Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 339–346

ISBN 978-83-60810-51-4

Sequential Optimization of γ-Decision Rules

Beata Zielosko
Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
Email: beata.zielosko@kaust.edu.sa
Institute of Computer Science
University of Silesia
39, Będzińska St., 41-200 Sosnowiec, Poland

***Abstract*—The paper is devoted to the study of an extension of dynamic programming approach which allows sequential optimization of approximate decision rules relative to length, coverage and number of misclassifications. Presented algorithm constructs a directed acyclic graph $\Delta_\gamma(T)$ which nodes are subtables of the decision table T. Based on the graph $\Delta_\gamma(T)$ we can describe all irredundant γ-decision rules with minimum length, after that among these rules describe all rules with maximum coverage, and among such rules describe all rules with minimum number of misclassifications. We can also change the set of cost functions and order of optimization. Sequential optimization can be considered as tool that help to construct simpler rules for understanding and interpreting by experts.**

I. Introduction

DECISION rules are one of popular ways for data representation used in machine learning and knowledge discovery. Exact decision rules can be overfitted, i.e., dependent essentially on the noise or adjusted too much to the existing examples. If decision rules are considered as a way of knowledge representation then instead of exact decision rules with many attributes, it is more appropriate to work with approximate decision rules which contain smaller number of attributes and have relatively good accuracy. Moreover, approximate decision rules often give better accuracy during classification process than exact decision rules. Therefore, approximate decision rules and also closely connected with them approximate reducts are studied intensively last years by H.S. Nguyen, Z. Pawlak, A. Skowron, D. Ślęzak and others [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

There are many approaches to the construction of decision rules and reducts: brute-force approach which is applicable to tables with relatively small number of attributes, genetic algorithms [13], [14], Apriori algorithm [15], simulated annealing [16], Boolean reasoning [17], [18], [19], separate-and-conquer approach (algorithms based on a sequential covering procedure) [5], [20], [21], [22], [23], [24], [25], [26], ant colony optimization [27], algorithms based on decision tree construction [7], [28], [29], [30], different kinds of greedy algorithms [6], [17]. Each method has different modifications, e.g., in the case of decision trees, we can use greedy algorithms based on different uncertainty measures (Gini index, entropy, etc.) for construction of decision rules.

In this paper, we present one more approach based on an extension of dynamic programming. We introduce an uncertainty measure that is the difference between number of rows in a given decision table and the number of rows labeled with the most common decision in this table. We fix a nonnegative threshold γ, and study so-called γ-decision rules that localize rows in subtables which uncertainty is at most γ. For each of such rules the number of misclassifications is at most γ.

We consider three cost functions: length, coverage and number of misclassifications. The choice of length is connected with the Minimum Description Length principle [31]. The rule coverage is important to discover major patterns in the data. Number of misclassifications is important from the viewpoint of accuracy of classification. Our approach allows sequential optimization of γ-decision rules relative to the mentioned cost functions.

Sequential optimization can be considered as tool whose help to construct rules which are simpler for understanding and interpreting by experts, e.g., among rules with maximum coverage we can find rules with minimum length. Such rules can be considered as part of knowledge and experts can easier analyze them.

Sequential optimization allows to find optimal rules relative to the considered cost functions, e.g., rules with minimum length and maximum coverage. This process can be seen as postprocessing of rules which helps to design classifiers. In case of totally optimal rules relative to the considered cost functions results of sequential optimization do not depend on the order of optimization. Besides, sequential optimization of system of decision rules also can help to discover some regularities or anomalies in data.

First results for decision rules based on dynamic programming approach were obtained in [32]. The aim of this study was to find one decision rule with minimum length for each row. In [33] we studied dynamic programming approach for exact decision rule optimization. In unpublished [34] we studied dynamic programming approach for approximate decision rule optimization and we used another uncertainty measure which is the number of unordered pairs of rows with different decisions in decision table T. In press [35] we presented procedures of optimization of irredundant γ-decision rules relative to the length and coverage, and in

press [36] – relative to the number of misclassifications. In this paper, we concentrate on sequential optimization of γ-decision rules relative to the length, coverage and number of misclassifications.

We present also results of experiments with some decision tables from UCI Machine Learning Repository [37] based on Dagger software system [38] created in King Abdullah University of Science and Technology (KAUST).

This paper consists of seven sections. Section II contains definitions of main notions. In Sect. III, we study a directed acyclic graph which allows to describe the whole set of irredundant γ-decision rules. In Sect. IV, we describe procedures of optimization of irredundant γ-decision rules relative to the length, coverage and number of misclassifications. In Sect. V, we discuss possibilities of sequential optimization of rules relative to a number of cost functions. Section VI contains results of experiments with decision tables from UCI Machine Learning Repository. Section VII contains conclusions.

II. Main Notions

In this section, we consider definitions of notions corresponding to decision table and decision rules.

A *decision table* T is a rectangular table with n columns labeled with conditional attributes $f_1, \ldots, f_n$. Rows of this table are filled by nonnegative integers which are interpreted as values of conditional attributes. Rows of T are pairwise different and each row is labeled with a nonnegative integer which is interpreted as a value of the decision attribute. It is possible that T is empty, i.e., has no rows.

A minimum decision value which is attached to the maximum number of rows in T will be called the *most common decision for* T. The most common decision for empty table is equal to 0.

We denote by $N(T)$ the number of rows in the table T and by $N_{mcd}(T)$ we denote the number of rows in the table T labeled with the most common decision for T. We will interpret the value $J(T) = N(T) - N_{mcd}(T)$ as *uncertainty* of the table T.

The table T is called *degenerate* if T is empty or all rows of T are labeled with the same decision. It is clear that $J(T) = 0$ if and only if T is a degenerate table.

A table obtained from T by the removal of some rows is called a *subtable* of the table T. Let T be nonempty, $f_{i_1}, \ldots, f_{i_k} \in \{f_1, \ldots, f_n\}$ and $a_1, \ldots, a_k$ be nonnegative integers. We denote by $T(f_{i_1}, a_1) \ldots (f_{i_k}, a_k)$ the subtable of the table T which contains only rows that have numbers $a_1, \ldots, a_k$ at the intersection with columns $f_{i_1}, \ldots, f_{i_k}$. Such nonempty subtables (including the table T) are called *separable subtables* of T.

