Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 195 (2008) 786–798
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Mining classification rules with Reduced MEPAR-miner Algorithm

Emel Kizilkaya Aydogan [a], Cevriye Gencer [b,*]

[a] Department of Industrial Engineering, Faculty of Engineering, Erciyes University, Kayseri, Turkey
[b] Department of Industrial Engineering, Faculty of Engineering and Architecture, Gazi University, Ankara, Turkey

Abstract

In this study, a new classification technique based on rough set theory and MEPAR-miner algorithm for association rule mining is introduced. Proposed method is called as 'Reduced MEPAR-miner Algorithm'. In the method being improved rough sets are used in the preprocessing stage in order to reduce the dimensionality of the feature space and improved MEPAR-miner algorithms are then used to extract the classification rules. Besides, a new and an effective default class structure is also defined in this proposed method. Integrating rough set theory and improved MEPAR-miner algorithm, an effective rule mining structure is acquired. The effectiveness of our approach is tested on eight publicly available binary and *n*-ary classification data sets. Comprehensive experiments are performed to demonstrate that Reduced MEPAR-miner Algorithm can discover effective classification rules which are as good as (or better) the other classification algorithms. These promising results show that the rough set approach is a useful tool for preprocessing of data for improved MEPAR-miner algorithm.

Keywords: Data mining; Classification rules; Attribute reduction; Rough set; Evolutionary programming

1. Introduction

Knowledge Discovery in Databases (KDD) has become a very attractive discipline both for research and industry within last few years. Its goal is to extract pieces of knowledge or 'patterns' from usually very large databases [1].

Rough set methodology provides a powerful tool for knowledge discovery from large and incomplete sets of data. A number of algorithms and systems have been developed based on the rough set theory [2]. Babu et al. [3] proposed a new learning approach integrating the activities of data abstraction, frequent item generation, compression, classification and the use of rough sets. Hassan and Tazaki [4] combined rough set theory and Genetic programming (GP) for deriving knowledge rules from medical database. Similarly, hybrid algorithms based on neural network and rough set theory are introduced in literature [5–7]. The main idea

* Corresponding author.
E-mail address: ctemel@gazi.edu.tr (C. Gencer).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2007.05.024

in these studies is accelerating and simplifying the process of using neural networks for mining knowledge. In addition, Pal and Mitra [8] introduced a new hybrid algorithm based on fuzzy sets and rough set theory for case generation, and showed efficiency of this algorithm on real-life data sets. Similarly, Wong et al. [9] proposed a method based on the concepts of Genetic Algorithm (GA) and SVD-QR method to construct an appropriate fuzzy system for pattern classification. Huang et al. [10] proposed a hybrid approach of rough set theory and genetic algorithm for fault diagnosis. Zhang et al. [11] introduced a hybrid classifier based on rough set theory and support vector machines called RS-SVMs to recognize radar emitter signals.

On the other hand, MEPAR-miner [12] is a new algorithm known as 'Multi-Expression Programming for Association Rule Mining' which is based on genetic programming and one of the most successful algorithm in association rule mining literature. But there are some drawbacks of this algorithm. Firstly, in the encoding structure, all of the attributes are used. Thus, the search space dimensions also get larger and the possibility of getting trap of local optima increases. Secondly, in the MEPAR-miner algorithm, a simple default class structure which depends on the mostly encountered class is used. It is observed that, on the test data no meaningful effect exists in the predictive accuracy operations of such default class structure. Finally, in the MEPAR-miner algorithm, since chromosome objective value calculation is time consuming (because of the reason that; for every terminal and function, all the training data sets are considered), solution time of simple genetic algorithm that is used may take a very long time. In order to eliminate these disadvantages, a Reduced MEPAR-miner Algorithm which has an effective encoding and default class structure and works according to parallel steady state genetic algorithm logic was improved.

This paper is organized as follows: In the first two sections, a brief description about rough set theory and MEPAR-miner algorithm is explained. The next section introduces our new detailed method. Following section presents some illustrative applications of our approach. Last section concludes our paper.

2. Rough set theory

Rough Set Theory (RST), introduced by Pawlak in the early 1980s, is a mathematical tool to deal with classification problems. It is based on the assumption that data and information are associated with every object of the universe of discourse. According to the definition given in Pawlak [13], a knowledge representation system or an information system is a pair $S = (U, A)$, where U is a non-empty, finite set of objects (called the universe), and A is a non-empty set of attributes.

The RS theory is based on the observation that objects may be indiscernible because of limited available information. For a subset of attributes $B \subseteq A$, the indiscernibility relation is defined by $IND(B)$ [13]:

$$IND(B) = \{(x, y) \in U^2 | a \in Ba(x) = a(y)\}.$$

$IND(B)$ is an equivalence relation on the set U. The relation $IND(B)$, $B \subseteq A$, constitutes a partition of U which is denoted $U/ND(B)$. If $(x, y) \in IND(B)$, then x and y are indiscernible by attributes from B. The equivalence classes of the B-indiscernibility relation are denoted $[X]_B$. For a subset $X \subseteq U$, the P-lower approximation of X can be defined as

$$\underline{B}X = \{x | [x]_{\mathrm{B}} \subseteq X\}.$$

Let $IND(B)$ and $IND(Q)$ be indiscernibility relations on U defined by the subset of attributes $B \subseteq A$ and $Q \subseteq \mathrm{A}$ respectively. An often-applied measure is the dependency degree of Q on B, which is defined as follows [13]:

$$\gamma_B(Q) = \frac{|POS_B(Q)|}{|U|}.$$

$POS_{\mathrm{B}}(\mathrm{Q}) = \cup_{X \in U/IND(Q)} \underline{B}X$, called a positive region of the partition $U/IND(Q)$ with respect to B.

