

Using ACO and Rough Set Theory to Feature Selection.

RAFAEL BELLO
PÉREZ
Computer Science
Department.
Universidad Central de
Las Villas
Santa Clara.
Cuba.

ANN NOWE,
PETER VRANČX
CoMo Lab, Computer
Science Dept
Vrije Universiteit
Brussel.
Brussel
Belgium

YUDEL GÓMEZ D
Computer Science
Department.
Universidad Central de
Las Villas
Santa Clara.
Cuba.

YAILÉ CABALLERO
Informatic Department
Universidad de
Camaguey
Camaguey
Cuba

Abstract: - In this paper we propose a model to feature selection based on ant colony and rough set theory (RST). The objective is to find the reducts. RST offers the heuristic function to measure the quality of one feature subset. We have studied three variants of ant's algorithms and the influence of the parameters on the performance both in terms of quality of the results and the number of reducts found. Experimental results show this hybrid approach shows interesting advantages when compared with other heuristic methods.

Key-Words: - Feature selection, reducts, ant model, rough set theory.

1 Introduction

Feature selection (FS) is useful to reduce the problem dimensionality problem; but also in improving the speed of data manipulation, and increasing the performance of processes by reducing the influence of noise features [8], [3], and [16].

FS methods search through of a set of nf features and try to find the best subset among the competing $2^{nf}-1$ candidate subsets according to some evaluation measure. All FS methods contain two important components: an Evaluation function (EF) used to evaluate a candidate feature subset and a Generation procedure (GP) to generate candidate sub-sets.

The EF tries to measure the discriminating ability of a subset of features, in order to distinguish the different class labels. It measures the goodness of a subset produced by some generation procedure. An optimal subset is always relative to a certain EF. The GP is a search algorithm to search through the feature space. Search strategies are important because the FS process may be time consuming and an exhaustive search for the "optimal" subset is impractical for even moderate sized problems. Examples of GP are blind and heuristic search, probabilistic methods and hybrid algorithms.

The approach presented in this paper is based on the Rough Sets Theory (RST) and Ant Colony Optimization (ACO). The former is used to build the EF and the latter to implement the GP using a filter approach based on forward selection.

RST was proposed by Z. Pawlak [12], it provides a technique for gaining insights into the data properties and it does not need any external information [18]. An important issue in the RST is FS based on the reduct concept. A reduct is a minimal set of features that preserves the partitioning of universe and hence the ability to perform classifications [11]. The use of reducts in the selection and reduction of attributes has been studied by various authors, among them are [11].

However, this beneficial alternative is limited because of the computational complexity of calculating reducts. The problem of finding a globally minimal reduct for a given information system is NP-hard. Good approximation methods for calculating reducts have been developed using heuristic search, among them [17], [1] and [4]. Another heuristic approach is ACO, which has been used in data mining tasks. In particular, R. Jensen and Q. Shen [7] have proposed a method which combines ACO and RS to find reducts with promising results.

Our research addresses the same issue. Our objective is to experimentally study the performance of three variants of the ACO applied to FS, and to analyze the parameter settings of the ACO. The setting of parameters is crucial for the performance of the algorithm [5] and [6].

In the following sections we present the study. First, we give the basic elements of ACO and RST in paragraph 2. After that the algorithms are

described in paragraph 3, and experiments and its results are showed in paragraphs 4.

2 About ACO and Rough set theory

ACO [5] and [6] was introduced by M. Dorigo, it is a metaheuristic used to guide other heuristics in order to obtain better solutions than those that are generated by local optimality methods. A colony of artificial ants cooperates to look for good solutions to discrete problems. Artificial ants are simple agents that incrementally build a solution by adding components to a partial solution under construction. Similar to real ants, artificial leave a pheromone trail while exploring the search space. This pheromone attracts other ants, and as such allows the ants to indirectly communicate with each other.

Ant System (AS) is the first ACO algorithm. It uses a random proportional rule to select the next component. This rule takes into account the pheromone intensity between the last and the next component and a heuristic value about the effect of adding the next component. Two parameters, α and β , determine the balance between the importance of the trail intensity and the heuristic value.