We denote by $E(T)$ the set of attributes from $\{f_1, \ldots, f_n\}$ which are not constant on T. For any $f_i \in E(T)$, we denote by $E(T, f_i)$ the set of values of the attribute f_i in T.

The expression

$$f_{i_1} = a_1 \wedge \ldots \wedge f_{i_k} = a_k \rightarrow d \qquad (1)$$

is called a *decision rule over* T if $f_{i_1}, \ldots, f_{i_k} \in \{f_1, \ldots, f_n\}$, and $a_1, \ldots a_k, d$ are nonnegative integers. It is possible that $k = 0$. In this case (1) is equal to the rule

$$\rightarrow d. \qquad (2)$$

Let $r = (b_1, \ldots, b_n)$ be a row of T. We will say that the rule (1) is *realizable for* r, if $a_1 = b_{i_1}, \ldots, a_k = b_{i_k}$. If $k = 0$ then the rule (2) is realizable for any row from T.

Let γ be a nonnegative integer. We will say that the rule (1) is a *γ-true for* T if d is the most common decision for $T' = T(f_{i_1}, a_1) \ldots (f_{i_k}, a_k)$ and $J(T') \leq \gamma$. If $k = 0$ then the rule (2) is a γ-true for T if d is the most common decision for T and $J(T) \leq \gamma$.

If the rule (1) is a γ-true for T and realizable for r, we will say that (1) is a *γ-decision rule for T and r*. Note that if $\gamma = 0$ we have an exact decision rule for T and r.

We will say that the rule (1) with $k > 0$ is an *irredundant* γ-decision rule for T and r if (1) is a γ-decision rule for T and r and the following conditions hold:

(i) $f_{i_1} \in E(T)$, and if $k > 1$ then $f_{i_j} \in E(T(f_{i_1}, a_1) \ldots (f_{i_{j-1}}, a_{j-1}))$ for $j = 2, \ldots, k$;

(ii) $J(T) > \gamma$, and if $k > 1$ then $J(T(f_{i_1}, a_1) \ldots (f_{i_j}, a_j)) > \gamma$ for $j = 1, \ldots, k-1$.

If $k = 0$ then the rule (2) is an *irredundant* γ-decision rule for T and r if (2) is a γ-decision rule for T and r, i.e., if d is the most common decision for T and $J(T) \leq \gamma$.

Let τ be a decision rule over T and τ be equal to (1).

The number k of conditions on the left-hand side of τ is called the *length* of this rule and is denoted by $l(\tau)$. The length of decision rule (2) is equal to 0.

The *coverage* of τ is the number of rows in T for which τ is realizable and which are labeled with the decision d. We denote it by $c(\tau)$. The coverage of decision rule (2) is equal to the number of rows in T which are labeled with the decision d.

The *number of misclassifications* of τ is the number of rows in T for which τ is realizable and which are labeled with decisions different from d. We denote it by $\mu(\tau)$. The number of misclassifications of the decision rule (2) is equal to the number of rows in T which are labeled with decisions different from d.

Proposition 1: in press [35] Let T be a nonempty decision table, r be a row of T and τ be a γ-decision rule for T and r which is not irredundant. Then by removal of some conditions from the left-hand side of τ and by changing the decision on the right-hand side of τ we can obtain an irredundant γ-decision rule $irr(\tau)$ for T and r such that $l(irr(\tau)) \leq l(\tau)$ and $c(irr(\tau)) \geq c(\tau)$.

Unfortunately, it is impossible to prove similar result for the number of misclassifications.

III. Directed Acyclic Graph $\Delta_\gamma(T)$

In this section, we present an algorithm that constructs a directed acyclic graph $\Delta_\gamma(T)$. Based on this graph we can describe the set of irredundant γ-decision rules for T and for each row r of T. Nodes of the graph are separable subtables of the table T. During each step, the algorithm processes one

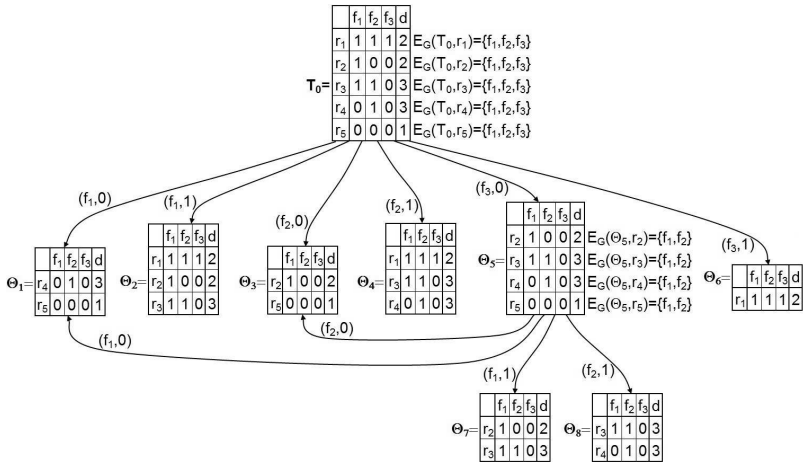

Fig. 1. Directed acyclic graph $G = \Delta_1(T_0)$

node and marks it with the symbol *. At the first step, the algorithm constructs a graph containing a single node T which is not marked with the symbol *.

Let the algorithm have already performed p steps. Let us describe the step $(p+1)$. If all nodes are marked with the symbol * as processed, the algorithm finishes its work and presents the resulting graph as $\Delta_\gamma(T)$. Otherwise, choose a node (table) Θ, which has not been processed yet. Let d be the most common decision for Θ. If $J(\Theta) \leq \gamma$ label the considered node with the decision d, mark it with symbol * and proceed to the step $(p+2)$. If $J(\Theta) > \gamma$, for each $f_i \in E(\Theta)$, draw a bundle of edges from the node Θ. Let $E(\Theta, f_i) = \{b_1, \ldots, b_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, b_1), \ldots, (f_i, b_t)$ respectively. These edges enter to nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$. If some of nodes $\Theta(f_i, b_1), \ldots, \Theta(f_i, b_t)$ are absent in the graph then add these nodes to the graph. We label each row r of Θ with the set of attributes $E_{\Delta_\gamma(T)}(\Theta, r) = E(\Theta)$. Mark the node Θ with the symbol * and proceed to the step $(p+2)$. The graph $\Delta_\gamma(T)$ is a directed acyclic graph. A node of such graph will be called *terminal* if there are no edges leaving this node. Note that a node Θ of $\Delta_\gamma(T)$ is terminal if and only if $J(\Theta) \leq \gamma$.