One of the major applications of rough set theory is the attribute reduction that is the elimination of attributes considered to be redundant while avoiding information loss. The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Using the dependency degree as a measure, attributes are removed so that the reduced set provides the same dependency degree as the original. In a decision system, a reduct is formally defined as a subset R of the conditional attribute set X such that

$\gamma_R(D) = \gamma_C(D)$, where D is the decision attributes set. A given dataset may have many reducts. Thus the set R of all reducts is defined as [13]:

$$R = \{R : R \subseteq C; \gamma_R(D) = \gamma_C(D)\}.$$

3. MEPAR-miner algorithm

MEPAR-miner algorithm is a relatively new classification technique for data mining, which is developed by Baykasoglu and Ozbakir [12].

In MEPAR-miner algorithm, the original MEP algorithm [14,15] is modified and adapted to extract classification rules. The original MEP chromosome representation, function and terminal sets are modified and redesigned to generate logical expressions. A logical expression represents a classification rule. Multiple rules can be combined together to form a set of decision rules for n-ary classification. The main structure of the rule list can be as follows:

IF antecedent$_1$ **THEN** class$_1$
ELSE IF antecedent$_2$ **THEN** class$_2$
...
...
ELSE class$_{\text{default}}$

Default class structure: The evaluation of rule list is started from the topmost rule and performed towards the default class until the instance exactly matches the corresponding rule. In the case that none of the rules in the list of rules matches a new instance, the new instance will be classified as the default class. In the MEPAR-miner approach default class is determined as the largest class in the data set.

Function and terminal sets: In MEPAR-miner algorithm, the chromosome structure consists of two parts: the first part is the terminal set and the second part is the function set that contains the logical functions. Terminal set consists of attribute, Relational Operator (RO) and value from the domain of corresponding attribute. In the given classification problem, the length of terminal genes is equal to the number of attributes. Each terminal gene is assigned a attribute randomly. After the attributes are assigned, the relational operators are set according to the type of the quality (categorical or continues).

The function set which is the second part of the chromosome is formed by logical operators. AND, OR operators have two argument pointers while NOT operator has only.

Function and terminal sets and also sample chromosome structure are illustrated in Table 1.

Algorithm type: Simple genetic algorithm is used. The chromosome with the best logical expression in population is copied to the next generation without changed (elitization).

Selection mechanism: Binary tournament selection procedure is used for evolution where two individuals are selected randomly from the current population and used for crossover and mutation operators.

Crossover: One point crossover is applied.

Mutation: Each symbol (terminal pointer, function, function pointer) in the chromosome may be target of the mutation operator. Some symbols in the chromosome are changed by mutation according to the predefined mutation probability. Random mutation point(s) within the chromosome is determined. If it is a terminal gene, then the terminal pointers are replaced by another relational operator and the attribute value

Table 1
Function and terminal sets

X_i	ith attribute
Relational operator	Type of attribute
$=$	Categorical attributes
$\leqslant, \geqslant$	Continues attributes
V_{xi}	Domain of ith attribute
Terminal set	$\{x_0 - \text{RO} - V_{xo}, x_1 - \text{RO} - V_{x1}, \ldots, x_n - \text{RO} - V_{xn}\}$
Function set	{AND, OR, NOT}

is modified according to the domain range. If the mutation point is a function gene, then logical function is replaced by another logical function. The pointers of mutated logical function, which point to the preceding genes, are reassigned.

4. Reduced MEPAR-miner Algorithm

Our approach of rough sets and multi-expression programming for mining classification rules consist of two major phases. The flow chart of this stage can be seen in Fig. 1.

4.1. Preprocessing stage

In this stage two main step is employed. These are discretization of Continuous Attributes and GA-based Attribute Reduction by Rough Sets as follows:

4.1.1. Discretization of continuous attributes

The continuous attributes in data sets are discretized by using Fayyad and Irani's (1993) entropy based discretization method (MDL) [16].

4.1.2. GA-based attribute reduction by rough sets

After discretization operation, in order to find attribute reducts, ROSETTA [17] which is used as a very powerful toolkit for rough set approach problems is used. In ROSETTA reduct sets determined by choosing GA based reducts and selector method.

An information table is sent to the integrated system for the GA-based reducts and selector. A rough set based software ROSETTA, developed by a team at the Norwegian University of Science and Technology, is employed to reduce the input attribute set and conduct the optimization operation of GA.

Implementing a genetic algorithm for computing minimal hitting sets as described by Vinterbo and Øhrn [18], the algorithm has support for both cost information and approximate solutions.

The algorithm's fitness function f is defined below, where δ is the set of sets corresponding to the discernibility function. The parameter α defines a weighting between subset cost and hitting fraction while ε is relevant in the case of approximate solution (see ROSETTA user's manual [17]):

$$f(B) = (1-\alpha) \times \frac{\cos t(A) - \cos t(B)}{\cos t(A)} + \alpha \times \min\left\{\varepsilon, \frac{|[S \text{ in } \delta | S \cap B \neq \emptyset]|}{|\delta|}\right\}.$$

The subsets B of A found through an evolutionary search driven by the fitness function and that are "good enough" hitting sets, i.e., that have a hitting fraction of at least ε, are collected in a "keep list". The size of the keep list $k$can be specified. Approximate solutions are controlled through two parameters, ε and k. The parameter ε signifies a minimal value for the hitting fraction while $k$denotes the number of extra keep lists in use by the algorithm. Each reduct in the returned reduct set has a support count associated with it. The support count is a measure of the "strength" of the reduct and might be evaluated by reduct selector. After conducting all fitness function evaluation iterations, the minimal subset of reducts extracted from the data set with respect to the target object can be examined.

