

# jRank – Ranking using Dominance-based Rough Set Approach

User's manual

# jRank – Ranking using Dominance-based Rough Set Approach

<http://www.cs.put.poznan.pl/mszelag/Software/jRank/jRank.html>

Marcin Szelaḡ, Roman Słowiński, Salvatore Greco, Jerzy Błaszczyszński, Szymon Wilk  
marcin.szelaḡ@cs.put.poznan.pl

July 18, 2013

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Basic Concepts of Dominance-based Rough Set Approach</b>	<b>3</b>
2.1	Problem Setting . . . . .	3
2.2	Pairwise Comparison Table (PCT) . . . . .	4
2.3	Rough Approximation of Preference Relations by Dominance Relation . . . . .	4
2.4	Quality of Approximation . . . . .	5
2.5	Induction of Decision Rules . . . . .	6
2.6	Application of Decision Rules and Exploitation of Preference Graphs . . . . .	7
2.7	Variable Consistency Dominance-based Rough Set Approaches . . . . .	8
<b>3</b>	<b>How to Use jRank</b>	<b>9</b>
3.1	Running jRank . . . . .	9
3.2	Illustrative Example . . . . .	13
3.2.1	Data File(s) . . . . .	13
3.2.2	Preference Information . . . . .	14
3.2.3	Experiment Configuration File . . . . .	14
3.2.4	Running jRank for the Illustrative Example . . . . .	15
3.2.5	Pairwise Comparison Table (PCT) . . . . .	16
3.2.6	Approximations of Preference Relations . . . . .	16
3.2.7	Induced Decision Rules . . . . .	17
3.2.8	Preference Graph . . . . .	18
3.2.9	Final Ranking . . . . .	18
3.2.10	Ranking Errors . . . . .	19
<b>4</b>	<b>Exemplary Applications of Dominance-based Rough Set Approach</b>	<b>19</b>
<b>5</b>	<b>Glossary</b>	<b>20</b>

## 1 Introduction

jRank is a decision support tool for solving multi-criteria choice and ranking problems. It is a command line Java application, based on java Rough Set (jRS) library, which implements methods of data analysis provided by the Dominance-based Rough Set Approach and Variable Consistency Dominance-based Rough Set Approaches. In the following Section 2, we describe the methodology applied to choice and ranking problems. Section 3 describes how to use jRank in general and for a particular multi-criteria ranking problem. Applications of DRSA can be found in Section 4 and in the surveys [17, 18, 19, 37, 39].

## 2 Basic Concepts of Dominance-based Rough Set Approach

Dominance-based Rough Set Approach (DRSA) is defined for problems with background knowledge about ordinal evaluations of objects from a universe, and about monotonic relationships between these evaluations, e.g., “the larger the mass and the smaller the distance, the larger the gravity” or “the greater the debt of a firm, the greater its risk of failure”. Precisely, the monotonic relationships are assumed between evaluation of objects on condition attributes and their evaluation on decision attribute. The monotonic relationships are also interpreted as monotonicity constraints, because the better the evaluation of an object, the better should be the decision class the object is assigned to. For this reason, classification problems of this kind are called ordinal classification problems with monotonicity constraints. Many real-world classification problems fall into this category. Typical examples are multiple criteria sorting and decision under uncertainty, where the order of value sets of attributes corresponds to increasing or decreasing order of preference of a decision maker. In these decision problems, the condition attributes are called *criteria*.

Although DRSA is a general methodology for reasoning about data describing ordinal classification problems with monotonicity constraints, in this manual, we shall use the vocabulary typical for multiple criteria choice and ranking problems [38].

### 2.1 Problem Setting

The data are given as an *information table*, where rows correspond to objects from a finite set  $A$  and columns correspond to attributes from a finite set  $C$ . The value set of attribute  $q_i \in C$  is denoted by  $V_{q_i}$ . Among attributes from  $C$  there has to be at least one *criterion*, i.e., an attribute with value set ordered according to some domain knowledge. We distinguish *cardinal criteria* (with values expressed on some interval or ratio scale) and *ordinal criteria* (with values expressed on some ordinal scale). The remaining attributes are called *regular attributes* (with values on some nominal scale).

The goal is to rank objects from set  $A$  from the best to the worst (i.e., to create a weak order of objects), or to choose the best objects.

Let us observe that the only objective information that can be used to compare objects is the *dominance relation*. However it is often the case that the dominance relation leaves many objects *incomparable*. Therefore, there is a need to acquire additional information that allows to compare considered objects pairwise. This information might be, e.g., given by the Decision Maker (DM), who comprehensively compares chosen objects from set  $A$ . We assume two possible forms of such *preference information*: a) pairwise comparisons of some objects from  $A$ , b) ranking (weak order) on some  $A^R \subseteq A$ . For example, let us consider the information table presented as Table 1.

Table 1: Exemplary information table with evaluations of house locations

Location	$q_1$ – Distance	$q_2$ – Price	$q_3$ – Comfort
$L_1$ – Poznan	3	60	Good
$L_2$ – Kapalica	35	30	Good
$L_3$ – Krakow	7	85	Medium
$L_4$ – Warszawa	10	90	Basic
$L_5$ – Wroclaw	5	60	Medium
$L_6$ – Malbork	50	50	Medium
$L_7$ – Gdansk	5	70	Medium

In this table, attributes Distance and Price are cardinal criteria (of cost type, i.e., the smaller the value, the better), while attribute Comfort is an ordinal criterion.

The pairwise comparisons of the DM concern assignment of pairs of objects from  $A$  to *preference relations*. We consider the simplest case, where there are two such relations:  $S$  – comprehensive outranking, and  $S^c$  – comprehensive non-outranking. Thus, the preference information in the form  $L_1 S L_6$ ,  $L_2 S L_4$ , and  $L_7 S^c L_1$  (which is equivalent to  $(L_1, L_6) \in S$ ,  $(L_2, L_4) \in S$ , and  $(L_7, L_1) \in S^c$ ) is read as: location  $L_1$  is at least as good as

location  $L_6$ , location  $L_2$  is at least as good as location  $L_4$ , and location  $L_7$  is NOT at least as good as location  $L_1$ .

The ranking given by the DM could be:  $L_1, L_7, L_2$  &  $L_6$ , which boils down to the following pairwise comparisons:  $L_1 S L_7, L_1 S L_2, L_1 S L_6, L_7 S L_2, L_7 S L_6, L_2 S L_6, L_6 S L_2$ , and  $L_6 S^c L_1, L_6 S^c L_7, L_2 S^c L_1, L_2 S^c L_7, L_7 S^c L_1$ .

In the following, we will denote by  $B \subseteq A \times A$  a set of pairs of objects for which the DM expressed his/her preferences, i.e., which were assigned by the DM either to  $S$  or to  $S^c$  preference relation.