Ant Colony System (ACS) is an advanced successor algorithm of AS. ACS uses an extra pseudorandom proportional rule to select the next component which parameters Alpha denotes the relative weight of trail intensity, Beta indicates the relative weight of heuristic value, and a new parameter q_0 (tunable parameter of transition rule) is introduced. Tuning the parameter q_0 allows modulation of the degree of exploration; with probability q_0 ants make the best possible move according to the learned pheromone trail and the heuristic information, while with probability $(1-q_0)$ it explores other solutions.

The RST is the second component in our hybrid approach. In RST, a Decision System is any information system of the form $SD=(U, A \cup \{d\})$ where U is a non-empty finite set of N objects called the Universe, A is a non-empty finite set of nf features and $d \notin A$ is the decision attribute. The basic concepts of RST are the lower and upper approximations of $X \subseteq U$; they are introduced with reference to an indiscernible relation $IND(B)$, where objects x and y belong to $IND(B)$ if and only if x and y are indiscernible from each other by features in B . Let $B \subseteq A$ and $X \subseteq U$. It can be proved that B defines an equivalence relation. The B -lower and B -upper approximations for all subset $X \subseteq U$ are defined by $B_*X = \{x : [x]_B \subseteq X\}$ and $B^*X = \{x :$

$[x]_B \cap X \neq \emptyset\}$ respectively, where $[x]_B$ denotes the equivalence class of x .

RST offers several measures about the Decision System. One of them is the quality of the approximation of classification (expression 1). It expresses the percentage of objects which can be correctly classified into the target classes $Y = \{Y_1, \dots, Y_n\}$ employing the set of features in B .

$$\gamma_B(Y) = \frac{\sum_{i=1}^n |B_*Y_i|}{|U|} \tag{1}$$

A reduct is a minimal set of attributes $B \subseteq A$ such that $IND(B) = IND(A)$. Therefore, a reduct is a minimal set of features such that the X -indiscernibility relation is con-served. That is $\gamma_A(Y) = \gamma_B(Y)$. Bell [1] shows that the computational cost of finding a reduct in the information system that is bounded above by nf^2N^2 , while the complexity in time of finding all the reducts is $O(2^{nf}J)$, where J is the computational cost required to find a reduct.

3 A hybrid model to feature selection

FS is an example of a difficult discrete problem. We represent this problem as a graph problem in the following way. Let $A = \{a_1, a_2, a_{nf}\}$ be a set of features. You can view this set as a network in which nodes represent features, and all nodes are connected by bidirectional links. Pheromone values τ_i are associated to nodes a_i . This is a difference between our approach and the approach proposed in [7] in which pheromone is associated with the links, which is common in ACO. The amount of pheromone on link a_i-a_j is function of the dependence degree of attribute a_j on feature a_i . In our approach we associate pheromone to nodes. The amount of pheromone is a function of the dependency of the feature associated to that node, to all other features. As a result the pheromone we associate to node a_i represents the absolute contribution of that feature to a reduct, rather than the contribution of a_i , given the fact that a_j was the previous feature included in the reduct.

The solution consists of reducts which have to be built step by step by the ants. Initially the ants are distributed over the nodes of the graph, and each has an empty subset which has to become a candidate reduct. The behavior of one single ant can be described as follows. In the first step, the ant is assigned to one of the nodes, in the next step it will move to some other node in the network. In doing

so, the ant performs a forward selection in which it expands its subset step-by-step by adding new features. To select the next node to visit the ant looks for all features that are not yet included in the subset and selects the next according to the rule of ACO. On the one hand it is attracted by the pheromone the other ants have already deposited in the graph and on the other hand by the heuristic function. The heuristic we use here is given by expression (1). This expression is also used to determine if the subset built so far is a reduct. Over time the quality of the subsets build by the ants will improve, which is supported by the monotonicity property of classical RST; these converge to near optimal reducts.

We have studied three variants of algorithm to FS based on ACO. The first is based on the Ant System (AS-RST-FS), the second is a modified Ant System (AS*-RST-FS) and the last is based on Ant Colony System (ACS-RST-FS). Following, we state several common aspects to the three variants:

a) Initial distribution of ants.

We have formulated the following rules to initial distribution of ants in each cycle of the algorithms:

- i) *If $m < nf$, then random initial distribution of ants;*
- ii) *If $m = nf$, then one ant is associated with each feature;*
- iii) *If $m > nf$, the first nf ants are distributed according to (ii) and the rest according to (i).*

b) Random proportional rule.

