ABHATH AL-YARMOUK: "*Basic Sci. & Eng.*" Vol. 21, No. 1, 2012, pp. 29-40

A Reduct Computation Approach Based on Ant Colony Optimization

Tamara Qablan*, Qasem A. Al-Radaideh and Sawsan Abu Shuqeir ***

Received on Aug. 31, 2010 **Accepted for publication on Oct. 19, 2011**

Abstract

Rough set theory provides an important concept for feature reduction called reduct. The cost of reduct set computation is highly influenced by the attribute set size of the dataset where the problem of finding reducts has been proven as an NP-hard problem. Ant Colony Optimization (ACO) as a meta-heuristic technique has been successfully applied to several combinatorial problems. This paper proposes an approach for reduct computation based on ACO methodology. The proposed approach has three main features: (1) the updated pheromone trials are directed to the nodes that visited by the ants rather than the visited edges connecting these nodes; (2) the pheromone trial values are limited between max and min trial limits; and (3) the heuristic value is evaluated dynamically during the ant search. To verify the proposed approach, several experiments are carried out on nine standard UCI datasets. . The results of experiments have showed that the proposed approach can produce a short reduct with less number of iterations in comparison to other ACO based feature reduction approaches.

Keywords: Data Mining, Rough Set Theory, Ant Colony Optimization, Reduct Computation.

Introduction

Data mining is an intelligent method to extract useful, interesting, interpretable, and novel patterns from large volume of data. To achieve the data mining goal, several techniques have been used and proposed such as Neural Network, Decision tree, Rule induction, Bayesian Belief Network, Genetic algorithm, and Rough Set Theory [1].

Data dimensionality reduction is one of the main open problems in the field of data mining. There are two main forms of attribute driven data dimensionality reduction, namely feature subset selection and feature partitioning. Several approaches were proposed and implemented for feature subset selection [2-5]. Feature subset selection, as a preprocessing step for data mining is the task of focusing the attention of an induction algorithm on subset of the given input attributes while ignoring the rest attributes.

Rough set theory was developed by Pawlak in 1982 as a mathematical tool for data analysis, to extract useful patterns from incomplete data-based information, find

* Department of Computer Science, AlHuson College, AlBalqaa Applied University, Irbid, Jordan

** Department of Computer Information Systems, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid 21163, Jordan.

dependence relationships among data, evaluate the important of attributes, discover the patterns of data, learn common decision-making rules, reduce all redundant features, and find the minimal reduct set to generate classification rules [6]. Rough set theory provides a mathematical tool that can be used to find out all possible feature subsets. Unfortunately, it has been proven that finding all possible features subsets is an NP-hard problem because there are 2N subsets for N features [7].

Reduct is the most important concept in rough set application to data mining. Reduct is denoted as the minimal subset of attributes that keep some important characteristics of the original set of attributes. For data classification, a reduct is the minimal attribute set preserving classification power of original dataset. Finding a reduct is similar to feature selection problem. There are several types of reduct, such as absolute reducts, relative reducts, optimal reducts, and approximate reducts. The problem of finding the minimal subset of features is called reduct computation [8-11]. This problem is an open research field where several researchers have proposed new techniques to find the reduct set.

Reduct computation techniques use either random search strategy or heuristics search strategy to find the significant feature from the original data set features [12] [13] [2] [14]. One of these techniques is the QuickReduct algorithm given in [15]. The algorithm searches for a minimal subset reduct without finding the others reduct subsets from the dataset. The search starts with empty subset then adds the most significant feature, which increases the rough set dependency degree, into the subset. These steps are performed until the dependency value reach to the maximum value in the dataset.

Caballero et al. [13] proposed three methods for feature selection based on some features of Rough set theory. The methods are called (MRSReduct, RSReduct, and RSRed*). The RSRed* algorithm is a heuristic algorithm based on the CORE concept and Rough set theory to calculate short reducts in acceptable times. In addition, several researches get the benefit of the Discernibility matrix structure to find optimal or approximate reducts [12] [2] [11] [16].

Ant Colony Optimization [17] is considered a new meta-heuristic approach that is successfully applied for solving many NP-hard combinatorial optimization problems such as traveling salesman problem (TSP). As for ant colony based attributes reduction, Liangjun et al. [4] introduced an approach for attribute reduction rough set theory based on ant colony optimization called (ACOAR). The approach followed the standard ACO algorithmic scheme for static combinatorial optimization problems. In the approach, at each cycle, every ant constructs a solution and then pheromone trails are updated. The termination conditions used are the maximum number of cycles or a given time limit. The iterations stop when termination condition is met.