## 2.2 Pairwise Comparison Table (PCT)

Pairwise comparisons of objects from set  $A$  are the basis for creation of the Pairwise Comparison Table (PCT). In order to built such a table, one calculates a difference of evaluations for each cardinal criterion. For ordinal criteria, as well as for regular attributes, one considers ordered pairs of evaluations. For the information table presented as Table 1 and for pairwise comparisons:  $L_1 S L_6, L_2 S L_4$ , and  $L_7 S^c L_1$ , one gets the following PCT:

Table 2: PCT for the house location problem

$(X, Y)$	$\Delta_{q_1}$	$\Delta_{q_2}$	$(q_3(X), q_3(Y))$	Relation
$(L_1, L_6)$	-47	10	(Good,Medium)	$S$
$(L_2, L_4)$	25	-60	(Good,Basic)	$S$
$(L_7, L_1)$	2	10	(Medium,Good)	$S^c$

Let us observe that attributes  $\Delta_{q_1}$  and  $\Delta_{q_2}$  from Table 2 are cost-type cardinal criteria since attributes  $q_1$  and  $q_2$  from Table 1 are cost-type cardinal criteria. Third attribute from Table 2 is no longer an ordinal criterion and it requires special treatment to enable comparison of ordered pairs of evaluations. This shall be explained in the following section.

Remark that the PCT is a decision table which includes examples of ordinal classification. The decision classes are relations  $S$  and  $S^c$ . Among attributes of PCT we have a set of *condition attributes* and one decision attribute  $d$ . Thus, this table constitutes an input preference information to be analyzed with the help of DRSA.

## 2.3 Rough Approximation of Preference Relations by Dominance Relation

Using DRSA, we want to represent (approximate) outranking and non-outranking relations in PCT by *granules of knowledge* generated by attributes. These granules are *dominance cones* in the attribute values space.

Let us consider set  $P \subseteq C$  of attributes. Moreover, let us distinguish three subsets of  $P$  denoted by  $P^N, P^O$ , and  $P^R$ , composed of cardinal criteria, ordinal criteria, and regular attributes, respectively.

Pair of objects  $(x, y)$  *dominates* pair of objects  $(w, z)$  with respect to (w.r.t.)  $P \subseteq C$  (shortly,  $(x, y)$   $P$ -dominates  $(w, z)$ ; denotation  $(x, y)D_P(w, z)$ ) if and only if (iff):

$$\begin{aligned} (x, y)D_{P^N}(w, z) &\Leftrightarrow \forall q_i \in P^N : \Delta_i(x, y) \succeq \Delta_i(w, z) \wedge \\ (x, y)D_{P^O}(w, z) &\Leftrightarrow \forall g_i \in P^O : g_i(x) \succeq g_i(w) \wedge g_i(y) \preceq g_i(z) \wedge \\ (x, y)I_{P^R}(w, z) &\Leftrightarrow \forall g_i \in P^R : g_i(x) = g_i(w) \wedge g_i(y) = g_i(z), \end{aligned}$$

where  $D_{P^N}, D_{P^O}$ , and  $I_{P^R}$  denotes dominance w.r.t. set  $P^N$ , dominance w.r.t. set  $P^O$ , and indiscernibility w.r.t. set  $P^R$ , respectively;  $\succeq$  and  $\preceq$  denote weak preference relation and inverse weak preference relation, respectively.

The dominance relation is reflexive and transitive, i.e., it is a partial preorder.

Given a set of attributes  $P \subseteq C$  and pair  $(x, y) \in A \times A$ , the granules of knowledge used for approximation in DRSA are:

- a set of pairs of objects dominating  $(x, y)$ , called  *$P$ -dominating set*,  
 $D_P^+(x, y) = \{(w, z) \in A \times A : (w, z)D_P(x, y)\},$

- a set of pairs of objects dominated by  $(x, y)$ , called *P-dominated set*,  
 $D_P^-(x, y) = \{(w, z) \in A \times A : (x, y) D_P(w, z)\}$ .

Let us recall that the dominance principle requires that a pair of objects  $(x, y)$  dominating pair  $(w, z)$  should also dominate  $(w, z)$  on the decision (i.e.,  $(x, y)$  should be assigned to at least as good decision class as  $(w, z)$ ). Pairs of objects satisfying the dominance principle are called *consistent*, and those which violate this principle are called *inconsistent*. Thus, inconsistency w.r.t. the dominance principle is observed when pair of objects  $(x, y) \in S^c$  dominates pair of objects  $(w, z) \in S$ .

The *P-lower approximation* of  $S$ , denoted by  $\underline{P}(S)$ , and the *P-lower approximation* of  $S^c$ , denoted by  $\underline{P}(S^c)$ , are defined as follows:

$$\begin{aligned}\underline{P}(S) &= \{(x, y) \in B : D_P^+(x, y) \subseteq S\}, \\ \underline{P}(S^c) &= \{(x, y) \in B : D_P^-(x, y) \subseteq S^c\}.\end{aligned}\tag{1}$$

The *P-upper approximation* of  $S$ , denoted by  $\overline{P}(S)$ , and the *P-upper approximation* of  $S^c$ , denoted by  $\overline{P}(S^c)$ , are defined as follows:

$$\begin{aligned}\overline{P}(S) &= \bigcup_{(x, y) \in S} D_P^+(x, y), \\ \overline{P}(S^c) &= \bigcup_{(x, y) \in S^c} D_P^-(x, y).\end{aligned}\tag{2}$$

The *P-boundary* of  $S$ , denoted by  $Bn_P(S)$ , and the *P-boundary* of  $S^c$ , denoted by  $Bn_P(S^c)$ , are defined as follows:

$$\begin{aligned}Bn_P(S) &= \overline{P}(S) - \underline{P}(S), \\ Bn_P(S^c) &= \overline{P}(S^c) - \underline{P}(S^c).\end{aligned}\tag{3}$$

The *P-lower* and *P-upper* approximations so defined satisfy the following *inclusion property*, for all  $P \subseteq C$ :

$$\begin{aligned}\underline{P}(S) \subseteq S \subseteq \overline{P}(S) \\ \underline{P}(S^c) \subseteq S^c \subseteq \overline{P}(S^c)\end{aligned}$$

The *P-lower* and *P-upper* approximations of  $S$  and  $S^c$  have an important *complementarity property*, according to which:

$$\begin{aligned}\underline{P}(S) = A - \overline{P}(S^c) \quad \text{and} \quad \overline{P}(S) = A - \underline{P}(S^c), \\ \underline{P}(S^c) = A - \overline{P}(S) \quad \text{and} \quad \overline{P}(S^c) = A - \underline{P}(S).\end{aligned}$$

Due to the above complementarity property,  $Bn_P(S) = Bn_P(S^c)$ .

## 2.4 Quality of Approximation

For every  $P \subseteq C$ , the *quality of approximation* of the classification  $\mathbf{Cl} = \{S^c, S\}$  by set of attributes  $P$  is defined as the ratio of the number of pairs of objects *P-consistent* with the dominance principle and the number of all the pairs of objects in  $B$ . Since the *P-consistent* pairs of objects are those which do not belong to the *P-boundary*  $Bn_P(S) = Bn_P(S^c)$ , the quality of approximation of the ordinal classification  $\mathbf{Cl}$  by set of attributes  $P$  can be written as:

$$\gamma_P(\mathbf{Cl}) = \frac{|\underline{P}(S) \cup \underline{P}(S^c)|}{|B|}.$$

In particular,  $\gamma_C(\mathbf{Cl})$  can be seen as a degree of consistency of the pairs of objects from  $B$ .