This rule is given by expression (2). It uses the quality of the approximation of the classification as heuristic function (η) to evaluate a subset B ($\eta(B) = \gamma_B(Y)$):

$$p_k(B^k, a_j) = \begin{cases} 0 & \text{if } a_j \in B^k \\ \frac{[\tau_j] * [\gamma_{B^k \cup \{a_j\}}(Y)]}{\sum_{a_j \in A - B^k} [\tau_j] * [\gamma_{B^k \cup \{a_j\}}(Y)]} & \text{if } a_j \in A - B^k \end{cases} \quad (2)$$

c) Initialization of trail.

Each attribute a_i has associated a pheromone value τ_i . We consider three alternatives to calculate initial pheromone values: (i) random values to τ_i ; (ii) $\tau_i(0) = 1/nf$; and (iii) $\tau_i(0) = \sigma_{A,D}(a_i)$, where $\sigma_{A,D}(a_i)$ is the relevance of feature a_i according to rough set theory.

d) Ant Stopping Criteria (ASC).

The ASC states when an ant finishes its activity in a cycle. Each ant k keeps adding one-feature-at-a-time to its current partial set B_k until $\gamma_B(Y) = \gamma_A(Y)$.

e) Process Stopping Criteria (PSC).

The process of finding sets B is a sequence of cycles ($NC = 1, 2 \dots$). In each cycle all ants build

their set B. The process is stopped (PSC = true) when the following condition is met: $NC > NC_{max}$.

f) Number of ants.
We tested several values for the parameter number of ants, $m = \{3, 9, 12, 15, 24, 33, 42, 48, 54 \text{ and } 60\}$. From this experimental study we infer the following rules of thumb to determine the number of ants as a function of the quantity of features ($m = f(nf)$), where $Round[x]$ denotes the closest integer to x :

- R1: *If $nf \leq 19$ then $m = nf$;*
- R2: *If $20 \leq nf \leq 49$ then*
*[If $0.666 * nf \leq 24$*
*then $m = 24$ else $m = Round[0.66 * nf]$]*
- R3: *If $nf \geq 50$ then*
*[If $0.5 * nf \leq 33$*
*then $m = 33$, else $m = Round[0.5 * nf]$]*

Following we will describe the algorithms. The first algorithm is based on AS.

Algorithm AS-RST-FS:

Initialization: $PSC \leftarrow False$ and $\tau_i(0) \leftarrow$ initial value $i = 1, \dots, nf$
Repeat

Initial distribution of ants (Each ant k is associated with one attribute a_i and $B_k \leftarrow \{a_i\}$)

Repeat

for $k = 1$ to m do if $ASC_k = False$ then Select the new feature a_i^* to add to B_k according to expression (2).

Update ASC_k .

Until ASC_k is True for all ants.

After all the ants have constructed their reduct the pheromones are updated:

a) $\tau_i \leftarrow (1 - \rho) * \tau_i$ for $i = 1, \dots, nf$ (evaporation process)

b) For $k = 1, \dots, m$ do For all $a_i \in B_k$ do $\tau_i \leftarrow \tau_i + \gamma_{B_k}(Y)$

Update PSC

Until PSC is True.

The second Algorithm, AS*-RST-FS, introduces some modifications. These affect the way in which the pheromone values are updated in three aspects: (i) After each ant has constructed her reduct τ_i is updated for all a_i in the reduct, (ii) After all ants have constructed their reducts τ_i is updated for all a_i in the best reducts (elitist strategy), and (iii) After all ants have constructed their reducts we applied an indirect evaporation for all τ_i by normalizing their values (indirect evaporation).

$$\tau_i \leftarrow \frac{\tau_i}{\sum_{j=1}^{nf} \tau_j} \quad (3)$$

The last algorithm, ACS-RST-FS, is based on the ACS model. In this case, the rule for selecting a new feature and the way to update pheromone are different.

Algorithm ACS-RST-FS:

Initialization of PSC to False and trail intensities ($\tau_i(0)$)

Repeat

Initial distribution of ants (Each ant k is associated with one attribute a_i and $B_k \leftarrow \{a_i\}$)

Repeat

For all ant k If $ASC_k = \text{False}$ then Select the new feature a_i^* to add to B_k ($B_k \leftarrow B_k \cup \{a_i^*\}$)

According to the pseudo-random proportional rule:

$$i = \begin{cases} \max(\tau_i * (\gamma_{B^* \cup \{a_i\}}(Y))^\beta) & \text{if } q \leq q_0 \\ I & \text{in other case} \end{cases} \quad (4)$$

where I is selected using the expression (2).

