# A STUDY ON ROUGH SET THEORY BASED DYNAMIC REDUCT FOR CLASSIFICATION SYSTEM OPTIMIZATION

Shampa Sengupta[1], Asit Kumar Das[2]

[1]Department of Information Technology, MCKV Institute of Engineering, Liluah, Howrah – 711 204, West Bengal, India
[2]Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah – 711 103, West Bengal, India

## ABSTRACT

*In the present day huge amount of data is generated in every minute and transferred frequently. Although the data is sometimes static but most commonly it is dynamic and transactional. New data that is being generated is getting constantly added to the old/existing data. To discover the knowledge from this incremental data, one approach is to run the algorithm repeatedly for the modified data sets which is time consuming. Again to analyze the datasets properly, construction of efficient classifier model is necessary. The objective of developing such a classifier is to classify unlabeled dataset into appropriate classes. The paper proposes a dimension reduction algorithm that can be applied in dynamic environment for generation of reduced attribute set as dynamic reduct, and an optimization algorithm which uses the reduct and build up the corresponding classification system. The method analyzes the new dataset, when it becomes available, and modifies the reduct accordingly to fit the entire dataset and from the entire data set, interesting optimal classification rule sets are generated. The concepts of discernibility relation, attribute dependency and attribute significance of Rough Set Theory are integrated for the generation of dynamic reduct set, and optimal classification rules are selected using PSO method, which not only reduces the complexity but also helps to achieve higher accuracy of the decision system. The proposed method has been applied on some benchmark dataset collected from the UCI repository and dynamic reduct is computed, and from the reduct optimal classification rules are also generated. Experimental result shows the efficiency of the proposed method.*

## KEYWORDS

*Dimension Reduction, Incremental Data, Dynamic Reduct, Rough Set Theory, PSO, classification System.*

## 1. INTRODUCTION

In today's e-governance age, everything is being done through electronic media. So huge data is generated and collected from various areas for which proper data management is necessary. Retrieval of some interesting information from stored data as well as time variant data is also a very challenging task. Extraction of meaningful and useful pattern from these large data is the main objective of data mining technique [1, 2]. Data mining techniques basically uses the concept of database technology [3] and pattern recognition [5, 6] principles. Feature selection [7, 8] and reduct generation [9, 10] are frequently used as a pre-processing step to data mining and knowledge discovery. For static data, it selects an optimal subset of features from the feature space according to a certain evaluation criterion. In recent years, dimension of datasets are growing rapidly in many applications which bring great difficulty to data mining and pattern recognition. As datasets changes with time, it is very time consuming or even infeasible to run

DOI : 10.5121/ijaia.2014.5403

repeatedly a knowledge acquisition algorithm. Rough Set Theory (RST) [10-14], a new mathematical approach to imperfect knowledge, helps to find the static as well as dynamic reduct. Dynamic reducts can put up better performance in very large datasets as well as enhance effectively the ability to accommodate noise data. The problem of attribute reduction for incremental data falls under the class of Online Algorithms and hence demands a dynamic solution to reduce re-computation. Liu [15] developed an algorithm for finding the smallest attribute set of dynamic reducts with increase data. Wang and Wang [16] proposed a distributed algorithm of attribute reduction based on discernibility matrix and function. Zheng et al. [17] presented an incremental algorithm based on positive region for generation of dynamic reduct. Deng [18] presented a method of attribute reduction by voting in a series of decision subsystems for generation of dynamic reduct. Jan G. Bazan et al. [19] presented the concept of dynamic reducts to solve the problem of large amount of data or incremental data.

Construction of classifier model for any decision system is an important job for many data mining applications. The objective of developing such a classifier is to classify unlabeled dataset into classes. Many particle swarm optimization algorithms [21-29] have been developed that deals only with continuous variables. This is a significant limitation because many optimization problems are there which featuring discrete variables in the problem domain. Typical problems are there in the space which deals with the ordering, grouping or arranging of discrete variables such as scheduling or routing problems. Therefore, the developing of particle swarm optimization algorithms that deals with discrete variables is important for such kind of problem. We propose a variant of Particle Swarm Optimization (PSO) algorithm applied to discrete data for rule selection in Data Mining. We will refer to this algorithm as the discrete Particle Swarm Optimization (DPSO) algorithm. The DPSO deals with discrete valued data, and its population of candidate solutions contains particles of different sizes. Although the algorithm has been specifically designed for optimized rule selection task, it is by no means limited to this kind of application. The DPSO algorithm may be applied to other optimization problems with little modification.
In the proposed method, a novel heuristic approach is proposed to find out a dynamic reduct of the incremental dataset using the concept of Rough Set Theory and a particle swarm optimization algorithm for selection of optimal classification rules from the whole dataset. To understand the concepts of dynamic data, a sample dataset is divided into two sub sets considering one as old dataset and other as new dataset. Using the concept of discernibility matrix and attribute dependency of Rough Set Theory reduct is computed from old dataset. Then to handle the new data or incremental data, previously computed reduct is modified wherever changes are necessary and generates dynamic reduct for the entire system. Then for designing a classification model for the entire decision system, primarily, all exhaustive rules are generated by considering the computed reduct using genetic algorithm technique available in RSES software tool [30]. For the larger dataset, if all decision rules are considered for data analysis then time complexity will be very high. For this reason a minimum set of interesting rules are generated here. Now discrete Particle Swarm Optimization (PSO) algorithm is applied on the initial rule set for selection of few optimal or near optimal classification rules. DPSO method identifies important or significant rules from initial classification rules without sacrificing predictive accuracy.

The details of the algorithm are provided in subsequent section.

The rest of the paper is organized as follows: Basic Concepts of Rough Set Theory is described in section 2. Section 3 demonstrated the process of generation of dynamic reduct and Section 4 introduces the DPSO algorithm proposed in this paper for the task of optimal rule selection. Section 5 reports experimental methodology, results and finally conclusion of the paper is stated in section 6.

# 2. BASIC CONCEPTS OF ROUGH SET THEORY

The rough set theory is based on indiscernibility relations and approximations. Indiscernibility relation is usually assumed to be equivalence relation, interpreted so that two objects are equivalent if they are not distinguishable by their properties. Given a decision system DS = (U, A, C, D), where U is the universe of discourse and A is the total number of attributes, the system consists of two types of attributes namely conditional attributes (C) and decision attributes (D) so that $A = C \cup D$. Let the universe $U = \{x_1, x_2... x_n\}$, then with any $P \subseteq A$, there is an associated P-indiscernibility relation IND(P) defined by equation (1).

