International Journal of Soft Computing 8 (1): 12-19, 2013
ISSN: 1816-9503

Dynamic Tabu Search for Dimensionality Reduction in Rough Set

Zalinda Othman, Azuraliza Abu Bakar, Salwani Abdullah, Mohd Zakree Ahmad Nazri and Nelly Anak Sengalang
Faculty of Information Sciences and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: This study proposed a Dynamic Tabu Search (DTSAR) that incorporates a dynamic Tabu list to solve an attribute reduction problem in Rough Set Theory. The dynamic Tabu list is used to skip the aspiration criteria and promote faster running times. A number of experiments have been conducted to evaluate the performance of the proposed technique with other published metaheuristic techniques, rough set and decision tree. DTSAR shows promising results on reduct generation time. It ranges between 0.20-22.18 min. For comparisons on the performances that are based on the number of produced reduct, DTSAR is on par with other metaheuristic techniques. DTSAR outperforms some techniques on certain datasets. Quality of classification rules generated by adopting DTSAR is comparable with those generated by Rough Set and Decision Trees Methods.

Key words: Tabu search, attribute reduction, rough set, computational intelligence, dynamic Tabu list

INTRODUCTION

In the last decade, dealing with data is not a problem for any organization as the amount of data is small. Today, there are too much of data that can be handled. Data can be from business transactions, educational reports, health reports, satellite images, scientific data and others. With these huge collections of data, a better solution for information retrieval is needed to improve the managerial decision making process. Data mining plays an important part to deliver a higher level of information, called knowledge, to the users. It is a task of discovering interesting patterns from large amount of data from databases (Han and Kamber, 2006). According to Han and Kamber, data mining consists of few stages such as cleaning, integration, selection, transformation, pattern evaluation and knowledge presentation. The initial stage is known as the data pre-processing and it includes cleaning, integration, selection, reduction and transformation. Data reduction in data pre-processing stage is one of the crucial steps in data mining. Its purpose is to reduce the data dimension into a smaller dimension but produces almost the same analytical result as the original data.

Dimensionality reduction is the study of a method to reduce the number of data dimensions that is used to describe the object. Its aim is to remove irrelevant and redundant data, thus reducing the computational cost and improving the data quality (Dash and Liu, 2008). In dimensionality reduction, the method of attribute subset selection is applied. According to Han and Kamber (2006), the aim of attribute subset selection is to find a minimum set such that the result given by the probability distribution of the data classes is almost the same to the original distribution obtained using all attributes. There are approaches used in dimension reduction, i.e., the heuristic methods and rough set theory based reducts computation.

The main focus of this research is to investigate Tabu search algorithm for dimensionality reduction problem in rough set theory and use the reduce dataset to generate good classification rules. This study proposes a rough set classifier model based on using dynamic Tabu list.

PRELIMINARIES

Dimensionality reduction: One of the important stages in data mining is the attribute reduction stage. There are several techniques to do attribute reduction such as data aggregation and compressions and numerousity and dimensionality reductions. Dimensions refer to the measurement of object perspective. Dimensionality reduction is the method for decreasing the object dimensions. Reducing the dimensions of object is also referred as reducing the number of attributes. The general objective is to remove irrelevant and redundant data and to reduce the computational cost and avoid data over-fitting. Dimensionality reduction will improve the quality of data for efficient data processing task such as pattern recognition and data mining (Dash and Liu, 2008). It is often classified into feature selection or extraction. In feature selection a subset of the original features are

Corresponding Author: Zalinda Othman, Faculty of Information Sciences and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

selected in the end while in feature extraction, the features are extracted using some mapping from the original set of features (Dash and Liu, 2008).

Feature extraction is defined as given a set of feature $S = \{v_1, v_2, ..., v_D\}$ which is then by finding a new set of features S' derived from a linear or non-linear mapping of S (Dash and Liu, 2008). The cardinality of $|S'| = d$ and $J(S') \geq J(T)$ for all derived set of features T with $|T| = d$ where J is the evaluation function and d is the parameter set by the user. Feature extraction is when all existing features are recombined to generate new features. Mapping in feature extraction is defined as transforming any original D dimensional feature vector to a new d dimensional feature vector.

Feature selection, meanwhile is defined as given a set of feature $S = \{v_1, v_2, ..., v_D\}$, then finding a subset S' of S with $|S'| = d$ such that $J(S') \geq J(T)$ for all $T \subset S$ and $|T| = d$ where J is the evaluation function and d is specified by the user. There are four components required by feature selection such as a generation or Search Strategy, Evaluation Method, a stopping criterion and/or Validation Method. From D original features, the generation of strategy is the process where a selected set of features combination is decided.