Moreover, for every  $P \subseteq C$ , the *accuracy of approximation* of relation  $S$  or  $S^c$  by set of attributes  $P$  is defined as the ratio of the number of pairs of objects belonging to the *P-lower* approximation and the *P-upper* approximation of a given relation. Accuracy of approximation  $\alpha_P(S)$ ,  $\alpha_P(S^c)$  can be written as:

$$\alpha_P(S) = \frac{|\underline{P}(S)|}{|\overline{P}(S)|}, \quad \alpha_P(S^c) = \frac{|\underline{P}(S^c)|}{|\overline{P}(S^c)|}.$$

## 2.5 Induction of Decision Rules

The dominance-based rough approximations of outranking and non-outranking relations can serve to induce a generalized description of pairs of objects from PCT in terms of “*if ... , then ...*” decision rules. The decision rules induced under a hypothesis that pairs of objects belonging to  $\underline{P}(S)$  (or  $\underline{P}(S^c)$ ) are positive examples, and all the others are negative ones, suggest a *certain* outranking relation (or *certain* non-outranking relation, respectively). On the other hand, the rules induced under a hypothesis that pairs of objects belonging to  $\overline{P}(S)$  (or  $\overline{P}(S^c)$ ) are positive examples, and all the others are negative ones, suggest a *possible* outranking relation (or *possible* non-outranking relation, respectively). Finally, rules induced under a hypothesis that pairs of objects belonging to boundary  $Bn_P(S) = Bn_P(S^c)$  are positive examples, and all the others are negative ones, are inconclusive, since they suggest an assignment to either  $S$  or  $S^c$ .

Let us consider set  $P \subseteq C$  of attributes. Let  $P = P^N \cup P^O \cup P^R$ , as in section 2.3. Moreover, let  $P = \{q_{i1}, \dots, q_{iz}\}$ ,  $P^N = \{q_{i1}, \dots, q_{ie}\}$ ,  $P^O = \{q_{ie+1}, \dots, q_{ip}\}$ , and  $P^R = \{q_{ip+1}, \dots, q_{iz}\}$ ,  $r_{i1}, \dots, r_{iz} \in \mathfrak{R}$ ,  $s_{ie+1}, \dots, s_{iz} \in \mathfrak{R}$ . It is meaningful to consider the following five types of decision rules:

- 1) *certain  $D_{\geq}$ -decision rules*, providing lower profile descriptions for objects belonging to  $\underline{P}(S)$ :  
if  $q_{i1}(x) - q_{i1}(y) \succeq r_{i1}$  and ... and  $q_{ie}(x) - q_{ie}(y) \succeq r_{ie}$  and  $q_{ie+1}(x) \succeq r_{ie+1}$  and  $q_{ie+1}(y) \preceq s_{ie+1}$  and ... and  $q_{ip}(x) \succeq r_{ip}$  and  $q_{ip}(y) \preceq s_{ip}$  and  $q_{ip+1}(x) = r_{ip+1}$  and  $q_{ip+1}(y) = s_{ip+1}$  and ... and  $q_{iz}(x) = r_{iz}$  and  $q_{iz}(y) = s_{iz}$ , then  $xSy$ ;
- 2) *possible  $D_{\geq}$ -decision rules*, providing lower profile descriptions for objects belonging to  $\overline{P}(S)$ :  
if  $q_{i1}(x) - q_{i1}(y) \succeq r_{i1}$  and ... and  $q_{ie}(x) - q_{ie}(y) \succeq r_{ie}$  and  $q_{ie+1}(x) \succeq r_{ie+1}$  and  $q_{ie+1}(y) \preceq s_{ie+1}$  and ... and  $q_{ip}(x) \succeq r_{ip}$  and  $q_{ip}(y) \preceq s_{ip}$  and  $q_{ip+1}(x) = r_{ip+1}$  and  $q_{ip+1}(y) = s_{ip+1}$  and ... and  $q_{iz}(x) = r_{iz}$  and  $q_{iz}(y) = s_{iz}$ , then possibly  $xSy$ ;
- 3) *certain  $D_{\leq}$ -decision rules*, providing upper profile descriptions for objects belonging to  $\underline{P}(S^c)$ :  
if  $q_{i1}(x) - q_{i1}(y) \preceq r_{i1}$  and ... and  $q_{ie}(x) - q_{ie}(y) \preceq r_{ie}$  and  $q_{ie+1}(x) \preceq r_{ie+1}$  and  $q_{ie+1}(y) \succeq s_{ie+1}$  and ... and  $q_{ip}(x) \preceq r_{ip}$  and  $q_{ip}(y) \succeq s_{ip}$  and  $q_{ip+1}(x) = r_{ip+1}$  and  $q_{ip+1}(y) = s_{ip+1}$  and ... and  $q_{iz}(x) = r_{iz}$  and  $q_{iz}(y) = s_{iz}$ , then  $xS^c y$ ;
- 4) *possible  $D_{\leq}$ -decision rules*, providing upper profile descriptions for objects belonging to  $\overline{P}(S^c)$ :  
if  $q_{i1}(x) - q_{i1}(y) \preceq r_{i1}$  and ... and  $q_{ie}(x) - q_{ie}(y) \preceq r_{ie}$  and  $q_{ie+1}(x) \preceq r_{ie+1}$  and  $q_{ie+1}(y) \succeq s_{ie+1}$  and ... and  $q_{ip}(x) \preceq r_{ip}$  and  $q_{ip}(y) \succeq s_{ip}$  and  $q_{ip+1}(x) = r_{ip+1}$  and  $q_{ip+1}(y) = s_{ip+1}$  and ... and  $q_{iz}(x) = r_{iz}$  and  $q_{iz}(y) = s_{iz}$ , then possibly  $xS^c y$ ;
- 5) *approximate  $D_{\leq}$ -decision rules*, providing simultaneously lower and upper profile descriptions for objects belonging to  $Bn_P(S) = Bn_P(S^c)$ :  
if  $q_{i1}(x) - q_{i1}(y) \succeq r_{i1}$  and ... and  $q_{ic}(x) - q_{ic}(y) \succeq r_{ic}$  and  $q_{ic+1}(x) - q_{ic+1}(y) \preceq r_{ic+1}$  and ... and  $q_{ie}(x) - q_{ie}(y) \preceq r_{ie}$  and  $q_{ie+1}(x) \succeq r_{ie+1}$  and  $q_{ie+1}(y) \preceq s_{ie+1}$  and ... and  $q_{ik}(x) \succeq r_{ik}$  and  $q_{ik}(y) \preceq s_{ik}$  and  $q_{ik+1}(x) \preceq r_{ik+1}$  and  $q_{ik+1}(y) \succeq s_{ik+1}$  and ... and  $q_{ip}(x) \preceq r_{ip}$  and  $q_{ip}(y) \succeq s_{ip}$  and  $q_{ip+1}(x) = r_{ip+1}$  and  $q_{ip+1}(y) = s_{ip+1}$  and ... and  $q_{iz}(x) = r_{iz}$  and  $q_{iz}(y) = s_{iz}$ , then  $xSy$  or  $xS^c y$ ,  
where  $N' = \{q_{i1}, \dots, q_{ic}\} \subseteq P^N$ ,  $N'' = \{q_{ic+1}, \dots, q_{ie}\} \subseteq P^N$ ,  $N'$  and  $N''$  are not necessarily disjoint,  $O' = \{q_{ie+1}, \dots, q_{ik}\}$ ,  $O'' = \{q_{ik+1}, \dots, q_{ip}\}$ ,  $O'$  and  $O''$  are not necessarily disjoint.

