Journal of Theoretical and Applied Information Technology
10th May 2013. Vol. 51 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BAT ALGORITHM FOR ROUGH SET ATTRIBUTE REDUCTION

[1]AHMED MAJID TAHA, [2]ALICIA Y.C. TANG

[1, 2] College of Graduate Studies, Universiti Tenaga Nasional, Malaysia
E-mail: [1]ahmedmt12002@yahoo.com , [2]aliciat@uniten.edu.my

ABSTRACT

Attribute reduction (AR) refers to the problem of choosing an optimal subset of attributes from a larger set of possible attributes that are most predictive for a given result. AR techniques have recently attracted attention due to its importance in many areas such as pattern recognition, machine learning and signal processing. In this paper, a new optimization method has been introduced called bat algorithm for attribute reduction (BAAR), the proposed method is based mainly on the echolocation behavior of bats. BAAR is verified using 13 benchmark datasets. Experimental results show that the performances of the proposed method when compared to other features selection methods achieve equal or better performance.

Keywords: *Attribute Reduction; Bat Algorithm; Nature Inspired; Rough Set.*

1. INTRODUCTION

Investigating how to select a subset of attributes from the original set of attributes while retaining an appropriately high accuracy in representing the original attributes are defined as attribute reduction. In real world problems, attribute reduction is a necessity due to the noisiness, misleading or irrelevant attributes [1]. By eliminating these attributes, extract knowledge from data can benefit greatly learning procedures and prediction tasks. The main goal of attribute reduction in data mining and machine learning is to enhance the predictive accuracy of a classification algorithms, decrease the dimensionality of feature space, pick up the comprehensibility, and the visualization of the induced concepts [2]. Attribute reduction problem confronted in many areas such as image recognition, bioinformatics, clustering, text categorization, systems monitoring as well as rule induction [1]. The importance of attribute reduction with the growing of high-dimensional data has become an essential task of a learning procedure.

The growing complication of real life problems has encouraged computer scientists to investigate for proficient problem-solving techniques. The behavior of ants, termites, bird's fishes, bees slime, moulds, and other creatures have enthused swarm intelligence investigators to create new optimization algorithms. Decentralized control and self-organization for those creatures are extraordinary features of swarm-based systems. New algorithms are emerging recently, these include bacterial foraging [3] fireflies algorithm [4], cockroaches infestation [5], slime moulds algorithm [6], and diverse bees algorithms.

Hill-climbing algorithms overwhelmingly fail to find optimal (even near optimal) solutions [7]. Since hill-climbing optimization method can fail and is easily trapped in local optima, much research efforts have shifted to meta-heuristics, for instance simulated annealing (SA) [7], ant colony optimization (ACO) [8], genetic algorithm (GA) [7], tabu search (TS) [9], particle swarm optimization (PSO) [10], and more recently Scatter search [11]. Metaheuristics algorithms can often obtain high quality solutions in reasonable time[12]. The former advantage guides the algorithm convergence to the optimality (exploitation) and the latter avoid the algorithm sticking in local optima in addition to the loss of diversity (exploration). The alignment between exploitation and exploration could lead to the global optimality achievement, accordingly selection efficient algorithms is a difficult task, if not impossible [4]. the selection requires extensive experience and knowledge of the problem of interest as well. Even so, there is no guarantee that an optimal or even suboptimal solution can be found [13]. It is well-known that attribute reduction is an NP-hard problem [14], the number of possible subsets is always start increasing exponentially for the reason that there are 2^N subsets for N features.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Therefore, it is necessary to consider efficient and effective meta-heuristic algorithms.

The organization of this paper is as follows: a rough set theory preliminary has been presented in Section 2. This followed by a detailed description of the bat algorithm in Section 3. The proposed approach to attribute reduction based on bat algorithm is discussed in Section 4. The section also includes the pseudo-code and parameters values of the presented algorithm. Experimental setup and results are described in Sections 5. Section 6 discusses and compares the various results. Section 7 concludes the paper, while future work and limitation are presented in Section 8.