Rough Set Theory: Rough Set Theory has been proposed by Pawlak in 1982 and is one of the powerful mathematical tools to deal with uncertainty and vagueness of data (Thangavel and Pethalakshmi, 2009). Rough set act as a classifier in data mining to compute a set of reducts which consists of indispensable attribute required for the decision. The main issue in rough set is to find a set of interesting attribute called reduct. It is known that calculation of reduct of an information system is a key problem in Rough Set Theory (Pawlak, 1991; Swiniarski and Skowron, 2003; Jensen and Shen, 2004). Detail of Rough Set Theory is given by Pawlak (1991).

Finding minimal reduct is NP-hard (Pawlak, 1991). In order to find minimal reduct, it is needed to locate and generate all possible solution reduct and choose the reduct with minimal cardinality. This procedure is only applicable for small datasets as bigger datasets will cause complex calculation of dependency measures and indiscernibility relation. There are many others approaches employing rough set for attribute reduction which are using the metaheuristic approaches such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), scatter search and Tabu search.

Tabu search: Fred Glover introduced Tabu search in 1986. It is one of the search space technique used to solve the non-linear problem. The word Tabu means something or action that is prohibited or banned due to religions and believes.

Tabu search is a memory-based approach and the most important component in Tabu search is its adaptive memory that allows more searching behaviour. This unavoidable element of Tabu search allows it to increase the memory so that it allows the status of Tabu changes from time to time. Tabu search is a metaheuristic that guides a local heuristic search procedure to explore the solution space beyond the local optima (Glover and Laguna, 1997).

The basic approach for Tabu search is based on the hill climbing algorithms. Tabu search has a concept of short term memory called Tabu list which is used to avoid the searching process trap in local optima. Tabu list is a list that contains the earlier solutions located in the search. When a local minimum is found, Tabu search allows non improving moves to be made and disallows any solutions that has already appeared in the Tabu list to be sampled again. This mean that the search is not repeated and will save a lot of times. Any improvements to the candidate solution are sure to be new un-sampled solutions.

Metaheuristic techniques in attribute reduction: There are number of researches done in rough set attribute reduction using metaheuristic techniques. Scatter search for rough Set Attribute Reduction (SSAR) was proposed by Wang *et al*. (2009). Their SSAR Method is according to the structure of the scatter search discussed earlier which consists the procedures of diversification generation, improvement, reference set update, subset generation, solution combination and intensification. The experimental result for SSAR shows that it performed better than other metaheuristic techniques in term of numbers of minimal reducts obtained. SSAR have also proven that it is much cheaper in computing the dependency degree function than other metaheuristic techniques.

According to Jensen and Shen (2003) in their research on finding rough set reduct with Ant Colony Optimization technique (ACO), it was found out that the proposed AntRSAR performed better in term of time spent to discover reducts than GenRSAR.

Particle Swam Optimization (PSO) was also used to find minimal reducts. To apply PSO to rough set attribute reduction, the particle's position was represented as binary bit strings of length N where N is the total attribute number. Every bit represents an attribute where 1 represent that the attribute is selected and vice versa.

An attribute reduction method based on GA with heuristic information was introduced by Shi and Fu where there are two elements required to generate attribute reducts by GA. The first element is the definition and implementation of the GA. For example, the solution of reduction problem must be represented as a genome/chromosome. Second element is the definition of the objective function or fitness function.

A Tabu search technique to solve the problem for attribute reduction in Rough Set Theory was proposed by (Hedar *et al*., 2006). The proposed Tabu search technique is a high level Tabu search with long term memory where it will invokes diversification and intensification search scheme besides the Tabu search neighbourhood search methodology. Even though the three mechanisms of diversification and intensification schemes do help to achieve a better performance, the drawback of these mechanisms would be the challenge to apply them in an appropriate time to avoid more unneeded complexity or premature convergence. The results obtained when applying Tabu search to attribute reduction are promising and showed low computation lost in the dependency degree function.

TABU SEARCH FOR ATTRIBUTE REDUCTION

In this study, a Tabu search based method called Dynamic Tabu Search Attribute Reduction (DTSAR) is proposed to deal with the attribute reduction problem in Rough Set Theory. First, the component of Tabu search for attribute reduction will be described and then the DTSAR algorithm will be formally stated.

