Control and Cybernetics

vol. 29 (2000) No. 4

Generation of reducts and rules in multi-attribute and multi-criteria classification

by

Robert Susmaga[(1)], Roman Słowiński[(1)], Salvatore Greco[(2)] and Benedetto Matarazzo[(2)]

[(1)]Institute of Computing Science, Poznań University of Technology, Piotrowo 3a, 60-965 Poznań, Poland
e-mails: robert.susmaga@cs.put.poznan.pl, slowinsk@sol.put.poznan.pl

[(2)]Faculty of Economics, Institute of Mathematics, University of Catania, Corso Italia 55, 95129 Catania, Italy

Abstract: The paper addresses the problem of analysing information tables which contain objects described by both attributes and criteria, i.e. attributes with preference-ordered scales. The objects contained in those tables, representing exemplary decisions made by a decision maker or a domain expert, are usually classified into one of several classes that are also often preference-ordered. Analysis of such data using the classic rough set methodology may produce improper results, as the original rough set approach is not able to discover inconsistencies originating from consideration of typical criteria, like e.g. product quality, market share or debt ratio. The paper presents the framework for the analysis of both attributes and criteria and a very promising algorithm for generating reducts. The algorithm presented is evaluated in an experiment with real-life data sets and its results are compared to those by two other reduct generating algorithms.

Keywords: intelligent information systems, rough sets theory, multi-attribute and multi-criteria classification, dominance relation, reducts of attributes and criteria, decision rules.

1. Introduction

As pointed out by Greco et al. (1998b), the Classic Rough Set Approach (CRSA) does not consider criteria, i.e. attributes with preference-ordered domains. In many real applications, however, the ordering properties of the considered attributes play an important role, e.g. in bankruptcy risk evaluation. Consider, for example, two firms, A and B, evaluated by a set of attributes including the 'debt ratio' (total debt/total assets). If A has a low value while B a high value then, from a bankruptcy risk point of view, A dominates B. Suppose, however, that the firm A has been assigned to a class of higher risk than the firm B. This is obviously inconsistent with the dominance principle. Within CRSA the two firms will be considered as just discernible and no inconsistency will be discovered.

Motivated by the above considerations, Greco et al. (1998a) have proposed a new rough set approach to the evaluation of the bankruptcy risk, in which the indiscernibility relation, used in CRSA, is substituted with a dominance relation. This new approach, called Dominance- based Rough Set Approach (DRSA), is general enough to be used with any classification problem involving both preference-ordered and preference-neutral attributes (see Greco et al., 1998b).

The rest of the paper is organized as follows. Section 2 presents the basic notions of the the Dominance-based Rough Set Approach (DRSA). Sections 3 includes remarks on reduct generation in DRSA and gives an illustrative example. Sections 4 describes the proposed reduct generating algorithm. Section 5 discusses the problems related to generating and applying decision rules in DRSA. The experimental evaluation of the algorithm is presented in Section 6. Finally, Section 7 summarizes the advantages of the DRSA in general and the reduct generating algorithm in particular.

2. Dominance-based Rough Set Approach (DRSA)

For algorithmic reasons, knowledge about objects (e.g. firms, patients) is often represented in the form of an information table. The rows of the table are labelled by *objects*, columns are labelled by *attributes* and entries of the table are *attribute-values*, called *descriptors*.

Formally, by an *information table* we understand a 4-tuple $S = \langle U, Q, V, f \rangle$, where U is a finite set of objects, Q is a finite set of *attributes* and *criteria*, $V = \bigcup_{q \in Q} V_q$, where V_q is the domain of attribute q, and $f : U \times Q \to V$ is a total function such that $f(x,q) \in V_q$ for every $(x,q) \in U \times Q$, called an *information function* (Pawlak, 1991). The set Q is, in general, divided into the set C of *condition attributes/criteria* and the set D of *decision attributes/criteria.* The notion of attribute differs from that of *criterion* because the scale (domain) of a criterion has to be ordered according to a decreasing or increasing preference, while the domain of an attribute is unrelated to preference.

Assuming that all condition attributes $q \in C$ are criteria, let S_q be an *outranking relation* (Roy, 1985) on U with respect to criterion q such that xS_qy means 'x is at least as good as y with respect to criterion q'. We suppose that S_q is a total preorder, i.e. a strongly complete and transitive binary relation, defined on U on the basis of the evaluations $f(x,q)$.

Assuming additionally that the set of decision attributes D (possibly a singleton $\{d\}$) induces a partition of U into a finite number of classes, let $\boldsymbol{Cl} = \{Cl_t,\ t \in T\}$, $T = \{1, \ldots, n\}$, be a set of these classes such that each ordered, i.e. for all $r, s \in T$, such that $r > s$, the objects from Cl_r are preferred (strictly or weakly, Roy, 1985) to the objects from Cl_s. More formally, if S is a *comprehensive outranking relation* on U, i.e. if for all $x, y \in U$, xSy means "x is at least as good as y", we suppose: $[x \in Cl_r,\ y \in Cl_s,\ r > s] \Rightarrow [xSy$ and *not* $ySx]$. The above are typical assumptions for a *multiple-criteria sorting problem*.

The key idea of rough sets is approximating one partition by another partition. In CRSA, the knowledge approximated is a partition of U into classes generated by the set of decision attributes/criteria, while the knowledge used for approximation is the partition of U into elementary sets of objects that are indiscernible with regard to a set of condition attributes. The elementary sets are perceived as 'granules of knowledge', which are further used for creating approximations.

In case of DRSA, where the set C may include both attributes and criteria, and where classes are preference-ordered, the knowledge approximated is a collection of upward and downward unions of classes and the 'granules of knowledge' are sets of objects implied by the dominance relation rather than the indiscernibility relation (Greco et al., 1998abc, 1999ab). This is the main difference between the CRSA and DRSA.

More precisely, the sets to be approximated in DRSA are the *upward* and *downward unions* of classes, which are defined as:

$$Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s, Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s,\ t = 1, \ldots, n.$$

The statement $x \in Cl_t^{\geq}$ means "x belongs at least to class Cl_t", while $x \in Cl_t^{\leq}$ means "x belongs at most to class Cl_t". Let us remark that $Cl_1^{\geq} = Cl_n^{\leq} = U$, $Cl_n^{\geq} = Cl_n$ and $Cl_1^{\leq} = Cl_1$. Furthermore, for $t = 2, \ldots, n$, we have: $Cl_{t-1}^{\leq} = U - Cl_t^{\geq}$ and $Cl_t^{\geq} = U - Cl_{t-1}^{\leq}$.

We say that x *dominates* y with respect to $P \subseteq C$, which is denoted as xD_Py, if xS_qy for all $q \in P$. Given $P \subseteq C$ and $x \in U$, the 'granules of knowledge' used for approximation in DRSA are:

- a set of objects dominating x, called *P-dominating set*, $D_P^+(x) = \{y \in U : yD_Px\}$,
- a set of objects dominated by x, called *P-dominated set*, $D_P^-(x) = \{y \in U : xD_Py\}$.