$\tau_i \leftarrow (1-\xi) * \tau_i + \xi * \tau_i(0)$, for feature a_i^* (local pheromone update rule)

Update ASC_k

Until ASC_k is True for all ants.

After all the ants have finished:

Select the best B_k and for all $a_i \in$

B_k do

$\tau_i \leftarrow (1-\rho) * \tau_i + \rho * \gamma_{B_k}(Y)$

Update PSC

Until PSC is True.

Considering the computation effort for finding either lower or upper approximations is $O(nf * N^2)$ according to [1] and [4], we can estimate the complexity of these algorithms as $O(NC * m * nf^2 * N^2)$, where NC is the number of cycles.

4 Experimental results

The aims of this experimental study are to evaluate the three previous algorithms, and to determine the rules of thumb for the setting of the parameters.

When we evaluated the initialization trail schemes previously presented alternatives (ii) and (iii) produce too low initial values. Using alternative (i) the algorithm generates slightly more reducts; this result is coherent with the fact that too low values for initial trails reduce the search space [6]. For this reason, further experiments were done using random [0, 1].

The algorithm performance was compared according to the resulting reduct quantity (RQ) and the average length of the resulting reduct set (RL);

the length of a reduct is defined as the number of features in the reduct. We applied these algorithms to FS in database from UCI Repository [2]. For each database, all tests have been carried out for $N_{\text{max}}=150$ cycles and were averaged over ten trials. We evaluated several values for parameters Alpha (α), Beta (β) and q_0 while ρ and ξ were held constant; that is $\rho=0.5$ and $\xi=0.1$, and $\alpha=\{0.1, 0.5, 1, 2, 5\}$, $\beta=\{0.1, 1, 2, 5\}$ and $q_0=\{0.3, 0.6, 0.9\}$.

	Breast Cancer	Exactly	Exactly 2	Heart	LED	Lung Cancer(1)	Lung Cancer(2)
nf	9	13	13	13	24	56	56
m	9	13	13	13	24	33	33
1	4(4) RQ=11 RL=4.6	6(1) RQ=1 RL=6	10(1) RQ=1 RL=10	6(2) RQ=2 RL=6	5(1) RQ=8 RL=19.4	4(33) RQ=138 RL=5	4(1) RQ=206 RL=6.8
2	4(2) RQ=5 RL=5	6(1) RQ=1 RL=6	10(1) RQ=1 RL=10	6(1) RQ=4 RL=7.5	5(1) RQ=4 RL=16.5	4(4) RQ=33 RL=6.9	5(2) RQ=60 RL=7.1
3	4(1) RQ=4 RL=4.8	6(1) RQ=1 RL=6	10(1) RQ=1 RL=10	7(3) RQ=4 RL=7.5	5(1) RQ=1 RL=5	4(7) RQ=29 RL=5.5	4(1) RQ=61 RL=6.1
4	4(2) RQ=2 RL=4	6(1) RQ=1 RL=6	10(1) RQ=1 RL=10	7(1) RQ=1 RL=7	5(1) RQ=1 RL=5	4(1) RQ=18 RL=5.8	6(2) RQ=42 RL=7.9
5	4(4) RQ=9 RL=4.6	6(1) RQ=1 RL=6	10(1) RQ=1 RL=10	6(1) RQ=10 RL=7.3	5(1) RQ=2 RL=13.5	4(69) RQ=202 RL=4.7	4(8) RQ=337 RL=6
6	4(4) RQ=17 RL=4.9	6(1) RQ=1 RL=6	10(1) RQ=1 RL=10	6(1) RQ=11 RL=7.7	5(1) RQ=6 RL=18	4(46) RQ=399 RL=6	4(8) RQ=643 RL=7.15

Table 1: Comparison between algorithms using data base from UCI Repository. Where **1** denotes AS-RST-FS with $\alpha=1$, $\beta=5$; **2** AS-RST-FS with $\alpha=5$, $\beta=1$; **3** AS*-RST-FS with $\alpha=1$, $\beta=5$; **4** AS*-RST-FS with $\alpha=5$, $\beta=1$; **5** ACS-RST-FS with $\beta=5$, $q_0=0.9$; **6** ACS-RST-FS with $\beta=1$, $q_0=0.3$.