Later, we will describe the procedures of optimization of the graph $\Delta_\gamma(T)$. As a result we will obtain a graph G with the same sets of nodes and edges as in $\Delta_\gamma(T)$. The only difference is that any row r of each nonterminal node Θ of G is labeled with a nonempty set of attributes $E_G(\Theta, r) \subseteq E(\Theta)$. It is possible also that $G = \Delta_\gamma(T)$.

Now, for each node Θ of G and for each row r of Θ, we describe the set of γ-decision rules $Rul_G(\Theta, r)$. We will move from terminal nodes of G to the node T.

Let Θ be a terminal node of G labeled with the most common decision d for Θ. Then

$$Rul_G(\Theta, r) = \{\rightarrow d\}.$$

Let now Θ be a nonterminal node of G such that for each child Θ' of Θ and for each row r' of Θ', the set of rules $Rul_G(\Theta', r')$ is already defined. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. For any $f_i \in E_G(\Theta, r)$, we define the set of rules $Rul_G(\Theta, r, f_i)$ as follows: $Rul_G(\Theta, r, f_i) = \{f_i = b_i \wedge \sigma \rightarrow s : \sigma \rightarrow s \in Rul_G(\Theta(f_i, b_i), r)\}$. Then

$$Rul_G(\Theta, r) = \bigcup_{f_i \in E_G(\Theta, r)} Rul_G(\Theta, r, f_i)$$

Theorem 1: in press [35] For any node Θ of $\Delta_\gamma(T)$ and for any row r of Θ, the set $Rul_{\Delta_\gamma(T)}(\Theta, r)$ is equal to the set of all irredundant γ-decision rules for Θ and r.

Example 3.1: To illustrate the presented algorithm we consider an example based on decision table T_0 depicted in Fig. 1. We set $\gamma = 1$, so during the construction of the graph $\Delta_1(T_0)$ we stop the partitioning of a subtable Θ of T_0 when $J(\Theta) \leq 1$. We denote $G = \Delta_1(T_0)$.

For each node Θ of the graph G and for each row r of Θ we describe the set $Rul_G(\Theta, r)$. We will move from terminal nodes of G to the node T_0. Terminal nodes of the graph G are Θ_1, Θ_2, Θ_3, Θ_4, Θ_6, Θ_7, Θ_8. For these nodes,
$Rul_G(\Theta_1, r_4) = Rul_G(\Theta_1, r_5) = \{\rightarrow 1\}$;
$Rul_G(\Theta_2, r_1) = Rul_G(\Theta_2, r_2) = Rul_G(\Theta_2, r_3) = \{\rightarrow 2\}$;
$Rul_G(\Theta_3, r_2) = Rul_G(\Theta_3, r_5) = \{\rightarrow 1\}$;
$Rul_G(\Theta_4, r_1) = Rul_G(\Theta_4, r_3) = Rul_G(\Theta_4, r_4) = \{\rightarrow 3\}$;
$Rul_G(\Theta_6, r_1) = \{\rightarrow 2\}$;
$Rul_G(\Theta_7, r_2) = Rul_G(\Theta_7, r_3) = \{\rightarrow 2\}$;
$Rul_G(\Theta_8, r_3) = Rul_G(\Theta_8, r_4) = \{\rightarrow 3\}$.
Now, we can describe the sets of rules attached to rows of Θ_5. This is a nonterminal node of G for which all children Θ_1, Θ_3, Θ_7, and Θ_8 are already treated. We have:
$Rul_G(\Theta_5, r_2) = \{f_2 = 0 \rightarrow 1, f_1 = 1 \rightarrow 2\}$;
$Rul_G(\Theta_5, r_3) = \{f_1 = 1 \rightarrow 2, f_2 = 1 \rightarrow 3\}$;
$Rul_G(\Theta_5, r_4) = \{f_1 = 0 \rightarrow 1, f_2 = 1 \rightarrow 3\}$;
$Rul_G(\Theta_5, r_5) = \{f_1 = 0 \rightarrow 1, f_2 = 0 \rightarrow 1\}$.
Finally, we can describe the sets of rules attached to rows of T_0:
$Rul_G(T_0, r_1) = \{f_1 = 1 \rightarrow 2, f_2 = 1 \rightarrow 3, f_3 = 1 \rightarrow 2\}$;
$Rul_G(T_0, r_2) = \{f_1 = 1 \rightarrow 2, f_2 = 0 \rightarrow 1, f_3 = 0 \wedge f_2 = 0 \rightarrow 1,\ f_3 = 0 \wedge f_1 = 1 \rightarrow 2\}$;
$Rul_G(T_0, r_3) = \{f_1 = 1 \rightarrow 2, f_2 = 1 \rightarrow 3, f_3 = 0 \wedge f_1 =$

$1 \to 2,\ f_3 = 0 \wedge f_2 = 1 \to 3\}$;
$Rul_G(T_0, r_4) = \{f_1 = 0 \to 1, f_2 = 1 \to 3, f_3 = 0 \wedge f_1 = 0 \to 1,\ f_3 = 0 \wedge f_2 = 1 \to 3\}$;
$Rul_G(T_0, r_5) = \{f_1 = 0 \to 1, f_2 = 0 \to 1, f_3 = 0 \wedge f_1 = 0 \to 1,\ f_3 = 0 \wedge f_2 = 0 \to 1\}$;

IV. Procedures of Optimization Relative to Length, Coverage and Number of Misclassifications

In this section, we describe procedures of optimization of irredundant γ-decision rules relative to the length, coverage and number of misclassifications.

First, we describe the procedure of optimization of the graph G relative to the length l. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^l_G(\Theta, r)$ of γ-decision rules with minimum length from $Rul_G(\Theta, r)$ and the number $Opt^l_G(\Theta, r)$ – the minimum length of a γ-decision rule from $Rul_G(\Theta, r)$.

We will move from the terminal nodes of the graph G to the node T. We will correspond to each row r of each table Θ the number $Opt^l_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by G^l.