Another ant colony based algorithm for feature selection was proposed by Yumin et al. in [18]. The algorithm adopted the mutual information based attribute significance as heuristic information. The algorithm started from the attribute core rather than from a random attribute as in Jensen and Shen algorithm [15]. The algorithm which is called RSFSACO can provide efficient solution to find a minimal subset of the attributes.

In this paper we proposed a heuristic reduct computation algorithm called Reduct Computation based on Ant Colony Optimization (RC_ACO). The algorithm provides an efficient approach to find minimal subset of features with less number of iterations in comparison to other ACO based algorithms. The proposed approach has three main features: (1) the updated pheromone trials are directed to the nodes that visited by the ants rather than the visited edges connecting these nodes; (2) the pheromone trial values are limited between max and min trial limits; and (3) the heuristic value is evaluated dynamically during the ant search. The results of experiments have showed that the proposed approach can produce a short reduct with less number of iterations in comparison to other ACO based feature reduction approaches. This leads to low computational effort in generating the reduct. In the experiment, the proposed algorithm achieved significant performance in comparison to other reduction algorithms such as ACOAR, AntRSAR, RSFSACO, and GenRSAR.

Rough Set Theory Preliminaries

Rough set theory [6] was developed in Poland in the early 1980s as a mathematical tool for knowledge discovery and data analysis, and concerns itself with the classificatory analysis of imprecise, uncertain or incomplete expressed in terms of data acquired from experience. The notion of classification is central to the approach; the ability to distinguish between objects, and consequently reason about partitions of the universe. Rough set theory has been adopted by many researches for several data mining tasks including classification [19].

Rough set theory is an extension of conventional set theory that provides approximations in decision making. A rough set is the approximation of an unclear intension by a pair of precise intensions which are known as upper and lower approximations. Equivalence classes are groups of objects which are indiscernible from each other [6].

In rough set theory, objects are perceived through the information that is available about them, that is, through their values for a predetermined set of attributes. In the case of inexact information, one has to be able to give and reason about rough classifications of objects. Basically, rough set theory deals with the approximation of sets that are difficult to describe with the available information.

The Rough set theory bears on the assumption that in order to define a set we need initially some information about elements of the universe. The information about elements are organized in a form of a decision system (DS) which is a pair of the form DS = (U, A {d}), where U = {x1, …, xn} is a nonempty finite set of objects called the universe, and A = {a1, …, ak} is a nonempty finite set of attributes. The member attributes of A are called conditional attributes and the attribute d is called decision attribute. Every attribute a ϵ A is a total function a:U→Va, where Va is the set of allowable values for the attribute a. In a practical rough set system Va is a discrete and finite set of values. An example of a decision system is presented in Table 1. The decision system S = (U, A U {d}) consists of five objects U = {x1, …, x5}, six conditional attributes A = {a1,…,a6}, and one decision attribute {d}.

Table 1: A Decision System *S* .

x_iU	a_1	a_2	a_3	a_4	a_5	a_6	**d**
x_1	2	1	2	2	1	1	**2**
x_2	1	1	1	1	2	2	**1**
x_3	3	1	2	2	1	1	**2**
x_4	1	1	2	1	3	2	**1**
x_5	2	2	2	1	3	1	**1**

The Indiscernibility Relation

For each possible subset of attributes B ⊆ A, a decision system gives rise to an equivalence relation called an Indiscernibility relation IND(B), where two objects (xi, xj) are members of the same equivalence class if and only if they cannot be discerned from each other on the basis of the set of attributes B. The formal definition of IND(B) is expressed as [8]:

$$IND(B)=\{(x_i, x_j)\ \ |U|\times|U|: a(x_i) = a(x_j)\ \ a \epsilon B\}.$$

The discernibility knowledge of the decision system is commonly recorded in a matrix called the discernibility matrix (DM). The DM is a symmetric |U|×|U| matrix with entries [cij] defined as [8]:

$$[c_{ij}] = \{a \epsilon A|\ if\ a(x_i) \neq a(x_j)\}\ \ a \epsilon\ \ A\ ;\ \text{otherwise}\}$$

The entry cij of the DM contains all the attributes that differentiate between two given objects xi and x_j.

