \subset (C - R)$
    if$\gamma_{RU\{x\}}$(D)> $\gamma_T$(D)
        T ← RU{x}
    R ← T
until$\gamma_R$(D)== $\gamma_C$(D)
return R
```

Figure:1 Quick Reduct

It calculates the dependency of each feature and selects the best one. The next best feature is added (that results in the greatest increase in dependency) at each iteration until the dependency of the reduct candidate equals the consistency of the dataset (1 if the dataset is consistent).This process, however, is not guaranteed to find a minimal and optimal reduct. It generates a close-to-minimal reduct, which is still useful in greatly reducing dataset dimensionality.

3.2 Entropy Based Feature Reduction

Entropy Based Reduction (EBR) method based on [6,19] is another approach to discover reduct. A similar approach has been adopted in [4] where an entropy measure is used for ranking the features. The approach relies on the observation that when the rough set dependency measure is maximized for a given subset, the entropy is minimized. The resulting entropy is 0 in case of consistent datasets when the dependency degree is 1. EBR is concerned with examining a dataset and determining those attributes that provide the most gain in information. The entropy of attribute A (which can take values a1...am) with respect to the conclusion C (of possible values c1...cn) is defined as:

$$E(A) = -\sum_{j=1}^{m} p(a_j)\sum_{i=1}^{m} p(c_i|a_j)\log_2 p(c_i|a_j) \quad (12)$$

Using this entropy measure the algorithm used in RSAR is modified (fig2) to deal with the subsets of attributes instead of individual attributes only. The algorithm selects the subset with the lowest resulting entropy, at each step. The search for the best feature subset is stopped when the resulting subset entropy is equal to the entropy of the full set of conditional attributes. However, the entropy measure is a m ore costly operation than that of dependency evaluation which may be an important factor when processing large datasets.

```
EBR (C)
C, the set of all conditional features;
R ← { }
do
    T← R
    $\forall x \subset (C - R)$
    if E($_{RU\{x\}}$)<E(T)
        T ← RU{x}
    R ← T
until E(R) == E(C)
return R
```

Figure:2 Entropy Based Reduct Algorithm

3.3 Genetic Based Reduct (GenRSAR)

Genetic Based Reduct (GenRSAR) approach employs a genetic search strategy which is quite effective for rapid search of large, nonlinear and poorly understood spaces [5]. Unlike classical feature selection strategies where one solution is optimized, a population of solutions can be modified at the same time [8]. This can result in several optimal (or close-to-optimal) feature subsets as output. A feature subset is represented as a binary string with length equal to the number of features present in the dataset. A zero or one in the j^{th} position in the chromosome denotes the absence or presence of the j^{th} feature in this particular subset. An initial population of chromosomes is created and genetic operators (crossover and mutation) are applied on this pool of feature subsets that generates the new feature subset pool. This pool is again evaluated by calculating the fitness of individuals using a suitable criterion function. This function evaluates the goodness of a feature subset; a larger value indicates a better subset. However the size of population, how they are created and which types of crossover and mutation used are some important issues.

The initial population consists of 100 randomly generated feature subsets, the probability of mutation and crossover set to 0.4 and 0.6 respectively, and the number of generations is set to 100. The fitness function is defined as follows:

$$\textit{fitness}(\mathrm{R}) = \gamma_{\mathrm{R}}(\mathrm{D}) * \frac{|\mathrm{C}|-|\mathrm{R}|}{|\mathrm{C}|} \tag{13}$$

3.4 Ant Colony Based Reduct (AntRSAR)

The ant colony optimization is a probabilistic technique inspired by the foraging behavior of real ants. The real ants find the shortest route due to deposition of pheromone in the path. Each ant probabilistically prefers to follow a direction rich in this chemical. ACO requires a problem to be represented as a graph. The Ant RSAR approach [7] reformulates the feature selection task into Ant Colony Optimization problem by representing the problem as a graph. Each node in a graph represents a feature and edges between them denoting the choice of the next feature. Initially each ant at each node traverses through the graph until a stopping criterion is satisfied. They constructed a path in order to search the optimal feature subset which is minimum number of nodes visited.

The heuristic desirability of traversal and edge pheromone levels are combined to form the so-called probabilistic transition rule, denoting the probability of an ant at feature i choosing to travel to feature j at time t:

$$P_{ij}^{K}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \subset J}[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} \tag{14}$$

where k is the number of ants, J_i^k the set of ant k's unvisited features, ηij is the heuristic desirability of choosing feature j when at feature i and τ_{ij}(t) is the amount of virtual pheromone on edge (i, j). The choice of α and β is determined experimentally.