2. ROUGH SET THEORY (RST)

Rough set theory (RST) was proposed in 1982 by Pawlak [15]. RST is a valid mathematic tool to handle imprecision (error rate), uncertainty and vagueness. RST is an extension of conventional set theory that supports approximations in decision making process. RST is one of the most effective techniques to attributes reduction, which can preserve the meaning of the features. It has been broadly applied in many domains *and proved its* value as a superior technique [16, 17]. RST implements attributes reduction task by using merely the granularity structure of the data, no further knowledge is needed. This uniqueness of the theory has been ascertained in works described by [8, 16]. The RST performance in previous work such as [9-11] has stimulated the authors to select RST as an evaluator for proposed solution. The theory of rough set is described *briefly* as *follow*: Let I = (U, A) be an information system, where U, described universe, is a nonempty set of a limited objects; *A* is a indiscernibility associate IND (*P*):

$$\text{IND}(P) = \{(x, y \in U^2 \mid \forall a \in P, a(x) = a(y)\} \quad (1)$$

If (x, y) ∈ IND (*P*): then *x* and *y* are indiscernible by features from *P*. The family of all equivalence classes of IND (*P*), that is, the partition identified by *P*, is denoted as *U/P*. An equivalence class of IND (*P*), that is, the block of the partition *U/P*, including x is denoted by $[x]_p$. The indiscernibility relative is the mathematical basis of rough set theory.

In rough set theory, the lower approximation and upper approximation are two essential operations. Given a random set $X \subseteq U$, the P-lower approximation of *X*, denoted as $\underline{P}X$, is the set of all elements of *U*, which can be definitely classified as elements of *X* based on the attribute set *P*. The P-upper approximation of *X*, denoted as *PX*, is the set of all elements of *U*, which can be probably classified as elements of *X* derived from the attribute set P. These two definitions can be expressed as:

$$\underline{P}X = \{x \mid [x]_p \subseteq X\} \quad (2)$$

$$\underline{P}X = \{x \mid [x]_p \cap X \neq \varnothing\} \quad (3)$$

A significant concern in data analysis is discovering dependencies between attributes. Dependency degree equation, given $P, Q \subseteq A$, the dependency degree is defined by

$$k = r_p(Q) = \frac{|POS_p(Q)|}{|U|} \quad (4)$$

Where |*Y*| is the is the cardinality of set *Y*. $POS_p(Q)$, called positive region, is defined by

$$POS_p(Q) = \bigcup_{X \in U/Q} \underline{P}X \quad (5)$$

The positive region includes all objects in *U* that can be exclusively classified to block of the partition *U/Q* by means of the knowledge in features *P*. The amount k can be used to calculate the dependency degree between *Q* and *P*. If $k = 1$, *Q* depends completely on *P* which means that all features values from *Q* are exclusively identified via values of features from *P*. If $0 < k$ 1, *Q* depends on *P* in a degree k. If $k = 0$, then *Q* does not depend on *P*. The evaluation of solution quality including three approaches; filter, wrapper and embedded, RST provide a filter based tool to extract knowledge from a field in a concise method. It can conserve the information content while reducing the quantity of attributes concerned. Based on degree of dependency, a reduct is represented by the following formula. Where R be a subset of C, then R is said to be a reduct if

$$r_R(D) = r_C(D) \wedge \forall R' \subset R, r_{R'}(D) < r_C(D) \quad (6)$$

Particularly, a reduct with minimum cardinality is called minimum reduct. The purpose of attribute reduction is to find a minimum reduct. Its purpose function is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\min_{R \in \mathbf{R}} |R| \qquad (7)$$

Where R is the set which consists of all reducts of C.

3. BAT ALGORITHM

The bat algorithm (BA) was first presented in [18] and it has been applied to benchmark functions, accordingly BA outperforms particle swarm optimization and genetic algorithms. BA has also been successfully applied to tough optimization problem such as motor wheel optimization problem [19], clustering problem [20], in addition to 8 well-known engineering optimization tasks [13]. BA implementations in the mentioned literature have attracted the authors to select this algorithm for attributes reduction task. Bats are animals that have wings and possess the capability of echolocation (also called biosonar). Echolocating animals emit calls to the environment and listen to the echoes of those calls. These echoes will be used to locate and identify the objects. Among all the bat species, microbats use echolocation extensively [21]. In microbats, echolocation is a type of sonar used to detect prey, avoid close obstacles in the dark, and locate roosting crevices. During echolocation these microbats emit a series of short, high-frequency sounds and listen for the echo that bounces back from the surrounding objects [22] as illustrated in figure 1. With this echo a bat can determine an object's size, shape, direction, distance, and motion. When the bats fly close to their prey, the rate of pulse emission can accelerate up to 200 pulses per second. A constant frequency in each pulse is also observed. The wavelengths of a pulse are in the same order of their prey sizes. The loudness for searching for prey is greater than when homing towards the prey [21]. In other words, the loudness decreases when getting nearer to the victims.