Solution representation: Tabu search attribute reduction uses binary value to represent the solution that represents the attribute subsets. The trial solution x is represented as a 0-1 vector and number of attribute of the solution x is equal to the number of condition attribute from the original dataset, denoted as |C|. If the solution subset have the value of 0, it means that the attribute is not contained in the attribute subset. Meanwhile if the trial solution subset value is 1, it means that the attribute is chosen to be inside the attribute subset.

Solution quality measurement: To measure the solution quality, the dependency degree γ measurement is used. Assuming that the current dependency degree for current solution attribute D is γ x (D) and the dependency degree for trial solution for attribute D is γ x' (D). Current solution x is better than the trial solution x' if:

- γ x (D)>γ x' (D) where the dependency degree for current solution x is more than the dependency degree for trial solution x'
- $\Sigma_i X_i < \Sigma_i X'_i$ if γ x (D) = γ x' (D) where the lesser number of attribute is accepted if both of the dependency degree for both solutions are the same

The parameters setting for the DTSAR algorithm are as follows:

- |TL|, size of Tabu list, min = 1, max = 5
- Length of neighbourhood solution = round (|C|/2)

Initial solution generation: In Tabu search, initial solution is created before the data reduction would be applied. The initial solution is constructed randomly, where each value in the attribute subset is assigned a value 1 or 0 randomly. For example, a dataset in Table 1 has five attributes, denoted as dataset A.

In order to get the initial solution, there are two concept need to follows:

- First, the number of attribute of the initial solution must be less than the attribute of the original dataset A which is the attribute for the initial solution <5
- The calculated dependency degree of original dataset A and the dependency degree have to be equal with the dependency degree of the initial solution where Dependency Degree (Initial solution) = Dependency Degree (Original dataset)

To generate initial solutions, it is firstly needed to calculate the dependency degree of each of the attributes. The equation to calculate the dependency degree measure is as follow (Pawlak, 1991) (Table 2):

$$Yp(Q) = \frac{|POSp(Q)|}{U} \quad (1)$$

Table 1: Dataset A

Patients	Headache	Muscle pain	Temperature	Flu
A1	1	1	0	0
A2	1	1	1	1
A3	1	1	1	1
A4	0	1	0	0
A5	0	0	1	0
A6	0	1	0	1

Attributes: patients, headache, muscle pain, temperature, flu; Objectives: A1, A2, A3, A4, A5, A6

Table 2: Datasets used in the experiment

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

If Yp (Q) = 1, it can be said that Q depend totally on P. If Yp (Q) <1, it can be said that Q depend partially on P.

The attribute for the highest dependency degree will be chosen and assigned as 1 and included in all related attribute subsets which cannot be removed at this stage. Other attributes will be assigned either 1 or 0 and randomly changed until the highest dependency value is achieved. The attribute subsets with highest dependency degree values are chosen as initial solutions and denoted as x. The solutions will be used in Tabu search attribute reduction.

Neighbourhood generation: The generation of neighborhood solution is by randomly selecting one attribute from the subset and switches the cell value of this attribute. If the selected attribute cell value is 1, it will be automatically changed to 0. This means that the attribute selection will not be in the attribute subset (e.g., remove one attribute). If the selected attribute cell value is 0, it will be automatically changed to 1. This means that the attribute is selected to be in the attribute subset (e.g., add one attribute). Appendix A shows the Tabu Search algorithm for generating the neighbourhood solution. The number of possible neighbourhood solution is dependent on the condition attribute, |C| which is equivalent to round (|C|/2) adopted from Hedar *et al.* (2006). This is done to avoid worsening the quality of reduction method when the problem size increases (Hedar *et al.*, 2006). The dependency degree for every generated neighborhood solution x (i) is calculated by γ (x (i)) based on the rough set dependency degree measurement and the number of attribute for that particular solution x (i) is calculated by #(x(i)). If the dependency of trial solution γ (x (i)) is more (better) than the dependency degree of the current solution γ (x) then the trial solution is accepted. If the dependency degree for both solutions is the same then the solution with lesser number of attribute will be accepted.