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

If $(x, y) \in$ IND (P), then x and y are indiscernible with respect to attribute set P. These indistinguishable sets of objects, therefore define an indiscernibilty relation referred to as the P-indiscernibility relation and the class of objects are denoted by $[x]_P$.

The lower approximation of a target set X with respect to P is the set of all objects which certainly belongs to X, as defined by equation (2).

$$\underline{PX} = \{x | [x]_P \subseteq X\} \quad (2)$$

The upper approximation of the target set X with respect to P is the set of all objects which can possibly belong to X, as defined by equation (3)

$$\overline{PX} = \{x | [x]_P \cap X \neq \emptyset\} \quad (3)$$

As rough set theory models dissimilarities of objects based on the notions of discernibility, a discernibility matrix is constructed to represent the family of discernibility relations. Each cell in a discernibility matrix consists of all the attributes on which the two objects have the different values. Two objects are discernible with respect to a set of attributes if the set is a subset of the corresponding cell of the discernibility matrix.

**(a) Discernibility Matrix and Core:** Given a decision system DS = (U, A, C, D), where U is the universe of discourse and A is the total number of attributes. The system consists of two types of attributes namely conditional attributes (C) and decision attributes (D) so that $A = C \cup D$. Let the universe $U = \{x_1, x_2... x_n\}$, then discernibility matrix $M = (m_{ij})$ is a $|U| \times |U|$ matrix, in which the element $m_{ij}$ for an object pair $(x_i, x_j)$ is defined by (4).

$$m_{ij} = \{a \in C : a(x_i) \neq a(x_j) \wedge (d \in D, d(x_i) \neq d(x_j))\} \quad (4)$$

where, i, j = 1, 2, 3... n

Thus, each entry (i, j) in the matrix S contains the attributes which distinguish the objects i and j. So, if an entry contains a single attribute say, $A_s$, it implies that the attribute is self sufficient to distinguish two objects and thus it is considered as the most important attribute, or core attribute. But in reality, several entries may contain single attribute, union of which is known as core CR of the dataset, as defined in (5).

$$CR = \cup\{m_{ij} | m_{ij} \neq \emptyset \text{ and } |m_{ij}| = 1, \forall i, j = 1, 2, \dots, n\} \quad (5)$$

**(b) Attribute Dependency and Reduct:** One of the most important aspects of database analysis or data acquisition is the discovery of attribute dependencies; that establishes a relationship by finding which variables are strongly related to which other variables. In rough set theory, the notion of dependency is defined very simply. Assume two (disjoint) sets of attributes, P and Q, and inquire what degree of dependency is present between them. Each attribute set induces an (indiscernibility) equivalence class structure. Say, the equivalence classes induced by P is $[x]_P$, and the equivalence classes induced by Q is $[x]_Q$. Then, the dependency of attribute set Q on attribute set P is denoted by $\gamma_P(Q)$ and is given by equation (6).

$$\gamma_P(Q) = \frac{\sum_{i=1}^{N} |\underline{P}X_i|}{|U|} \qquad (6)$$

Where, $Q_i$ is a class of objects in $[x]_Q$ ; $\forall$ i = 1, 2, …, N.

A reduct can be thought of as a sufficient set of attributes to represent the category structure and the decision system. Projected on just these attributes, the decision system possesses the same equivalence class structure as that expressed by the full attribute set. Taking the partition induced by decision attribute D as the target class and R as the minimal attribute set, R is called the reduct if it satisfies (7). In other words, R is a reduct if the dependency of decision attribute D on R is exactly equal to that of D on whole conditional attribute set C.

$$\gamma_R(D) = \gamma_C(D) \qquad (7)$$

The reduct of an information system is not unique. There may be many subsets of attributes which preserve the equivalence-class structure (i.e., the knowledge) expressed in the decision system.

**(c) Attribute Significance:** Significance of an attribute a in a decision table *A= (U, CUD)* (with the decision set *D*) can be evaluated by measuring the effect of removing of an attribute a$\in$C from the attribute set C on the positive region. The number γ(C, D) expresses the degree of dependency between attributes C and D. If attribute 'a' is removed from the attribute set C then the value of (γ(C, D)) will be changed.

So the significance of an attribute a is defined as

$$\sigma_{(C,D)}^{a} = \frac{\gamma(C,D) - \gamma(C - \{a\}, D)}{\gamma(C,D)} \qquad (8)$$

**(d) Dynamic Reduct:** The purpose of dynamic reducts is to get the stable reducts from decision subsystems. Dynamic reduct can be defined in the following direction.

**Definition 1:** If *DS* = (*U*, *A*, *d*) is a decision system, then any system *DT* = (*U'*, *A*, *d*) such that *U'* ⊆ *U* is called a subsystem of *DS*. By P (*DS*) we denote the set of all subsystems of *DS*. Let *DS* = (U, *A*, *d*) be a decision system and *F* ⊆ P (*DS*). By *DR* (*DS*, *F*) we denote the set *RED (DS)* $\cap \bigcap_{DT \in F}$ *RED (DT).* Any elements of *DR* (*DS*, *F*) are called an F-dynamic reduct of *DS*.
So from the definition of dynamic reducts it follows that a relative reduct of DS is dynamic if it is also a reduct of all sub tables from a given family of F.

**Definition 2:** Let *DS* = (U, *A*, *d*) be a decision system and *F* ⊆ P (*DS*). By *GDR* (*DS*, *F*) we denote the set

$$\bigcap_{DT \in F} RED\ (DT)$$

Any elements of *GDR* (*DS*, *F*) are called an F generalized dynamic reduct of *DS*. From the above definitions of generalized dynamic reduct it follows that any subset of *A* is a generalized dynamic reduct if it is also a reduct of all sub tables from a given family F.