Our research shows the relation between parameters α and β has the strongest impact on the quantity of generated subsets by AS-RST-FS and AS*-RST-FS algorithms. We think this is the key relation in these algorithms. Results show that regardless of the method to calculate $\tau_i(0)$, values {1, 5} are key values for parameters α and β ; the combination ($\alpha=1$, $\beta=5$) found more than twice the number of reducts than the combination ($\alpha=5$, $\beta=1$) in the case of AS-RST-FS, and about 1.5 time in the case of AS*-RST-FS algorithm. Another interesting result is the combination ($\alpha=1$, $\beta=5$) that generates shorter reducts than ($\alpha=5$, $\beta=1$).

Similarly, there is a key relation is between β and q_0 in the ACS-RST-FS algorithm, in this case the key values are {0.3, 0.9} for q_0 and {1, 5} for β , we use $\alpha=1$. The combination ($\beta=1$, $q_0=0.3$) generates about 3 times more reducts than combination ($\beta=5$, $q_0=0.9$), regardless of the quantity of the ants and the

number of cycles. But, in this case the combination ($\beta=5, q_0=0.9$) obtained shorter reducts than other combinations. Also, the effect of increasing the number of ants needed is bigger in the case of combinations ($\alpha=1, \beta=5$) and ($\beta=1, q_0=0.3$) than other combinations.

In Table 1, we present the algorithm performances. Where, $n1(n2)$ denotes $n2$ shortest reducts with $n1$ features; RQ is the resulting quantity of reducts and RL is the average length of the resulting reducts. We only present information about the two shortest reducts. The database Lung Cancer (1) was discretized by using the method EntropyScaler from Rosetta [10], others containing real valued features have been discretized as [7]. The algorithm based on ACS obtains the best results. The average number of cycles to find the shortest reduct was 15, 21 and 27 for ACS-RST-FS, AS-RST-FS and AS*-RST-FS respectively.

In their earlier work [7], R. Jensen and Q. Shen have proposed the algorithm AntRSAR to find reducts, they showed promising results when comparing the results of this method with other methods to build reducts such as RSAR (using Quickreduct), [15], and GenRSAR (genetic algorithm based), and suggested more experimentation and further investigation to study the influence of the parameters. In Table 2, we present a comparison between ACS-RST-FS, AntRSAR, RSAR and GenRSAR algorithms (data for the last three are taken from [7]) with respect to the length of minimal reduct found by algorithms. These results show that the algorithms based on ACO obtain the best results. In the case of ACS-RST-FS, 21 cycles were enough to obtain these results while the results reported back in [7] were obtained in 250 cycles.

Algorithm	Heart	Exactly	Exactly 2	LED	Lung Cancer
nf	13	13	13	24	56
RSAR	7	9	13	12	4
GenRSAR	6	6	10	6	6
AntRSAR	6	6	10	5	4
ACS-RST-FS	6	6	10	5	4

Table 2: Comparison with other algorithms.

Moreover, we have compared the algorithms performance with the performance of other methods to select features. These are LEX, UMDAf3 and RSReduct algorithms.

LEX algorithm was proposed by Santiesteban [14], it reports the best results among others in the pattern recognition area.

We have defined the method UMDAf3 for calculating reducts starting from the integration of the adaptability function $f3$ reported in [17] and the UMDA method (Univariate Marginal Distribution Algorithm) [9].

Moreover, we have formulated a greedy algorithm called RSReduct that starts with an empty set of attributes and builds reducts by selecting features according to the heuristic function $f(ai)=H(ai)+C(ai)$, where terms $H(ai)$ and $C(ai)$ were proposed by Piñeiro in [13].

In Table 3 we present a comparison of the performance of these three algorithms and ACS-RST-FS using the data base Lung Cancer discretized using the method EntropyScaler from Rosetta [10]. We selected the Lung Cancer database because this has the largest quantity of features. Again, the algorithm based on ACO has the best results.