Tabu list: The purpose of Tabu list is to avoid generating improper solutions such as all attributes are considered (all zero) and all attributes are not considered (all one) and to avoid revisiting the same solutions (Glover and Laguna, 1997; Hedar *et al.*, 2006). Tabu list plays an important role in search of high quality solutions. The size of Tabu list in this research matched the work done by Hedar *et al.* (2006) where the min Tabu list is 1 and the maximum Tabu list is 5. Preliminary experiments have been performed by Hedar *et al.* (2006) before setting the Tabu list size and the experiments indicated that the current setting of Tabu list is good enough to escape from local optima. This research will use the same Tabu list size as proposed (Hedar *et al.*, 2006) because similar 13 datasets are used in the Tabu search reduction experiment.

The main difference between the Tabu search and (Hedar *et al.*, 2006) is that in this research short term memory dynamic Tabu list was used to keep the recency only (only the recent changed attributes) whilst (Hedar *et al.*, 2006) used static Tabu list to keep track of the recent visited solution. The main point of using dynamic Tabu list is to reduce the computational time and to increase the efficiency by releasing some attribute from Tabu, based on the attached Tabu duration which is different from one attribute to another.

Appendix B shows the part of dynamic Tabu list in the proposed algorithm. The size of dynamic Tabu list will keep changing from time to time. Unlike static Tabu list, dynamic Tabu list does not need to wait until all 5 spaces of Tabu list to be fully filled to remove a list. Dynamic Tabu list are more flexible because for each of the Tabu list, an attribute index tabu tuner, att_indx_tt(i) of 1-5 will be randomly assigned to it. After every iteration, the attribute index tabu tuner, att_indx_tt(i) will decrease by −1. If the attribute index tabu tuner, att_indx_tt(i) is 0, the value for the Tabu list is discarded. The benefit of dynamic Tabu list is to skip the aspiration criteria because if an improving solution is found and the particular solutions are already in the dynamic Tabu list, they will be discarded based on the attribute index tabu tuner = 0. It is still possible to accept the improving solution if the same improving solution is re-generated further in the next iterations. Furthermore, the usage of dynamic Tabu list can help to shorten the running time of Tabu search.

Termination criterion: Tabu search will terminate the neighbourhood generation when the dependency degree of current solution γ (x (i)) = 1. The generated solution with dependency degree γ (x (i)) = 1 is the reduct or reduce dataset of the current dataset. The data reduction obtained by using Tabu search attribute reduction is called reduct. Reduct obtained is then used to obtain classification rules and accuracy which is then imported into the rough set analysis tool to produce the classification rules.

Development of rough classifier: The classification rules are used to generate the accuracy of the classification rules. The accuracy of the classification rules is based on the confusion matrix. Confusion matrix is a useful tool for analyzing how well the classifier can recognize tuples of different classes (Han and Kamber, 2006). In the example for dataset A, two classes are obtained, where they can be stated in term of positive tuples (Flu = Yes) and negative tuples (Flu = No). True positives refer to the positive tuples that were correctly labeled by the classifier whereas false positives are the negative tuples that were labeled incorrectly and false negatives are the positive tuples that were incorrectly labeled. Positive is the number of positive (Yes) tuples and negative is the number of negative (No) tuples. The accuracy of the classification rules will then be calculated as follows:

$$\text{Accuracy} = \text{Sensitivity}\frac{\text{Positive}}{\text{Positive+Negative}} + \text{Specificity}\frac{\text{Positive}}{\text{Positive+Negative}}$$

Where:

$$\text{Sensitivity} = \frac{\text{True positive}}{\text{Positive}}$$

And:

$$\text{Specificity} = \frac{\text{True negative}}{\text{Negative}}$$

NUMERICAL EXPERIMENTS

This study explains the analysis of the experimental results of the research works. The experiments are developed using JAVA programming language. About 13 well-known datasets from UCI machine learning, namely M of N, Exactly, Exactly2, Heart, Vote, Credit, Mushroom, LED, Letters, Derm, Derm2, WQ and Lung datasets are used in the experiment as shown in Table 2. Two experiments are conducted to compare the performance of the proposed DTSAR. Firstly, DTSAR is compared against other well-know attribute reduction techniques and secondly, DTSAR is compared with rough set and decision tree methods. The results of the experiments are explained in the following paragraphs. The Tabu search attribute reduction code was executed 20 times for each datasets with different initial solutions. The results of reduct are reported in Table 3 and 4. In Table 3, the number of running times for each datasets varies from 0.20-22.18 min. There are significant different for running time taken to produce reduct for dataset with attributes 20-25. The datasets Mushrooms have 22 attributes while LED has 24 attributes. Even though there are only 2 different attributes between the two datasets, the run time are different by approximately 12.38 min. This is due to the DTSAR algorithm is trying to filter the best reduct for each dataset until the dependency is equal to 1 and will only then terminate the process. As the dependency degree does not equal to 1, the searches continue. This does explain why the complexity of data does affect the run times. Looking at dataset letters in particular where the number of attributes is 25 but the run time is 0.07 min, this is due to the fact that in letters dataset, there is only 26 number of objects found in the contents. The same goes to the dataset Lung where even though the number of attributes is the highest at 56 but the run time is only 0.20 because the number of object for the dataset is 32 only.