Time complexity of computation of all reducts is NP-Complete. Also, the intersection of all reducts of subsystems may be empty. This idea can be sometimes too much restrictive, so more general notion of dynamic reducts are described. They are called (F, ε) dynamic reducts, where ε > 0. The set *DR (DS, F)* of all (F, ε) dynamic reducts is defined by

$$DR_{\varepsilon}^{(DS)} = \{C \in RED\ (DS, d): \ \frac{card(DT \in F : C \in red(DT, d))}{Card\ (F)} \geq 1 - \varepsilon\}$$

# 3. DYNAMIC REDUCT GENERATION USING ROUGH SET THEORY

Various concepts of rough set theory like discernibility matrix, attribute significance and attribute dependency are applied together to compute dynamic reducts of a decision system. The term dynamic reduct is used in the sense that the method computes a set of reducts for the incremental data very quickly without unnecessarily increasing the complexity since they are sufficient to represent the system and subsystems of it. Based on the discernibility matrix M and the frequency value of the attributes, the attributes are divided [20] into the core set CR and noncore set NC for old subsystem $DS_{old}$. Next, highest ranked element of NC is added to the core CR in each iteration provided the dependency of the decision attribute D on the resultant set increases for the old subsystem ; otherwise it is ignored and next iteration with the remaining elements in NC is performed. The process terminates when the resultant set satisfies the condition of equation (7) for the old subsystem and is considered as an initial reduct RED_OLD. Then backward attribute removal process is applied for each noncore attribute x in the generated reduct RED_OLD, it is checked whether (7) is satisfied using RED_OLD – {x}, instead of R. Now if it is satisfied, then x is redundant and must be removed. Thus, all redundant attributes are removed and final reduct RED_OLD is obtained.

To generate the dynamic reduct, discernibility matrix is constructed for the new subsystem $DS_{new}$ and frequency values of all conditional attributes are calculated. Now the previously computed reduct (RED_OLD) from the old dataset is applied to new dataset for checking whether it can preserve the positive region in the new data set i.e., whether the dependency value of the decision attribute on that reduct set is equal to that of the decision attribute on the whole conditional attribute set. If the condition is satisfied, then that reduct set is considered as dynamic reduct (DRED). Otherwise; according to the frequency values obtained using [16] of the conditional attributes, higher ranked attribute is added to the most important attribute set in each iteration provided attribute dependency of the resultant set increases and subsequently a reduct is formed after certain iteration when dependency of the decision attribute on the resultant set is equal to that of the decision attribute on the whole condition attribute set for the new subsystem. Then backward attribute removal process is applied for generation of final dynamic reduct of the system. In this process, significance value of each individual attribute is calculated using equation (8) except that most important attribute set in a reduct. If the significance value of a particular attribute is zero, then that attribute is deleted from the reduct. In this way, all redundant attributes are removed and finally dynamic reduct is generated by modifying the old reducts for the entire data.

The proposed method describes the attribute selection method for the computation of reducts from old data and dynamic reduct set DRED for entire data considering incremental data.

Algorithm1 generates initial reduct for the old decision system $DS_{old}$ = (U, A, C, D) and Algorithm2 generates dynamic reduct for the entire data, by considering the old data as well as incremental data.

**Algorithm1**: Initial_Reduct_Formation ($DS_{old}$, CR, NC)
Input: $DS_{old}$, the decision system with C conditional attributes and D decisions with objects x, CR, the core and NC, the non-core attributes
Output: RED_OLD, initial reduct

```
Begin
        RED_OLD = CR   /* core is considered as initial reduct*/
        NC_OLD = NC /* take a copy of initial elements of NC*/
       /*Repeat-until below forward selection to give one reduct*/
        Repeat
            x = highest ranked element of NC_OLD
            If (x = $\phi$) break   /*if no element found in NC*/
            If ($\gamma_{RED\_OLD \cup \{x\}}$ (D) > $\gamma_{RED\_OLD}$ (D))
               {
                 RED_OLD = RED_OLD $\cup$ {x}
                NC_OLD = NC_OLD - {x}
               }
        Until ($\gamma_{RED\_OLD}$(D) = $\gamma_C$(D))
          // apply backward removal
        For each x in (RED_OLD – CR)
              If ($\gamma_{RED\_OLD - \{x\}}$(D) = = $\gamma_C$(D))
                   RED_OLD = RED_OLD - {x}
       Return (RED_OLD);
End
```

**Algorithm2**: Dynamic_Reduct_Formation (DS, C, D)
//An algorithm for computation of dynamic reducts for incremental data
Input: DS = {$DS_{new}$}, the new decision system with C conditional attributes and D decisions attribute and reduct RED_OLD obtained from 'Reduct Formation' algorithm for the old dataset ($DS_{old}$).
Output: Dynamic reduct (DRED), reduct of $DS_{old} \cup DS_{new}$

```
Begin
    If (($\gamma_{(RED\_OLD)}$ (D) = $\gamma_{(C)}$ (D))
        {
          DRED = RED_OLD
          Return DRED
        }
    Else {
         NC = C - RED_OLD
         CR = DRED  /*initial reduct is considered as core reduct of new system */
         Repeat
            DRED = RED_OLD   /* Old reduct is considered as core */
            x = highest frequency attribute of NC
            If ($\gamma_{DRED \cup \{x\}}$(D) > $\gamma_{DRED}$(D))
              {
                DRED = DRED $\cup$ {x}
                 NC = NC - {x}
            }
        Until ($\gamma_{DRED}(D) = \gamma_C(D)$)
        // apply backward removal
    For each highest ranked attribute x in (DRED – CR) using (8)
        If ($\gamma_{DRED - \{x\}}(D) = = \gamma_C(D)$)
            DRED = DRED - {x}
      Return (DRED);
  }  /* end of else*/
End
```

# 4. OPTIMUM CLASSIFICATION SYSTEM GENERATION

The algorithm presented in this section is based on DPSO algorithm [21]. The proposed algorithm deals with generation of optimized classification rules from discrete valued dataset, which is typically a rule mining problem. Primarily, all exhaustive rules are generated from the entire dataset by considering its reduct using Genetic Algorithm method available in RSES2.2.2 software tool [30]. Then the DPSO method identifies important or significant rules from all initial available classification rules without sacrificing predictive accuracy. Every rule has two parts, conditional part and decision part, conditional part comprises of some conditional attributes with their values and decision part has decision attribute with the corresponding decision value or class. In a rule, each conditional attribute with the corresponding value is termed as rule component. So a rule is formed by some rule components. In PSO, population of candidate solutions contains particles of different sizes. Here population of candidate solutions is generated from initial rule set. Each particle represents the antecedent part of a rule by considering the conclusion part is fixed at the beginning of the execution and represents a target attribute value. In this case to discover the optimal rule set, predicting different target/decision values, the algorithm has to run several times, one for each decision value. There are N (No of initial rules) particles in a swarm. The length of each particle may vary from 1 to n, where n is the number of unique rule component present in the initial rule set, which is made by considering only the antecedent part of each rule. Each particle $R_i$ keeps a record of the best position it has ever attained. This information is kept in a distinguished particle labeled as $B_i$. The swarm also keeps the information of the global best position ever attained by any particle in the swarm. This information is also kept in a separated particle labeled G. G is equal to the best Bi present in the swarm.