The parameter defines a weight between the subset costs and hitting fraction is set at 0.4 in the case of approximate solutions. Other genetic operating parameters are set as follows:

Population size	70
Length of Chromosome	22
Size of keep list	256
Selection operator	Elitism
Crossover operator	Single point operator
Crossover probability	0.3
Mutation probability	0.05
Fitness parameter	Discernibility function

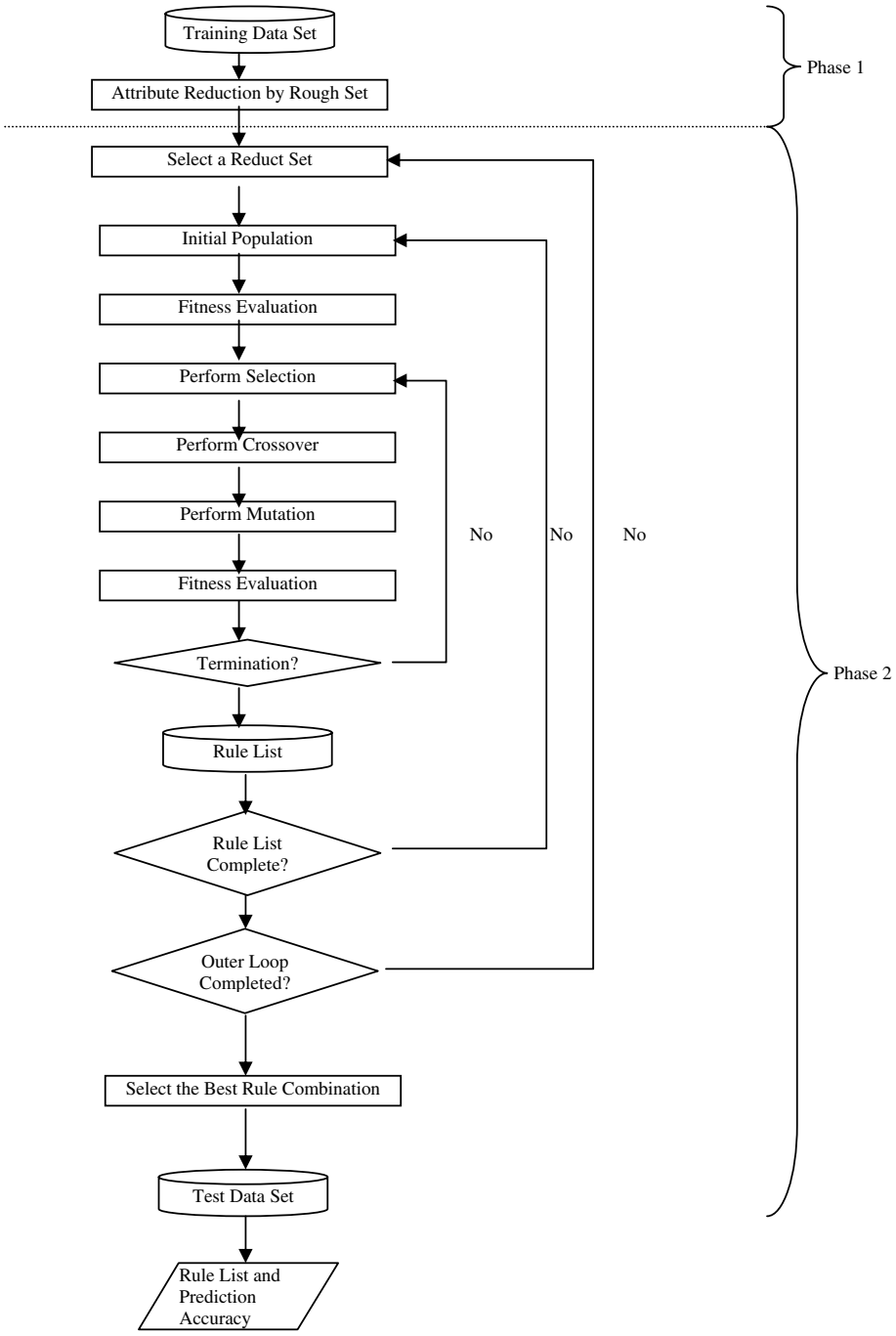

Fig. 1. Flowchart of the algorithm.

4.2. Extract classification rules stage

In this stage parallel steady state genetic algorithm (pSSGA) is used for extract classification rules. The reducts produced by ROSETTA are used as inputs into the Reduced MEPAR-miner Algorithm. The chromosome representation is modified according to reducts and the algorithmic structure of MEPAR-miner is improved by introducing a new operators and default class structure. Sample chromosome structure can be seen in Fig. 2.