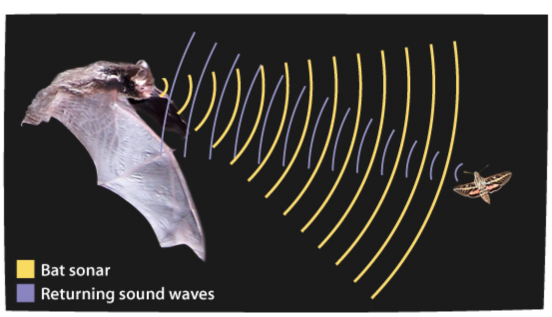

Fig. 1: Bat Sonar

The idea of the BA is to mimic the bats when catching their prey. BA enforcement is more complicated than many other meta-heuristic algorithms [18] in that each agent (bat) is assigned a set of interacting parameters such as position, velocity, pulse rate, loudness, and frequencies. This interaction affects the quality of a solution and the time needed to obtain such solution.

The principle of bat algorithm is as follows. A swarm of bats is assumed to fly randomly with velocity V_i at position X_i with a fixed frequency f, varying wavelength λ and loudness A_0 to search for a victim. They have capability to adjust the wavelength of their emitted pulses in addition to regulate the rate of pulse emission $r \in [0, 1]$, relying on the closeness of their target. Although the loudness can be different in many ways, the loudness differs from a large (positive) A_0 to a minimum constant value $A_{\min}$. The frequency f is in the range $[f_{\min}, f_{\max}]$ corresponds to a range of wavelengths $[\lambda_{\min}, \lambda_{\max}]$. For example, a frequency range of [20 kHz, 500 kHz] corresponds to a range of wavelengths from 0.7mm to 17mm.

4. BAT ALGORITHM FOR ATTRIBUTE REDUCTION

4. 1 Frequency Representation

The frequency will be a positive integer or a float number depending on the selected minimum and maximum frequency. As a matter of fact, frequency is fixed and it calculates throw equation (7). The choice of minimum and maximum frequency values are based on the domain of interest. As illustrated in equation (8) the frequency does affect the velocity. In the original design of BA, β value is constant, but through experiments it was found that if this value function is set random between 0 and 1 will lead to more diversity in the velocity, thus leading to more solutions in the search space.

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \qquad (7)$$

4.2 Velocity Representation

The velocity of each bat is represented as a positive integer number. Velocity suggests how many of the bat's attributes should be changed at a certain moment in time. The bats will communicate with each other through the global best, in order to be the same like that of the global best position (solution). Experiment results (Section 5) show that the algorithm can obtain good but not optimal solution. Besides, the algorithm will take longer time to obtain good solution. It is observed that the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

velocity may reach up to 15 in some cases, which indicates 15 attributes might be changed at certain times. One of particle swarm algorithm founders Kennedy stated that when the maximum velocity is too high, particles might fly away thus missing the path to good solutions [26]. This phenomenon persists when the maximum velocity is too low, where particles will have difficulty escaping from locally optimal regions. So we adopt this approach for the proposed algorithm and we set maximum velocity V_{max} = (1/3) * N where N is the number of features, the value is selected based on the data in [10]. Equation (2) is used to adjust the velocity during each iteration.

$$v_i^t = v_i^{t-1} + (x_i^t - x_*) f_i \qquad (8)$$

4.3 Position Representation

In the proposed algorithm each bat's position is formulated as binary bit strings of length N, where N is the total number of features. Each bit represents a feature, the '1' means the corresponding feature is selected and the '0' means not selected. Each position is a subset of features and the position is updated according to equation (9).