## 4.1. Encoding of the Particles for the Proposed DPSO Algorithm

Each rule is formed by some rule components and each rule component is identified by a unique positive integer number or index. These indices, vary from 1 to n, where n is the number of unique rule components present in the initial rule set. Each particle is subsets of indices without repetition. For example, corresponding to the rules $R_1$, $R_2$ and $R_3$ given below, the rule components are (A=1), (A=2), (B=3), (B=4) and (C=5) which are indexed as 1, 2, 3, 4 and 5 respectively.

$R_1$= {A=1, B=3, C=5}
$R_2$= {A=2, B=4}
$R_3$= {A=2, C=5}

Where, A, B, C are the conditional attributes, N (Number of initial rules) = 3. Here initial swarm representing candidate solution could looks as follows:

$R_1$= {1, 3, 5}
$R_2$= {2, 4}
$R_3$= {2, 5}.

### 4.2. The Initial Population for the Proposed DPSO Algorithm

The initial population of particles is generated as follows.

Population of candidate solutions is generated from initial rule set. Each particle represents the antecedent part of a rule by considering the conclusion part as fixed at the beginning of the execution and represents a target attribute value. Rule encoding process is described in section 4.1. By considering the rule encoding process the particles are formed for each rule generated using GA based classification method.

### 4.3. Velocities

The DPSO algorithm does not use a vector of velocities. It works with proportional likelihoods. Basically, the idea of proportional likelihood used in the DPSO algorithm is almost similar with the idea of velocity used in the standard PSO algorithm. Each particle is associated with a 2 × n array of proportional likelihoods, where 2 and n represents number of rows and number of columns respectively. In this standard proportional likelihood array, each element in the first row of $V_i$ represents the proportional likelihood based on which a rule component be selected. The second row of $V_i$ has the indices of the rule components which is associated with the respective proportional likelihoods of the first row of the vector $V_i$. There is a one-to-one correspondence between the columns of this array. At the beginning, all elements in the first row of $V_i$ are set to 1, e.g., $V_i$ = {{1, 1, 1, 1, 1 }, {1,2,3,4,5}.

After the initial population of particles is generated, this array is always updated before a new configuration for the particle associated to it is made. The updating process is based on $R_i$, $B_i$ (particle's previous best position) and G (global best position) and works as follows.

In addition to $R_i$, Bi and G, three constant updating factors, namely, a, b and c are used to update the proportional likelihoods $v_{(i,d)}$. These factors determine the strength of the contribution of $R_i$, $B_i$ and G to the adjustment of every coordinate $v_{(i,d)}$ € $V_{i.}$ Parameter values of a, b and c is chosen experimentally.

### 4.4 Generating New Particles for the Proposed DPSO Algorithm

The proportional likelihood array $V_i$ is then used to sample a new configuration of particle $R_i$ that is, the particle associated to it. First, for a particle with $V_i$, all indices present in $R_i$ have their corresponding proportional likelihood increased by 'a'. Similarly, all indices present in $B_i$ and G have their corresponding proportional likelihood increased by 'b' and 'c' respectively. Now each element of the first row of the array $V_i$ is then multiplied by a uniform random number between 0 and 1. A new random number is generated for every single multiplication performed. The new particle is then defined by ranking the columns in $V_i$ by the values in its first row. That is, the elements in the first row of the array are ranked in a decreasing order of value and the indices of the rule components (in the second row of $V_i$) follow their respective proportional likelihoods.

Thus for example, after all the steps if a particle i has length 3 and the particle associated to the array

$$V_i = \begin{array}{ccccc} 0.74 & 0.57 & 0.50 & 0.42 & 0.20 \\ 5 & 4 & 3 & 1 & 2 \end{array}$$

Then first 3 indices from the second row of Vi would be selected to compose the new particle. That is, $R_i$ = {*, *, *} the indices (rule components) 5, 4 and 3 would be selected to compose the new particle, i.e., $R_i$ = {5, 4, 3}. Note that indices that have a higher proportional likelihood are, on average, more likely to be selected. If indices are the rule components with same attribute are selected, then higher confidence rule component is selected. In that case new particle size is also changes in the generation. The updating of $R_i$, $B_i$ and G is identical to what is described earlier. In this way new particles are formed generation by generation.

### 4.5. Design of Fitness Function

The purpose of this experiment was to evaluate the classification accuracy and comprehensibility by the number of rules in the data set, and the average number of rule conditions per rule. The fitness function f (Ri) of any particle i is computed as follows. Optimized rule selection process can be performed by DPSO as a multi objective problem to maximize the confidence (association rule mining concept) of a rule to achieve higher classification accuracy as well as minimizing the length of a rule. The goal is to see whether DPSO can select optimized set of rules to achieve a higher classification accuracy rate from initial set of rules.

In this regard following fitness function is considered.
$F_i$=α*$rule_i$_confidence+ (1-α)*1/ ($rule_i$_length)

**$rule_i$ _confidence:** A rule like, $R_i \rightarrow \boldsymbol{(C_{i1} = a)\wedge(C_{i2} = b)\ldots\wedge(C_{in} = c)} \rightarrow \boldsymbol{(D = d_i)}$ has the condition part $C_i$ where conditional attributes are associated with value, and the decision part D has the decision values $\boldsymbol{d_i}$. So here the rule maps $C_i \rightarrow \boldsymbol{d_i}$.

Then the confidence of the rule is conf ($R_i$) = number of rows in *dataset* that match $C_i$ and have class label $\boldsymbol{d_i}$**/** number of rows in *dataset* that match only $C_i$.