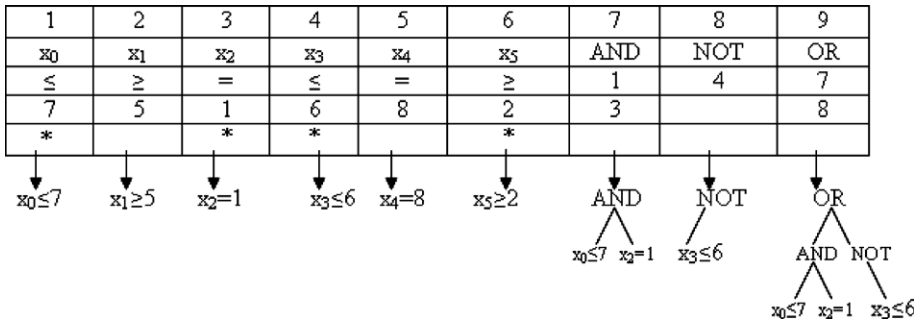

Fig. 2. Sample chromosome structure.

Default class structure: In this study, a new default class structure is also proposed. If we calculate the misidentified parts for each class by using the training data and we choose the class, which has the maximum misidentified as the default class, we can identify maximum amount of test data correctly. This follows:

$$\text{Default class} = \max_i(FN_i), \quad i = \text{number of class}.$$

Function and terminal sets: As first part of the chromosome, each terminal gene is assigned a attribute randomly. After the attributes are assigned, the relational operators are set according to the type of the quality (categorical or continues). After terminal genes are formed, the genes that have the attributes randomly selected from the reduct set are marked.

The function set, which is the second part of the chromosome is formed by logical operators and marked genes.

For each gene in the chromosome a fitness value is calculated for all different classes. The class with the highest fitness value is assigned as the label of that gene. After fitness evaluation, each gene has its own fitness value and associated class label. Highest fitness value and its label represent the fitness value and the class of the chromosome.

In rule classifier systems, there are two distinct approaches to individual or particle representation: the Michigan and the Pittsburgh approaches [19]. In the Michigan approach, each individual encodes a single rule whereas in the Pittsburgh approach each individual encodes a set of rules. In this work, Michigan coding approach is used.

Fitness function: The rule evaluation function must not only consider instances correctly classified but also the one left to classify and the wrongly classified ones. This is why the following four possible concepts are taken into consideration [20].

- True Positive (TP): The number of instances covered by the rule that are correctly classified, i.e., its class matches the training target class.
- False Positive (FP): The number of instances covered by the rule that are wrongly classified, i.e., its class differs from the training target class.
- True Negative (TN): The number of instances not covered by the rule, whose class differs from the training target class.
- False Negative (FN): The number of instances not covered by the rule, whose class matches the training target class.

Sensitivity (S_e) measures the fraction of actual positive examples that are correctly classified:

$$S_e = \text{TP}/(\text{TP} + \text{FN}). \tag{1}$$

Specificity (S_p) measures the fraction of actual negative examples that are correctly classified:

$$S_p = \text{TN}/(\text{TN} + \text{FP}). \tag{2}$$

By using these concepts, the fitness function is defined as follows [20]:

$$\text{Fitness} = S_e \times S_p. \tag{3}$$

The value of the fitness function is in the range of 0–1. The fitness value is 1 when all of the instances are correctly classified by the rule.

Genetic operators

Algorithm type: In this study, Steady State Genetic Algorithm (SSGA) is used. SSGA was introduced by Whitley and Kauth in 1988 [21]. It is different from the generational model in that there is typically one single new member inserted into the new population and generally the worst chromosome is removed from the population.

Selection mechanism: Binary tournament selection procedure is used for evolution where two individuals are selected randomly from the current population and used for crossover and mutation operators.

Crossover: Two parents are selected and recombined according to the predefined crossover probability (p_m) during crossover. In this work, one point crossover is applied. Two parent chromosomes are randomly selected in the mating pool. A crossover point is randomly determined to perform the recombination process as shown in Fig. 3. The first part of parent-1 and the second part of parent-2 are recombined to produce two offspring and one of them is selected randomly.

In Fig. 3, the crossover point is selected as the 4th position of the parent chromosomes which is shown by a thick line.

After one point crossover, two offspring are obtained as shown in Fig. 4.

Mutation: Each symbol (terminal pointer, function, function pointer) in the chromosome may be target of the mutation operator. By mutation, some symbols in the chromosome are changed according to the predefined mutation probability. Random mutation point(s) within the chromosome is determined. If it is a terminal gene and marked gene formed by the attributes within the reduct set, then the terminal pointers are