The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed for a certain number of times, then the process halts and outputs the best feature subset encountered. If neither condition holds, then the pheromone is updated, a new set of ants are created and the process iterates once more. To tailor this mechanism to find rough set reduct, it is necessary to use the dependency measure as the stopping criterion. This means that an ant will stop building its feature subset when the dependency of the subset reaches the maximum for the dataset (the value 1 for consistent datasets). The dependency function may also be chosen as the heuristic desirability measure, but this is not necessary.

3.5 Particle Swarm Based Reduct (PSO-RSAR)

Particle swarm algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. The PSO-RSAR[1] discover the best feature combinations by observing changes in positive region as the particles proceed through the search space. The approach defined the search space of m dimensions for the reduction problem where T = (U,C,D,V,*f*) decision table, C, the set of condition attributes, consists of m attributes. Accordingly, each particle's position is represented as a binary string of length m. Each dimension of the particle's position maps one condition attribute. The domain for each dimension is limited to 0 or 1. The value '1' means the corresponding attribute is selected while '0' means not selected. Each position can be "decoded" to a potential reduction solution, a subset of C. The particle's position is a series of priority levels of the attributes. The sequence of the attribute will not be changed during the iteration. But after updating the velocity and position of the particles, the particle's position may appear as real values such as 0.4, etc. It is meaningless for the reduction. Therefore, a discrete particle swarm optimization [14] is introduced for this combinatorial problem. During the search procedure, each individual is evaluated using the fitness. According to the definition of rough set reduct, the reduction solution must ensure that the decision ability is the same as the primary decision table and the number of attributes in the feasible solution is kept as low as possible. The algorithm first evaluates whether the potential reduction solution satisfies POSE = Upos or not (E is the subset of attributes represented by the potential reduction solution). If it is a feasible solution, the number of 1's in it is calculated. The solution with the lowest number of 1's is selected. For the particle swarm, the lesser the number of 1's in its position, the better the fitness of the individual. POSE = Upos is used as the criterion of the solution validity. As a summary, the particle swarm model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d-dimension problem space to search the new solutions, where the fitness *f* can be measured by calculating the number of condition attributes in the potential reduction solution. Each particle has a position represented by a position-vector pi (i is the index of the particle), and a velocity represented by a velocity-vector vi. Each particle remembers its own best position so far in a vector Bpi, and its j-th dimensional value is bpij. The best position-vector among the swarm so far is then stored in a vector Gp, and its j-th dimensional value is gpj. When the particle moves in a state space restricted to zero and one on each dimension, the change of probability with time steps is defined as follows:

$$P(pij(t) = 1) = f(pij(t\text{-}1), vij(t\text{-}1), Bpij(t\text{-}1), Gpj(t\text{-}1)) \tag{15}$$

Where the probability function is

$$sig(v_{ij}(t)) = \frac{1}{1 + e^{-v_{ij}(t)}} \tag{16}$$

At each time step, each particle updates its velocity and moves to a new position according to the following equation

$$vij(t) = w.vij(t\text{-}1) + \varphi_1 r_1(Bp(t\text{-}1) – p_{ij}(t\text{-}1) + \varphi_2 r_2(Gp(t\text{-}1) – p_{ij}(t\text{-}1) \tag{17}$$

$$pij(t) = \begin{cases} 1 \; if \; \rho < sig(v_{ij}(t)) \\ 0 \qquad otherwise \end{cases} \tag{18}$$

Where φ_1 is a positive constant, called as coefficient of the self-recognition component, φ_2 is a positive constant, called as coefficient of the social component. r1 and r2 are random numbers in the interval [0,1]. The variable w is called the inertia factor, whose value is typically setup to vary linearly from 1 to near 0 during the iterated processing.$\rho$is a random number in the closed interval [0, 1]. In this step, a particle decides where to move next, considering its current state, its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm.

3.5 Bee Colony Based Reduct (BeeRSAR)

Artificial Bee Colony (ABC) algorithm [9] for real parameter optimization that simulates the foraging behaviour of bee colony for unconstrained optimization problems[10,11,12,13]. For solving constrained optimization problems, a constraint handling method was incorporated with the algorithm. In a real bee colony, there are some tasks performed by specialized individuals. These specialized bees try to maximize the nectar amount stored in the hive by performing efficient division of labour and self-organization. BeeRSAR is an attempt to implement ABC algorithm for feature reduction. In this approach the bees select the feature subsets at random and calculate their fitness and find the best one in each iteration. This procedure is repeated for number of iterations to find the optimal subset. In the first step employed bees produce the feature subset in random. N random numbers are generated between 1 and N and assigned to each employed bee. The feature subset is constructed by performing round operation on these random numbers, and then extracting only the unique numbers from the set. The second step produces the new source for employed bees whose total number equals half of the number food sources by equation (19).