Comparisons of reduct using dtsar with other metaheuristic methods: Table 4 shows the results comparisons for the current research by using Tabu search which applying dynamic Tabu list with other results that applied metaheuristic techniques for attribute reduction. The number of runs to achieve the number is represented as superscripts in parentheses. The number without superscripts means that for all the 20 run times, only that particular number appear. DTSAR outperforms GenRSAR for all the tested data except for heart dataset.

Table 3: Running time obtained from DTSAR

Dataset	No. of attributes	DTSAR	Running time (min)
M of N	13	6	0.56
Exactly	13	6	0.54
Exactly2	13	10	1.42
Heart	13	$6^{(15)}7^{(5)}$	0.82
Vote	16	$8^{(19)}9^{(1)}$	0.28
Credit	20	$8^{(10)}9^{(6)}10^{(4)}$	6.65
Mushroom	22	$4^{(16)}5^{(4)}$	9.80
LED	24	$5^{(17)}6^{(3)}$	22.18
Letters	25	$8^{(19)}9^{(1)}$	0.07
Derm	34	$6^{(16)}7^{(4)}$	4.12
Derm2	34	$8^{(4)}9^{(15)}10$	4.45
WQ	38	$12^{(4)}13^{(15)}14$	9.67
Lung	56	$5^{(15)}6^{(5)}$	0.20

Table 4: Results of DTSAR with other methods based on number of attributes

Datasets	No. of attributes	AntRSAR	SimRSAR	GenRSAR	TSAR	SSAR	DTSAR
M of N	13	6	6	$6^{(6)}7^{(12)}$	6	6	6
Exactly	13	6	6	$6^{(10)}7^{(10)}$	6	6	6
Exactly2	13	10	10	$10^{(9)}11^{(11)}$	10	10	10
Heart	13	$6^{(18)}7^{(2)}$	$6^{(29)}7^{(1)}$	$6^{(18)}7^{(2)}$	6	6	$6^{(15)}7^{(5)}$
Vote	16	8	$8^{(15)}9^{(15)}$	$8^{(2)}9^{(18)}$	8	8	$8^{(19)}9^{(1)}$
Credit	20	$8^{(12)}9^{(4)}10^{(4)}$	$8^{(18)}9^{(1)}11^{(1)}$	$10^{(6)}11^{(14)}$	$8^{(13)}9^{(5)}10^{(2)}$	$8^{(9)}9^{(8)}10^{(3)}$	$8^{(10)}9^{(6)}10^{(4)}$
Mushroom	22	4	4	$5^{(1)}6^{(5)}7^{(14)}$	$4^{(17)}5^{(3)}$	$4^{(12)}5^{(8)}$	$4^{(16)}5^{(4)}$
LED	24	$5^{(12)}6^{(4)}7^{(3)}$	5	$6^{(1)}7^{(3)}8^{(16)}$	5	5	$5^{(17)}6^{(3)}$
Letters	25	8	8	$8^{(8)}9^{(12)}$	$8^{(17)}9^{(3)}$	$8^{(5)}9^{(15)}$	$8^{(19)}9^{(1)}$
Derm	34	$6^{(17)}7^{(3)}$	$6^{(12)}7^{(8)}$	$10^{(6}11^{(14)}$	$6^{(14)}7^{(6)}$	6	$6^{(16)}7^{(4)}$
Derm2	34	$8^{(3)}9^{(17)}$	$8^{(3)}9^{(7)}$	$10^{(4)}11^{(16)}$	$8^{(2)}9^{(14)}10^{(4)}$	$8^{(2)}9^{(18)}$	$8^{(4)}9^{(15)}10$
WQ	38	$12^{(2)}13^{(7)}14^{(11)}$	$13^{(16)}14^{(4)}$	16	$12^{(1)}13^{(13)}14^{(6)}$	$13^{(4)}14^{(16)}$	$12^{(4)}13^{(15)}14$
Lung	56	4	$4^{(7)}5^{(12)}6^{(1)}$	$6^{(8)}7^{(12)}$	$4^{(6)}5^{(13)}6^{(1)}$	4	$5^{(15)}6^{(5)}$