As can be seen from above, a decision rule consists of a conjunction of elementary conditions and a decision.

Since a decision rule is a kind of implication, a *minimal* rule is understood as an implication such that there is no other implication with the premise of at least the same weakness (in other words, a rule using a subset of elementary conditions and/or weaker elementary conditions) and the same conclusion. We shall consider minimal decision rules only.

Rule  $r$  *covers* a pair of objects  $(x, y) \in A \times A$  iff this pair satisfies all elementary conditions of  $r$ . Moreover, rule  $r$  is *supported* by a pair of objects  $(x, y) \in B$  iff this pair satisfies all elementary conditions of  $r$  and belongs to the relation suggested by  $r$ .

The rules of type 1) and 3) represent certain knowledge extracted from data, while the rules of type 2) and 4) represent possible knowledge; the rules of type 5) represent doubtful knowledge, because they are supported by inconsistent pairs of objects only.

One can consider the following basic characteristics of decision rule  $r$ :

- *support*, denoted by  $\text{supp}(r)$ , and defined as the number of pairs in  $B$  that support  $r$ ,
- *strength*, denoted by  $\sigma(r)$ , and defined as  $\sigma(r) = \text{supp}(r)/|B|$ ,
- *confidence* (also called *certainty factor*), denoted by  $\text{conf}(r)$ , and defined as the ratio of  $\text{supp}(r)$  and the number of pairs of objects that satisfy all elementary conditions of  $r$ ,
- *coverage factor*, denoted by  $\text{cf}(r)$ , and defined as the ratio of  $\text{supp}(r)$  and the number of pairs of objects that belong to the relation suggested by  $r$ .

Apart from the above basic characteristics, one might consider some *Bayesian confirmation measures* which reflect the attractiveness of a rule [26].

Let us consider, e.g., a rule of type 1). A pair of object  $(x, y) \in B$  supporting decision rule  $r$  is a *base* of this rule if  $q_{i1}(x) - q_{i1}(y) = r_{i1}$  and  $\dots$  and  $q_{ie}(x) - q_{ie}(y) = r_{ie}$  and  $q_{ie+1}(x) = r_{ie+1}$  and  $q_{ie+1}(y) = s_{ie+1}$  and  $\dots$  and  $q_{ip}(x) = r_{ip}$  and  $q_{ip}(y) = s_{ip}$  and  $q_{ip+1}(x) = r_{ip+1}$  and  $q_{ip+1}(y) = s_{ip+1}$  and  $\dots$  and  $q_{iz}(x) = r_{iz}$  and  $q_{iz}(y) = s_{iz}$ . Similar definitions hold for other types of rules. A decision rule having at least one base is called *robust*. Identification of supporting objects and bases of robust rules is important for interpretation of the rules in multiple criteria decision analysis.

A certain (possible) decision rule suggesting assignment to relation  $S$ , or to relation  $S^c$ , is *discriminant* if it covers only pairs of objects from the lower (upper) approximation of the respective relation. A rule is *non-redundant* if removing any of its conditions causes that it is no more discriminant.

Set of certain (possible) rules suggesting assignment to relation  $S$ , or to relation  $S^c$ , is *complete* iff each pair of objects  $(x, y)$  belonging to  $\underline{P}(S)$ , or to  $\underline{P}(S^c)$ , ( $\overline{P}(S)$ , or to  $\overline{P}(S^c)$ ,) is covered by at least one rule from this set. A set of decision rules is *minimal* if it is complete and *non-redundant*, i.e., exclusion of any rule from this set makes it incomplete.

Note that the syntax of decision rules induced from rough approximations defined using dominance cones, is using consistently this type of granules. Each condition profile defines a dominance cone in multi-dimensional evaluation space. The cones are positive for  $D_{\geq}$ -rules and negative for  $D_{\leq}$ -rules.

Let us also remark that dominance cones corresponding to condition profiles can originate in any point of evaluation space, without the risk of being too specific. Thus, contrary to traditional granular computing, the evaluation space need not to be discretized.

Induction of rules from dominance-based rough approximations is performed with the VC-DomLEM algorithm, proposed in [5, 6]. This is a sequential covering algorithm that induces a *minimal* set of rules.

Alternatively, one can consider an *exhaustive* set of rules. However, it is computationally hard to generate such set of rules. Therefore, we adapted the idea described in [8] in order to use *implicit (virtual) exhaustive* set of rules. For example, in order to verify if there exists at least one certain rule for relation  $S$  that covers pair  $(e_1, e_2) \in A \times A$ , it is enough to check if there exists at least one pair of objects  $(x, y) \in \underline{P}(S)$  that *contributes* to assign pair  $(e_1, e_2)$  to  $S$ . Let  $R = \{q_i \in P : (e_1, e_2)D_{\{q_i\}}(x, y)\}$ , where  $P \subseteq C$  (set of *compatible attributes*). Then, pair  $(x, y) \in \underline{P}(S)$  contributes to assign pair  $(e_1, e_2)$  to  $S \Leftrightarrow \forall (w, z) \notin \underline{P}(S) : (w, z) \notin D_R^+(x, y)$ .

## 2.6 Application of Decision Rules and Exploitation of Preference Graphs

In jRank it is possible to use both explicit minimal set of rules as well as implicit exhaustive set of rules. In both cases, one can apply certain or possible rules on  $A \times A$  to construct a *preference graph*. It is also possible to perform application of rules on some other set of pairs, say  $T \times T$ , where objects from set  $T$  are described by the same attributes as those of  $A$ . However, for the sake of simplicity, in the following, we assume that the application of rules concerns set  $A \times A$ .

The vertices of constructed preference graph  $G = \langle V, A \rangle$  correspond to objects, while arcs correspond to  $S$  and  $S^c$  preference relations between the objects. Thus,  $G$  contains an  $S$ -arc ( $S^c$ -arc) between object  $x$  and  $y$  iff pair of objects  $(x, y) \in A \times A$  is covered by a decision rule suggesting assignment to relation  $S$  ( $S^c$ ). It is also possible to label the arcs by weights that represent *satisfaction degrees* of the preference relations. For instance, if a single possible rule suggesting assignment to relation  $S$ , with confidence equal to 0.9, covers pair  $(x, y)$ , then the  $S$ -arc between  $x$  and  $y$  may be given weight 0.9. Moreover, in case of many covering rules, the respective arc may be given a weight which somehow aggregates confidence of these covering rules, e.g., by taking maximum confidence.

The preference graph resulting from application of decision rules is exploited by a *ranking method* in order to obtain final ranking (weak order) on  $A$ . This ranking is a solution to a ranking problem. In the case of a choice problem, the solution is the set of objects which share the first place in the final ranking.