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj} \tag{19}$$

Where $\varphi_{ij}$is a uniformly distributed real random number within the range [-1,1], k is the index of the solution chosen randomly from the colony (k = int(rand * N) + 1), j = 1, . . .,D and D is the dimension of the problem. After producing vi, this new solution is compared to xith solution and the employed bee exploits the better source. In the third step of the algorithm, an onlooker bee chooses a food source with a higher probability and produces a new source in selected food source site. As for the employed bee, the better source is decided to be exploited. The fitness value for each bee is calculated, using the following equation:

$$fit_i = \begin{cases} \dfrac{1}{1 + f_i} & if \; f_i \geq 0 \\ 1 + abs(f_i) & otherwise \end{cases} \tag{20}$$

The probability is calculated by means of fitness value using the following equation.

$$P_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j} \tag{21}$$

Where fit_i is the fitness of the solution xi. After all onlookers are distributed to the sources, sources are checked whether they are to be abandoned. If the number of cycles that a source cannot be improved is greater than a predetermined limit, the source is considered to be exhausted. The employed bee associated with the exhausted source becomes a scout and makes a random search in problem domain by the following equation.

$$x_{ij} = x_j^{min} + (x_j^{max} - x_j^{min}) * rand \tag{22}$$

4. Fire Fly Based Reduct (FA_RSAR)

Nature inspired algorithms are more powerful for optimization problems especially NP hard problems. This paper presents a novel approach for dimensionality reduction based on firefly algorithm (FA). Firefly algorithm (FA) [21] is inspired by biochemical and social aspects of real fireflies. Real fireflies produce a short and rhythmic flash that helps them in attracting (communicating) their mating partners and also serves as protective warning mechanism. FA formulates this flashing behavior with the objective function of the problem to be optimized. The following three rules are idealized for basic formulation of FA (3) all fireflies are unisex so that fireflies will attract each other regardless of their sex. (2) Attractiveness is proportional to their brightness, which decreases as distance increases between two flies. Thus the less bright one will move towards the brighter one. In case it is unable to detect more brighter one it will move randomly. (3) The brightness of a firefly is determined by the landscape of the objective function. Figure 1 describes the basic FA algorithm.

```
Objective function $f(x)$, $x = (x_1, \ldots\ldots x_4)^T$
Generate initial population of fireflies $x_i$ (i=1,2,...n)
Light intensity $I_i$ at $x_i$ is determined by $f(x_i)$
Define light absorption coefficient $\gamma$
While(t>MaxGeneration)
for i=1:n all n fireflies
  for j=1:i all n fireflies
    if ($I_j$>$I_i$), Move firefly i towards j in d-dimension;
    endif
   Attractiveness varies with distance r via exp[-$\gamma$r]
   Evaluate new solution and update light intensity
  end for j
end for i
Rank the fireflies and find the current best
end while
Post process results and visualization
```

Fig. 3 Firefly Algorithm

The basic FA assumes there exists n fireflies $x_{i,}$ i=1,2,….,n initially positioned randomly in the space and intensity i o f each firefly is associated with the objective function $f(x), i.e. I = \alpha f(x)$. Only firefly with higher flash intensity attracts the other one i.e. I_i>I_j, j=1,2,…n, j≠i. Attractiveness or the brightness of firefly varies with the distance between firefly i and firefly j i.e. $r_{ij} = d(x_i,x_j)$. In addition the light intensity I decrease with the distance from its source and it is also absorbed in the air. Thus most of the fireflies are visible only to a limited distance.

The proposed feature selection algorithm (FA_RSAR) incorporates the basic behavior of FA with RST to improve the performance of feature selection procedure. The algorithm initially starts with n fireflies $x_{i,}$ i=1,2,….,n corresponding to each feature in conditional feature set C. The intensity I_i of each firefly x_i is initialized with its dependency with D, set of decision features:

$$I_i= \gamma_{xi}(D) \qquad (23)$$

Each firefly i find its distance with other firefly j and determine the increment in intensity of j with the movement of i towards j, using equation 24 &25 respectively, where I_j>I_i, j=1,2,…n, j≠i.