For any $P \subseteq C$ we say that $x \in U$ belongs to $Cl_t^{\geq}$ *without any ambiguity* if $x \in Cl_t^{\geq}$ and for all the objects $y \in U$ dominating x with respect to P, we have $y \in Cl_t^{\geq}$, i.e. if $D_P^+(x) \subseteq Cl_t^{\geq}$. Furthermore, we say that $y \in U$ *might belong* to $Cl_t^{\geq}$ if there existed at least one object $x \in Cl_t^{\geq}$ such that y dominates x with respect to P, i.e. if $y \in D_P^+(x)$.

Thus, with respect to $P \subseteq C$, the set of all objects belonging to $Cl_t^{\geq}$

by $\underline{P}(Cl_t^{\geq})$, and the set of all objects that could belong to $Cl_t^{\geq}$ constitutes the *P-upper approximation* of $Cl_t^{\geq}$, denoted by $\overline{P}(Cl_t^{\geq}$:

$$\underline{P}(Cl_t^{\geq}) = \{x \in U : D_P^+(x) \subseteq Cl_t^{\geq}\},$$
$$\overline{P}(Cl_t^{\geq}) = \bigcup_{x \in Cl_t^{\geq}} D_P^+(x), \text{ for } t = 1, \ldots, n.$$

Analogously, using $D_P^-(x)$ one defines *P-lower* and *P-upper approximation* of $Cl_t^{\leq}$:

$$\underline{P}(Cl_t^{\leq}) = \{x \in U : D_P^-(x) \subseteq Cl_t^{\leq}\},$$
$$\overline{P}(Cl_t^{\leq}) = \bigcup_{x \in Cl_t^{\leq}} D_P^-(x), \text{ for } t = 1, \ldots, n.$$

The *P-boundaries* (*P*-doubtful regions) of $Cl_t^{\geq}$ and $Cl_t^{\leq}$ are defined as:

$$Bn_P(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq} - \underline{P}(Cl_t^{\geq}),$$
$$Bn_P(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq} - \underline{P}(Cl_t^{\leq}), \text{ for } t = 1, \ldots, n.$$

Due to complementarity of the rough approximations (Greco et al., 1998b), the following property holds:

$$Bn_P(Cl_t^{\geq}) = Bn_P(Cl_{t-1}^{\leq}), \text{ for } t = 2, \ldots, n,$$
$$\text{and } Bn_P(Cl_t^{\leq}) = Bn_P(Cl_{t+1}^{\geq}), \text{ for } t = 1, \ldots, n-1.$$

For every $P \subseteq C$ we define the *quality of approximation of partition* $\boldsymbol{Cl}$ by set of attributes and criteria P, or in short, *quality of sorting*, as:

$$\gamma_P(\boldsymbol{Cl}) = \frac{\text{card}(U - (\bigcup_{t \in T} Bn_P(Cl_t^{\leq})))}{\text{card}(U)} = \frac{\text{card}(U - (\bigcup_{t \in T} Bn_P(Cl_t^{\geq})))}{\text{card}(U)}.$$

The quality expresses the ratio of all P-correctly sorted objects to all objects in the table.

Each minimal subset $P \subseteq C$ such that $\gamma_P(\boldsymbol{Cl}) = \gamma_C(\boldsymbol{Cl})$ is called a *reduct* of $\boldsymbol{Cl}$ and denoted by $RED_{\boldsymbol{Cl}}$. Let us remark that an information table can have more than one reduct. The intersection of all reducts is called the *core* and denoted by $CORE_{\boldsymbol{Cl}}$. The problem of generating reducts is interesting but difficult, as the complexity of the problem is NP-complete (Skowron & Rauszer, 1994). In Section 4, we present one of the most effective algorithms for generating all reducts of information systems.

The dominance-based rough approximations of upward and downward unions of classes are also used to induce generalized descriptions of objects in form of '*if* ... *then* ...' decision rules. The common ground for the problems of generating/analysing rules and reducts is that the rules are usually generated using only a subset of attributes/criteria and not all of them. Clearly, a subset of attributes/criteria that maintains the quality of sorting is much-desired for

3. Computation of dominance-based reducts

The reduct generating algorithms that are based on the notion of discernibility list/matrix, consist of two easily separated phases. Phase I creates and processes the list of differences between pairs of objects that are to be distinguished, and Phase II performs the actual search for reducts. Because the type of the final result of this algorithm is only influenced by its first phase, the algorithm may be easily adapted to produce various kinds of results. For example, it can generate reducts in the CRSA framework (i.e. classic rough sets approach reducts) as well as in the DRSA framework (as presented in this paper). It can also be used to produce exhaustive sets of decision rules, because rules are generated as minimal subsets of conditions that allow to distinguish some objects of a given class from every object of another class (in this sense the decision rules are strongly related to reducts).

What is even more interesting, the most complex part of the algorithm from the computational point of view is the Phase II which, in turn, is absolutely independent of the type of results that are to be generated. It is this part of the algorithm that actually solves the NP-hardness, which underlies the problem of generating all reducts (the complexity of the problem in Phase I is, on the other hand, merely polynomial).

The two-phase structure of the algorithm considerably helps in its development, because the algorithm may be adapted to generate different kinds of results, such as reducts in DRSA or CRSA, by modifications to its Phase I. On the other hand, its computing time may be independently improved by developments of the Phase II.

In its current form, the FRGA algorithm differs from the early discernibility matrix based algorithm RGA (Skowron & Rauszer, 1992; Tannhäuser, 1994) in two aspects. Firstly, its Phase I has been adapted to the framework of the DRSA, so that it can handle both indiscernibility and dominance between objects from U (Susmaga et al., 1999). Secondly, its Phase II underwent a development and allows much quicker computation, which is due to the minimality tests that have been introduced in (Susmaga, 1998b).

From the computational point of view, which is assumed in the discernibility based family of algorithms, it is advantageous to consider a differently formulated, but an equivalent definition of reducts. According to the definition above, the reduct is a minimal subset of attributes/criteria that preserves the value of the quality of sorting. A re-formulation of the definition is as follows.

Let C be the set of attributes/criteria and $x, y \in U$ denote two objects such that $x \in Cl_r$, $y \in Cl_s$, $r > s$, none of the objects is dominated by another and at least one of the objects belongs to a lower approximation of any union of classes. Let $P \subset C$. If $yDPx$ then the set P caused a conflict between x and y, because x is now dominated by y, i.e. an object that belongs to a class of lower preference. To prevent conflicts, the set P must contain:

- at least one (preference-neutral) attribute q, such that $f(x, q) \neq f(y, q)$, or

Other attributes/criteria may be discarded from P, thus allowing for the reduction of information without introducing conflicts between objects. Obviously, in the process of reduct generation all appropriate pairs of objects of a different decision must be analysed. A pair is appropriate only if at least one object of the pair does not belong to any $Bn_C(Cl_t^{\geq})$ for $t = 2, \ldots, n$. For each such pair, the set of all attributes that satisfy above conditions is established. All the resulting sets are stored in a list called the Dominance Retaining List (*DRL*).