Let Θ be a terminal node of G. Then we correspond the number

$$Opt^l_G(\Theta, r) = 0$$

to each row r of Θ.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number

$$Opt^l_G(\Theta, r) = \min\{Opt^l_G(\Theta(f_i, b_i), r) + 1 : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G^l}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^l_G(\Theta(f_i, b_i), r) + 1 = Opt^l_G(\Theta, r)\}.$$

Theorem 2: in press [35] For each node Θ of the graph G^l and for each row r of Θ, the set $Rul_{G^l}(\Theta, r)$ is equal to the set $Rul^l_G(\Theta, r)$ of all γ-decision rules with the minimum length from the set $Rul_G(\Theta, r)$.

Now, we describe the procedure of optimization of the graph G relative to the coverage c. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^c_G(\Theta, r)$ of γ-decision rules with maximum coverage from $Rul_G(\Theta, r)$ and the number $Opt^c_G(\Theta, r)$ – the maximum coverage of a γ-decision rule from $Rul_G(\Theta, r)$.

We will move from the terminal nodes of the graph G to the node T. We will correspond to each row r of each table Θ the number $Opt^c_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by G^c.

Let Θ be a terminal node of G and d be the most common decision for Θ. Then we correspond to each row r of Θ the number $Opt^c_G(\Theta, r)$ that is equal to the number of rows in Θ which are labeled with the decision d.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number

$$Opt^c_G(\Theta, r) = \max\{Opt^c_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G^c}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^c_G(\Theta(f_i, b_i), r) = Opt^c_G(\Theta, r)\}.$$

Theorem 3: in press [35] For each node Θ of the graph G^c and for each row r of Θ, the set $Rul_{G^c}(\Theta, r)$ is equal to the set $Rul^c_G(\Theta, r)$ of all γ-decision rules with the maximum coverage from the set $Rul_G(\Theta, r)$.

Detailed descriptions of the procedures of optimization of irredundant γ-decision rules relative to the length and the coverage with examples, and proofs of Theorems 2 and 3 can be found in press [35].

Now, we describe the procedure of optimization of the graph G relative to the number of misclassifications μ. For each node Θ in the graph G, this procedure corresponds to each row r of Θ the set $Rul^\mu_G(\Theta, r)$ of γ-decision rules with the minimum number of misclassifications from $Rul_G(\Theta, r)$ and the number $Opt^\mu_G(\Theta, r)$ – the minimum number of misclassifications of a γ-decision rule from $Rul_G(\Theta, r)$.

We will move from the terminal nodes of the graph G to the node T. We will correspond to each row r of each table Θ the number $Opt^\mu_G(\Theta, r)$ and we will change the set $E_G(\Theta, r)$ attached to the row r in Θ if Θ is a nonterminal node of G. We denote the obtained graph by G^μ.

Let Θ be a terminal node of G and d be the most common decision for Θ. Then we correspond to each row r of Θ the number $Opt^\mu_G(\Theta, r)$ which is equal to the number of rows in Θ which are labeled with decisions different from d.

Let Θ be a nonterminal node of G and all children of Θ have already been treated. Let $r = (b_1, \ldots, b_n)$ be a row of Θ. We correspond the number

$$Opt^\mu_G(\Theta, r) = \min\{Opt^\mu_G(\Theta(f_i, b_i), r) : f_i \in E_G(\Theta, r)\}$$

to the row r in the table Θ and we set

$$E_{G^\mu}(\Theta, r) = \{f_i : f_i \in E_G(\Theta, r), Opt^\mu_G(\Theta(f_i, b_i), r) = Opt^\mu_G(\Theta, r)\}.$$

Theorem 4: in press [36] For each node Θ of the graph G^μ and for each row r of Θ, the set $Rul_{G^\mu}(\Theta, r)$ is equal to the set $Rul^\mu_G(\Theta, r)$ of all γ-decision rules with the minimum number of misclassifications from the set $Rul_G(\Theta, r)$.
Detailed description of the procedure of optimization of irredundant γ-decision rules relative to the number of misclassifications with example, and proof of Theorem 4 can be found in press [36].

V. Sequential Optimization

Theorems 2, 3 and 4 show that for a given decision table T and row r of T, we can make sequential optimization of rules relative to the length, coverage and number of misclassifications. We can find all irredundant γ-decision rules for T and r with minimum length, after that among these rules find all rules with maximum coverage, and finally among the obtained

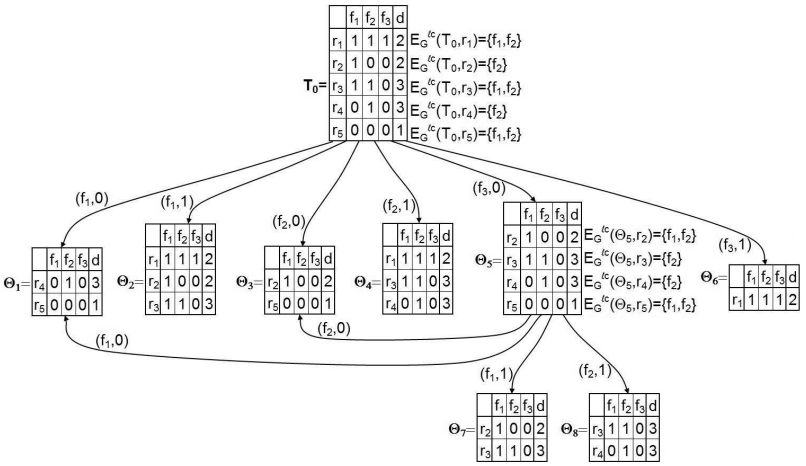

Fig. 2. Graph G^{lc}

rules find all rules with minimum number of misclassifications. We can use an arbitrary set of cost functions and an arbitrary order of optimization.

We have three cost functions: length l, coverage c, and number of misclassifications μ. Let F be one of sets $\{l, c, \mu\}$, $\{l, c, \}$, $\{l, \mu\}$, and $\{c, \mu\}$. An irredundant γ-decision rule τ for T and r is called *totally optimal relative to the cost functions from* F if, for each cost function $f \in F$, the value $f(\tau)$ is minimum if $f \in \{l, \mu\}$ or maximum if $f = c$ among all irredundant γ-decision rules for T and r. In particular, we will say that an irredundant γ-decision rule for T and r is *totally optimal relative to the length and coverage* if it has minimum length and maximum coverage among all irredundant γ-decision rules for T and r. We can describe all totally optimal rules relative to the cost functions from F using the procedures of optimization relative to these cost functions.