Attribute reduction

The reduct concept of A refers to the minimal selection of attributes that preserves the Indiscernibility relation computed on the basis of the full set of attributes. Formally, a reduct of a decision system is any subset B ⊆ AU{*d*} such that IND(B) = IND(A) and IND(B – {*a*}) IND(A) for every *a* ϵ B. The decision system may have several reducts and the core set of the decision system consists of the attributes that exist in all reducts (i.e., the intersection of the reducts). The set of reducts for the decision system S presented in table 1, which are generated exhaustively, are ({a_1, a_2}, {a_4}, {a_5}). Here, the core of *S* is empty.

The Rough Set

The starting point of rough set theory is the Indiscernibility relation generated by information about objects of interest. let X ⊆ U be a target set that we wish to represent using attribute set B; this target set can be approximated using only information contained within B by constructing the B-lower (B-L) and B-Upper (B-U) approximations of X. These two approximations can be expressed respectively as [5]:

$B\text{-}L(X) = \{x \mid [x]_B \subseteq X\}$
$B\text{-}U(X) = \{x \mid [x]_B \cap X \neq \emptyset\}$
The set *BNB (X) = B-U(X) – B-L(X)* is called the *B-boundary region* of X.

Dependency of attributes

We have two types of dependency for attributes the totally dependency and the partial dependency. A set of attributes D has a total on a set of attributes C denoted by C=>D if all values of attributes from D are uniquely determined by values of attributes from C. On the other hand the partial dependency means that only some values of D are determined by values of C. Formally dependency can be defined in the following way. Let D and C be subsets of A. D depends on C in a degree K (0<=*k*<=1) denoted as $C \Rightarrow_k D$, if k=1 we say that D depends totally on C, and if *k*=0 then D does not depend on C, and if 0< *k* <1 then we say that D depends partially (in degree of *K*) on C [4] [5].

Where

$$K = \frac{|POS_C(D)|}{|U|}$$

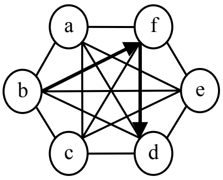
$$POS_C(D) = \underset{X \in U/D}{U} C^*(X)$$

The expression $POS_c(D)$ is called the positive region of the partition U/D with respect to C.

Ant Colony Optimization for Reduct Computation

To accomplish the task, the attribute reduction problem is represented as a complete graph *G* = (*V*, *E*) where node (*V*) represents a condition attribute and the edge (*E*) represents a path to the next attribute. Each node in the graph is assigned a heuristic information *h* and a pheromone trial *t.* An illustrative example is depicted in figure 1 which is adopted from [4].

The proposed algorithm for reduct computation (RC_ACO) follows the standard ACO algorithm that is used for static combinatorial optimization problems but with new features. This algorithm performs as follow: at each cycle, every ant constructs a solution and then the pheromone trials for each ant are updated with a value which depends on the solution constructed by the ants. The ant that constructs the best solution will have the higher amount of pheromone update. The algorithm stops iterating when the maximum number of cycles is reached or when the algorithm reaches a reduct number that is less or equal to the optimal one. The Pseudo-code of the complete algorithm (RC_ACO) is presented in figure 2.



Full Attribute Set = { a,b,c,d,e,f}
Reduced Attribute Set= {b,f,d}

Figure 1: An illustration for constructing a solution [4]

```
Algorithm: (RC_ACO)
Initializing data: $t$=0.5, $l_{gb}$=|C| and calculate $r_C(D)$
Repeat {
For each ant k in 1 … num_ant do
{ // begin for ant
  - $R_{save}$=empty
  - L= lgb, l=0;
  - Construct solution as the following
  - Select the first attribute $a_{first}$ randomly
  - R=R $\cup$ { $a_{first}$ }
  - $l$=1
- While ($r_R(D)$!=$r_C(D)$ and $l$<=$l_{gb}$) do
{ - select next attribute $a_n$ from unselected list of attributes using formula (1)
- $l$=$l$+1
- R=R $\cup$ { $a_n$}
}
- If ($l$<L) then $R_{save}$=R and L=$l$
} //end for ant
Update pheromone trial as follows:
  if (L< $l_{gb}$) then
  $R_{star}$=$R_{save}$
  $l_{gb}$=L
for every a $\epsilon$ $R_{star}$ do
  $t$(a)=$t$(a)+q/ρ* $l_{gb}$
for every a $\epsilon$ C and a $\notin$ $R_{star}$ do
  $t$(a)= ρ* $t$(a)
} Until (terminated condition is reached)
```

Figure 2: The steps of the RC_ACO algorithm.