The idea of the *DRL* resembles that of the discernibility list, which, in turn, originates from the discernibility matrix (Skowron & Rauszer, 1992; Susmaga, 1998a; Tannhäuser, 1994). The Discernibility List is a special case of the *DRL* computed for a system in which there exist exclusively preference-neutral attributes.

After being created, the *DRL* is processed using the law of absorption—elements that are supersets of some other elements are discarded from the list. Finally, the list is sorted in the ascending order of the cardinality of its elements, producing the Sorted, Absorbed Dominance Retaining List (*SADRL*). The *SADRL* may be directly used for generating reducts, which are found as all minimal subsets of attributes/criteria that have non-empty intersections with each element of the *SADRL*. It must be stressed that both absorbing and sorting the *DRL* has the only objective of improving the effectiveness of the reduct generating algorithm but does not affect the actual result.

To give a better idea of the introduced approach, we shall present an example. Let U be a set of nine examples, $x_1, \ldots, x_9$, described by four condition criteria c_1, c_2, c_3, c_4 and a decision criterion d. Let us also assume that domains of criteria c_1, c_2, c_3, c_4, d, are: the set $\{0, 1, 2\}$, the interval $[0, 1]$, the interval $[1, 5]$, the set $\{1, 2, 3\}$, and the set $\{\mathrm{I}, \mathrm{II}, \mathrm{III}\}$, respectively. Finally, the preference is increasing with the values, e.g. 1 is preferred to 0 on criterion c_1, 4 is preferred to 3 on criterion c_3 and III is preferred to II on criterion d. In general, criteria may take values from different domains, including non-numerical ones, providing that the preference of all the values is clearly defined. For computational reasons, such values are usually translated into numeric ones. In the current example, we assume that the domains have already been translated.

The descriptions of objects x_i by values of criteria c_j and d, i.e. the information function $f : U \times Q \to V$, is given in Table 1.

The decision criterion d takes three different values, so it implies three classes of objects:

- $Cl_{\mathrm{I}} = \{x_8, x_9\}$,
- $Cl_{\mathrm{II}} = \{x_1, x_2, x_3\}$,
- $Cl_{\mathrm{III}} = \{x_4, x_5, x_6, x_7\}$.

With such a criterion, the following unions of classes may be considered:

- "at most I" ($Cl_{\mathrm{I}}^{\leq}$) versus "at least II" ($Cl_{\mathrm{II}}^{\geq}$),

	c_1	c_2	c_3	c_4	d
x_1	0	0.4	0	2	II
x_2	0	0.4	3	1	II
x_3	1	0.9	0	3	II
x_4	1	0.9	0	3	III
x_5	0	0.9	3	3	III
x_6	0	0.9	3	2	III
x_7	1	0.4	3	3	III
x_8	1	0.1	5	3	I
x_9	2	0.9	5	2	I

Table 1. Exemplary information table

The contents of the class unions is as follows:

- "at most I": $Cl_{\mathrm{I}}^{\leq} = Cl_{\mathrm{I}} = \{x_8, x_9\}$,
- "at most II": $Cl_{\mathrm{II}}^{\leq} = Cl_{\mathrm{I}} \cup Cl_{\mathrm{II}} = \{x_1, x_2, x_3, x_8, x_9\}$,
- "at least II": $Cl_{\mathrm{II}}^{\geq} = Cl_{\mathrm{II}} \cup Cl_{\mathrm{III}} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$,
- "at least III": $Cl_{\mathrm{III}}^{\geq} = Cl_{\mathrm{III}} = \{x_4, x_5, x_6, x_7\}$.

Now, let us examine and present the dominance between objects with regard to the set C of all the condition criteria. This is done by presenting for each pair of objects, (x_i, x_j), one of the following:

- '$\geq$', if x_i dominates x_j,
- '$\leq$', if x_i is dominated by x_j,
- '$=$', if x_i is identical with x_j, and
- '$\neq$', otherwise.

The dominance information is collected and presented in Table 2.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_1	$=$	$\neq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\neq$	$\leq$
x_2	$\neq$	$=$	$\neq$	$\neq$	$\leq$	$\leq$	$\leq$	$\neq$	$\leq$
x_3	$\geq$	$\neq$	$=$	$=$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$
x_4	$\geq$	$\neq$	$=$	$=$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$
x_5	$\geq$	$\geq$	$\neq$	$\neq$	$=$	$\geq$	$\neq$	$\neq$	$\neq$
x_6	$\geq$	$\geq$	$\neq$	$\neq$	$\leq$	$=$	$\neq$	$\neq$	$\leq$
x_7	$\geq$	$\geq$	$\neq$	$\neq$	$\neq$	$\neq$	$=$	$\neq$	$\neq$
x_8	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$	$=$	$\neq$
x_9	$\geq$	$\geq$	$\neq$	$\neq$	$\neq$	$\geq$	$\neq$	$\neq$	$=$

Table 2. The matrix of dominance information in the example

Having established the dominance, it is easy to create the dominating and dominated sets for a given object x_i, i.e. the set of objects that dominate x_i, $D_C^+(x_i)$, and the set of objects dominated by x_i, $D_C^-(x_i)$. An object x_j belongs to $D_C^+(x_i)$ if there is '$=$' or '$\leq$' in row x_i and column x_j of the dominance table. On the other hand, an object x_j belongs to $D_C^-(x_i)$ if there is '$=$' or '$\geq$' in

- x_1: $D_C^+(x_1) = \{x_1, x_3, x_4, x_5, x_6, x_7, x_9\}$, $D_C^-(x_1) = \{x_1\}$;
- x_2: $D_C^+(x_2) = \{x_2, x_5, x_6, x_7, x_9\}$, $D_C^-(x_2) = \{x_2\}$;
- ...
- x_9: $D_C^+(x_9) = \{x_9\}$, $D_C^-(x_9) = \{x_1, x_2, x_6, x_9\}$.

Finally, the approximations of class unions with regard to all condition criteria (i.e. the set C) may be defined. In the example we have:

- $\underline{C}(Cl_{\mathrm{I}}^{\leq}) = \{x_8\}$, $\overline{C}(Cl_{\mathrm{I}}^{\leq}) = \{x_1, x_2, x_6, x_8, x_9\}$, $Bn_C(Cl_{\mathrm{I}}^{\leq}) = \{x_1, x_2, x_6, x_9\}$,
- $\underline{C}(Cl_{\mathrm{II}}^{\leq}) = \{x_1, x_2, x_8\}$, $\overline{C}(Cl_{\mathrm{II}}^{\leq}) = \{x_1, x_2, x_3, x_4, x_6, x_8, x_9\}$, $Bn_C(Cl_{\mathrm{II}}^{\leq}) = \{x_3, x_4, x_6, x_9\}$,
- $\underline{C}(Cl_{\mathrm{II}}^{\geq}) = \{x_3, x_4, x_5, x_7\}$, $\overline{C}(Cl_{\mathrm{II}}^{\geq}) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_9\}$, $Bn_C(Cl_{\mathrm{II}}^{\geq}) = \{x_1, x_2, x_6, x_9\}$,
- $\underline{C}(Cl_{\mathrm{III}}^{\geq}) = \{x_5, x_7\}$, $\overline{C}(Cl_{\mathrm{III}}^{\geq}) = \{x_3, x_4, x_5, x_6, x_7, x_9\}$, $Bn_C(Cl_{\mathrm{III}}^{\geq}) = \{x_3, x_4, x_6, x_9\}$.