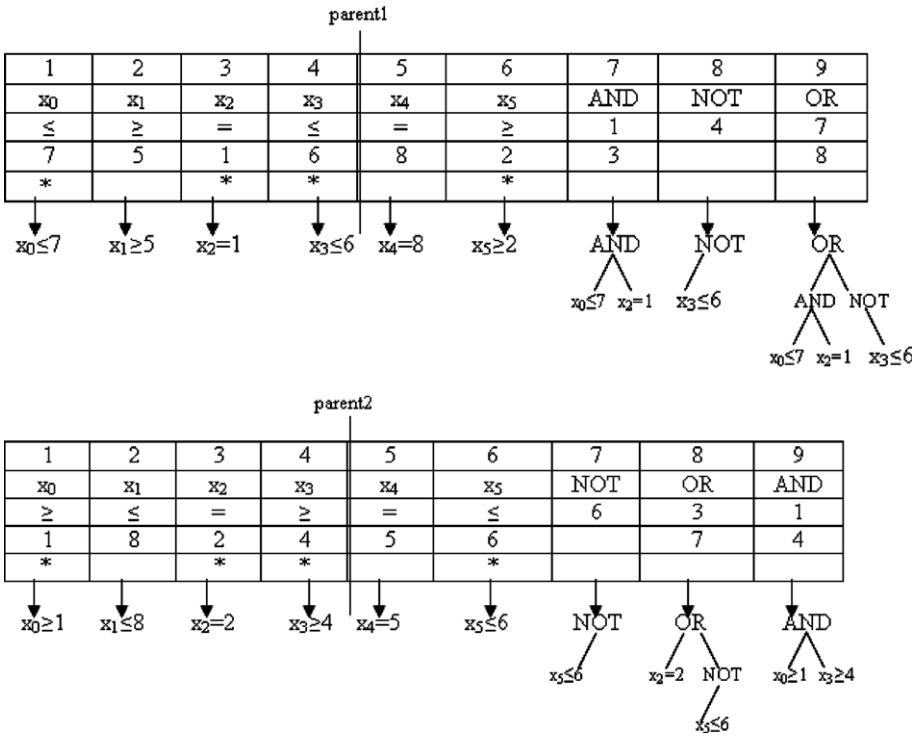

Fig. 3. Parent chromosomes before crossover.

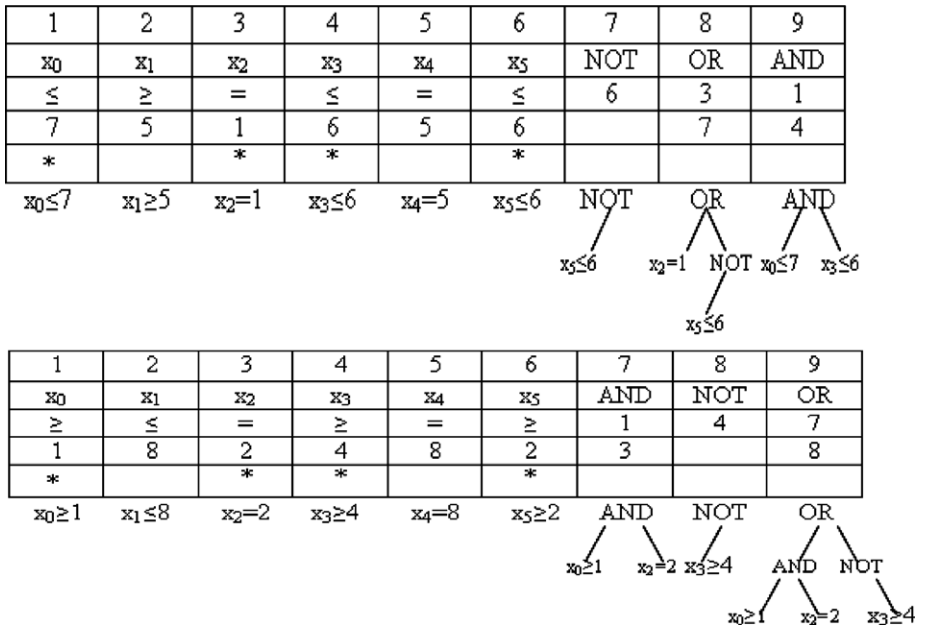

Fig. 4. Offspring after crossover.

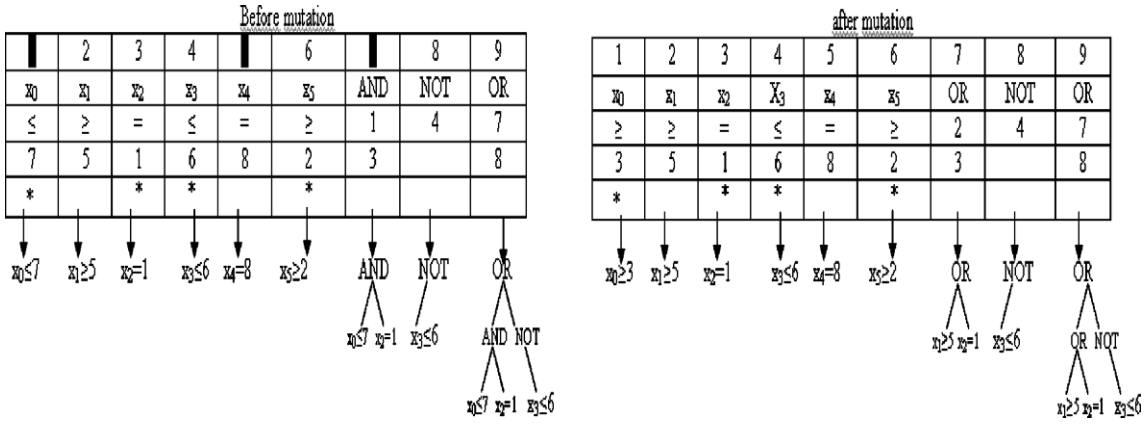

Fig. 5. Chromosome before and after mutation operation.

replaced by another relational operator and the attribute value is modified accordingly to be within the domain range. If the terminal gene chosen is not a marked gene, it is not mutated and taken as it is because it won't have any effects on the calculations. If the mutation point is a function gene, then the logical function is replaced by another logical function. The pointers of mutated logical function, which point to the preceding genes, are reassigned (Fig. 5).

5. Experimental study

The datasets used to verify the usefulness of the presented methodology are obtained from the UCI Machine Learning Repository [22]. The results of experiments we reported on the following datasets: Wisconsin Breast Cancer Data Set, Ljubljana Breast Cancer Data Set, Tic-Tac-Toe Data Set, CRX Data Set, Nursery Data Set, Cleveland Data Set, Iris Data Set, Lymphography Data Set.