We consider six ranking methods that can be applied to the preference graph. They yield a weak order of objects from set  $A$ . These methods are: Net Flow Score (NFS), Repeated Net Flow Score (RNFS), and four variants of the former two [7, 43]. We denote the methods by:

1.  $NFS(S, S^c)$ ,
2.  $RNFS(S, S^c)$ ,
3.  $NFS(S^*, S^{c*})$ ,
4.  $RNFS(S^*, S^{c*})$ ,
5.  $NFS(P^*, P^{-1*})$ ,
6.  $RNFS(P^*, P^{-1*})$ .

In fact, the four variants of the basic methods are obtained by preprocessing the preference graph, either by calculating *transitive closure* of  $S$  and  $S^c$  relations (methods 3 and 4) or by first taking the *asymmetric part* of  $S$  and  $S^c$ , and only then calculating the transitive closure of both relations (methods 5 and 6). In case of relation  $S$ , calculation of the asymmetric part boils down to the following rule: if  $(x, y) \in S$  and  $(y, x) \in S$ , then remove both arcs from the graph. In case of relation  $S^c$ , calculation of the asymmetric part boils down to the following rule: if  $(x, y) \in S^c$  and  $(y, x) \in S^c$ , then remove both arcs from the graph.

During calculation of the net flow score it is possible to consider weights of  $S$  and  $S^c$ -arcs [10].

According to [25], the first ranking method appears to be the only one that is neutral, strictly monotonic and independent of circuits. It is also faithful [7], which means that it preserves a weak order. The properties of the remaining ranking methods still need to be verified.

## 2.7 Variable Consistency Dominance-based Rough Set Approaches

In DRSA, lower approximation of relation  $S$  or  $S^c$  contains only consistent pairs of objects. Such a lower approximation is defined as a sum of dominance cones that are subsets of the approximated relation. In practical applications, however, such a strong requirement may result in relatively small lower approximations. Therefore, several extensions of DRSA have been proposed. These extensions relax the condition for inclusion to the lower approximation. Variable Consistency Dominance-based Rough Set Approaches (VC-DRSA) include to lower approximations pairs of objects which are sufficiently consistent. Different measures of consistency may be applied in VC-DRSA. Given a user-defined threshold value, extended lower approximation of considered relation is defined as a set of pairs of objects for which the consistency measure satisfies that threshold.

Several definitions of VC-DRSA have been considered in the literature so far. In the first papers concerning VC-DRSA [15, 24], consistency (of objects) have been calculated using rough membership measure [31, 45]. Then, in order to ensure monotonicity of lower approximation with respect to the dominance relation, the idea of the first papers have been extended in the work [1]. Recently, it has been pointed out that it is reasonable to require that the consistency measure used in the definition of lower approximation satisfies some properties of monotonicity [4]. Resulting variable consistency approaches, employing monotonic consistency measures, are called Monotonic Variable Consistency Dominance-based Rough Set Approaches [2, 3, 4].

In jRank, four different consistency measures can be applied to calculate extended lower approximations that serve as a basis for induction of decision rules. These are: rough membership measure  $\mu$ , measure  $\epsilon^*$ , and measure  $\epsilon'$  [4]. Each weight (satisfaction degree) of an  $S$ -arc (or  $S^c$ -arc) in the preference graph (resulting from the application of induced decision rules) can be calculated by taking maximum strength of the rules that cover corresponding pair of objects and suggest assignment to relation  $S$  (or, respectively,  $S^c$ ). Let  $T \in \{S, S^c\}$ . Moreover, let consider application of consistency measure  $\epsilon$  when calculating extended lower approximations. Then, the strength of a single induced rule  $r$  suggesting assignment to relation  $T$  can be calculated as  $1 - \hat{\epsilon}_T(r)$  or as  $(1 - \hat{\epsilon}_T(r))cf(r)$ , where  $\hat{\epsilon}_T(r)$  is the value of the rule consistency measure  $\hat{\epsilon}_T$  defined in [42] and  $cf(r)$  denotes coverage factor of rule  $r$  defined as the ratio of the number of pairs of objects supporting  $r$  and the cardinality of relation  $T$ .



## 3 How to Use jRank

### 3.1 Running jRank

jRank can be downloaded from <http://www.cs.put.poznan.pl/mszelag/Software/jRank/jRank.html>. It is a highly configurable command-line application written in Java programming language, distributed as ZIP archive. After unpacking the archive, one should get the following structure:

- `experiments` – storage directory for settings, input and output files for different experiments,
- `lib` – contains Java Rough Set (jRS) library (JAR archive),
- `jRank.info` – file containing a list of all jRank parameters,
- `readme.txt` – file containing useful information concerning the jRank software.

jRank works in a per-experiment mode. This means that in order to run calculations, one should first create an *experiment configuration file* called `experiment.properties` and prepare input data – ISF file(s) containing the information table(s) to be processed. The flow of an experiment is as follows:

1. *learning information table* is read from a given “learning data file”,
2. on the basis of pairwise comparisons or ranking given in the experiment configuration file, PCT is created; PCT is written to a given ISF file,
3. approximations of relation  $S$  and  $S^c$  are calculated for the PCT; they are written to a given \*.apx file,
4. in case when explicit minimal set of decision rules is used, certain/possible decision rules are induced from the lower/upper approximations of  $S$  and  $S^c$ ; these rules are written to a given \*.rules file,
5. *test information table* is read from a given “test data file” (if “test data file” is different than “learning data file”); objects from the test information table have to be evaluated by the same attributes as those of the learning information table,
6. decision rules are applied to all pairs of objects from the test information table; resulting preference graph is written to a given \*.graph file,
7. chosen ranking method is applied to the preference graph and the final ranking (weak order) is written to a given \*.ranking file,
8. a report concerning ranking errors is written to a \*.txt file.

Assume that current path is jRank main directory and we want to create and run experiment called “my-Experiment”. To achieve this goal, we need to perform the following steps (in parentheses, we give respective Windows command-line commands):

1. open `experiments` directory (`cd experiments`),
2. create `myExperiment` subdirectory (`mkdir myExperiment`),
3. open `myExperiment` subdirectory (`cd myExperiment`),
4. place ISF file with learning information table in `myExperiment` directory,
5. if used, place ISF file with test information table in `myExperiment` directory,
6. create experiment configuration file in `myExperiment` directory; it has to be called `experiment.properties`,
7. edit `experiment.properties` file according to our preferences,
8. return to `experiments` directory (`cd ..`),
9. run our experiment, using its name (`jRank myExperiment`).

To ease configuration of an experiment, not all parameters need to be defined in the `experiment.properties` file. Many of the parameters have reasonable or even recommended default values, as can be seen in `default.properties` master configuration file. This file is always read by jRank before user-defined experiment configuration file. Such an order has two goals. First, the user does not have to give values for the parameters with default values. Second, it is possible to override default value of any parameter by assigning to it a new value.

Experiment configuration file is composed of any number of lines, each containing one `key = value` entry, and optional comment after `#` sign. In case when a given `key` appears both in master configuration file `default.properties` and in experiment configuration file `experiment.properties`, the `value` from the latter overrides the `value` from the former (i.e., the default value).