As it can be easily observed, $Bn_C(Cl_{\mathrm{I}}^{\leq}) = Bn_C(Cl_{\mathrm{II}}^{\geq})$ and $Bn_C(Cl_{\mathrm{II}}^{\leq}) = Bn_C(Cl_{\mathrm{III}}^{\geq})$. If so, all objects belonging to the doubtful regions are:

$$\begin{aligned} &Bn_C(Cl_{\mathrm{I}}) \cup Bn_C(CL_{\mathrm{II}}) = Bn_C(Cl_{\mathrm{II}}^{\geq}) \cup Bn_C(Cl_{\mathrm{III}}^{\geq}) \\ &= \{x_1, x_2, x_3, x_4, x_6, x_9\}. \end{aligned}$$

All the remaining objects, i.e.: x_5, x_7 and x_8, do not enter any of the doubtful regions and, as such, determine the quality of sorting. Because in the example there are only three (out of nine) such objects, the quality of sorting $\gamma_C(\boldsymbol{Cl})$ equals $3/9 \approx 0.33$. The identified objects x_5, x_7, x_8 are especially important in the reduct generation process, because in the reduced table they must remain outside every doubtful region.

Because the dominance relation is monotonic, removing an attribute/criterion will not affect existing dominance between pairs of objects. A dominance, however, may be produced between so far incomparable pairs (i.e. pairs described with '$\neq$' in the dominance table). As long as domination between any two objects coming from doubtful regions is not produced, the quality of sorting is not changed. However, as soon as one of the objects x_5, x_7, x_8 would begin dominating an object from a higher class or become dominated by an object from a lower class, a situation referred to as the conflict would occur. In result, the quality of sorting would decrease. To prevent this, if a subset of condition attributes/criteria is to satisfy the definition of a reduct, it must not cause conflicts that would involve objects x_5, x_7 or x_8.

In the example, a conflict would be created if:

- x_5 (class Cl_{III}) began to be dominated by any object from $Cl_{\mathrm{I}} \cup Cl_{\mathrm{II}}$,
- x_7 (class Cl_{III}) began to be dominated by any object from $Cl_{\mathrm{I}} \cup Cl_{\mathrm{II}}$,
- x_8 (class Cl_{I}) began to dominate any object from $Cl_{\mathrm{II}} \cup Cl_{\mathrm{III}}$.

The algorithm presented solves the problem by computing the list of at-objects. Let us consider the object x_5. The corresponding line from the dominance table is:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_5	$\geq$	$\geq$	$\neq$	$\neq$	$=$	$\geq$	$\neq$	$\neq$	$\neq$

To maintain the current level of quality of sorting, we must control the relation between x_5 and every object from the lower classes, i.e. from $Cl_{\mathrm{I}} \cup Cl_{\mathrm{II}} = \{x_1, x_2, x_3, x_8, x_9\}$. Because out of these, x_1 and x_2 are already dominated by x_5, it is enough to consider only three objects: x_3, x_8 and x_9. Let us focus on x_3. Appropriate descriptions of x_3 and x_5 are as follows:

	c_1	c_2	c_3	c_4	d
x_3	1	0.9	0	3	II
x_5	0	0.9	3	3	III

Notice that the only criterion that prevents x_5 (class Cl_{III}) from being dominated by x_3 (class Cl_{II}) is the attribute c_3. If this criterion were removed from the decision table, a conflict would be created and the quality of sorting would decrease. In result, c_3 must be included in every reducts of the decision table. To remember this, the presented reduct generating algorithm stores the set $\{c_3\}$ as an element of the Dominance Retaining List (*DRL*). The remaining elements of *DRL* are created by analysing all other potential conflicts.

In the example, the subsets of criteria preventing conflicts between the following pairs of objects are:

- (x_5, x_3): $\{c_3\}$, (x_5, x_8): $\{c_2\}$, (x_5, x_9): $\{c_4\}$,
- (x_7, x_3): $\{c_3\}$, (x_7, x_8): $\{c_2\}$, (x_7, x_9): $\{c_4\}$,
- (x_8, x_1): $\{c_2\}$, (x_8, x_2): $\{c_2\}$, (x_8, x_3): $\{c_2\}$, (x_8, x_4): $\{c_2\}$, (x_8, x_5): $\{c_2\}$, (x_8, x_6): $\{c_2\}$, (x_8, x_7): $\{c_2\}$.

In result, the *DRL* consists of the following 13 elements: $\{c_3\}, \{c_2\}, \{c_4\}, \{c_3\}, \{c_2\}, \{c_4\}, \{c_2\}, \{c_2\}, \{c_2\}, \{c_2\}, \{c_2\}, \{c_2\}, \{c_2\}$. Clearly, there is a great deal of redundancy in this list. As it turns out, however, all the repeated elements may be removed from *DRL* in the process called absorption. In general, absorption allows to remove elements that are not minimal with regard to inclusion, i.e. elements that are supersets of some other elements. In the example, all *DRL* elements are singletons, so only removal of repeated elements takes place. The resulting absorbed *DRL* (*ADRL*) consists merely of 3 unique elements: $\{c_2\}$, $\{c_3\}$ and $\{c_4\}$.

According to the reduct generating algorithm, the *ADRL* should now be sorted with regard to the cardinality of its elements, producing the Sorted *ADRL* (i.e. *SADRL*). Because in the example all elements are of the same cardinality, the list may be treated as sorted.

The *SADRL* is finally used in the reduct generation process. A reduct is a subset of attributes/criteria that has a non-empty intersection with all elements reduct may also have non-empty intersection with all elements of SADRL. In the example there exists only one such subset, namely the set $\{c_2, c_3, c_4\}$.

4. The Fast Reduct Generating Algorithm

The actual algorithm used for generating dominance-based reducts is an adaptation of the Fast Reduct Generating Algorithm, FRGA, which is a representative of the family of algorithms based on the notion of discernibility matrix. The modification concerns the first phase of the original algorithm, in which the procedure for creating the Absorbed Discernibility List is substituted with that of creating the *SADRL*.

The algorithm consists of two phases: Phase I creates, absorbs and sorts the list of necessary conflict-preserving attributes, while Phase II generates reducts using the resulting list. The most important part of the algorithm, as far as its computational effectiveness is concerned, is the Fast Prime Implicant Test (FPI) introduced in Susmaga (1998b).

The algorithm is presented in Fig. 1.