$$r_{ij}=\min(\gamma_C(D)- \gamma(_{xi,,xj})(D)) \qquad (24)$$

$$Inc_j=\gamma_{(xi,xj)}(D)-I_{xj} \qquad (25)$$

Each firefly i thus move towards its best mating partner j having minimum distance with i and movement results in greatest increase in dependency. If any firefly that does not able to find any best matting partner, the intensity of firefly i is absorbed by the system and it will be invisible to all other fireflies in the space. The movement among flies thus results in subsets of fireflies with increase in dependency. Each subset is then evaluated for stopping criterion i.e. minimality and required dependency (1 for consistent dataset). The algorithm follows the same procedure for new groups of fireflies generated in previous iteration and determines the intensity I_{ij} of each group x_{ij} by equation (25) until the stopping criterion is satisfied. Figure 4 presents the proposed FA_RSAR algorithm.

$$I_{ij}= \gamma(_{xi,,xj})(D) \qquad (26)$$

```
FA_RSAR(C,D)
C, the set of all conditional features
D, the set of decision features
Objective function R = {$X : X \subseteq C, \gamma_X(D)= \gamma_C(D)$}
Generate initial population of fireflies $x_i$ (i=1,2,...n)
corresponding to each conditional feature
Light intensity $I_i$ at $x_i$ is determined by $I_{xi}=\gamma_{xi}(D)$
F=C
while ($\gamma_X(D)!=\gamma_C(D)$)
  F'=F
  F = [ ]
  for i=1:F' fireflies
     for j=1:F' fireflies
       find the best matting partner j for i th at
       satisfies the following conditions
       (i)   Intensity of j is greater than intensity
             of i, i.e. ($I_j$>$I_i$)
       (ii)  Distance between i and j should be
             minimum in terms of distance between
             $\gamma(_{xi,,xj})(D)$ and $\gamma_C(D)$
       (iii) Movement of i towards j increases the
             intensity of j i.e. $\gamma_{(xi,xj)}(D)>I_{xj}$ and
     end for j
     Move firefly i towards j i.e $x_{ij}$.
     $I_{ij}= \gamma(_{xi,,xj})(D)$
     F = F $\cup$ $x_{ij}$
  end for i
  Evaluate each $x_{ij}$ in F for dependency i.e.
  $\gamma$ $x_{ij}(D) = = \gamma_C(D)$ and minimality
end while
```

Fig. 4 FA_RSAR Algorithm

Two critical aspects of feature selection problem are the degree of optimality (in terms of subset size and corresponding dependency degree) and time required to achieve this optimality. Existing methods achieved success in either of these aspects such as Quick Reduct and EBR methods finds reduct in less time but not guaranteed to find a minimal subset[1,7,20] whereas GenRSAR, AntRSAR, PSO-RSAR and BeeRSAR methods improve the performance but consume more time [7,20]. The proposed FA_RSAR algorithm works on both these aspects. It provides a stochastic approach that overcomes the shortcomings of conventional hill climbing approach and infeasibility of complete search to identify optimal subset. The algorithm can identify several optimal or close to optimal features subset as compared to classical feature selection strategies where one solution is optimized. Moreover it does not require any random parameter assumptions hence produces the consistent results.

5. Experiments and Results

The algorithm described in previous section is applied in medical domain to find minimal attribute set that classify the object without deterioration of classification quality. Four different medical datasets were obtained from UCI machine learning data repository [2] to evaluate the performance of proposed algorithm. Table 1 shows the details of datasets used in this paper.

Table 1 Detail of Data Sets Used for Experiment

Data Set Name	Total Number of Instances	Total Number of Features
Dermatology	366	34
Cleveland Heart	300	13
Lung Cancer	32	56
Wisconsin	699	09

Table 2 Results of various methods

Data set	Dermatology	Cleveland Heart	Lung Cancer	Wisconsin
Total Features	34	13	56	09
RSAR	10	7	4	5
EBR	10	7	4	5
AntRSAR	8-9	6-7	4	5
GenRSAR	10-11	6-7	6-7	5
PSORSAR	7-8	6-7	4	4-5
BeeRSAR	7	6	4	4
FA_RSAR	7	3	5	4

Table 2 shows the reduct results in terms of size of the reduct found by various methods. Experimental result shows that the proposed FA_RSAR algorithm produces better results as compared to RSAR, EBR, AntRSAR and GenRSAR and performs at par with PSORSAR and BeeRSAR methods. Moreover the search technique applied in FA_RSAR algorithm consumes less time to find subset without compromising the results as compared to PSORSAR and BeeRSAR methods. Hence the experimental study proves that FA_RSAR outperforms the other feature selection methods.

6. Conclusion

Feature selection is a valuable preprocessing technique for applications involving huge data. It refers to the problem of selecting minimal attribute set that are most predictive to represent the original attributes in data set. The paper discussed the strengths and weaknesses of various existing feature selection methods. These methods either fail to find optimal data reductions or require more time to achieve better results. The paper presents a new feature selection algorithm (FA_RSAR) that incorporates the basic behavior of Firefly Algorithm with RST to improve the performance. The performance of the proposed algorithm is evaluated by comparing it with other methods using medical datasets. The experimental results prove that our algorithm exhibits consistent and better performance in terms of time and optimality as compared to other methods.