Tables 3, 4, and 5 contain description of all jRank parameters. In case when a parameter value is a disk path, directories have to be separated by `/` or `\` sign. At least `learningDataFile` has to be given. Other disk paths can be set automatically using `learningDataFile`. Disk paths can be absolute or relative to the experiment's directory.

Table 3: jRank parameters

Parameter	Default value	Remarks
<code>learningDataFile</code>	–	disk path to the learning ISF data file
<code>testDataFile</code>	–	disk path to the test ISF data file; if not given, then assumed to be equal to <code>learningDataFile</code>
<code>pctFile</code>	–	disk path to the ISF file where PCT created using given preference information will be saved
<code>pctApxFile</code>	–	disk path to the file where approximations of $S$ and $S^c$ relations in PCT will be saved
<code>pctRulesFile</code>	–	disk path to the file where rules for PCT will be saved
<code>preferenceGraphFile</code>	–	disk path to the file where preference graph resulting from application of rules to all pairs of objects from the test data file will be saved
<code>rankingFile</code>	–	disk path to the file where ranking of all objects from the test data file will be saved
<code>referenceRanking</code>	–	comma-separated list of places; at each place there can be just one or many objects' numbers, separated by white spaces (objects' numbers start from 1); if this parameter is defined, then <code>pairs</code> are ignored
<code>pairs</code>	–	comma-separated list of compared pairs; each entry consists of a pair of examples (comma-separated, in <code>{}</code> ) and $S/S^c$ flag indicating the result of comparison, e.g., <code>{1, 2} S</code> , <code>{2, 4} S<sup>c</sup></code>
<code>consistencyMeasure</code>	<code>epsilon</code>	indicator of consistency measure [4] used to calculate lower approximations of $S$ and $S^c$ relations; has to belong to set <code>{epsilon, epsilon*, epsilon', rough-membership}</code> , where subsequently listed values encode measures: $\epsilon$ , $\epsilon^*$ , $\epsilon'$ , and $\mu$
<code>consistencyMeasureThreshold</code>	0.0	threshold for consistency measure; floating-point value from interval <code>[0,1]</code> for consistency measures <code>epsilon</code> , <code>epsilon*</code> , and <code>rough-membership</code> , or from interval <code>[0,X]</code> for consistency measure <code>epsilon'</code> , where <code>X</code> can be greater than 1

Table 4: jRank parameters

Parameter	Default value	Remarks
<code>typeOfRules</code>	<code>certain</code>	type of considered decision rules – either <b>certain</b> or <b>possible</b> ; possible rules may only be used for such value of consistency measure threshold, which for used consistency measure implies classical DRSA
<code>consideredSetOfRules</code>	<code>minimal</code>	used set of decision rules – either <b>minimal</b> (explicit minimal set of rules induced by VC-DomLEM algorithm) or <b>exhaustive</b> (virtual exhaustive set of rules)
<code>satisfactionDegreesInPreferenceGraph</code>	<code>fuzzy</code>	type of satisfaction degrees (weights) in the preference graph – either <b>fuzzy</b> (i.e., from interval $[0,1]$ ) or <b>crisp</b> (i.e., from set $\{0,1\}$ ); fuzzy satisfaction degrees cannot be used for DRSA + exhaustive set of possible rules + rough membership
<code>fuzzySatisfactionDegreeCalculationMethod</code>	<code>max-credibility</code>	method of calculating fuzzy satisfaction degrees in preference graphs, if such satisfaction degrees are used; for a given pair of objects, can be maximum of credibility over covering rules ( <b>max-credibility</b> ) or maximum product of credibility and coverage factor over covering rules ( <b>max-credibility-x-coverage-factor</b> )
<code>rankingProcedure</code>	<code>nfs</code>	ranking method used for exploitation of the preference graph; has to belong to set $\{nfs, rnfs, nfs-*, rnfs-*, nfs-p-*, rnfs-p-*\}$ , where subsequently listed values encode ranking methods: $NFS(S, S^c)$ , $RNFS(S, S^c)$ , $NFS(S^*, S^{c*})$ , $RNFS(S^*, S^{c*})$ , $NFS(P^*, P^{-1*})$ , $RNFS(P^*, P^{-1*})$
<code>dominance</code>	<code>pareto</code>	dominance relation used to calculate approximations of $S$ and $S^c$ relations in PCT – either <b>pareto</b> or <b>lorenz</b> ; safe option is to choose Pareto dominance since for Lorenz dominance [36] the data must meet special requirements
<code>dominanceForPairsOfOrdinalValues</code>	<code>classic</code>	indicator of the considered definition of dominance for pairs of ordinal values in PCT – either <b>classic</b> (recommended) or <b>strict</b>
<code>negativeExamplesTreatmentForVCDRSA</code>	<code>only-inconsistent</code>	strategy of covering negative examples by rules in case of VC-DRSA [5]; can be <b>only-inconsistent</b> (recommended), <b>only-inconsistent-and-boundary</b> , or <b>any</b>
<code>ruleConditionsSelectionMethodInVCDomLEM</code>	<code>mix</code>	strategy of rule conditions selection in VC-DomLEM algorithm – either <b>mix</b> (denotes that each rule can employ elementary conditions built using evaluations of different pairs of objects) or <b>base</b> (denotes that each rule has to have a base, i.e., it can employ only elementary conditions built using evaluations of a single pair of objects)

Table 5: jRank parameters

Parameter	Default value	Remarks
<code>allowEmptyRulesInVCDomLEM</code>	<code>true</code>	tells if VC-DomLEM algorithm is allowed to induce rules with empty condition part if their consistency is good enough – either <code>true</code> or <code>false</code>
<code>useEdgeRegionsInVCDomLEM</code>	<code>false</code>	tells if only pairs of objects from EDGE region [5] or rather all (sufficiently consistent) pairs of objects from considered lower/upper approximation should be used when creating rule’s conditions in VC-DomLEM algorithm – either <code>true</code> or <code>false</code>
<code>optimizeRuleConsistencyInVCDomLEMwrt</code>	<code>approximation</code>	set of pairs of objects with respect to which value of rule consistency measure optimized in VC-DomLEM algorithm is calculated; can be lower (or upper) approximation of the preference relation for which a certain (or possible, respectively) rule is generated or entire preference relation (for certain rules only) – either <code>approximation</code> or <code>set</code>
<code>writeDominationInformation</code>	<code>true</code>	tells if sections [P-dominating sets] and [P-dominated sets] should be present in generated *.apx file – either <code>true</code> or <code>false</code>
<code>writeRulesStatistics</code>	<code>true</code>	tells if rules’ statistics should be present in generated *.rules file – either <code>true</code> or <code>false</code>
<code>writeLearningPositiveExamples</code>	<code>true</code>	tells if learning positive examples of each rule should be listed inside [RULESINFO] section of generated *.rules file – either <code>true</code> or <code>false</code> ; works only when <code>writeRulesStatistics</code> is <code>true</code>
<code>precision</code>	<code>-1</code>	integer value; denotes precision of floating-point numbers; set <code>-1</code> in order to print floating-point numbers as they are, without rounding

### 3.2 Illustrative Example

Suppose that a Belgian citizen is willing to buy a house in Poland. He wants to rank 11 alternatives described by three criteria: distance from the nearest airport (in km), price (in thousands of EUR) and comfort. The alternatives (objects) are gathered in the *learning information table* presented as Table 6.