To describe process of sequential optimization we set $G = \Delta_\gamma(T)$ and first, we consider the case when F contains two cost functions. Without the loss of generality we can assume that $F = \{l, c\}$.

We apply the procedure of optimization relative to the coverage to the graph G. As a result we obtain the graph G^c and, for each row r of T, the value $Opt^c_G(T, r)$ which is equal to the maximum coverage of an irredundant γ-decision rule for T and r.

After that, we apply the procedure of optimization relative to the length to the graph G. As a result we obtain the graph G^l. Finally, we apply the procedure of optimization relative to the coverage to the graph G^l. As a result we obtain the graph G^{lc} and, for each row r of T, the value $Opt^c_{G^l}(T, r)$ which is equal to the maximum coverage of an irredundant γ-decision rule for T and r among all irredundant γ-decision rules for T and r with minimum length.

One can show that a totally optimal relative to the length and coverage irredundant γ-decision rule for T and r exists if and only if $Opt^c_G(T, r) = Opt^c_{G^l}(T, r)$. If the last equality holds then the set $Rul_{G^{lc}}(T, r)$ is equal to the set of all totally optimal relative to the length and coverage irredundant γ-decision rules for T and r.

It is clear that the results of sequential optimization of irredundant decision rules for T and r relative to the length and coverage does not depend on the order of optimization (length+coverage or coverage+length) if and only if there exists a totally optimal relative to the length and coverage irredundant γ-decision rule for T and r. We can find all irredundant γ-decision rules for T and r with minimum length and after that among these rules find all rules with maximum coverage. We can use an arbitrary order of optimization, e.g., coverage and length.

Example 5.1: Figure 2 presents the directed acyclic graph G^{lc} obtained from the graph G (see Fig. 1) after sequential optimization relative to the length and coverage. Using the graph G^{lc} we can describe for each row r_i, $i = 1, \ldots, 5$, of the table T_0 the set $Rul_{G^{lc}}(T_0, r_i)$ of all irredundant 1-decision rules for T_0 and r_i which have maximum coverage among all irredundant 1-decision rules for T_0 and r_i with minimum length. We will give also the value $Opt^c_{G^l}(T_0, r_i)$ which is equal to the maximum coverage of a 1-decision rule for T_0 and r_i among all irredundant 1-decision rules for T_0 and r_i with minimum length. This value was obtained during the procedure of optimization of the graph G^l relative to the coverage. We have:
$Rul^{lc}_G(T_0, r_1) = \{f_1 = 1 \to 2, f_2 = 1 \to 3\}$, $Opt^c_{G^l}(T_0, r_1) = 2$;
$Rul^{lc}_G(T_0, r_2) = \{f_1 = 1 \to 2\}$, $Opt^c_{G^l}(T_0, r_2) = 2$;
$Rul^{lc}_G(T_0, r_3) = \{f_1 = 1 \to 2, f_2 = 1 \to 3\}$, $Opt^c_{G^l}(T_0, r_3) = 2$;
$Rul^{lc}_G(T_0, r_4) = \{f_2 = 1 \to 3\}$, $Opt^c_{G^l}(T_0, r_4) = 2$;
$Rul^{lc}_G(T_0, r_5) = \{f_1 = 0 \to 1, f_2 = 0 \to 1\}$, $Opt^c_{G^l}(T_0, r_5) = 1$.
Values $Opt^c_G(T_0, r_i)$ (obtained by the procedure of optimization the graph G relative to the coverage) are the same as $Opt^c_{G^l}(T_0, r_i)$, for $i = 1, \ldots, 5$. Therefore, for $i = 1, \ldots, 5$, $Rul_{G^{lc}}(T_0, r_i)$ is the set of all totally optimal relative to the length and coverage irredundant 1-decision rules for T_0 and r_i, and we have such rules for each row of T_0.

Now we consider the case when $F = \{l, c, \mu\}$. At the beginning, we will make the same steps as in the case $F = \{l, c\}$. As a results we obtain the graph G^{lc} and values $Opt^c_G(T, r)$, $Opt^c_{G^l}(T, r)$ for any row r of T.

After that, we apply the procedure of optimization relative to the number of misclassifications to the graph G. As a result we obtain the graph G^μ and, for each row r of T, the value $Opt^\mu_G(T, r)$ which is equal to the minimum number of misclassifications of an irredundant γ-decision rule for T and r.

Finally, we apply the procedure of optimization relative to the number of misclassifications to the graph G^{lc}. As a result we obtain the graph $G^{lc\mu}$ and, for each row r of T, the value $Opt^\mu_{G^{lc}}(T, r)$ which is equal to the minimum number of misclassifications of an irredundant γ-decision rule for T and r among all irredundant γ-decision rules for T and r with maximum coverage, and among all irredundant γ-decision rules for T and r with minimum length.

One can show that a totally optimal relative to l, c and μ irredundant γ-decision rule for T and r exists if and only if $Opt^c_G(T, r) = Opt^c_{G^l}(T, r)$ and $Opt^\mu_G(T, r) = Opt^\mu_{G^{lc}}(T, r)$. If these equalities hold then the set $Rul_{G^{lc\mu}}(T, r)$ is equal to the set of all totally optimal relative to l, c and μ irredundant γ-decision rules for T and r.

It is clear that the results of sequential optimization of irredundant decision rules for T and r relative to the length, coverage and number of misclassifications do not depend on the order of optimization (l+c+μ, l+μ+c, c+l+μ, c+μ+l, μ+l+c, or μ+c+l) if and only if there exists a totally optimal relative to l, c and μ irredundant γ-decision rule for T and r.

For decision table T_0 depicted in Fig. 1, the graph $G^{lc\mu}$ is the same as the graph G^{lc} presented in Fig. 2: $E_{G^{lc}}(\Theta_5, r_i) = E_{G^{lc\mu}}(\Theta_5, r_i)$ for $i = 2, \ldots, 5$, and $E_{G^{lc}}(T_0, r_i) = E_{G^{lc\mu}}(T_0, r_i)$ for $i = 1, \ldots, 5$. However, totally optimal relative to l, c and μ irredundant 1-decision rules exist only for rows r_2 and r_5 of the table T_0.