Variable \Rightarrow Algorithms \Downarrow	Length of shortest reduct	QR	Average LR
LEX	4(1)5(57)6(92)	150	5.6
UMDAf3	4(1)5(17)	18	4.9
RSReduct	4(1)	1	4
ACS-RST-FS	4(69)5(122)	202	4.7

Table 3: Comparison of ACS-RST-FS and other algorithms.

4 Conclusion

We have presented a study about the possibilities of applying a hybrid model (ant model + rough sets) to feature selection. This research includes, as an important issue, the setting of ant parameters.

We have studied three alternatives of algorithms, the first one based on Ant system model (AS-RST-FS), the second one on Ant system plus an elitist strategy (AS*-RST-FS) and the third one based on Ant Colony System (ACS-RST-FS). We can conclude ACS-RST-FS performs better than others.

Also, we have determined that the values of parameters α and β in AS-RST-FS and AS*-RST-FS algorithms and parameters β and q_0 in ACS-RST-FS algorithm are crucial; they have the most important effect on the algorithm performances.

The comparison of the hybrid model with other methods to calculate reducts in RST shows that ant model based algorithms yield good results. This ratifies results obtained by R. Jensen and Q. Shen in [7]. In this research, we have obtained additional results about the set-ting of the parameters of the ant model.

Acknowledgments

The authors would like to thank VLIR (Vlaamse InterUniversitaire Raad, Flemish Interuniversity Council, Belgium) for supporting this work under the IUC Program VLIR-UCLV.

References:

- [1] D. Bell and J. Guan. Computational methods for rough classification and discovery. *Journal of ASIS* 49, 5, pp. 403-414. 1998.
- [2] C.L. Blake and C.J. Merz. *UCI repository of machine learning databases*, 1998. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [3] M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial intelligence* 151, pp. 155-176. 2003.
- [4] J.S. Deogun, S.K. Choubey and H. Sever. Feature selection and effective classifiers. *Journal of ASIS* 49, 5, pp. 423-434. 1998.
- [5] M. Dorigo, G. D. Caro and L. Gambardella. Ant algorithms for Discrete optimization. *Artificial Life* 5(2), pp. 137-172. 1999.
- [6] M. Dorigo and T. Stutzle. *Ant Colony Optimization*. MIT Press. 2004.
- [7] R. Jensen and Q. Shen. Finding rough set reducts with ant colony optimization. *Proceedings of UK Workshop on Computational Intelligence*, pp. 15-22. 2003.
- [8] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition* 33, pp. 25-41. 2000.
- [9] H. Mühlenbein. The equation for the response to selection and its use for prediction. *Evolutionary Computation* 5(3), pp. 303-346, 1998.
- [10] A. Ohm. ROSETTA Technical Reference Manual. Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, 2001.
- [11] S.K. Pal and A. Skowron, (Eds).. *Rough Fuzzy Hybridization: a new trend in decision-making*. Springer-Verlag, 1999.
- [12] Z. Pawlak. Rough sets. *International Journal of Information & Computer Sciences* 11, 341-356, 1982.
- [13] P. Piñero, L. Arco, M.M. García, Y. Caballero. Two New Metrics for Feature Selection in Pattern Recognition. *Lectures Notes in computer Science (LNCS 2905)*, pp. 488-497. Springer, Verlag, Berlin Heidelberg. New York. ISSN 0302-9743. ISBN 3-540-20590-X. 2003.
- [14] Y. Santiesteban and A. Pons, LEX: LEX: a new algorithm to calculate typical testors. *Mathematics Sciences Journal*, Vol. 21, No. 1, 2003.
- [15] Q. Shen and A. Chouchoulas. Combining rough sets and data-driven fuzzy learning. *Pattern Recognition*, 32(12):2073, 1999.
- [16] J. Stefanowski. An experimental evaluation of improving rule based classifiers with two approaches that change representations of learning examples. *Engineering Applications of Artificial Intelligence* 17, pp. 439-445. 2004.
- [17] J. Wroblewski. Finding minimal reducts using genetic algorithms. In Wang, P.P. (Ed). *Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences*, North Carolina, USA, p. 679, pp. 186-189. 1995.
- [18] Z.B. Xu, J.Y. Liang, C.Y. Dang, K.S. Chin. Inclusion degree: a perspective on measures for rough set data analysis. *Information Sciences* 141, pp. 227-236. 2002.