$$x_i^t = x_i^{t-1} + v_i^t \qquad (9)$$

4.4 Loudness Representation

Loudness A_i is represented as numbers of features which will be changed at certain time in local search according to equation (4), where A_i^t is the average loudness of all the bats at certain iteration, while $\varepsilon \in [-1,1]$. The sound loudness (A) also has a range, i.e., between the maximum loudness and minimum loudness. If we assume that A_{max} = 3 and A_{min} = 1, this refers to when the bat getting closer to the target it begins to reduce the number of features from 3 features to 2 features then become a single feature. Sound loudness value plays an important role in obtaining the solution.

$$x_{new} = x_{old} + \varepsilon A^t \qquad (10)$$

Through iterations the loudness value will decrease if the bat started approaching the best solution. Equation (11) shows that the amount of decrease is determined by α value which plays a similar role as cooling factor of a cooling schedule in the simulated annealing algorithm. After a few experiments we found that the appropriate values for A_{max} are 2 or 3 depending on the dataset dimension. In this work we set the maximum loudness equal to 2 and the minimum loudness equal to 1.

$$A_i^{t+1} = \alpha A_i^t \qquad (11)$$

4.5 Pulse Rate Representation

Pulse rate r_i value will play a role in whether a local search procedure around the global best solution should be conducted or skipped. The higher value of pulse rate will reduce the likelihood of conducting a local search around the global best and vice versa. Accordingly when the bat approaching the best solution, pulse rate value will increase and subsequently influencing the skip of a local search around the global best. The amount of increase is determined by the γ value as defined in the equation (12).

$$r_i^{t+1} = r_i^0[1-\exp(-\gamma t)] \qquad (12)$$

4.6 Objective Function

The solution will straighten out of RST, Where $\xi_R(D)$ is the classification quality of condition attribute set R relative to decision D, $|R|$ refer to the length of elected attribute subset. $|C|$ is the total number of features. δ and φ are two parameters corresponding to the importance of classification quality and subset length, δ ∈ [0, 1], φ = 1- δ, the objective function is calculated according to equation (13).

$$Sol_Q = \delta.\xi_R(D) + \varphi.\frac{|C|-|R|}{|C|} \qquad (13)$$

This formula denotes the classification quality and feature subset length have different importance. We adopt this approach based on the work done in [10, 23], they states that classification quality is more significance than the size of subset, as a result both parameters have been set as follow: δ = 0.9, φ = 0.1. The high δ guarantees that the best position is at least a real rough set reduction. The quality of each position is calculated according to equation 8, the goal is to maximize fitness values. Fig. 2 presents the pseudo-code for BAAR.

---

1 Initialize parameters: A, A_{min}, A_{max}, r, f_{min}, f_{max}, P_{max}, I_{max}, V_{max}, V_{min}, Φ, δ, γ, α

2 Generate a swarm with P_{max} bats

3 Calculate cost function for all bats

4 Find the current best bat (x_*)

Journal of Theoretical and Applied Information Technology
10th May 2013. Vol. 51 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

While stop condition not met **Do**
For i = 1 **to** P_{max} **Do**
Frequency $f_i = f_{\min} + (f_{\max} - f_{\min})\beta$
Velocity $v_i^t = v_i^{t-1} + (x_i^t - x_i) f_i$
If $(V_i > V_{max})$ **Then** $(V_i = V_{max})$ **End-If**
If $(V_i > V_{min})$ **Then** $(V_i = V_{min})$ **End-If**
Locations $x_i^t = x_i^{t-1} + v_i^t$
If (Rand > r_i) **Then**
calculate εA^t
If $(\varepsilon A^t > A_{max})$ **Then** $(\varepsilon A^t = A_{max})$ **End-If**
If $(\varepsilon A^t > A_{min})$ **Then** $(\varepsilon A^t = A_{min})$ **End-If**
Generate a local solution around the best solution (x_*) $[x_{new} = x_{old} + \varepsilon A^t]$
End-If
Calculate εA^t
If $(\varepsilon A^t > A_{max})$ **Then** $(\varepsilon A^t = A_{max})$ **End-If**
If $(\varepsilon A^t > A_{min})$ **Then** $(\varepsilon A^t = A_{min})$ **End-If**
Generate a new solution around the current solution $[x_{new} = x_{old} + \varepsilon A^t]$
If (Rand < A_i) & ($f(x_i) < f(x_*)$)
Consent the new solution
Increase r_i $r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)]$
Decrease A_i $[A_i^{t+1} = \alpha A_i^t]$
End-If
End-For
Find the current best solution (x_*)
End-While