Considering complexities of the presented algorithms, it is possible to show (see analysis of similar algorithms in [7], page 64) that the time complexities of algorithms which construct the graph $\Delta_\gamma(T)$ and make sequential optimization of γ-decision rules relative to the length, coverage and number of misclassifications, are bounded from above by polynomials on the number of separable subtables of T, and the number of attributes in T. In [39] it was shown that the number of separable subtables for decision tables with attributes from a restricted infinite information systems is bounded from above by a polynomial on the number of attributes in the table. Examples of restricted infinite information system were considered, in particular, in [7].

VI. Experimental Results

We studied a number of decision tables from UCI Machine Learning Repository [37]. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. In some tables there were equal rows with, possibly, different decisions. In this case each group of identical rows was replaced with a single row from the group with the most common decision for this group. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute.

Let T be one of these decision tables. We consider for this table the value of $J(T)$ and values of γ from the set $\Gamma(T) = \{\lfloor J(T) \times 0.01 \rfloor, \lfloor J(T) \times 0.2 \rfloor, \lfloor J(T) \times 0.3 \rfloor\}$. These parameters can be found in Table I, where column "Rows" contains number of rows, column "Attr" contains number of conditional attributes, column "$J(T)$" contains difference between number of rows in decision table and number of rows with the most common decision for this decision table, column "$\gamma \in \Gamma(T)$" contains values from $\Gamma(T)$.

Tables II, III and IV present results of sequential optimization of irredundant γ-decision rules relative to the length, coverage and number of misclassifications for corresponding values of γ. Columns "avg_l", "avg_c", "avg_μ" contain average length, average coverage and average number of misclassifications after three steps of corresponding order of optimization. We did experiments for all possible combination of order of optimization (l+c+μ, l+μ+c, c+l+μ, c+μ+l, μ+l+c, or μ+c+l) but because of limited number of pages we present results only for order μ+l+c (column μ+length+coverage") and l+μ+c (column "length+μ+coverage"). For example, for case of optimization μ+l+c we make three steps of optimization – relative to the number of misclassifications, then relative to the length, and then relative to the coverage. After that, we find the average length, the average coverage and the average number of misclassifications of rules after three steps of optimization.

In our experiments, for data sets which are in bold (Table II, Table III, Table IV), for all possible combination of order of optimization relative to l, c, μ, we found the same values of average length, average coverage and average number of misclassifications after three steps of optimization. It means that for such data sets each row has at least one totally optimal rule relative to the length, coverage and number of misclassifications. For the rest of data sets the number of rows for which exist totally optimal rules relative to l, c, μ is less than the number of all rows in the considered decision table.

TABLE I
Parameters of decision tables and values of $\gamma \in \Gamma(T)$

Name of decision table	Rows	Attr	$J(T)$	$\gamma \in \Gamma(T)$		
				0.01	0.2	0.3
Adult-stretch	16	4	4	0	0	0
Agaricus-lepiota	8124	22	3916	39	783	1174
Balance-scale	625	4	337	3	67	101
Breast-cancer	266	9	76	0	15	22
Cars	1728	6	518	5	103	155
Flags	193	26	141	1	28	42
Hayes-roth-data	69	4	39	0	7	11
House-votes-84	279	16	92	0	18	27
Lymphography	148	18	67	0	13	20
Nursery	12960	8	8640	86	1728	2592
Shuttle-landing	15	6	6	0	1	1
Soybean-small	47	35	30	0	6	9
Spect-test	169	22	8	0	1	2
Teeth	32	8	22	0	4	6
Tic-tac-toe	958	9	332	3	66	99
Zoo	59	16	40	0	8	12

TABLE II
SEQUENTIAL OPTIMIZATION FOR $\lfloor \gamma = J(T) \times 0.01 \rfloor$

Decision table	$\lfloor \gamma = J(T) \times 0.01 \rfloor$					
	μ+length+coverage			length+μ+coverage		
	avg_l	avg_c	avg_μ	avg_l	avg_c	avg_μ
Adult-stretch	1.250	7.000	0.000	1.250	7.000	0.000
Agaricus-lepiota	1.182	1370.132	0.000	1.176	1366.455	0.063
Balance-scale	2.942	5.875	0.277	2.290	17.760	1.448
Breast-cancer	2.665	7.038	0.000	2.665	7.038	0.000
Cars	2.351	332.964	0.094	2.087	335.046	0.817
Flags	1.933	6.394	0.000	1.627	6.896	0.306
Hayes-roth-data	2.145	6.522	0.000	2.145	6.522	0.000
House-votes-84	2.538	65.409	0.000	2.538	65.409	0.000
Lymphography	1.993	15.169	0.000	1.993	15.169	0.000
Nursery	2.953	1537.605	1.208	2.274	1642.249	18.201
Shuttle-landing	1.400	1.867	0.000	1.400	1.867	0.000
Soybean-small	1.000	12.234	0.000	1.000	12.234	0.000
Spect-test	1.479	53.550	0.000	1.479	53.550	0.000
Teeth	2.261	1.000	0.000	2.261	1.000	0.000
Tic-tac-toe	3.017	66.580	0.000	3.004	66.643	0.038
Zoo	1.559	10.525	0.000	1.559	10.525	0.000

TABLE III
SEQUENTIAL OPTIMIZATION FOR $\lfloor \gamma = J(T) \times 0.2 \rfloor$

Decision table	$\lfloor \gamma = J(T) \times 0.2 \rfloor$					
	μ+length+coverage			length+μ+coverage		
	avg_l	avg_c	avg_μ	avg_l	avg_c	avg_μ
Adult-stretch	1.250	7.000	0.000	1.250	7.000	0.000
Agaricus-lepiota	1.182	1367.037	0.000	1.000	1229.795	16.594
Balance-scale	1.000	92.312	32.688	1.000	92.312	32.688
Breast-cancer	2.447	9.835	0.711	1.068	20.417	6.831
Cars	1.604	355.564	8.760	1.250	412.850	21.817
Flags	2.166	6.057	0.021	1.000	9.005	4.275
Hayes-roth-data	1.565	7.217	0.667	1.565	7.217	0.667
House-votes-84	2.487	62.305	0.086	1.000	110.771	6.602
Lymphography	2.081	14.534	0.027	1.007	18.703	5.047
Nursery	1.667	1877.841	137.459	1.000	2289.867	878.133
Shuttle-landing	1.333	1.933	0.067	1.133	2.000	0.267
Soybean-small	1.000	12.234	0.000	1.000	12.234	0.000
Spect-test	1.485	53.556	0.000	1.107	60.183	0.284
Teeth	2.000	1.000	0.304	1.174	1.000	1.957
Tic-tac-toe	2.611	34.732	10.520	1.292	113.691	47.364
Zoo	1.593	10.593	0.000	1.000	11.712	1.746

We can also observe that data sets for which exists totally optimal relative to l, c, μ irredundant γ-decision rule for T and r, are different when the value of γ is changing.