Pheromone Trials and Heuristic Information

In this step the pheromone trial *t* at each vertex (attribute) in the graph is initialized to equal amount of pheromone. In the next step, every ant constructs its own solution; after that the pheromone trial for each vertex in the solution is updated with a value which depends on the solution constructed by the ants. Typically, the vertices of the best solution will have the higher amount of pheromone update.

Heuristic information is calculated dynamically during constructing a solution. Formally, let $\{a_1,\ldots,a_4\}$ be a set of attributes and let ant$_1$ be an ant. Suppose ant$_1$ selected $\{a_2, a_3\}$ attributes during constructing its solution, and while ant$_1$ is still constructing its solution, it selected attribute a_1 as the next attribute, the heuristic information at this moment is given as follows:

$h(a_2, a_3, a_1) = |\mathrm{POS}_{\{a2, a3, a1\}}(D)| / |U|$.

Constructing a Solution

In RC_ACO algorithm each ant starts from a randomly selected attribute, and then it selects the next attribute from those unselected attributes depending on the property value p as given in the next formula [17].

$$p_j^k(t) = \frac{t_j^\alpha(t) * h_j^\beta(t)}{\sum_{i\in u} t_i^\alpha(t) * h_i^\beta(t)}$$

Where k denotes the ant number,

t denotes the iteration numbers,

u denotes the set of attribute that are not visited yet,

t^α is the pheromone value,

h^β is the heuristic information for the selected attribute j by ant k at time t,

α and β are parameters that control the pheromone trials and the heuristic information.

The construction process is stopped when one of the following two termination conditions is met. (1) The cardinality of the solution is less than or equal the cardinality obtained so far. (2) The maximum number of cycles is achieved. The first condition means that a better solution has been constructed, so the process can be terminated. While the second condition means that there is no better solution can be constructed.

Pheromone Update

After each ant has constructed a solution, pheromone trails are updated on each vertex. The ant that constructs the best solution (*bs*) will have the higher amount of pheromone update. In RC_ACO, pheromone is updated according to the following formula: [4]

$$t_j(t+1) = \rho t_j(t) + {}^{q}\!/_{l_{bs}}$$

Where t_j is the amount of pheromone on vertex j at time t,

$t_j(t+1)$ is the amount of pheromone on vertex *j* at next iteration,
$\rho \in [0,1]$ is a decay constant used to simulate the evaporation of pheromone,
q is a given constant, and
l_{bs} is the number of attribute in the best solution.

While for other vertices, the pheromone trails are updated using the following formula: [4]

$$t_{j\in u}(t+1) = \rho t_{j\in u}(t)$$

Where *u* is the set of vertices that do not belong to the best solution. Pheromone trials are updated according to *min_max* technique [20]. To avoid search stagnation situation if the relative differences of pheromone trails are too extreme:

If $t_j > t_{max}$ then $t_j = t_{max}$ and
If $t_j < t_{min}$ then $t_j = t_{min}$

Experiments and Evaluation

For the experimental purposes, the parameters of RC_ACO algorithm are set to the following values where these parameters were determined based on some preliminary runs of the algorithm:

$\alpha = 1.0$, $\beta=0.1$, $\rho=0.9$, $q=0.1$, $t=0.5$, $t_{max}=1.0$, $t_{min} = 0.001$, number of ants = 10, and the maximum number of cycles = 50.

The algorithm is tested using 9 standard UCI datasets [21]. We compared the first five dataset with two algorithms, AntRSAR [15] and ACOAR [4], where the rest dataset are compared with RSFSACO algorithm [18]. For experimental purposes, several approaches in this area set the number of runs to 20 runs. In the case of the proposed approach, the program was executed 20 times for each data set. This number was adopted in order to compare the performance of the proposed approach with other approaches. The results for the first five dataset are given in table1. For each algorithm, the minimum reduct obtained at every run is given. The upper case number in parentheses represents the number of runs that were required to obtain this reduct.

Table 2: Comparison of AntRSAR, ACOAR and RC_ACO.