The main part of the algorithm is a breadth-first search for minimal subsets of attributes/criteria that have non-empty intersections with all elements of the *SADRL*. The current solution is the set Red_i, which initially contains only one, empty subset, $Red_0 := \{\emptyset\}$. In each iteration i of the loop, the set Red_i is confronted with C_i (the i'th element of the *SADRL*) and split in two disjoint parts: the set of elements that have a non-empty intersection with C_i and the set of those which have not. An element with a non-empty intersection does not have to be modified in any way and is simply stored in S_i. An element with an empty intersection, on the other hand, has to be augmented with successive elements of C_i, giving rise to a family of attribute subsets, the union of which is stored in T_i. Elements of T_i are subsequently checked for minimality by the $FPI(R)$ and those that passed the test are finally stored in Red_{i+1}.

The main computational difficulty of the augmenting procedure is to avoid generating subsets of attributes/criteria that are not minimal with regard to inclusion. It also turns out to be the main challenge of the algorithm, because the algorithm's overall computing time is most strongly influenced by testing for minimality. In a simple approach minimality testing may be achieved by checking if for each $R \in Red_i$ there exists another $R' \in Red_i$ such that $R' \subseteq R$. If so, the set R is not minimal with regard to inclusion and should be discarded from Red_i.

A much better approach consists in discovering redundant attributes/criteria in R (Skowron & Rauszer, 1992; Tannhäuser, 1994). An element $q \in R$ ($R \in Red_i$) is not redundant in R if there exists $C_j \in SADRL$, $j < i$, such that $R \cap C_j = \{q\}$, otherwise the element is redundant. The subset R that contains at least one redundant element is not minimal with regard to inclusion. Compu-

Input: A set of objects U ($|U| = N$);
the objects are described by values of attributes from the set Q.

Output: The set K of all reducts for the set U.

PHASE I—creation of the Sorted, Absorbed Dominance Retaining List (*SADRL*)

Step 1
Create the Dominance Retaining List by storing subsets C_i of those attributes and criteria that retain dominance between all appropriate pairs of objects
The resulting list contains elements $(C_1, C_2, \ldots, C_D)$.

Step 2
Absorb the created Dominance Retaining List by eliminating empty and non-minimal elements from *DRL*:
$ADRL := \{C_i \in DRL : C_i \neq \emptyset,\ C_i$ is unique in $ADRL$
and for no $C_j \in ADRL : C_j \subset C_i\}$.
The resulting, absorbed list contains elements $(C_1, C_2, \ldots, C_d)$, and usually $d \ll D$.

Step 3
Sort the *ADRL* in the ascending order of its element cardinality (create *SADRL*).

PHASE II—-a Breadth-First Search for reducts

Step 1
$Red_0 := \{\emptyset\}$.

Step 2
For every $i = 1..d$ compute:
$S_i := \{R \in Red_{i-1} : R \cap C_i \neq \emptyset\}$.
$T_i := \bigcup_{q \in C_i} \bigcup_{R \in Red_{i-1} : R \cap C_i = \emptyset} \{R \cup \{q\}\}$.
$MIN_i := \{R \in T_i : FPI(R) = true\}$.
$Red_i := S_i \cup MIN_i$.

The final result is $K := R_d$.

Figure 1. The Fast Reduct Generation Algorithm (FRGA)

more effective in practical applications. The testing used in the presented algorithm has been introduced and evaluated in (Susmaga, 1998b) and is briefly characterized below.

Each family F of subsets $R^1, R^2, \ldots, R^P$ ($P = |C_i|$) from T_i has a predecessor $R' \in Red_{i-1}$ such that $R^1 = R' \cup \{q^1\}$, $R^2 = R' \cup \{q^2\}$, $\ldots$, $R^P = R' \cup \{q^P\}$, where the attributes $q^1, q^2, \ldots, q^P$ are elements of C_i ($C_i = \{q^1, q^2, \ldots, q^P\}$). Because in iteration $i-1$ the elements of R_{i-1} all passed the minimality test, that means that for each $q \in R'$ there exists a $C \in SADRL_{1..i-1}$ such that $R' \cap C = \{q\}$. What remains to do is to check if the same holds for each

$R^1, R^2, \ldots, R^P$ the following auxiliary list *Sing* is created:

$$Sing = \{(C_j, a) : C_j \in SADRL_{1..i}, \ |R^j \cap C_j| = \{a\}\}.$$

The test for inclusion minimality for a given R^j ($R^j = R' \cup \{q^j\}$) consists in looking for elements $(C, a) \in Sing$ such that $q^j \notin C$. If $q^j \notin C$ then $R^j \cap C$ remains equal to $\{a\}$, which means that a is not redundant in R^j. If there are no redundant attributes in R^j then R^j is minimal with regard to inclusion, otherwise it is not minimal. Additionally, a set R^j that is a singleton (contains only one element) is always minimal.

More on effective tests of inclusion minimality may be found in Susmaga (1998b, 2000).

5. Generation and application of decision rules in DRSA

The dominance-based rough approximations of upward and downward unions of classes can also serve to induce a generalized description of objects from the information table in terms of '*if* ... *then* ...' decision rules. The common ground for the problems of generating/analysing rules and reducts is that the rules are usually generated using a subset of attributes/criteria and not all of them. Clearly, a non-conflicting subset of attributes/criteria is much-desired for this purpose, so this is where a dominance-based reduct may be directly applied.

Actually, all the details related to assessing and applying the decision rules are beyond the scope of this paper. As a result, the current section introduces the definition of rules only for the sake of completeness of the presentation of Dominance-based Rough Set Approach.

For a given upward or downward union of classes, $Cl_t^{\geq}$ or $Cl_s^{\leq}$, the decision rules induced under a hypothesis that objects belonging to $\underline{P}(Cl_t^{\geq})$ or $\underline{P}(Cl_s^{\leq})$ are *positive* and all the others *negative*, suggest an assignment to "at least class Cl_t" or to "at most class Cl_s", respectively; on the other hand, the decision rules induced under a hypothesis that objects belonging to the intersection $\overline{P}(Cl_s^{\leq}) \cap \overline{P}(Cl_t^{\geq})$ are *positive* and all the others *negative*, are suggesting an assignment to some classes between Cl_s and Cl_t ($s < t$).