Table 6: Information table for the house location problem

Location	$q_1$ – Distance	$q_2$ – Price	$q_3$ – Comfort
$L_1$ – Poznan	3	60	Good
$L_2$ – Kapalica	35	30	Good
$L_3$ – Krakow	7	85	Medium
$L_4$ – Warszawa	10	90	Basic
$L_5$ – Wroclaw	5	60	Medium
$L_6$ – Malbork	50	50	Medium
$L_7$ – Gdansk	5	70	Medium
$L_8$ – Kornik	50	40	Medium
$L_9$ – Rogalin	15	50	Basic
$L_{10}$ – Lublin	8	60	Good
$L_{11}$ – Torun	100	50	Medium

Obviously, Distance and Price are cost-type cardinal criteria. Comfort is a gain-type ordinal criterion with the following order in the domain: Good  $\succ$  Medium  $\succ$  Basic, where  $\succ$  denotes a strict preference relation.

#### 3.2.1 Data File(s)

As the first step you should create an ISF file containing data from the considered learning information table. For this purpose, you may use any plain text editor in order to type the text shown below.

```

**ATTRIBUTES
+ location: (nominal) description
+ distance: (integer)
+ price: (integer)
+ comfort: [Basic, Medium, Good]

**PREFERENCES
distance: cost
price: cost
comfort: gain

**EXAMPLES
Poznan          3          60          Good
Kapalica        35          30          Good
Krakow          7           85          Medium
Warszawa        10          90          Basic
Wroclaw         5           60          Medium
Malbork         50          50          Medium
Gdansk          5           70          Medium
Kornik          50          40          Medium
Rogalin         15          50          Basic
Lublin          8           60          Good
Torun           100         50          Medium

**END

```

Remark that + sign in front of an attribute name is an indicator of activity (in turn, - means that the attribute should not be taken into account).

Now, save the file as `Houses11.isf` in `experiments/Houses11` directory. For this example, we are going to apply decision rules to the set of all pairs of objects from Table 6. Therefore, the *test information table* is the same as the learning information table. However, if needed, ISF file containing data from the considered test information table can be created just in the same way.

### 3.2.2 Preference Information

Assume that the decision maker has compared pairwise locations  $L_1 - L_7$  in the way presented in Fig. 1.

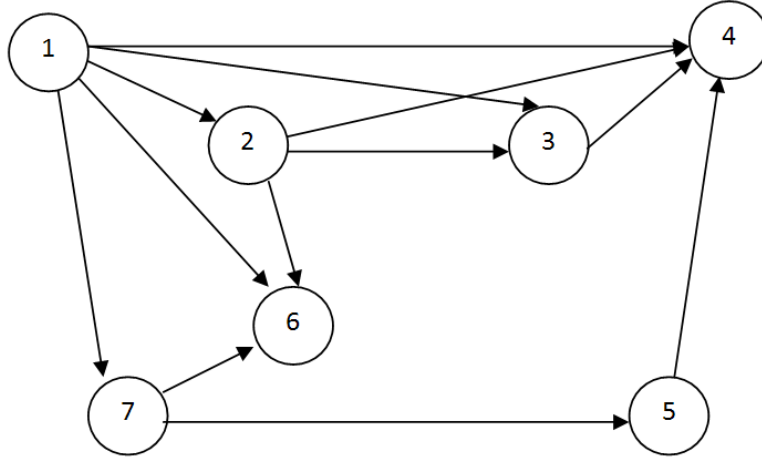


Figure 1: Preference information for the house location problem

An arc between two locations denotes preference relation  $S$ . Moreover, we assume that the given preference information is symmetric, i.e.,  $(x, y) \in S \Leftrightarrow (y, x) \in S^c$ .

### 3.2.3 Experiment Configuration File

Now, you need to create experiment configuration file. For this purpose, you may again use any plain text editor in order to type the text shown below.

```
learningDataFile = Houses11.isf
```

```
pairs = {1,2} S, {1,3} S, {1,4} S, {1,6} S, {1,7} S, {2,3} S, {2,4} S, {2,6} S,
        {3,4} S, {5,4} S, {7,5} S, {7,6} S, {2,1} Sc, {3,1} Sc, {4,1} Sc, {6,1} Sc,
        {7,1} Sc, {3,2} Sc, {4,2} Sc, {6,2} Sc, {4,3} Sc, {4,5} Sc, {5,7} Sc, {6,7}
        Sc
```

Make sure that all pairwise comparisons are on the same line and save the file as `experiment.properties` in `experiments/Houses11` directory.

Let us observe that in the considered example, the content of `experiment.properties` file is very much simplified (only two parameters defined). Remaining parameters are assigned default values from `experiments/default.properties` master configuration file.

### 3.2.4 Running jRank for the Illustrative Example

Once configuration file is ready, it is now possible to run our experiment. While in the `experiments` directory, in the available command-line type the command `jRank Houses11`. In the command line you should be able to see output information of jRank. Assuming that jRank is installed in directory `C:\jRank`, the output should be:



C:\jRank\experiments>

Lack of [WARNING] or [ERROR] entries in the above log suggests that jRank was able to perform required calculations. As a result, in the `Houses11` directory there should appear the following new text files:

- `Houses11_partialPCT.isf` – contains PCT created for a given set of pairwise comparisons,
- `Houses11_partialPCT.apx` – contains approximations of relations  $S$  and  $S^c$  in PCT,
- `Houses11_partialPCT.rules` – contains decision rules generated by VC-DomLEM algorithm,
- `Houses11.graph` – contains preference graph in a format accepted by `Gvedit` and `dotty` from `Graphviz`,
- `Houses11.ranking` – contains final ranking (weak order) of house locations,
- `Houses11_report.txt` – contains information about ranking errors.

Since the names of the created files had not been specified during creation of an experiment configuration file, they were set automatically. The suffix `_partialPCT` is used to indicate that we deal with a “partial” PCT, i.e., a PCT that is created for just some, not all possible, pairs of objects from the learning information table.

In the following, we discuss how to interpret the results. In the files generated by jRank, objects are identified by the number of row from learning/test information table, and pairs of objects are identified by the number of row from PCT. Thus, numbers present in `*.isf` file with PCT correspond to rows of the learning information table. Numbers present in `*.graph` and `*.ranking` files correspond to rows of the test information table. Numbers present in `*.apx` and `*.rules` files correspond to rows of PCT. All the numbers start from one.

### 3.2.5 Pairwise Comparison Table (PCT)

The content of the ISF file with the created PCT is rather self-explanatory. We should just notice the presence of two “extra” attributes:

- `Pair_of_examples` – used to store pair of numbers; these numbers reflect objects from the learning information table,
- `Comprehensive_preference_grade` – used to store comprehensive preference relation; relation  $S$  is encoded by value 0.0, while relation  $S^c$  is encoded by value  $-1.0$ .