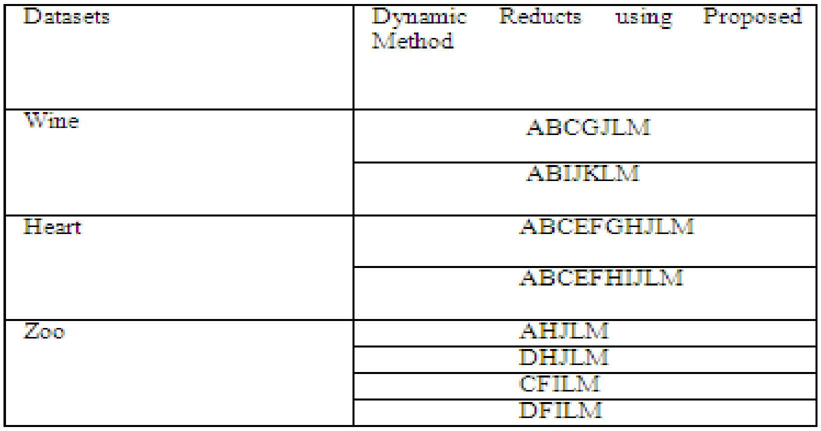
Here relative importance of the rule confidence and the length of the rule are considered. Rule confidence is set larger than length of a rule because the classification performance is assumed more important than the number of conditional attributes present in a rule i.e. rule length.

The objective is to find the fittest rules with which it is possible to classify the data set as belonging to one of the classes with an acceptable accuracy. Here α=0.8 is considered.

## 5. EXPERIMENTAL RESULTS AND DISCUSSION

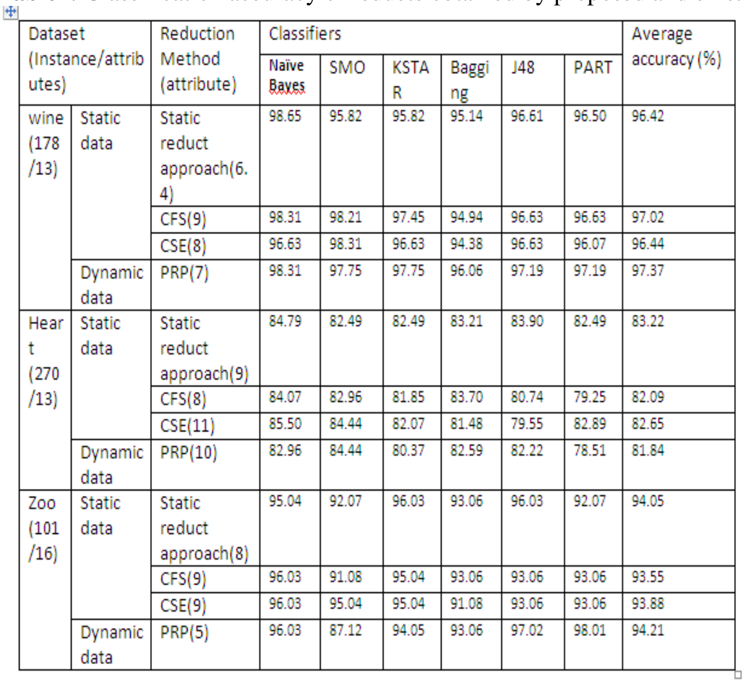
The method is applied on some benchmark datasets obtained from UCI repository 'http: //www.ics.uci.edu/mlearn/MLRepository'. Here the attributes are abbreviated by letters A, B, and so on, starting from their column position in the dataset. The wine dataset contains 178 instances and 13 conditional attributes. In our method, for computation of dynamic reduct the wine dataset is divided into 2 sub tables considering randomly 80% of data as old data and other 20% of data is new data. Reduct is calculated for the old data using Algorithm1.Then based on previous reducts, the proposed algorithm worked on new data and generates two dynamic reducts {{ABCGJLM}, {ABIJKLM}} for the whole dataset. Similarly dynamic reducts are calculated for the heart and Zoo dataset. Reducts are also calculated for the modified data set using static data approach. All results are given in Table 1. Accuracies of the reduct of our proposed algorithm (PRP) are calculated and compared with existing attribute reduction techniques like 'Correlation-based Feature Selection' (CFS) and 'Consistency-based Subset Evaluation' (CSE), from the 'weka' tool [31] as shown in Table 2. The proposed method, on average, contains lesser number of attributes compared to CFS and CSE and at the same time achieves higher accuracy, which shows the effectiveness of the method.

**Table1..** Dynamic reducts of datasets



| Datasets | Dynamic Reducts using Proposed Method |
|---|---|
| Wine | ABCGJLM |
| | ABIJKLM |
| Heart | ABCEFGHJLM |
| | ABCEFHIJLM |
| Zoo | AHJLM |
| | DHJLM |
| | CFILM |
| | DFILM |

**Table2:** Classification accuracy of reducts obtained by proposed and existing method



| Dataset (Instance/attributes) | | Reduction Method (attribute) | Classifiers | | | | | | Average accuracy (%) |
|---|---|---|---|---|---|---|---|---|---|
| | | | Naïve Bayes | SMO | KSTAR | Bagging | J48 | PART | |
| wine (178 /13) | Static data | Static reduct approach(6.4) | 98.65 | 95.82 | 95.82 | 95.14 | 96.61 | 96.50 | 96.42 |
| | | CFS(9) | 98.31 | 98.21 | 97.45 | 94.94 | 96.63 | 96.63 | 97.02 |
| | | CSE(8) | 96.63 | 98.31 | 96.63 | 94.38 | 96.63 | 96.07 | 96.44 |
| | Dynamic data | PRP(7) | 98.31 | 97.75 | 97.75 | 96.06 | 97.19 | 97.19 | 97.37 |
| Heart (270 /13) | Static data | Static reduct approach(9) | 84.79 | 82.49 | 82.49 | 83.21 | 83.90 | 82.49 | 83.22 |
| | | CFS(8) | 84.07 | 82.96 | 81.85 | 83.70 | 80.74 | 79.25 | 82.09 |
| | | CSE(11) | 85.50 | 84.44 | 82.07 | 81.48 | 79.55 | 82.89 | 82.65 |
| | Dynamic data | PRP(10) | 82.96 | 84.44 | 80.37 | 82.59 | 82.22 | 78.51 | 81.84 |
| Zoo (101 /16) | Static data | Static reduct approach(8) | 95.04 | 92.07 | 96.03 | 93.06 | 96.03 | 92.07 | 94.05 |
| | | CFS(9) | 96.03 | 91.08 | 95.04 | 93.06 | 93.06 | 93.06 | 93.55 |
| | | CSE(9) | 96.03 | 95.04 | 95.04 | 91.08 | 93.06 | 93.06 | 93.88 |
| | Dynamic data | PRP(5) | 96.03 | 87.12 | 94.05 | 93.06 | 97.02 | 98.01 | 94.21 |

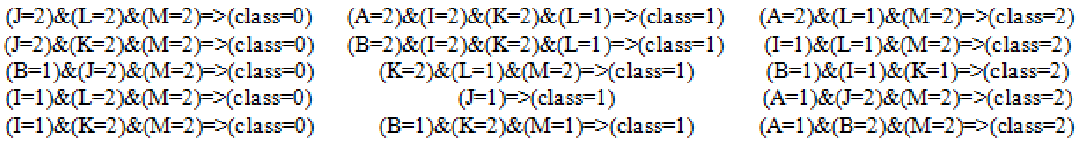
Now, randomly a reduct is selected from the reduct set for generation of optimal classification rules. At first all exhaustive rules are generated from the entire dataset using Genetic Algorithm method [30]. Then the proposed DPSO method selects important or significant rules from all initially generated classification rules without sacrificing predictive accuracy.