References

[1] B. Yue et al.(2007) “A New Rough Set Reduct Algorithm Based on Particle Swarm Optimization”, IWINAC, Part I, LNCS 4527, pp. 397–406.

[2] Blake, C.L., and Merz C.J. UCI Repository of machine learning databases. Irvine, University of California, 1998. http://www.ics.uci.edu/`mlearn/.

[3] Chouchoulas, A. and Shen, Q. (2001) ‘Rough set-aided keyword reduction for text categorization’, Applied Artificial Intelligence, Vol. 15, No. 9, pp. 843-873.

[4] Dash, M. and Liu, H. (1997) ‘Feature Selection for Classification’, Intelligent Data Analysis, Vol. 1, No. 3, pp. 131-156.

[5] Holland, J. (1975) Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbour.

[6] Jensen, R. and Shen, Q. (2001) 'A Rough Set-Aided System for Sorting WWW Bookmarks', In N. Zhong et al. (Eds.), Web Intelligence: Research and Development, pp. 95-105.

[7] Jensen, R. and Shen, Q. (2003) 'Finding Rough Set Reducts with Ant Colony Optimization', Proceedings UK Workshop on Computational Intelligence, pp. 15–22.

[8] Kudo, M. and Skalansky, J. (2000) 'Comparison of algorithms that select features for pattern classifiers', Pattern Recognition, Vol. 33, No. 1, pp. 25-41.

[9] Karaboga, D. (2005) 'An idea based on honey bee swarm for numerical optimization', Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[10] Karaboga, D. and Basturk, B (2006) 'An Artificial Bee Colony (ABC) algorithm for numeric function optimization', In IEEE Swarm Intelligence Symposium 2006, Indiana, USA.

[11] Karaboga, D. and Basturk, B. (2007a) 'A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm', Journal of Global Optimization, Vol. 39, No. 3, pp. 459–471.

[12] Karaboga, D. and Basturk, B. (2007b) 'Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems' Foundations of Fuzzy Logic and Soft Computing', LNCS, Springer-Verlag, Vol. 4529, pp. 789–798.

[13] Karaboga, D. and Basturk, B. (2008) 'On the performance of Artificial Bee Colony (ABC) algorithm', Applied Soft Computing, Vol. 8, No. 1, pp. 687–697.

[14] Liu, H. Abraham, A. and Li, Y. (2009) 'Nature Inspired Population- Based Heuristics for Rough Set Reduction', Rough Set Theory, SCI, Springer-Verlag, Vol. 174, pp. 261-278.

[15] Pawlak, Z. (1982) 'Rough Sets', International Journal of Computer and Information Sciences, Vol. 11, pp. 341–356.

[16] Pawlak, Z. (1991) Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers.

[17] Pawlak, Z. (1993) 'Rough Sets: Present State and The Future', Foundations of Computing and Decision Sciences, Vol. 18, pp. 157–166.

[18] Pawlak, Z. (2002) 'Rough Sets and Intelligent Data Analysis', Information Sciences, Vol. 147, pp. 1–12.

[19] Quinlan, J.R. (1993) C4.5: Programs for Machine Learning, The Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

[20] Suguna, N. and Thanuskodi, K.(2010), 'A Novel Rough Set Reduct Algorithm for Medical Domain Based on Bee Colony Optimization', Journal of Computing, Volume2, Issue6,June 2010, ISSN 2151-9617

[21] Yang, X.,(2009), 'Firefly Algorithm for Multimodal Optimization', SAGA 2009, LNCS 5792,pp.169-178,2009.

[22] Zang et.al.,(2010), "A Review of Nature Inspired Algorithm" Journal of Bionic Engineering 7 Suppl. (2010) s232-s237.

First Author: Dr. Hema Banati completed her Ph. D in 2006 after pursuing her Master of Computer Applications both from University of Delhi. She is an associate professor, Department of Computer Science, Dyal Singh College, University of Delhi and is actively supervising Ph.D students at Department of Computer Science, University of Delhi. Her research interests include Web engineering, software engineering, Human Computer Interfaces, multiagent systems, E-commerce and E-learning. She has 31 publications to her credit

Second Author: Monika Bajaj is a research scholar in Computer Science Department at University of Delhi. Her research interests include Web engineering, Human Computer Interfaces and E - commerce. She has 5 publications to her credit.