DTSAR overall outperformed GenRSAR, SimRSAR, TSAR in the Derm2 dataset based on the number of reduct which appears more frequently in the run times. DTSAR is better than AntRSAR with small differences in 3 datasets, namely LED, Derm2 and WQ. For Lung dataset, AntRSAR outperformed DTSAR. The results for other datasets between AntRSAR and DTSAR shown that they are comparable, since there are no significant different in the results.

For Vote, Derm2 and WQ datasets, it is shown that DTSAR outperforms SimRSAR. As for comparisons with SSAR, DTSAR has performed better for Mushroom, Letters, Derm2 and WQ datasets. Meanwhile with TSAR, DTSAR outperformed TSAR for Letters, Derm, Derm2 and WQ datasets. DTSAR has also performed better than SSAR for Mushroom, Letters, Derm2 and WQ datasets. As a conclusion, DTSAR has been shown to outperform other metaheuristic techniques in term of reduct. This proved that applying dynamic Tabu list as part of the Tabu search approach does generate a better yet comparable result with TSAR and with other various metaheuristic approaches also.

Comparisons of reduct with rough set and decision tree: Referring to the Table 5, it can be seen that the reduct from credit, mushroom and LED datasets outperformed the rough set and decision tree techniques. As for rough set, it outperformed DTSAR in five datasets which are Heart, Letters, Lung, Vote and WQ datasets. The same number of reduct was obtained from DTSAR and rough set techniques in Exactly, Exactly2 and M of N datasets. Looking at the overall reduct obtained from DTSAR and decision tree techniques, it can be shown that DTSAR has performed better for all datasets except for Letters dataset which decision tree has outperformed DTSAR. Figure 1 shows the comparisons of reduct obtain from DTSAR and other techniques.

Comparisons of reduct accuracy with rough set and decision tree: Looking in detail into the classification accuracy value between the three techniques in Table 6, decision tree outperformed the other two techniques by five datasets namely Credit, Derm, Derm2, Letters and Vote. The classification accuracy is very useful because from the accuracy obtain, it is possible to know if certain reduct is reliable or not reliable. For example, the Derm dataset have the best reduct in rough set theory but the accuracy for Derm dataset is very low if compared to DTSAR and decision tree which is 53.49%. This means that the reduct on by rough set is not reliable. This condition is the same for Letters, Heart, Derm2 and Exactly2 datasets. These four dataset produce a better reduct in rough set but the classification accuracy obtained is lower than other methods. As for Letters

Table 5: Results of DTSAR with other methods based on number of reduct

	No. of reduct		
Datasets	DTSAR	Rough set	Decision tree
Credit	$8^{(10)}9^{(6)}10^{(4)}$	15	18
Derm	$6^{(16)}7^{(4)}$	4	19
Derm2	$8^{(4)}9^{(15)}10$	5	18
Exactly	6	6	13
Exactly2	10	10	11
Heart	$6^{(15)}7^{(5)}$	3	10
Letters	$8^{(19)}9^{(1)}$	2	3
Lung	$5^{(15)}6^{(5)}$	4	14
M of N	6	6	13
Mushroom	$4^{(16)}5^{(4)}$	6	20
Led	$5^{(17)}6^{(3)}$	6	21
Vote	$8^{(19)}9^{(1)}$	4	12
WQ	$12^{(4)}13^{(15)}14$	1	21

Table 6: Comparisons of techniques based on accuracy

	Accuracy (%)		
Datasets	DTSAR	Rough set	Decision tree
Credit	81.80	77.00	89.70
Derm	92.07	53.49	100.00
Derm2	88.00	67.00	100.00
Exactly	100.00	100.00	94.20
Exactly2	100.00	63.50	91.83
Heart	96.25	65.28	93.22
Letters	88.46	0.00	100.00
Lung	96.87	100.00	100.00
M of N	100.00	100.00	100.00
Mushroom	94.87	100.00	98.80
Led	100.00	100.00	100.00
Vote	88.79	78.51	89.80
WQ	97.00	99.05	92.45

dataset, the accuracy of reduct from rough set is 0%. This means that using rough set to measure the reduct for that dataset is not reliable. Referring to Table 6, the number of reducts obtain for Letters is 2 and 1 for WQ but both produced different value in accuracy. Although, the reduct for WQ is 1 but the accuracy of reduct is 99.05%. The accuracy may differ in term of how well the rough set classifier can recognize tuples for the datasets (Han and Kamber, 2006) which can be recognized as true positive, false positive, true negative and false negative. All the tuples are defines based on the four classes mentioned and are dependent on how well the rough set classifier can recognize whether the tuples are correctly or incorrectly labeled in the tested datasets.