To illustrate the method, wine dataset with the reduct (ABIJKLM) is considered. First, the 178 records with 3 distinct classes (0, 1, 2) in the wine data set were divided into 10 almost equal sized folds. The computational experiments involved a 10-fold cross-validation method [13].The folds were randomly generated but under the following regulation. The proportion of objects of different classes in every single fold is same to the one found in the original data set containing all 178 records. Each of the 10 folds is used once as test set and the remaining of the data set is used as training set. The purpose of the experiment was to evaluate the generalization ability, measured as the classification accuracy on test set. Each training set was used for generation of initial classification rules. For each initial rule set, DPSO algorithm generates optimized set of classification rules according to their fitness which is used to classify the test set (each 10 test folds) accordingly. DPSO selects only the best particle in each run as the rule.

As the DPSO algorithm is stochastic algorithm, 30 independent runs for the algorithm were performed for every single fold and for every decision classes. The average number of rules by the rule selection algorithms has always been rounded to the nearest integer. The population size (initial rule set) used for the algorithm is on average 52 and the search stops in one run after 100 iterations. Other choices of parameter values were a = 0.10, b = 0.12 and c = 0.14. Parameter values were fixed experimentally. The results of experiment were as follows.

For Wine on average 12 rules are selected as optimized rules from 52 rules. In some of the folds 100% accuracy is also achieved. And finally using these 12 rules on test set 98.96% of average classification accuracy is achieved.Using fewer rules, DPSO algorithm obtained on average, a better predictive accuracy than the classification performed using all the initial classification rules for few classes. For the initial rule set average accuracy values on test set for 3 classes (0, 1, 2) were 100, 100, 97.9 while using optimized rule set corresponding accuracy values on test set were 100, 96.9, 100. The results also indicate that not only the predictive accuracy is good, but also the number of rule conditions or rule length is relatively short (on average each optimal rule contains two less rule components compare to initial rules) and also number of rules reduces dramatically. Few samples (out of 52) of initial rules and optimal rules obtained by the proposed method for the wine dataset are shown in a tabular form in Table3 and Table4 respectively. Flowchart of actions of generating the initial classification rules using GA method and optimal classification rules using PSO method along with their classification accuracy are also given. And Table5 shows average classification accuracy achieved by the proposed method for some benchmark datasets.

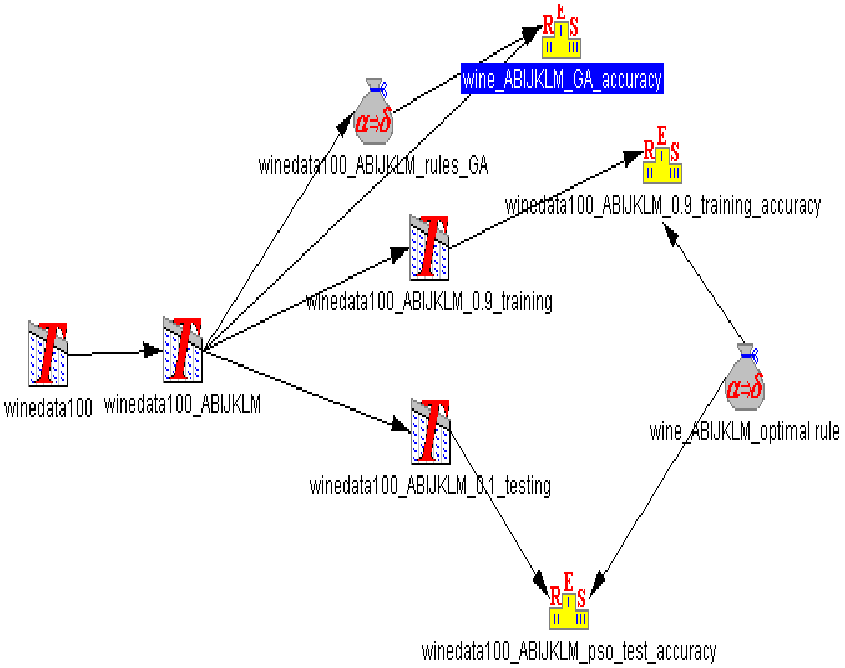
Table3: Sample of initial learning from the full wine data set using GA

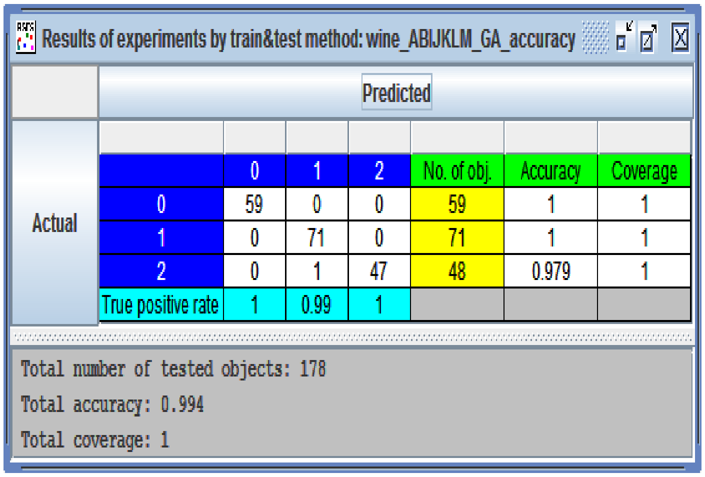


| | | |
|---|---|---|
| (J=2)&(L=2)&(M=2)=>(class=0) | (A=2)&(I=2)&(K=2)&(L=1)=>(class=1) | (A=2)&(L=1)&(M=2)=>(class=2) |
| (J=2)&(K=2)&(M=2)=>(class=0) | (B=2)&(I=2)&(K=2)&(L=1 )=>(class=1) | (I=1)&(L=1)&(M=2)=>(class=2) |
| (B=1)&(J=2)&(M=2)=>(class=0) | (K=2)&(L=1)&(M=2)=>(class=1) | (B=1)&(I=1)&(K=1)=>(class=2) |
| (I=1)&(L=2)&(M=2)=>(class=0) | (J=1)=>(class=1) | (A=1)&(J=2)&(M=2)=>(class=2) |
| (I=1)&(K=2)&(M=2)=>(class=0) | (B=1)&(K=2)&(M=1)=>(class=1) | (A=1)&(B=2)&(M=2)=>(class=2) |