Dataset	#Attributes	AntRSAR	ACOAR	RC_ACO
Vote	16	8	8	8
Credit	20	$8^{(12)}9(4)10^{(4)}$	$8^{(16)}9^{(4)}$	8
Mushroom	22	4	4	4
Drem	34	$6^{(17)}7^{(3)}$	6	6
Heart	13	$6^{(18)}7^{(2)}$	6	6

In the RC_ACO algorithm, for each run there are 50 iterations which mean the maximum number of cycles required for each ant to find the solution. In practice, ants may find the solution before completing all iterations. For example, for the vote dataset, the RC_ACO found the solution in 8.20 iterations in average over the 20 runs of the algorithm.

The information presented in table 2 indicates for example that for the *Drem* dataset which has 34 attributes, the AntRSAR algorithm reduced the number of attributes from 34 attributes to 6 attributes in 17 runs and 7 attributes in 3 runs with an average 57 iterations whereas the RC_ACO algorithm reduced the number of attributes to 6 attributes in all the 20 runs, with an average number of iterations of 23.95.

From the results in table 2, we notice, in addition that, RC_ACO achieved better results in comparison to AntRSAR using the Credit and Drem datasets in term of the size of the generated reduct, and it achieved the same results for the rest datasets. On the other hand RC_ACO achieved similar results in comparison with ACOAR except for the credit dataset. Moreover, the experimental results in table 3 show that the RC_ACO algorithm achieved better results in comparison to RSFSACO using the Mushroom and Vote datasets and it achieved the same results for the rest datasets in term of the reduct size.

Table 3: Comparison of RSFSACO and RC_ACO

Dataset	#Attributes	RSFSACO	RC_ACO
Breast-cancer	10	4	4
Chess-king	37	29	29
Zoo	17	5	5
Vote	16	9	8
Mushroom	22	$4^{(13)}5^{(7)}$	4

The average number of iterations required to obtain the minimal reducts by RC_ACO in comparison to the ACOAR approach over the 20 runs are presented in Table 4, where the other datasets are represented in table 5.

Table 4: Average Number of iterations to obtain the minimal reducts by RC_ACO comparing with ACOAR.

Dataset	#Attributes	ACOAR	RC_ACO
Vote	16	33	8.2
Credit	20	72	13.7
Mushroom	22	34	14.4
Drem	34	57	23.95
Heart	13	30	5.95

Table 5: Average number of iterations to obtain the minimal reducts by RC_ACO.

Dataset	RC_ACO
Breast-cancer	4
Chess-king	5.35
zoo	4.2
vote	8.2
mushroom	14.4

Conclusion

In this paper we proposed a new algorithm called RC_ACO for reduct computation. The algorithm is an ant colony optimization (ACO) based approach. In the proposed algorithm, the pheromone update process is performed on the nodes that are visited by the ants rather than on the edges connecting these nodes, where these pheromone trials are limited between max and min limits. The efficiency of this approach comes from evaluating the heuristic value during the ant search. As a conclusion, the proposed algorithm has outperformed other ACO based feature reduction approaches. The results demonstrate that the proposed approach can provide competitive solutions in generating short reduct efficiently using less number of iterations.

طريقة لايجاد المختزل باستخدام خوارزمية مستعمرة النمل

تمارا القبلان وقاسم احمد الردايده وسوسن أبوشقير

ملخص

توفر نظرية المجموعات التقريبية مفهوما هاما لاختزال عدد الميزات يدعى "المختزل" (Reduct). تكلفة عملية حساب او ايجاد المختزل تتأثر بشكل كبير بعدد الميزات حيث تم اثبات ان عملية حساب المختزل تعتبر من المسائل المعقدة. تعتبر خوارزمية مستعمرة النمل الأمثل (ACO) أسلوب استرشادي مناسب لحل هذا النوع من المسائل حيث تم تطبيقها بنجاح على عدة مسائل اندماجي توافيقية. هذه الورقة تقترح طريقة جديد لحساب المختزل تعتمد على خوارزميات مستعمرة النمل (ACO). للطريقة المقترحة ثلاثة ميزات: (1) تعديل قيمة الفيرومون توضع على العقدة التي تزورها النمله وليس على الحواف كما في الطرق الاخرى. (2) قيم الفيرومون محدودة بين قيم دنيا وقيم فصوى. (3) القيمة الاسترشادية تقيم بشكل مرن اثناء عملية البحث للنمل. للتحقق من كفاءة الطريقة المقترحة تم تنفيذ

العديد من التجارب باستخدام تسعة مجموعات بيانات قياسية من موقع UCI وقد أظهرت نتائج التجارب أن الطريقة المقترحة يمكن أن تنتج مختزل قصيربعدد اقل من التكرارات في الخوارزمية كما هو موجود في الخوارزميات المقترحة الاخرى حيث تفوقت الطريقة المقترحة على غيرها من الطرق المعتمدة على خوارزميات مستعمرة النمل المستخدمة في ايجاد المختزل.