Assuming that for each continuously valued $q \in C$, (i.e. V_q is quantitative) and for each $x, y \in U$, $f(x,q) \geq f(y,q)$ implies xS_qy (i.e. V_q is preference-ordered), the following three types of decision rules can be considered:

1. $D_{\geq}$-*decision rules* with the following syntax:
   if $f(x,q_1) \geq r_{q1}$ *and* $f(x,q_2) \geq r_{q2}$ *and* ... $f(x,q_p) \geq r_{qp}$, *then* $x \in Cl_t^{\geq}$, where $P = \{q_1, \ldots, q_p\} \subseteq C$, $(r_{q1}, \ldots, r_{qp}) \in V_{q1} \times V_{q2} \times \ldots \times V_{qp}$ and $t \in T$;
2. $D_{\leq}$-*decision rules* with the following syntax:
   if $f(x,q_1) \leq r_{q1}$ *and* $f(x,q_2) \leq r_{q2}$ *and* ... $f(x,q_p) \leq r_{qp}$, *then* $x \in Cl_t^{\leq}$, where $P = \{q_1, \ldots, q_p\} \subseteq C$, $(r_{q1}, \ldots, r_{qp}) \in V_{q1} \times V_{q2} \times \ldots \times V_{qp}$ and

3. $D_{\geq\leq}$-*decision rules* with the following syntax:
if $f(x,q_1) \geq r_{q1}$ *and* $f(x,q_2) \geq r_{q2}$ *and* $\ldots$ $f(x,q_k) \geq r_{qk}$ *and* $f(x,q_{k+1}) \leq r_{qk+1}$ *and* $\ldots$ $f(x,q_p) \geq r_{qp}$, *then* $x \in Cl_s \cup Cl_{s+1} \cup \ldots \cup Cl_t$, where $O' = \{q_1,\ldots,q_k\} \subseteq C$, $O'' = \{q_{k+1},\ldots,q_p\} \subseteq C$, $P = O' \cup O''$, O' and O'' not necessarily disjoint, $(r_{q1},\ldots,r_{qp}) \in V_{q1} \times V_{q2} \times \ldots \times V_{qp}$, $s,t \in T$ such that $s < t$.

As it is possible that $\{q_1,\ldots,q_k\} \cap \{q_{k+1},\ldots,q_p\} \neq \emptyset$, in the condition part of a $D_{\geq\leq}$-decision rule we can have '$f(x,q) \geq r_q$' and '$f(x,q) \leq r'_q$', where $r_q \leq r'_q$, for some $q \in C$. Moreover, if $r_q = r'_q$, the two conditions boil down to '$f(x,q) = r_q$'.

Since each decision rule is an implication, by a *minimal* decision rule we understand such an implication that there is no other implication with an antecedent (i.e. the *if* part) of at least the same weakness and a consequent (i.e. the *then* part) of at least the same strength.

A set of decision rules is *complete* if it fulfils the following conditions:

- each $y \in \underline{C}(Cl_t^{\geq})$ supports at least one $D_{\geq}$-decision rule of the type '*if* $f(x,q_1) \geq r_{q1}$ *and* $f(x,q_2) \geq r_{q2}$ *and* $\ldots$ $f(x,q_p) \geq r_{qp}$, *then* $x \in Cl_r^{\geq}$', with $r,t \in \{2,\ldots,n\}$ and $r \geq t$,
- each $y \in \underline{C}(Cl_t^{\leq})$ supports at least one $D_{\leq}$-decision rule of the type '*if* $f(x,q_1) \leq r_{q1}$ *and* $f(x,q_2) \leq r_{q2}$ *and* $\ldots$ $f(x,q_p) \leq r_{qp}$, *then* $x \in Cl_u^{\leq}$', with $u,t \in \{1,\ldots,n-1\}$ and $u \leq t$,
- each $y \in \overline{C}(Cl_s^{\leq}) \cap \overline{C}(Cl_t^{\geq})$ supports at least one $D_{\geq\leq}$-decision rule of the type '*if* $f(x,q_1) \geq r_{q1}$ *and* $f(x,q_2) \geq r_{q2}$ *and* $\ldots$ $f(x,q_k) \geq r_{qk}$ *and* $f(x,q_{k+1}) \geq r_{qk+1}$ *and* $\ldots$ $f(x,q_p) \geq r_{qp}$, *then* $x \in Cl_v \cup Cl_{v+1} \cup \ldots \cup Cl_z$', with $s,t,v,z \in T$ and $s \leq v < z \leq t$.

Let us remark that application of any complete set of decision rules on the objects from the information table results in either exact or approximate reassignment of these objects to the classes Cl_t, $t \in T$. This may be explained in greater detail as follows. For an object $x \in U$, reassignment means the intersection of all unions of classes suggested by the consequents of rules matched (supported) by x. Given a complete set of rules, and an object $y \in U$, such that $y \notin Bn_C(Cl_s^{\leq})$ and $y \notin Bn_C(Cl_s^{\geq})$ for any $s \in T$, the following situations may occur:

- $y \in Cl_t$, $t = 2,\ldots,n-1$; then there exists at least one $D_{\geq}$-decision rule whose consequent is $x \in Cl_t^{\geq}$, and at least one $D_{\leq}$-decision rule whose consequent is $x \in Cl_t^{\leq}$;
- $y \in Cl_1$; then there exists at least one $D_{\leq}$-decision rule whose consequent is $Cl_1^{\leq}$;
- $y \in Cl_n$; then there exists at least one $D_{\geq}$-decision rule whose consequent is $x \in Cl_n^{\geq}$.

In all the above situations, application of the complete set of rules to object y will result in exact reassignment of y to class Cl_t. Similarly, for each object

$t \leq [<]\, t1$, which means that y belongs exclusively to boundaries $Bn_C(Cl_v^{\geq})$, $v = s+1, \ldots, t$, and $Bn_C(Cl_z^{\leq})$, $z = s, \ldots, t-1$, there exists at least one $D_{\geq\leq}$-decision rule whose consequent is $x \in Cl_s \cup Cl_{s+1} \cup \ldots \cup Cl_t$. Thus, in result of application of the complete set of rules to object y, the object will be assigned (approximately) to classes $Cl_s \cup Cl_{s+1} \cup \ldots \cup Cl_t$.

A set of minimal decision rules is called *minimal* if it is complete and non-redundant, i.e. exclusion of any rule from this set makes it non-complete. Many induction strategies can be adopted to obtain a set of decision rules (e.g. Stefanowski & Vanderpooten, 1994):

- generation of a *minimal* description, i.e. a minimal set of rules,
- generation of an *exhaustive* description, i.e. all possible minimal rules for a given information table,
- generation of a characteristic *description*, i.e. a set of minimal rules covering relatively many objects each, but in total not necessarily all objects from U.

It is worth mentioning that the reduct generating algorithms may be easily adapted to the task of generating exact decision rules. Especially adaptation to the problem of generating exhaustive rule sets is easy. This is because reducts and rules are, from a certain point of view, a global and a local solution to the same problem (which may be formulated either in CRSA or DRSA).

Let us now consider both reducts and rules in CRSA. The set of all reducts is in fact the set of all minimal subsets of attributes that allow to distinguish all objects belonging to lower approximations of different classes from other objects. In this sense it is a global problem.

When generating exhaustive sets of decision rules, on the other hand, we are looking for minimal subsets of conditions that allow to distinguish particular objects, provided that these objects belong to lower approximations of classes, from all objects belonging to different lower approximations of class unions. So this is a kind of local problem, as it regards a given object. In consequence, the computation must be repeated for each object from any lower approximation (the resulting rules must also be finally checked for minimality, but this does not influence the way they are generated).