Fig. 2: Pseudo Code For BAAR

5. EXPERIMENTAL RESULTS

The proposed method is tested on a computer running Window Vista with core 2 duo 2.0 GHZ processor and 2 GB memory. Algorithm based on BAAR was programmed in MATLAB and applied to 13 well-known datasets with diverse numbers of attributes and cases, 10 of them from UCI [24]. The rest are m-of-n, exactly, and exactly2 [25]. Table 1 described the datasets. The BAAR MATLAB code for each dataset was run 20 times with different initial solutions as suggested by [7]. BAAR was terminated after 250 iterations, same as [7, 8].

Table 1. Description Of The Data Sets Used In Experiments.

Datasets	No. of features	No. of samples
Lung	56	32
WQ	38	521
Derm2	34	358
Derm	34	366
Letters	25	26
LED	24	2000
Mushroom	22	8124
Credit	20	1000
Vote	16	300
Heart	13	294
Exactly2	13	1000
Exactly	13	1000
M-of-N	13	1000

BAAR setting parameters has been summarized with their assigned values in Table 2. These values are based on our numerical experiments and the original algorithm presented in [18]. BAAR is compared with five other attribute reduction algorithms as listed below.

1. Simulated annealing for rough set attribute reduction (SimRSAR) [7].
2. Ant colony optimization for rough set attribute reduction (AntRSAR) [8].
3. Genetic algorithm for rough set attribute reduction (GenRSAR) [8].
4. Tabu search for attribute reduction (TSAR) [9]
5. Scatter search for attribute reduction (SSAR) [11]

Journal of Theoretical and Applied Information Technology
10th May 2013. Vol. 51 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Parameter	Symbol	Value
Sound Loudness	A	0.9
Pulse Rate	r	0.5
Minimum Frequency	f_{min}	0
Maximum Frequency	f_{max}	2
No. of Population	P_{max}	25
No. of Iteration	I_{max}	250
Increase Pulse Rate Value	γ	0.9
Decrease Sound Loudness Value	α	0.9
Maximum Loudness	A_{max}	2
Minimum Loudness	A_{min}	1
Maximum Velocity	V_{max}	(1/3) * N
Minimum Velocity	V_{min}	1
Weighting Value	δ	0.9
Weighting Value	Φ	0.1

Table 2. Parameters Values Used In The Tests.

Minimal reduct comparison is provided in Table 3, all algorithms have the same number of runs for each dataset, except the results of SimRSAR for Heart, Vote and Drem2 datasets for which the number of runs are 30, 30 and 10, respectively. For each algorithm, the size of reduct obtained at every run is given. Between the brackets is the total number of runs that this cardinality is achieved. The number of features without brackets denotes that the method could obtain this number of features for all runs.

6. RESULTS DISCUSSION

This work ascertains that the performance of BAAR is superior to many existing algorithms that were applied to the same datasets. The results obtained for heart dataset exclusive minimal reduct are better than the best known solution obtained by other methods. Moreover, these good solutions are obtained in all experiment runs, attributes that have been selected for heart dataset are 1, 4, 5, 7 and 8. BAAR could find the best known minimal reducts

Table 3. Minimal Reduct Comparison.