References

[1] Han J. and Kambar M., 'Data Mining: Concepts and techniques', 2nd edition, Morgan Kaufmann Publishers, ISBN 1-55860-901-6. 2006.

[2] Pei X. and Wang Y., “An Approximate Approach to Attribute Reduction”, *International Journal of Information Technology*. 2006.

[3] Liu F., and Lu S., “A Feature Selection Algorithm Based on Discernibility Matrix”, Springer-Verlag Berlin Heidelberg. 2007.

[4] Liangjun K. , Zuren F., and Zhigang R., “An efficient ant colony optimization approach to attribute reduction in rough set theory”, *Pattern Recognition Letters*, Vol 29(2008), 1351–1357..

[5] Parthaláin N. and Shen Q., “Exploring the boundary region of tolerance rough sets for feature selection, *Pattern Recognition,* 42(2009), 655 – 667.

[6] Pawlak Z, *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht. ISBN 0-7923-1472-7, (1991).

[7] Skowron A. and Rauszer C., “The discernibility matrices and functions in information systems”, in Słowiński R. (ed.), Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory. Kluwer, Dortrecht. pp. 331-362. (1992).

[8] Yiyu Y., Yan Z., and Jue W., “On Reduct Construction Algorithms, Rough Sets and Knowledge Technology”, *In Proceedings of the First International Conference of Rough Set and Knowledge Technology (RSKT 2006)*, LNAI 4062, pp. 297-304. 2006.

[9] Lin T.Y. and Yin P., “Heuristically Fast Finding of the Shortest Reducts”, In Proceedings of Rough Sets and Current Trends in Computing. LNAI 3066, pp. 465–470. 2004.

[10] Keyun H., Lili D., Yuchang L., and Chunyi S., “Sampling for Approximate Reduct in very Large Datasets”. Computer Science Department, Tsinghua University, Beijing 100084, P.R.China. (2000).

[11] Al-Radaideh Q. A., Md Nasier S., Selamat M.H. and Ibrahim H., “Approximate Reduct Computation by Rough Sets based Attribute Weighting”, *In Proceedings of 2005 International IEEE Conference on Granular Computing*, 2(2005), 383-386.

[12] Wang R., Miao D., Hu G., “Discernibility Matrix Based Algorithm for Reduction of Attributes”, *In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (WI-IATW ‘06), (2006), 477-480.

[13] Caballero Y., Alvarez D., Bello R., and Garcia M., “Feature Selection Algorithms using Rough Set Theory”, *In Proceedings of the Seventh International*

Conference on Intelligent Systems Design and Applications, IEEE, (2007), 407-411.

[14] Jensen R., Shen Q., and Tuson A., "Finding Rough Set Reducts with SAT", *In Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing RSFDGrC,* LNAI 3641, (2005) 194-203.

[15] Jensen R., Shen, Q., "Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches". *IEEE Transaction on Knowledge and Data Engineering*. 16(12) (2004) 1457–1471.

[16] Tan S., Cheng X. and Xu H., "An Efficient global optimization approach for Rough Set Based Discernibility Reduction", *International Journal of Innovative Computing, Information and Control*, (2006) 725-736.

[17] Dorigo M., Maniezzo, V., Colorni, A., "Ant system: Optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man, and Cybernetics*. 26 (1996), 29–41.

[18] Yumin C., Duoqian M., and Ruizhi W., "A rough set approach to feature selection based on ant colony optimization", *Pattern Recognition Letters*. 31(2010), 226-233.

[19] Bazan J., Nguyen H.S., Nguyen S.H., Synak P., Wróblewski J., "Rough Set Algorithms in Classification Problem". In: L. Polkowski, S. Tsumoto and T.Y. Lin (eds.): Rough Set Methods and Applications. Physica-Verlag, Heidelberg, New York, (2000) 49 – 88.

[20] Stützle T. and Hoos H., "MAX-MIN ant system". *Future Generation Computer Systems.* 16(8) (2000), 889–914.

[21] Blake C., and Merz C., "UCI repository of machine learning database", University of California, Irvine, Department of Information and Computer Sciences. 1998.