Despite some differences between generating reducts and rules, it is important to stress that the main computational mechanism is actually the same—searching for minimal subsets of attributes that maintain discernibility between pre-specified objects. The seemingly impairing incompatibility, namely the fact that in generating rules we are looking for subsets of conditions defined on attributes and not for subsets of attributes, may be easily solved by the introduction of attribute binarization (Ziarko & Shan, 1995). After binarization, each attribute in the binarized information table corresponds to an ⟨attribute,value⟩ pair in the original table. As such, the pair univocally represents a simple condition, which is the basic building block of a decision rule. In result, from the decision rules may be viewed as the same problem and, as such, solved using similar algorithms.

6. Experimental evaluation of the Fast Reduct Generation Algorithm

The computational capabilities of the presented algorithm are best illustrated with results of an experiment. The experiment focused on generating reducts (rather than decision rules), as from the computational point of view it is a more demanding problem. The data sets used in this experiment were real-life data sets of various origin, created for scientific purposes and used in different experiments and analyses (e.g. Stefanowski & Słowiński, 1997).

Table 3 presents the characteristics of the data sets. There are basically only four different data sets, but as many as 22 various configurations of condition

Data Set Conf.	#Objects	#Classes	#Attr+Crit	#Attr	#Crit
Cars-ac/0	159	6	43	28	15
Cars-ac/1	159	6	43	12	31
Cars-ac/2	159	6	43	8	35
Cars-ac/3	159	6	43	8	35
Cars-ac/4	159	6	43	16	27
Cars-c	159	6	43	0	43
Urod2-a	343	2	33	33	0
Urod2-ac/1	343	2	33	3	30
Urod2-ac/2	343	2	33	6	27
Urod2-c	343	2	33	0	33
Livdpl-a	80	2	22	22	0
Livdpl-ac	80	2	22	19	3
Livdpl-c	80	2	22	0	22
Eswl-a/1	500	2	28	33	0
Eswl-ac/1	500	2	28	25	3
Eswl-c/1	500	2	28	0	33
Eswl-a/2	500	3	28	33	0
Eswl-ac/2	500	3	28	25	3
Eswl-c/2	500	3	28	0	33
Eswl-a/3	500	6	28	33	0
Eswl-ac/3	500	6	28	25	3
Eswl-c/3	500	6	28	0	33

Table 3. Characteristics of the data sets configurations

attributes and criteria were constructed. The column *Data Set/Dec. No* denotes the data set configuration. The exact number of attributes and criteria in a particular configurations is given in columns *#Attr* and *#Crit.*

It must be also finally stressed that the experiments were conducted with algorithm and that no claim is made as to any potential applications of these particular results. As it can be observed, in some data configurations the cardinality of the total outcome reaches thousands of reducts. Analysis of such a number of reducts would be difficult in most applications. In such difficult cases the search for reducts should be accompanied by additional constraints concerning the reducts to be generated.

An approach in which subsets of constrained reducts may be generated is presented in e.g. (Susmaga, 1999).

It is additionally worth noting that changing the preference-neutral attributes into criteria decreases the number of generated reducts.

Table 4 presents the results of reduct generation in form of the number of reducts computed for each data configuration and the computing time in seconds of CPU. Entries '######' denote that a particular algorithm did not

Data Set Conf.	#Reducts	FRGA [s]	RGA [s]	ROM [s]
Cars-ac/0	26519	1.63E+01	4.77E+02	######
Cars-ac/1	11246	8.90E+00	1.58E+02	######
Cars-ac/2	6659	7.27E+00	1.72E+02	######
Cars-ac/3	7228	7.87E+00	1.73E+02	######
Cars-ac/4	11004	1.01E+01	3.31E+02	######
Cars-c	4297	5.97E+00	9.57E+01	######
Urod2-a	38207	9.65E+01	######	######
Urod2-ac/1	11918	2.37E+01	3.26E+03	######
Urod2-ac/2	7873	1.78E+01	1.31E+03	######
Urod2-c	4	9.13E+00	9.35E+00	######
Livdpl-a	1295	1.30E+00	6.62E+01	######
Livdpl-ac	1058	1.12E+00	4.06E+01	######
Livdpl-c	47	4.80E–01	5.10E–01	4.47E+02
Eswl-a/1	963	1.43E+01	4.24E+01	######
Eswl-ac/1	210	1.34E+01	1.40E+01	######
Eswl-c/1	1	1.34E+01	1.35E+00	2.17E+03
Eswl-a/2	809	1.44E+01	2.30E+01	######
Eswl-ac/2	166	1.42E+01	1.46E+01	######
Eswl-c/2	1	1.42E+01	1.41E+00	1.86E+03
Eswl-a/3	587	1.50E+01	2.04E+01	######
Eswl-ac/3	119	1.47E+01	1.60E+01	######
Eswl-c/3	1	1.41E+01	1.39E+01	1.91E+03

Table 4. Computing times [in seconds of the CPU] of three reduct generating algorithms for each of the data set configuration. Entries '######' mean that the computing time exceeds the assumed limit of 1.00E+04 seconds.

terminate in reasonable time of 10,000 seconds of CPU, i.e. almost 3 hours, and was terminated. The actual computing time remains in this case unknown (but it certainly exceeds 10,000 seconds). The column *FRGA* presents the compaper. As a comparison, the column *RGA* presents the times obtained by another algorithm of the discernibility matrix family (Skowron & Rauszer, 1992; Tannhäuser, 1994), and the column *ROM* presents the times obtained by a completely different reduct generating algorithm, which was described in Romański (1988).

It is interesting that both the discernibility based algorithms outperform strongly the other approach. Out of the two, the FRGA seems to be noticeably better, which is especially clear when more difficult data sets are analyzed.

The computing platform in all the experiments was a SUN SPARCstation running at 110 MHz.

7. Conclusions

The purpose of this paper was to introduce and experimentally evaluate a very promising algorithm for generating all reducts in the framework of the new, Dominance-based Rough Set Approach. The DRSA is an interesting alternative to CRSA, over which it manifests several advantages.

The first one is the ability of handling criteria, preference-ordered classes and inconsistencies in the set of decision examples, which CRSA was not able to discover, namely inconsistencies in the sense of violation of the dominance principle. The second advantage is the ability to analyse information tables with continuously valued criteria without explicit pre- discretization phase. This property is very important because the process of discretization, depending on the actual method employed, may lead to different and not always consistent results of the CRSA (Słowiński & Słowiński, 1990). The third advantage of the DRSA lies in a richer syntax of decision rules induced from rough approximations, which may now include conditions involving both equality as well as inequality. Last but not least, the syntax of the DRSA rules is much more comprehensible to practitioners and makes the representation of knowledge more synthetic since minimal sets of decision rules are usually smaller than minimal sets of decision rules resulting from CRSA.

The presented reduct generated algorithm is general enough to compute reducts both in the CRSA as well as in the DRSA. It is also a very fast algorithm and undoubtedly outperforms other methods as far as the computing times are concerned.