Sequential optimization can be considered as a problem of multi-criteria optimization with hierarchically dependent criteria. For example, if the length of rules is the most important criterium and we would like to construct short rules, length should be considered as the first cost function in order of optimization. Based on results presented in Tables II, III, IV we can find the minimum values of average length (column "avg_l" in order "length+μ+coverage") and the minimum values of average number of misclassifications (column "avg_μ" in order "μ+length+coverage"), for considered γ.

For Tables II, III, IV we can observe also:

- The length of irredundant γ-decision rules is nonincreasing when γ in increasing (column "avg_l" in order "length+μ+coverage"). For the order "μ+length+coverage" we can find exceptions as Flags, Lymphography, Spect-test and Zoo in Table III, and Flags, House-votes-84 and Lymphography in Table IV.
- The average coverage (column "avg_c" in order "length+μ+coverage" is the same or greater than average coverage in order "μ+length+coverage". The exception is data set Agaricus-lepiota.
- The average number of misclassifications (column "avg_μ") is nondecreasing when the value of γ is increasing.
- In order of optimization "length+μ+coverage" μ is the second cost function, so values in the column "avg_μ" in this case are usually greater than in column "avg_μ" and order "μ+length+coverage".

TABLE IV
SEQUENTIAL OPTIMIZATION FOR $\lfloor \gamma = J(T) \times 0.3 \rfloor$

Decision table	$\lfloor \gamma = J(T) \times 0.3 \rfloor$					
	μ+length+coverage			length+μ+coverage		
	avg_l	avg_c	avg_μ	avg_l	avg_c	avg_μ
Adult-stretch	1.250	7.000	0.000	1.250	7.000	0.000
Agaricus-lepiota	1.182	1364.978	0.000	1.000	1229.795	16.594
Balance-scale	1.000	92.312	32.688	1.000	92.312	32.688
Breast-cancer	2.203	10.741	1.312	1.008	22.060	7.699
Cars	1.444	364.206	15.238	1.000	472.579	43.421
Flags	2.487	6.171	0.093	1.000	9.005	4.275
Hayes-roth-data	1.565	7.087	1.145	1.000	9.609	5.304
House-votes-84	2.753	45.667	0.226	1.000	110.771	6.602
Lymphography	2.270	9.338	0.189	1.000	18.784	5.135
Nursery	1.667	1757.600	271.200	1.000	2289.867	878.133
Shuttle-landing	1.333	1.933	0.067	1.133	2.000	0.267
Soybean-small	1.000	12.234	0.000	1.000	12.234	0.000
Spect-test	1.485	52.864	0.000	1.030	63.172	0.414
Teeth	1.913	1.000	0.522	1.000	1.000	2.739
Tic-tac-toe	2.000	64.551	16.590	1.029	152.585	63.856
Zoo	1.593	10.593	0.000	1.000	11.712	1.746

We can consider γ as an upper bound on the number of misclassifications of irredundant γ-decision rules (see column $\gamma \in \Gamma(T)$ in Table I). Results in Tables II, III, IV show that average values of the minimum number of misclassifications are often less than upper bound on the number of misclassifications given by γ.

Experiments were done using software system Dagger [38]. It is implemented in C++ and uses Pthreads and MPI libraries for managing threads and processes respectively. It runs on a single-processor computer or multiprocessor system with shared memory. Parameters of computer which was used for experiments are following: desktop with 2 Xeon x5550 processors running at 2.66 GHz (each with 4 cores and 8 threads) all sharing 16 GB of RAM. The longest time of the preformed experiments (for $\gamma \in \Gamma(T)$) was for the decision table Flags – 181 min. For the decision table Agaricus-lepiota – 36 min., for the Nursery – 3 min.

VII. CONCLUSIONS

We studied an extension of dynamic programming approach for the sequential optimization of γ-decision rules relative to the length, coverage and number of misclassifications. The considered approach allows to describe the whole set of irredundant γ-decision rules and optimize these rules sequentially relative to arbitrary subset and order of cost functions. So, we can consider the problem of multi-criteria optimization of decision rules with hierarchically dependent criteria.

Results of sequential optimization of irredundant γ-decision rules for T and r depend on the order of optimization if there

are no totally optimal relative to l, c, μ irredundant γ-decision rules for T and r.

Sequential optimization of irredundant γ-decision rules and construction of totally optimal rules can be considered as tools which support design of classifiers. To predict the value of decision attribute for a new object we can use in a classifier only totally optimal rules or rules with the maximum coverage, etc. Short rules which cover many objects can be useful also in knowledge discovery to represent knowledge extracted from decision tables. In this case, rules with smaller number of descriptors are more understandable.

Future study will be connected with the construction of classifiers and scalability for the presented approach.

ACKNOWLEDGMENT

The author would like to thank you Prof. Mikhail Moshkov, Dr. Igor Chikalov and Talha Amin for possibility to use results of Dagger software system.

REFERENCES

[1] J. G. Bazan, H. S. Nguyen, T. T. Nguyen, A. Skowron, and J. Stepaniuk, "Synthesis of decision rules for object classification," in *Incomplete Information: Rough Set Analysis*, E. Orłowska, Ed. Heidelberg: Physica-Verlag, 1998, pp. 23–57.

[2] J. Błaszczyński, R. Słowiński, and R. Susmaga, "Rule-based estimation of attribute relevance," in *RSKT 2011*, ser. LNCS, vol. 6954. Springer, 2011, pp. 36–44.