Wisconsin Breast Cancer Data Set (WBCD): The WBCD data set consists of 699 instances. Each instance consists of 9 continuous attributes. The measurements of attributes are assigned an integer value between 1 and 10. The data set contains 16 instances with missing values. Because of the small number of missing data, these cases are discarded from data set and remaining 683 cases are used.

Ljubljana Breast Cancer Data Set: Ljubljana breast cancer data set consists of 286 instances. Each instance consists of 9 categorical attributes. The data set contains 9 instances with missing values. Because of the small number of missing data, these cases are discarded from data set and the remaining 277 cases are used.

Tic-Tac-Toe Data Set: This database encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "win for x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row"). The data set consists of 958 instances and contains 9 categorical attributes, each corresponding to one tic-tac-toe square. There is not any missing attribute value in the data set.

CRX Data Set: CRX data set contains credit card applications. CRX data set is studied because it has a mix of attributes; continues, nominal with small number of values and nominal with larger number of values. The data set consists of 690 instances and 15 attributes. There are 37 cases with missing attribute values. In this study, if an attribute value is missing in any case, the whole case is not omitted but only the missing attribute for which the value is missing.

Nursery Data Set: Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. The data set consists of 12,960 instances and 8 attributes. All of the attributes are categorical. There are no missing attribute values.

Cleveland Heart Disease Data Set: Cleveland heart disease data set consists of 303 instances. This database contains 75 attributes but all published experiments refer to use a subset of 13 of them. There are 6 missing attribute values. In this study, if an attribute value is missing in any case, the whole case is not omitted but only the attribute for which the value is missing.

Lymphography Data Set: Lymphography data set consists of 148 instances. Each instance consists of 18 attributes. There is not any missing attribute value in the data set.

Iris Data Set: The data set consists of 150 instances and 4 attributes. All of the attributes are continuous. There are no missing attribute values.

The main characteristics of the data sets are summarized in Table 2.

Table 2
Main characteristics of the data sets

Data set	#cases	#categorical attributes	#continuous attributes	#classes
Wisconsin breast cancer (WBCD)	683	–	9	2
Ljubljana breast cancer (LBCD)	282	9	–	2
Tic-Tac-Toe	958	9	–	2
CRX	690	9	6	2
Nursery	12,960	8	–	5
Cleveland heart disease	303	8	5	5
Iris	150	–	4	3
Lymphography	148	18	–	4

Table 3
Predictive accuracies of Reduced MEPAR-miner Algorithm

Predictive accuracy (%)	Data sets							
	CRX	Nursery	Iris	Ljubljana BC	Tic-Tac-Toe	Wisconsin BC	Cleveland HD	Lymphography
Max	1	1	1	1	0.94791	1	0.93333	1
Average	**0.97815**	**0.99560**	**0.97333**	**0.92857**	**0.91542**	**0.99705**	**0.91616**	**0.93794**
Min	0.95588	0.98611	0.93333	0.89285	0.875	0.98529	0.87878	0.76666
Standard deviation	1.71	0.52	5.44	4.45	2.53	0.62	3.96	8.16

The comparative study is carried out across the predictive accuracies. The predictive accuracy of the classifier measures the proportion of correctly classified instances [23]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

In data sets, predictive accuracy is measured by a well-known 10-fold cross-validation procedure. Each data set is divided into 10 partitions and the algorithm is run once for each partition. Each time, a different partition is used as test set and the other 9 partitions are grouped together to build training set. The predictive accuracies on the test set of the 10 runs are averaged and reported as the predictive accuracies. Also standard deviations of the corresponding predictive accuracies are calculated.

In Table 3, 10-fold cross-validation results are summarized. Average, maximum and minimum predictive accuracies on the test data sets and standard deviations are reported.

5.1. Experimental setup

In the algorithm presented, parameter settings are set as follows:

Algorithm type	Steady state genetic algorithm
Selection mechanism	Binary tournament selection
Population size	50
Length of chromosome	25
Crossover operator	One point operator
Crossover probability	0.75
Mutation probability	0.30
Number of generation	100
Number of inner loop	10

5.2. Performance comparisons

5.2.1. Comparison with classical machine learning algorithms

Two rule-based machine-learning algorithms C4.5 rules [24], i.e., J48 in WEKA [25], and PART [26] as well as a statistical classifier Naïve Bayes [27] are also applied to the datasets for comparison. In order to being able to compare results under same circumstances 10-fold cross validation is applied to data sets. Ten runs are carried out for each data set. Minimum, maximum, average accuracies and standard deviation for test data are presented in Table 4.

Seven out of eight data sets, the performance of our algorithm exceeds the current ones. Only in the Tic-Tac-Toe data set, the standard deviation of our algorithm is lower than the PART algorithm while our average accuracy value is slightly lower than PART algorithm value.

5.2.2. Comparison with several other rule-based classifiers

The performance of this algorithm is compared with several other rule-based classifiers from the recent literature. The results of comparison are shown in Table 5.

As shown in Table 5, Reduced MEPAR-miner Algorithm discovered rules with better predictive accuracies than the other algorithms in seven data sets namely Ljubljana breast cancer, Wisconsin breast cancer, CRX, Nursery, Cleveland heart disease, Iris, Lymphography data sets. In Tic-Tac-Toe data set, the predictive accuracy of Reduced MEPAR-miner Algorithm is slightly worse than CN2 and MEPAR-miner. Predictive accuracy of Reduced MEPAR-miner Algorithm is more than 91%, which is generally not the case for the other compared algorithms.