In the case of Letters with accuracy equal to 0% which mean that the produced reduct is wrongly labeled by the classifier, hence giving a smaller yet 0% to its accuracy. Figure 1 shows the comparisons on the three techniques in term of accuracy.

Comparisons of reduct classification rules with rough set and decision tree: From the reduct obtained, rules are then generated. From Table 7, it is clear that although the result for reduct is worst for decision tree technique but it has the least rules generation for all datasets. There are some significant differences between DTSAR and rough

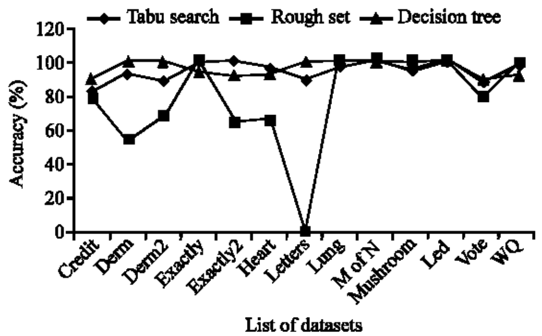

Fig. 1: Comparisons of techniques based on accuracy

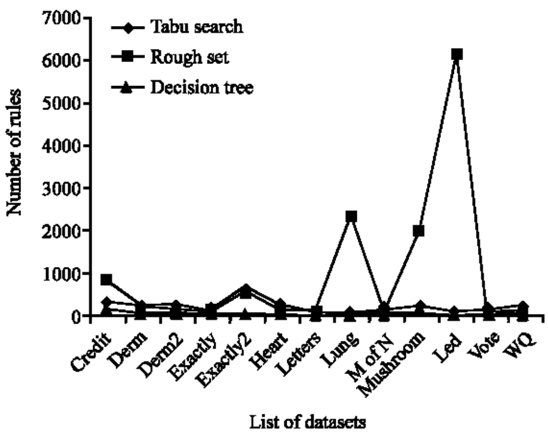

Fig. 2: Comparisons of techniques based on number of rules

set techniques. Even though rough sets do outperformed DTSAR in generating reduct for five datasets, the rule generated by Lung dataset is too huge until it reaches 2417 rules. DTSAR and rough set techniques produce the same reduct for datasets Exactly, Exactly2 and M of N dataset as in Table 4. However, the rules generated by DTSAR are more than the rules generated by rough set.

Overall, the rules generated by rough set are too many and is too complex to analyze compared to DTSAR and decision tree. Figure 2 shows the comparisons of number of rules generated from reduct obtain from DTSAR and other techniques.

Overall, among the three techniques, it can be summarized that DTSAR and rough set is comparable in term of the reduct and classification accuracy obtained except for WQ dataset. Meanwhile, the rough set technique outperformed DTSAR in reduct very significantly.

Table 7: Comparisons of techniques based on number of rules

Datasets	No. of rules DTSAR	Rough set	Decision tree
Credit	340	879	179
Derm	174	16	13
Derm2	186	18	18
Exactly	64	64	91
Exactly2	648	576	27
Heart	231	20	17
Letters	23	117	1
Lung	23	2417	7
M of N	64	64	69
Mushroom	163	2046	169
Led	11	6209	19
Vote	125	14	15
WQ	254	3	5

CONCLUSION

The work done in this research is based on Tabu search approach towards attribute reduction using rough set theory. The approach introduced in this study is the dynamic Tabu list and has been applied to the rough set attribute reduction. There are 13 well-known datasets being used throughout this work. In this research, reducts obtained represent the data reduction for the datasets, the dimension of the dataset is reduced into a smaller size. The results of the research show that dynamic Tabu list that is introduce to the Tabu search attribute reduction does give promising results.

The dynamic Tabu list is use to skip the aspiration criteria and to promote faster running times. However, the results obtained are comparable with earlier researches on Tabu search technique with static Tabu list. If the results are significantly better, it could be use as a hybrid with other metaheuristic techniques for future attribute reduction purposes.