Datasets	BAAR	SimRSAR	AntRSAR	GenRSAR	TSAR	SSAR
M-of-N	6	6	6	$6^{(6)}7^{(12)}$	6	6
Exactly	6	6	6	$6^{(10)}7^{(10)}$	6	6
Exactly2	10	10	10	$10^{(9)}11^{(11)}$	10	10
Heart	5	$6^{(29)}\ 7^{(1)}$	$6^{(18)}\ 7^{(2)}$	$6^{(18)}7^{(2)}$	6	6
Vote	8	$8^{(15)}\ 9^{(15)}$	8	$8^{(2)}9^{(18)}$	8	8
Credit	8	$8^{(18)}\ 9^{(1)}\ 11^{(1)}$	$8^{(12)}\ 9^{(4)}\ 10^{(4)}$	$10^{(6)}11^{(14)}$	$8^{(13)}9^{(5)}10^{(2)}$	$8^{(9)}\ 9^{(8)}\ 10^{(3)}$
Mushroom	4	4	4	$5^{(1)}6^{(5)}7^{(14)}$	$4^{(17)}5^{(3)}$	$4^{(12)}\ 5^{(8)}$
LED	5	5	$5^{(12)}\ 6^{(4)}\ 7^{(3)}$	$6^{(1)}7^{(3)}8^{(16)}$	5	5
Letters	$8^{(18)}9^{(2)}$	8	8	$8^{(8)}9^{(12)}$	$8^{(17)}9^{(3)}$	$8^{(5)}\ 9^{(15)}$
Derm	$6^{(13)}7^{(7)}$	$6^{(12)}\ 7^{(8)}$	$6^{(17)}\ 7^{(3)}$	$10^{(6)}11^{(14)}$	$6^{(14)}7^{(6)}$	6
Derm2	$9^{(12)}\ 10^{(8)}$	$8^{(3)}\ 9^{(7)}$	$8^{(3)}\ 9^{(17)}$	$10^{(4)}11^{(16)}$	$8^{(2)}9^{(14)}10^{(4)}$	$8^{(2)}\ 9^{(18)}$
WQ	$12^{(2)}\ 13^{(11)}14^{(7)}$	$13^{(16)}\ 14^{(4)}$	$12^{(2)}\ 13^{(7)}\ 14^{(11)}$	16	$12^{(1)}13^{(13)}14^{(6)}$	$13^{(4)}\ 14^{(16)}$
Lung	$4^{(10)}\ 5^{(6)}\ 6^{(4)}$	$4^{(7)}\ 5^{(12)}\ 6^{(1)}$	4	$6^{(8)}7^{(12)}$	$4^{(6)}5^{(13)}6^{(1)}$	4

Journal of Theoretical and Applied Information Technology
10th May 2013. Vol. 51 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

for all tested datasets except for Derm2. For 8 of the total datasets presented in Table 3, BAAR gains the best minimal reducts in all runs. BAAR outperforms SimRSAR in 5 datasets, equal in 5 datasets, worse in 2 datasets and comparative in 1 dataset. BAAR outperforms AntRSAR in 4 datasets, equal in 5 datasets and worse in 4 datasets. BAAR outperforms GenRSAR in all datasets. BAAR outperforms TSAR in 4 datasets, equal in 4 datasets, comparative in 3 datasets and worse in 2 datasets. BAAR outperforms SSAR in 5 datasets, equal in 5 datasets and worse in 3 datasets. BAAR can be competitive or better than existing methods in data mining optimization such as genetic algorithm and ant colony optimization. The results also show that BAAR has strong inherent search ability in the problem space that can efficiently find minimal reducts. Nevertheless, BAAR sometimes could not obtain the best known solutions same as others method just as there is no single method that always tend to select the best solution for all datasets, but BAAR outperforms in term of obtaining unique solutions such as in heart dataset.

7. CONCLUSIONS

This paper discusses a novel attribute reduction method based on BA and RST. To find out whether the proposed algorithm can locate an optimal reduct, numerical experiments has been conducted on 13 well-known datasets. Comparisons with other attribute reduction algorithms have revealed that BAAR has a superior performance. BAAR exploits the strength of existing (successful) algorithms with an added feature inspired by the echolocation behavior of micro bats. The novelty comes from the combined use of the exiting algorithmic steps and the solicited essential feature of the bats to produce good results. With regard to the parameters used, BAAR has 14 parameters. This is the highest number of parameters generated when compared to other benchmarking methods. TSAR has 7 parameters, SSAR has 5 parameters, GenRSAR and SimRSAR has 3 parameters for both, and lastly AntRSAR has 2 parameters. As a conclusion for this study, BAAR method provides a promising attribute reduction technique.

8. LIMITATION AND FUTURE WORK

The parameter setting in the proposed algorithm is rather static. It is recommended that the BAAR be integrated with a method to flexibly tuning the parameters in order to make it mutable or dynamic. Accordingly the algorithm is able to deal with each single dataset in a different way, relying on its characteristics such as the dimension of a dataset. Further investigation will focus on two other aspects, namely the classification accuracy and the time consumed for the above runs to achieve better minimal reduct. Applying BAAR to more datasets is currently on the go. The result obtained can be useful for deeper analysis of the said theory especially from the perspective of its behavior and its shortcomings (if any). Collectively, all the future works will contribute to the natural extension of the proposed BAAR, making it a smarter and more robust technique for attribute reduction.