Table 4
Reduced MEPAR-miner algorithm comparative results

Data Set		NaiveBayes	PART	C4.5	Reduced MEPAR-miner
Ljubljana breast cancer	Average	72.69704	69.4064	74.28079	**92.857**
	Standard deviation	7.737207	7.628285	6.050664	4.45
	Minimum	51.72414	48.27586	53.57143	89.285
	Maximum	89.65517	86.2069	86.2069	100
Iris	Average	94	94.67	93.33	**97.333**
	Standard deviation	4.92	6.13	5.44	5.44
	Minimum	86.67	86.67	86.67	93.333
	Maximum	100	100	100	100
Wisconsin breast cancer	Average	97.19648	94.69462	95.00828	**99.705**
	Standard deviation	1.711341	2.514011	2.730619	0.62
	Minimum	92.85714	88.57143	87.14286	98.529
	Maximum	100	100	100	100
Nursery	Average	90.50021	98.66554	96.21913	**99.560**
	Standard deviation	0.389248	0.354198	0.291374	0.52
	Minimum	90.04083	97.91146	95.82293	98.611
	Maximum	91.17113	99.13774	96.64094	100
Tic-Tac-Toe	Average	69.64232	**93.85296**	85.28103	91.542
	Standard deviation	4.402807	3.076832	3.184181	2.53
	Minimum	58.94737	83.33333	75.78947	87.5
	Maximum	82.10526	100	93.75	94.791
Cleveland heart disease	Average	56.38065	51.40215	52.05914	**91.616**
	Standard deviation	7.13104	7.636893	6.688745	3.96
	Minimum	40	25.80645	33.33333	87.878
	Maximum	70	70	66.66667	93.333
Lymphography	Average	84.26	80.26	78.15	**93.794**
	Standard deviation	9.03	8.49	10.63	8.16
	Minimum	69.23	69.23	61.54	76.666
	Maximum	100	93.33	100	100
CRX	Average	77.85507	84.44928	85.56522	**97.815**
	Standard deviation	4.181567	4.349558	3.95653	1.71
	Minimum	66.66667	73.91304	73.91304	95.588
	Maximum	86.95652	94.2029	92.75362	100

6. Conclusion

Based on rough set theory, we present a new algorithm that provides an efficient and robust mechanism for the classification of data mining problems.

Reduced MEPAR-miner Algorithm ensures the calculation of the optimal sets of attributes and uses fixed length linear strings of chromosomes to represent logical expression trees of different shapes and sizes. Reduced MEPAR-miner Algorithm introduces the combination of rough set theory, genetic algorithms and genetic programming advantages based on this feature.

The proposed Reduced MEPAR-miner Algorithm has been validated upon eight datasets obtained from the UCI Machine Learning Repository. The performance of the proposed algorithm is compared in two fold. Firstly, rule-based machine-learning algorithms and a statistical classifier are used for comparison. According to the comparison, it is shown that proposed algorithm outperforms to other algorithms except to one dataset (Tic-tac-toe). Secondly, our algorithm is compared with several other rule-based classifiers from the recent literature. Similarly, Reduced MEPAR-miner Algorithm gives good results except one dataset (Tic-tac-toe). Comparisonal results show that the proposed method produces comprehensible classification rules with good predicting accuracies for the datasets, which are competitive as compared to existing classifiers in literature.

Table 5
Comparison of predictive accuracies of data mining algorithms

Data set	Classifier	Accuracy
Ljubljana BC	Ant-Miner [28]	75.28 ± 2.24
	CN2 [28]	67.69 ± 3.59
	MEPAR-miner [12]	90.63 ± 4.48
	Reduced MEPAR-miner Algorithm	$\mathbf{92.857 \pm 4.45}$
Wisconsin BC	Ant-Miner [28]	96.04 ± 0.93
	CN2 [28]	94.88 ± 0.88
	MEPAR-miner [12]	99.41 ± 0.76
	Reduced MEPAR-miner Algorithm	$\mathbf{99.705 \pm 0.62}$
Tic-Tac-Toe	Ant-Miner [28]	73.04 ± 2.53
	CN2 [28]	$\mathbf{97.38 \pm 0.52}$
	MEPAR-miner [12]	94.47 ± 1.31
	Reduced MEPAR-miner Algorithm	91.542 ± 2.53
Cleveland HD	Ant-Miner [28]	59.67 ± 2.50
	CN2 [28]	57.48 ± 1.78
	MEPAR-miner [12]	87.78 ± 3.51
	Reduced MEPAR-miner Algorithm	$\mathbf{91.616 \pm 3.96}$
CRX	C4.5 [29]	91.79 ± 2.1
	Double C4.5 [29]	90.78 ± 1.2
	C4.5/AG [29]	91.66 ± 1.8
	MEPAR-miner [12]	96.96 ± 2.50
	Reduced MEPAR-miner Algorithm	$\mathbf{97.815 \pm 1.71}$
Nursery	C4.5 [29]	95.4 ± 1.2
	Double C4.5 [29]	97.23 ± 1.0
	C4.5/AG [29]	96.77 ± 0.7
	MEPAR-miner [12]	95.83 ± 1.80
	Reduced MEPAR-miner Algorithm	$\mathbf{99.560 \pm 0.52}$
Iris	DCC [30]	96.73
	GP-Co [30]	95.3
	GGP [30]	91.04
	Reduced MEPAR-miner Algorithm	$\mathbf{97.333 \pm 5.44}$
Lymphography	CN2 [31]	81.6
	MLP [31]	81.6
	DIMLP [31]	80.4
	SIM [31]	86.2
	Reduced MEPAR-miner Algorithm	$\mathbf{93.794 \pm 6.1}$