Future research concerning the same domain will be directed towards developing further effective algorithms in the dominance-based framework. This concerns, first of all, adaptation of the presented algorithm for generating reducts to the problem of generating decision rules and a proper experimental evaluation of the algorithm.

Acknowledgements. The research of the first two authors has been supported by the State Committee for Scientific Research (KBN), research grant port from the Italian Ministry of University and Scientific Research (MURST). Most of experimenting required for preparing and creating this paper was conducted at the Supercomputing and Networking Centre of Poznań, Poland.

References

CHMIELEWSKI, M. and GRZYMALA-BUSSE, J. (1995) Global discretization of continuous attributes as preprocessing for machine learning. In: Lin, T.Y. and Wildberger, A. (eds.), *Soft Computing*. Simulation Councils Inc., San Diego, CA, 294–301.

DIMITRAS, A.I., ZANAKIS, S. and ZOPOUNIDIS, C. (1996) A survey of business failures with an emphasis on prediction methods and industrial applications. *European Journal of Operational Research*, **90**, 487–513.

FAYYAD, U.M. and IRANI, K.B. (1992) On handling the continuous-valued attributes in decision tree generation. *Machine Learning*, **8**, 87–102.

GRECO, S., MATARAZZO, B. and SŁOWIŃSKI, R. (1998a) A new rough set approach to evaluation of bankruptcy risk. In: Zopounidis, C. (ed.), *Operational Tools in the Management of Financial Risk*. Kluwer Academic Publishers, Dordrecht, 121–136.

GRECO, S., MATARAZZO, B. and SŁOWIŃSKI, R. (1998b) A new rough set approach to multicriteria and multiattribute classification. In: Polkowski, L. and Skowron, A. (eds.), *Rough Sets and Current Trends in Computing*. LNAI 1424, Springer-Verlag, Berlin, 60–67.

GRECO, S., MATARAZZO, B. and SŁOWIŃSKI, R. (1998c) Rough approximation of a preference relation in a pairwise comparison table. In: Polkowski, L. and Skowron, A. (eds.), *Rough Sets in Knowledge Discovery*, Vol. 2. Physica-Verlag, Heidelberg/New York, 13–36.

GRECO, S., MATARAZZO, B. and SŁOWIŃSKI, R. (1999a) Rough Approximation of Preference Relation by Dominance Relations. *European Journal of Operational Research*, **117**, No. 2, 63–83.

GRECO, S., MATARAZZO, B. and SŁOWIŃSKI, R. (1999b) The use of rough sets and fuzzy sets in MCDM. Chapter 14 in: Gal, T., Hanne, T. and Stewart, T. (eds), *Advances in Multiple Criteria Decision Making*. Kluwer Academic Publishers, Boston/Dordrecht/London, 14.1–14.59.

NGUYEN, H.S. (1998) Discretization problems for rough set methods. In: Polkowski, L. and Skowron, A. (eds.), *Rough Sets and Current Trends in Computing*. LNAI 1424, Springer-Verlag, Berlin, 545–552.

PAWLAK, Z. (1982) Rough sets. *International Journal of Information & Computer Sciences*, **11**, 341–356.

PAWLAK, Z. (1991) *Rough Sets. Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Dordrecht.

ROMAŃSKI, S. (1988) Operations on Families of Sets for Exhaustive Search Given a Monotonic Function. In: Beeri, C., Smith, J.W. and Dayal, U. (eds.), *Proceedings of the 3rd International Conference on Data and Knowl-*

ROY, B. (1985) *Méthodologie Multicritère d'Aide à la Décision.* Economica, Paris.

SKOWRON, A. and RAUSZER, C. (1992) The Discernibility Functions Matrices and Functions in Information Systems. In: Słowiński, R. (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory.* Kluwer Academic Publishers, Dordrecht, 331–362.

SŁOWIŃSKI, R. (ed.) (1992) *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publishers, Dordrecht.

SŁOWIŃSKI, K. and SŁOWIŃSKI, R. (1990) Sensitivity analysis of rough classification. *International Journal of Man-Machine Studies*, **32**, 693–705.

SŁOWIŃSKI, R. and VANDERPOOTEN, D. (1997) Similarity relation as a basis for rough approximations. In: Wang, P.P. (ed.), *Advances in Machine Intelligence & Soft Computing.* Duke University Press, Durham, NC, 17–33.

SŁOWIŃSKI, R. and VANDERPOOTEN, D. (1999) A generalised definition of rough approximations. *IEEE Transactions on Data and Knowledge Engineering*, in press.

SŁOWIŃSKI, R. and ZOPOUNIDIS, C. (1995) Application of the rough set approach to evaluation of bankruptcy risk. *Intelligent Systems in Accounting, Finance and Management*, **4**, 27–41.

STEFANOWSKI, J. and SŁOWIŃSKI, K. (1997) Rough Sets a as Tool for Studying Attribute Dependencies in the Urinary Stones Treatment Data. In: Lin, T.Y. and Cercone, N. (eds.), *Rough Set and Data Mining.* Kluwer Academic Publishers, Dordrecht, 177–198.

STEFANOWSKI, J. and VANDERPOOTEN, D. 1994) A general two-stage approach to inducing rules from examples. In: Ziarko, W. (ed.), *Rough Sets, Fuzzy Sets and Knowledge Discovery.* Springer-Verlag, Berlin, 317–325.

SUSMAGA, R. (1998a) Experiments in Incremental Computation of Reducts. In: Skowron, A. and Polkowski, L. (eds.), *Rough Sets in Data Mining and Knowledge Discovery.* Springer-Verlag, Berlin, 530–553.

SUSMAGA, R. (1998b) Effective Tests for Minimality in Reduct Generation. *Foundations of Computing and Decision Sciences*, Poznań, Poland, **23**, No. 4, 219–240.

SUSMAGA, R. (1999) Computation of Minimal Cost Reducts. In: Raś, Z. and Skowron, A. (eds.), *Foundations of Intelligent Systems.* Lecture Notes in Artificial Intelligence, **1609**, Springer-Verlag, Berlin, 448–456.

SUSMAGA, R. (2000) New Tests for Inclusion Minimality in Reduct Generation. *Foundations of Computing and Decision Sciences*, **25**, No. 2, 121–146.

SUSMAGA, R., SŁOWIŃSKI, R., GRECO, S. and MATARAZZO, B. (1999) Computation of reducts for multi-attribute and multi-criteria classification. In: Kłopotek, M. and Michalewicz, M. (eds.), *Proceedings of the VIIIth Workshop on Intelligent Information Systems*, Ustroń, Poland, June 14–

TANNHÄUSER, M. (1994) Efficient Reduct Computation (in Polish). M.Sc. thesis, Institute of Mathematics, Warsaw University, Warsaw.

ZIARKO, W. and SHAN, N. (1995) KDD R: A Comprehensive System for Knowledge Discovery in Databases Using Rough Sets. In: Lin, T.Y. and Wildberger, A.M. (eds.), *Soft Computing*. Society for Computer Simulation, San Diego, 298–301.