ACKNOWLEDGEMENT

The project is funded by Universiti Kebangsaan Malaysia Arus Perdana Grant UKM-AP-ICT-19-2010.

APPENDIX

Appendix A (Neighbourhood generation algorithm)
Initialization step:

Generate initial solution randomly, x;
Calculate dependency degree for the initial solution x, γ (x);
Calculate the number of attribute for the initial solution x, #(x);
Set best solution, xbest←x;

```
 Set the number of attribute for xbest, #(xbest) = #(x);
 Set the dependency degree of xbest, γ (xbest) ← γ (x);
 Set C = the number of condition attributes in the current datasets
Set maximum number of neighborhood, Max_n = |C|/2;
Set maximum Tabu duration, Max_TT = 5; Tabu list length TList = 5;
    Set   Attribute index, att_indx = 0, attribute index Tabu tuner, att_indx_tt = 0
Do while (dependency degree not equal to one)
Neighborhood step
        for i = 1 to Max_n do
         Generate a neighborhood solution which is not in TList, by randomly select one
         attribute att_indx(i) from x and flip-flop it's cell, x (i)
        Calculate dependency degree for the neighborhood solution x (i), γ (x (i));
        Calculate the number of attribute for neighborhood solution x(i), #(x(i));
        If ( i >1)
                   if (γ (x (i)) > γ (x (i - 1)))
                    x←x (I);
                    γ (x)←γ (x (i));
                     #(x)← #( x(i));
                   (Update Tabu list)
                   else if ((γ (x (i)) == γ (x (i-1))) and (#(x (i)) <#(x (i-1))))
                         x← x(i);
                      γ (x)← γ (x(i));
                      #(x) ← #(x (i));
                 (Update Tabu list)
             endif
       end-for
Update the best solution
         if (γ (x) >= γ (xbest))
                         xbest ← x;
                      γ (xbest)← γ (x);
                      # (xbest) ← #(x);
end while;
     Return the best solution, xbest, γ (xbest) and #(xbest);
```

Appendix B (Dynamic tabu list algorithm)

```
Update Tabu list
    For j = 1 to Tlist,
      att_indx_tt(j) = att_indx_tt(j)-1;
     if(att_indx_tt(j))==0    then remove att_indx_tt(j) form Tlist,
  endfor
  Generate random number r between [1, 5];
 Attribute index tabu tuner,  att_indx_tt(i) = r
 Add att_indx(i) and att_indx_tt(i) to Tlist,
Tlist = Tlist,+1;
```

REFERENCES

Dash, M. and H. Liu, 2008. Dimensionality reduction. http://www.public.asu.edu/~huanliu/papers/dm07.pdf.

Glover, F. and M. Laguna, 1997. Tabu search Kluwer Academic Publishers, Boston, ISBN: 0-7923-9965-X, Pages: 408.

Han, J. and M. Kamber, 2006. Data Mining: Concepts and Techniques 2nd Edn., Morgan Kauffman Publishers, USA., ISBN 1558609016.

Hedar, A.R., J. Wang and M. Fukushima, 2006. Tabu search for attribute reduction in rough set theory. Soft Comp. A Fusion Found. Methodol. Appl., 12: 909-918.

Jensen, R. and Q. Shen, 2003. Finding rough set reducts with ant colony optimization. Proceedings of the UK Workshop on Computational Intelligence, September 1-3, 2003, Bristol, UK., pp: 15-22.

Jensen, R. and Q. Shen, 2004. Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE Trans. Knowledge Data Eng., 16: 1457-1471.

Pawlak, Z., 1991. Rough Sets: Theoretical Aspects of Reasoning About Data. 1st Edn., Kluwer Academic Publishers, UK., ISBN: 978-0792314721.

Swiniarski, R.W. and A. Skowron, 2003. Rough set methods in feature selection and recognition. Pattern Recogn. Lett., 24: 833-849.

Thangavel, K. and A. Pethalakshmi, 2009. Dimensionality reduction based on rough set theory: A review Applied Soft Comput., 9: 1-12.

Wang, J., A.R. Hedar, G. Zheng and S. Wang, 2009. Scatter search for rough set attribute reduction. Proceedings Scatter search for rough set attribute reduction. Proceedings of the International Joint Conference on Computational Sciences and Optimization-Volume 01, April 24-26, 2009, Sanya, Hainan, pp: 531-535.