Table4: Sample of optimal classification rules using PSO

(A=2)&(I=2)&(K=2)=>(class=0)
(K=2)&(M=2)=>(class=0)
(K=2)=>(class=1)
(B=1)&(K=2)=>(class=1)
(J=1)=>(class=1)
(A=1)&(I=2)=>(class=1)
(A=1)&(K=2)=>(class=1)
(J=2)&(K=1)=>(class=2)
(L=1)&(K=1)=>(class=2)

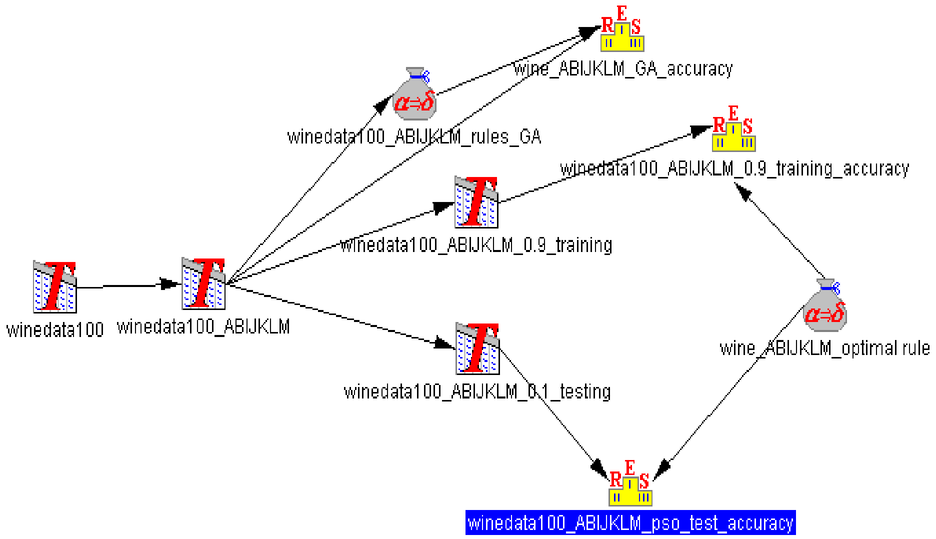

wine_ABIJKLM_GA_accuracy
winedata100_ABIJKLM_rules_GA
winedata100_ABIJKLM_0.9_training_accuracy
winedata100_ABIJKLM_0.9_training
winedata100
winedata100_ABIJKLM
wine_ABIJKLM_optimal rule
winedata100_ABIJKLM_0.1_testing
winedata100_ABIJKLM_pso_test_accuracy

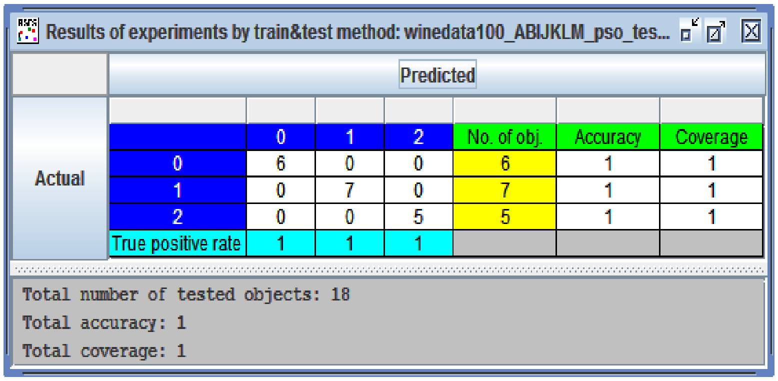

Results of experiments by train&test method: wine_ABIJKLM_GA_accuracy

| | | Predicted | | | | | |
|---|---|---|---|---|---|---|---|
| Actual | | 0 | 1 | 2 | No. of obj. | Accuracy | Coverage |
| | 0 | 59 | 0 | 0 | 59 | 1 | 1 |
| | 1 | 0 | 71 | 0 | 71 | 1 | 1 |
| | 2 | 0 | 1 | 47 | 48 | 0.979 | 1 |
| | True positive rate | 1 | 0.99 | 1 | | | |

```
Total number of tested objects: 178
Total accuracy: 0.994
Total coverage: 1
```

Fig. 1. Flowchart of actions of generating classification rules and accuracy using Genetic Algorithm for wine data using the reduct ABIJKLM


wine_ABIJKLM_GA_accuracy
winedata100_ABIJKLM_rules_GA
winedata100_ABIJKLM_0.9_training_accuracy
winedata100_ABIJKLM_0.9_training
winedata100
winedata100_ABIJKLM
wine_ABIJKLM_optimal rule
winedata100_ABIJKLM_0.1_testing
winedata100_ABIJKLM_pso_test_accuracy


Results of experiments by train&test method: winedata100_ABIJKLM_pso_tes...

| | | Predicted | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | No. of obj. | Accuracy | Coverage |
| Actual | 0 | 6 | 0 | 0 | 6 | 1 | 1 |
| | 1 | 0 | 7 | 0 | 7 | 1 | 1 |
| | 2 | 0 | 0 | 5 | 5 | 1 | 1 |
| | True positive rate | 1 | 1 | 1 | | | |

```
Total number of tested objects: 18
Total accuracy: 1
Total coverage: 1
```

Fig. 2. Flowchart of actions of generating optimal classification rules and accuracy using PSO algorithm for wine dataset using the reduct ABIJKLM for a single fold.