[3] I. Brzezińska, S. Greco, and R. Słowiński, "Mining pareto-optimal rules with respect to support and confirmation or support and anti-support," *Eng. Appl. of AI*, vol. 20, no. 5, pp. 587–600, 2007.

[4] I. Chikalov, M. Moshkov, and B. Zielosko, "Online learning algorithm for ensemble of decision rules," in *RSFDGrC 2011*, ser. LNCS. Heidelberg: Springer, 2011, vol. 6743, pp. 310–313.

[5] K. Dembczyński, W. Kotłowski, and R. Słowiński, "Ender: a statistical framework for boosting decision rules," *Data Min. Knowl. Discov.*, vol. 21, no. 1, pp. 52–90, 2010.

[6] M. Moshkov, M. Piliszczuk, and B. Zielosko, *Partial Covers, Reducts and Decision Rules in Rough Sets - Theory and Applications*. Heidelberg: Springer, 2008.

[7] M. Moshkov and B. Zielosko, *Combinatorial Machine Learning - A Rough Set Approach*. Heidelberg: Springer, 2011.

[8] H. S. Nguyen and D. Ślęzak, "Approximate reducts and association rules - correspondence and complexity results," in *RSFDGrC 1999*, ser. LNCS, vol. 1711. Springer, 1999, pp. 137–145.

[9] Z. Pawlak, *Rough Sets - Theoretical Aspects of Reasoning about Data*. Dordrecht: Kluwer Academic Publishers, 1991.

[10] ——, "Rough set elements," in *Rough Sets in Knowledge Discovery*. Heidelberg: Physica-Verlag, 1998, pp. 10–30.

[11] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Inf. Sci.*, vol. 177, no. 1, pp. 3–27, 2007.

[12] A. Skowron, "Rough sets in KDD," in *16th World Computer Congress, IFIP'2000, Proc. Conf. Intelligent Information Processing, IIP'2000*. Beijing: House of Electronic Industry, 2000, pp. 1–17.

[13] D. Ślęzak and J. Wróblewski, "Order based genetic algorithms for the search of approximate entropy reducts," in *RSFDGrC 2003*, ser. LNCS. Springer, 2003, vol. 2639, pp. 308–311.

[14] J. Wróblewski, "Finding minimal reducts using genetic algorithm," in *Proc. of the Second Annual Joint Conference on Information Sciences*. Wrightsville Beach, NC, 1995, pp. 186–189.

[15] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. of the 20th International Conference on Very Large Data Bases, VLDB '94*. Morgan Kaufmann, 1994, pp. 487–499.

[16] R. Jensen and Q. Shen, "Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 12, pp. 1457–1471, 2004.

[17] H. S. Nguyen, "Approximate boolean reasoning: foundations and applications in data mining," in *T. Rough Sets*, ser. LNCS. Springer, 2006, vol. 4100, pp. 334–506.

[18] Z. Pawlak and A. Skowron, "Rough sets and boolean reasoning," *Inf. Sci.*, vol. 177, no. 1, pp. 41–73, 2007.

[19] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*. Dordrecht: Kluwer Academic Publishers, 1992, pp. 331–362.

[20] A. An and N. Cercone, "Elem2: a learning system for more accurate classifications," in *Proc. of the 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*. London, UK: Springer-Verlag, 1998, pp. 426–441.

[21] J. Błaszczyński, R. Słowiński, and M. Szeląg, "Sequential covering rule induction algorithm for variable consistency rough set approaches," *Inf. Sci.*, vol. 181, no. 5, pp. 987–1002, 2011.

[22] P. Clark and T. Niblett, "The cn2 induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, 1989.

[23] J. Fürnkranz, "Separate-and-conquer rule learning," *Artif. Intell. Rev.*, vol. 13, no. 1, pp. 3–54, 1999.

[24] J. Fürnkranz and P. A. Flach, "Roc 'n' rule learning–towards a better understanding of covering algorithms," *Mach. Learn.*, vol. 58, no. 1, pp. 39–77, 2005.

[25] J. W. Grzymała-Busse, "Lers – a system for learning from examples based on rough sets," in *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, 1992, pp. 3–18.

[26] R. S. Michalski, "A theory and methodology of inductive learning," *Artif. Intell.*, vol. 20, no. 2, pp. 111–161, 1983.

[27] U. Boryczka and J. Kozak, "New algorithms for generation decision trees-Ant-Miner and its modifications," in *Foundations of Computational Intelligence (6)*. Springer, 2009, pp. 229–262.

[28] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *Proc. of the Fifteenth International Conference on Machine Learning, ICML '98*. Morgan Kaufmann Publishers Inc., 1998, pp. 144–151.

[29] S. Michalski and J. Pietrzykowski, "iAQ: A program that discovers rules," AAAI-07 AI Video Competition, 2007. [Online]. Available: http://videolectures.net/aaai07_michalski_iaq/

[30] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[31] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[32] B. Zielosko, M. Moshkov, and I. Chikalov, "Optimization of decision rules based on methods of dynamic programming," *Vestnik of Lobachevsky State University of Nizhny Novgorod*, vol. 6, pp. 195–200, 2010, (in Russian).

[33] T. Amin, I. Chikalov, M. Moshkov, and B. Zielosko, "Dynamic programming approach for exact decision rule optimization," in *Rough Sets and Intelligent Systems. Professor Zdzisław Pawlak in Memoriam*, A. Skowron and Z. Suraj, Eds. Springer, 2012, in press.

[34] ——, "Dynamic programming approach to optimization of approximate decision rules," *Inf. Sci.*, submitted for publication.

[35] ——, "Dynamic programming approach for partial decision rule optimization," *Fundam. Inform.*, 2012, in press.

[36] ——, "Optimization of approximate decision rules relative to number of misclassifications," in *Proc. of the 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*. Springer, 2012, in press.

[37] A. Asuncion and D. J. Newman, "UCI Machine Learning Repository," 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/

[38] A. Alkhalid, T. Amin, I. Chikalov, S. Hussain, M. Moshkov, and B. Zielosko, *Dagger: A tool for analysis and optimization of decision trees and rules*. Blue Herons, 2011, pp. 29–39.

[39] M. Moshkov and I. Chikalov, "On algorithm for constructing of decision trees with minimal depth," *Fundam. Inform.*, vol. 41, no. 3, pp. 295–299, 2000.