References

[1] K. Cios, W. Pedrycz, R. Swiniarski, Data Mining Methods for Knowledge Discovery, vol. 33, ABD, Kluwer Academic Publishers, 2000.
[2] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences (2006).
[3] T.R. Babu, M. Murty, V.K. Agrawal, Hybrid learning scheme for data mining applications, in: HIS'04: 4th International Conference on Hybrid Intelligent Systems, 2005, pp. 266–271.
[4] Yasser Hassan, E. Tazaki, Induction of knowledge using evolutionary rough set theory, Cybernetics and Systems: An International Journal (2003).
[5] Renpu Li, Zheng-ou Wang, Mining classification rules using rough sets and neural networks, Computing, Artificial Intelligence and Information Technology (2003).
[6] S. Jaroslaw, K. Kierzkowska, Hybrid Classifier Based on Rough Sets and Neural Networks (2003). http://www.elsevier.nl/locate/entcs/volume82.
[7] Yasser Hassan, E. Tazaki, Shin Egava, Kazuho Suyama, Rough Neural Classifier System, IEEE, 2002.
[8] K. Pal Sankar, P. Mitra, Case generation using rough sets with fuzzy representation, IEEE Transactions on Knowledge and Data Engineering 16 (2004) 292–300.
[9] C. Wong, Lin Bo-Chen, C. Chen, Fuzzy rules extraction by a hybrid method for pattern classification, in: Annual Conference of the North American Fuzzy Information Processing Society, vol. 3, 2001, pp. 1798–1803.

[10] C.L. Huang, T.S. Li, T.K. Peng, A hybrid approach of rough set theory and genetic algorithm for fault diagnosis, International Journal of Advanced Manufacturing Technology (2005) 119–127.
[11] G. Zhang, Z. Cao, Y. Gu, A hybrid classifier based on rough set theory and support vector machines, Lecture Notes in Computer Science (2006) 1287–1296.
[12] A. Baykasoğlu, L. Özbakir, MEPAR-miner: multi-expression programming for classification rule mining, European Journal of Operational Research, 2006, corrected proof, available from: <www.sciencedirect.com>.
[13] Z. Pawlak, Rough Sets Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, 1991.
[14] M. Oltean, D. Dumitrescu, Multi Expression Programming, Technical report, UBB-01-2002, Babeş-Bolyai University, Cluj-Napoca, Romania, 2002, available from: <www.mep.cs.ubbcluj.ro>.
[15] M. Oltean, C. Grosan, Evolving digital circuits using multi expression programming, in: R. Zebulum et al. (Eds.), NASA/DoD Conference on Evolvable Hardware, 24–26 June, Seattle, IEEE Press, NJ, 2004, pp. 87–90.
[16] U. Fayyad, K.B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, in: Proc. of 13th International Joint Conference on Artificial Intelligence, 1993, pp. 1022–1027.
[17] Norwegian University of Science and Technology (2001) ROSETTA version 1.4.4.1. Available from: <www.idi.ntnu.no/~alex/rosetta>.
[18] S. Vinterbo, A. Øhrn, Minimal approximate hitting sets and rule templates, International Journal of Approximate Reasoning 25 (2) (2000) 123–143.
[19] A. Freitas, Survey of evolutionary algorithms for data mining and knowledge discovery, in: A. Ghosh, S. Tsutsui (Eds.), Advances in Evolutionary Computation, Springer-Verlag, 2001.
[20] R.S. Parpinelli, H.S. Lopes, A.A. Freitas, An ant colony based system for data mining: applications to medical data, in: Proc. Genetic and Evolutionary Computation Conf. (GECCO-2001), Morgan Kaufmann, San Francisco, California, 2001, pp. 791–798.
[21] D. Whitley, J. Kauth, Genitor: a different genetic algorithm, in: Proc. Rocky Mountain Conf. Artificial Intelligence, Denver, CO, 1988, p. 118.
[22] UCI (University of California at Irvine) Machine Learning Repository. Available from: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
[23] A.C. Tan, D. Gilbert, An empirical comparison of supervised machine learning techniques in bioinformatics, in: Proceedings of the First Asia Pacific Bioinformatics Conference, 2003.
[24] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman, San Mateo, CA, 1993.
[25] H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques With Java Implementations, Morgan Kaufman, San Mateo, CA, 1999.
[26] E. Frank, I.H. Witten, Generating accurate rule sets without global optimization, in: Proc. 15th Int. Conf. Machine Learning (ICML'98), 1998, pp. 144–151.
[27] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: Proc. 11th Conf. Uncertainty in Artificial Intelligence, San Mateo, CA, 1995, pp. 338–345.
[28] R.S. Parpinelli, H.S. Lopes, A.A. Freitas, Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computation 6 (4) (2002) 321–332.
[29] D.R. Carvalho, A.A. Freitas, New results for a hybrid decision tree/genetic algorithm for data mining, in: J. Garibaldi (Ed.), Proc. 4th Int. Conf. on Recent Advances in Soft Computing (RASC-2002), Nottingham Trent University, 2002, pp. 260–265.
[30] K.C. Tan, Q. Yu, T.H. Lee, A distributed evolutionary classifier for knowledge discovery in data mining, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and reviews 35 (2) (2005).
[31] P. Luukka, Similarity classifier using measure derived from Yu's norms applied to medical data sets, in: IEEE International Conference on Fuzzy Systems, Vancouver, Canada, 2006.