Table5: Classification accuracy of reduct based dataset using proposed method

| Datasets | No of objects | No of attributes | Dynamic Reduct | No. of initial rule using GA | Average no of optimal rules using DPSO | Average classification accuracy of DPSO |
|---|---|---|---|---|---|---|
| Wine | 178 | 14 | ABIJKLM | 52 | 12 | 98.96 |
| Zoo | 101 | 17 | AHJLM | 23 | 10 | 98.70 |
| Pima-indian-diabetes- | 768 | 9 | BFH | 97 | 22 | 98.90 |
| Austra | 690 | 15 | ABEJ | 906 | 200 | 99.80 |

## 6. CONCLUSION

The paper describes a new method of attribute reduction for incremental data using the concepts of Rough Set theory. Even if the data is not completely available at a time, the algorithm first finds the reduct of existing data and finally compute the dynamic reduct without reusing the data that has already arrived rather it extract some knowledge from the arrived data which is used together with newly available data. The proposed dimension reduction method used only the concepts of rough set theory which does not require any additional information except the decision system itself. Since, reduct generation is a NP-complete problem, so different researchers' use different heuristics to compute reducts used for developing classifiers. Dynamic reducts are very important for construction of a strong classifier. So considering this dynamic reduct for the entire data, proposed DPSO method produces optimal classification rules for the data set. The discovered rules are with of high accuracy and comprehensibility. Using fewer rules, DPSO algorithm obtained on average, a better predictive accuracy than the classification performed using all the initial classification rules for few classes. In the future dynamic rules can be selected from the incremental data itself by using the set of dynamic reducts and finally ensemble them to generate an efficient dynamic classifier.

## REFERENCES

[1] Han, and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2001.

[2] I. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2nd edition, 2005

[3] Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends Viviana E. Ferraggine , Jorge H. Doorn , Laura C. Rivero, ISBN-10: 1605662429 ISBN-13: 978-1605662428

[4] D. Freedman, R. Pisani, and R. Purves. Statistics. W. Norton & Company, 3rd edition, September 1997.

[5] Devijver, P.A., and Kittler, J. (1982) Pattern Recognition: A Statistical Approach Englewood Cliffs, NJ: Prentice Hall.

[6] Della Pietra,S., Della Pietra, V., and Lafferty, J. (1997) "Inducing features of random fields." IEEE transactions on pattern Analysis and Machine Intelligence, 19(4), pp. 380-393.

[7] Ethem Alpaydin Introduction to Machine Learning.PHI, 2010

[8] T. M. Mitchell. Machine Learning. McGraw-Hill, August 1997.

[9] R. Jensen, QiangShen, "Fuzzy-Rough Attribute Reduction with Application to Web Categorization, Fuzzy Sets and Systems" Vol.141, No.3, pp.469-485, 2004

[10] N.Zhong and A. Skowron, "A Rough Set-Based Knowledge Discovery Process", Int. Journal of Applied Mathematics and Computer Science. 11(3), 603-619, 2001. BIME Journal, Volume (05), Issue (1), 2005

[11] Pawlak, Z.: "Rough sets.: International journal of information and computer sciences," Vol, 11, pp. 341-356 (1982)

[12] Pawlak, Z.: "Rough set theory and its applications to data analysis," Cybernetics and systems 29 (1998) 661-688, (1998)

[13] K. Thangavel, A. Pethalakshmi. "Dimensionality reduction based on rough set theory : A review" Journal of Applied Soft Computing, Volume 9, Issue 1, pages 1 -12, 2009.

[14] N.Zhong and A. Skowron, "A Rough Set-Based Knowledge Discovery Process", Int. Journal of Applied Mathematics and Computer Science., 11(3), 603-619, 2001.

[15] Z.T,Liu.: "An incremental arithmetic for the smallest reduction of attributes"Acta Electro nicasinicia,vol.27,no.11,pp.96—98,1999

[16] J.Wang and J.Wang,"Reduction algorithms based on discernibility matrx:The order attributes method.Journal of computer Science and Technology,vol.16.No.6,2001,pp.489-504

[17] G.Y.Wang,Z.Zheng and Y.Zhang"RIDAS-A rough set based intelligent data analysis system" Proceedigs of the 1st International conference on machine Learning and Cybernatics,Beiing,Vol2,Feb,2002,pp.646-649.

[18] D.Deng,D.Yan and J.Wang,"parallel Reducts based on Attribute significance",LNAI6401,2010,pp.336-343.

[19] G..Bazan ,"Dynamic reducts and statistical Inference" Proceedings of the 6th International conference on Information Processing and Management of uncertainity in knowledge based system,July 125, Granada,Spain,(2),1996pp.1147-1152

[20] Asit Kumar Das, Saikat Chakrabarty, Shampa Sengupta "Formation of a Compact Reduct Set Based on Discernibility Relation and Attribute Dependency of Rough Set Theory" Proceedings of the Sixth International Conference on Information Processing – 2012 August 10 - 12, 2012, Bangalore, Wireless Network and Computational Intelligence Springer pp 253-261.

[21] Elon S. Correa, Alex A. Freitas , Colin G. Johnson "A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatics Data Set" in the proceedings of GECCO'06, July 8-12, 2006, Seattle, Washington, USA pp35-42.

[22] T. Blackwell and J. Branke. "Multi-swarm optimization in dynamic environments." In Lecture Notes in Computer Science, volume 3005, pages 489-500. Springer-Verlag, 2004

[23] E. S. Correa, M. T. Steiner, A. A. Freitas, and C. Carnieri. "Using a genetic algorithm for solving a capacity p-median problem." Numerical Algorithms, 35:373-388, 2004.

[24] A. A. Freitas. Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag, October 2002.

[25] G. Kendall and Y. Su. "A particle swarm optimisation approach in the construction of optimal risky portfolios." In Proceedings of the 23rd IASTED International Multi-Conference on Applied Informatics, pages 140-145, 2005. Artificial intelligence and applications

[26] J. Kennedy and R. C. Eberhart. "A discrete binary version of the particle swarm algorithm." In Proceedings of the 1997 Conference on Systems, Man, and Cybernetics, pages 4104-4109, Piscataway, NJ, USA, 1997. IEEE.

[27] J. Kennedy and R. C. Eberhart. Swarm Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001

[28] R. Poli, C. D. Chio, and W. B. Langdon. "Exploring extended particle swarms: a genetic programming approach." In GECCO'05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pages 169-176, New York, NY, USA, 2005. ACM Press.

[29] Y. Shi and R. C. Eberhart. "Parameter selection in particle swarm optimization." In EP'98: Proceedings of the 7th International Conference on Evolutionary Programming, pages 591-600, London, UK, 1998. Springer-Verlag.

[30] Bazan J., Szczuka M., The RSES Homepage, http://alfa.mimuw.edu.pl/˜rses

[31] WEKA: Machine Learning Software, http://www.cs.waikato.ac.nz/~ml/