REFERENCES:

[1] Jensen, R., Combining rough and fuzzy sets for feature selection, in School of Informatics. 2005, University of Edinburgh: Edinburgh.
[2] Liu, H. and H. Motoda, feature selection for knowledge discovery and data mining. Vol. 454. 1998. 244
[3] Passino, K.M., Biomimicry of bacterial foraging for distributed optimization and control. Control Systems, IEEE, 2002. 22(3): p. 52-67.
[4] Yang, X.-S., Nature-Inspired Metaheuristic Algorithms. First ed. 2008: Luniver Press.
[5] Havens, T.C., et al. Roach Infestation Optimization. in Swarm Intelligence Symposium, 2008. SIS 2008. IEEE. 2008.
[6] Monismith, D.R. and B.E. Mayfield. Slime Mold as a model for numerical optimization. in Swarm Intelligence Symposium, 2008. SIS 2008. IEEE. 2008.
[7] Jensen, R. and Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. Knowledge and Data Engineering, IEEE Transactions on, 2004. 16(12): p. 1457-1471.
[8] Jensen, R. and Q. Shen. Finding Rough Set Reducts with Ant Colony Optimization. in Proceedings of the 2003 UK Workshop on Computational Intelligence. 2003.
[9] Hedar, A.-R., J. Wang, and M. Fukushima, Tabu search for attribute reduction in rough set theory. Soft Computing - A Fusion of Foundations, Methodologies and Applications, 2008. 12(9): p. 909-918.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] Wang, X., et al., Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters, 2007. 28(4): p. 459-471.

[11] Jue, W., et al. Scatter Search for Rough Set Attribute Reduction. in Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on. 2009.

[12] Talbi, E.-G., Metaheuristics: From Design to Implementation. 2009: Wiley

[13] Pawlak, Z., Rough sets. International Journal of Parallel Programming, 1982. 11(5): p. 341-356.

[14] Swiniarski, R.W. and A. Skowron, Rough set methods in feature selection and recognition. Pattern Recognition Letters, 2003. 24(6): p. 833-849.

[15] Duan, Q., et al., Personalized Web Retrieval based on Rough-Fuzzy Method. Journal of Computational Information Systems, 2007.

[16] Yang, X.-S. and A.H. Gandomi, Bat Algorithm: A Novel Approach for Global Engineering Optimization. Engineering Computations, 2012. 29(5).

[17] Pawlak, Z., Rough Sets: Theoretical Aspects of Reasoning about Data 1 edition ed. 1991: Springer.

[18] Yang, X.-S., A New Metaheuristic Bat-Inspired Algorithm Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), J. González, et al., Editors. 2010, Springer Berlin / Heidelberg. p. 65-74.

[19] Bora, T.C., L.S. Coelho, and L. Lebensztajn, Bat-Inspired Optimization Approach for the Brushless DC Wheel Motor Problem. Magnetics, IEEE Transactions on, 2012. 48(2): p. 947-950.

[20] Khan, K., A. Nikov, and A. Sahai, A Fuzzy Bat Clustering Method for Ergonomic Screening of Office Workplaces Third International Conference on Software, Services and Semantic Technologies S3T 2011, D. Dicheva, Z. Markov, and E. Stefanova, Editors. 2011, Springer Berlin / Heidelberg. p. 59-66.

[21] Altringham, J.D., T. Mcowat, and L. Hammond, Bats:Biology and Behaviour. 1998: Oxford Univesity Press.

[22] Parpinelli, R.S. and H.S. Lopes, New inspirations in swarm intelligence: a survey. International Journal of Bio-Inspired Computation, 2011. 3(1): p. 1-16.

[23] Bae, C., et al., Feature selection with Intelligent Dynamic Swarm and Rough Set. Expert Systems with Applications, 2010. 37(10): p. 7026-7032.

[24] BLAKE, C.L. and C.J. MERZ, UCI repository of machine learning databases. University of California, Irvine, 1998.

[25] Raman, B. and T.R. Ioerger, Instance Based Filter for Feature Selection. 2002.

[26] Kennedy, J. The particle swarm: social adaptation of knowledge. in Evolutionary Computation, 1997., IEEE International Conference on. 1997.