### 3.2.6 Approximations of Preference Relations

In jRS, the syntax of `*.apx` files is common for different applications of DRSA. Since the primary purpose of the `*.apx` file format was to store approximations of unions of ordered decision classes [15], some used keywords does not fit well in our context, where there are only two decision classes (i.e., the preference relations). However, for the sake of compatibility, we decided not to change these keywords. Therefore, `At least 0.0` should be read as outranking relation  $S$ , while `At most -1.0` should be read as non-outranking relation  $S^c$ .

At the beginning of the `*.apx` file there are two sections: `[P-dominating sets]` and `[P-dominated sets]`. In these sections, for each pair of objects, one may find pairs of objects that belong to its dominating and dominated set.

In the `[Approximations]` section, one may find lists of pairs of objects that belong to lower approximation, upper approximation, and boundary of each preference relation.

In the `[Accuracy of approximation]` section there are given four values for each relation: the *accuracy of approximation* of the relation by the considered set of attributes, the number of pairs in the relation, the number of pairs in the lower approximation of the relation, and the number of pairs in the upper approximation of the relation.

Finally, last section `[Quality of sorting]` contains the value of the *quality of approximation* of the classification  $CI=\{S^c, S\}$  by the considered set of attributes.



### 3.2.7 Induced Decision Rules

Rules induced by VC-DomLEM algorithm are written to a \*.rules file. In this file, the most important sections are [RULES] and [RULESINFO]. The former contains a list of induces rules, the latter contains statistics of the rules. In case of the house location problem, [RULES] section is the following:

```
[RULES]
#Certain at least rules
1: {PAIR(Evaluations_on_comfort) D (Good,Medium)} => (
  Comprehensive_preference_grade >= 0.0) |CERTAIN, AT_LEAST, 0.0|
2: {PAIR(Evaluations_on_comfort) D (Medium,Basic)} => (
  Comprehensive_preference_grade >= 0.0) |CERTAIN, AT_LEAST, 0.0|
3: [DIFF(Evaluations_difference_on_distance) <= -32] => (
  Comprehensive_preference_grade >= 0.0) |CERTAIN, AT_LEAST, 0.0|
#Certain at most rules
4: {(Medium,Good) D PAIR(Evaluations_on_comfort)} => (
  Comprehensive_preference_grade <= -1.0) |CERTAIN, AT_MOST, -1.0|
5: {(Basic,Medium) D PAIR(Evaluations_on_comfort)} => (
  Comprehensive_preference_grade <= -1.0) |CERTAIN, AT_MOST, -1.0|
6: [DIFF(Evaluations_difference_on_distance) >= 32] => (
  Comprehensive_preference_grade <= -1.0) |CERTAIN, AT_MOST, -1.0|
```

Obviously, # sign starts a comment. As can be seen above, there are three certain rules for each preference relation. For instance, the first rule should be decoded as: if  $\text{Comfort}(x) \succeq \text{Good}$  and  $\text{Comfort}(y) \preceq \text{Medium}$  then  $xSy$ ; the sixth rule should be read as: if  $\text{Distance}(x) - \text{Distance}(y) \geq 32$  then  $xS^c y$ .

Now, let us consider the statistics of the first rule. The [RULESINFO] section contains the following entry for this rule:

```
1: {PAIR(Evaluations_on_comfort) D (Good,Medium)} => (
  Comprehensive_preference_grade >= 0.0) |CERTAIN, AT_LEAST, 0.0|
LearningPositiveExamples: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12
Support: 7
SupportingExamples: 2, 3, 4, 5, 6, 7, 8
Strength: 0.2916666666666667
Confidence: 1.0
CoverageFactor: 0.5833333333333334
Coverage: 7
CoveredExamples: 2, 3, 4, 5, 6, 7, 8
NegativeCoverage: 0
NegativeCoveredExamples:
InconsistencyMeasure: 0.0
EpsilonPrimMeasure: 0.0
```

Most of the statistics are self-explanatory. **InconsistencyMeasure** denotes value of  $\epsilon$  consistency measure, while **EpsilonPrimMeasure** denotes value of  $\epsilon'$  consistency measure [4]. Both values are calculated for the dominance cone corresponding to the conjunction of rule's conditions. Since the first rule is a certain one, **LearningPositiveExamples** contains a list of pairs of objects from the lower approximation of relation  $S$ . This tag is present since the default value of parameter **writeLearningPositiveExamples** is true.

### 3.2.8 Preference Graph

Preference graph resulting from application of induced rules to all pairs of objects from Table 6 is written to a \*.graph file. This file has a format understood by **Gvedit** and **dotty** from **Graphviz** package. In order to view the preference graph visually, you should install Graphviz on your system, run **GVedit**, open \*.graph file, and choose **Run** option. In the dialog that appears, you should choose **Layout Engine** and **Output File Type**. Recommended values for this two parameters are **circo** and **jpg** or **png**, respectively, as shown in Fig. 2.

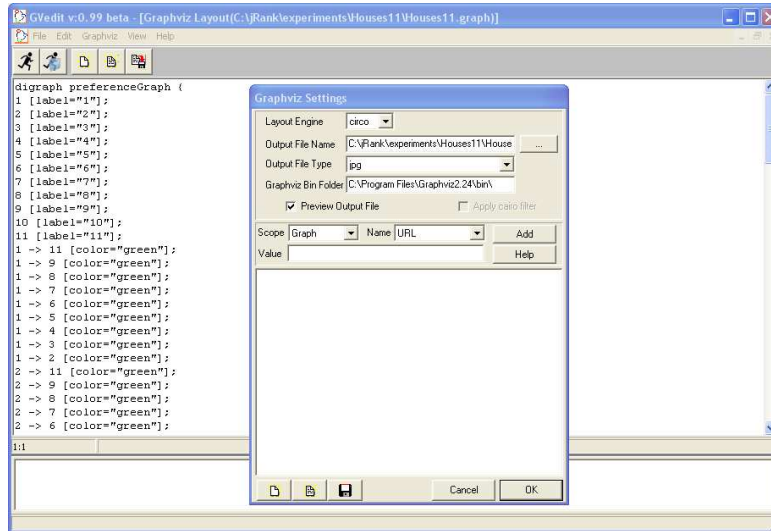


Figure 2: GVedit settings for the house location problem

After acceptance of the settings, GVedit generates visual representation of the preference graph. In case of the house location problem, it should resemble this of Fig. 3.

Green arcs denote outranking relation  $S$ ; red arcs denote non-outranking relation  $S^c$ .

### 3.2.9 Final Ranking

Final ranking of considered objects (weak order) is written to a \*.ranking file. It is obtained by applying to the preference graph chosen ranking method. In our example, we applied a default method, i.e., the Net Flow Score. The resulting ranking is the following:

1:	1	18.0
2:	10	16.0
3:	2	14.0
4:	3, 5, 7	4.0
5:	6, 8	-10.0
6:	4, 9	-12.0
7:	11	-16.0

As can be seen above, the ranking consists of seven ranks, ordered according to the net flow score (the last value in each row). The best house location is the first one, i.e., Poznan. The worst one is Torun. There are three ties – at fourth, fifth, and sixth place. For instance, fourth place is shared by locations 3, 5, and 7, i.e., by Krakow, Wroclaw, and Gdansk.

One can easily observe the correspondence between the final ranking and the preference graph from Fig. 3. House location no. 1 has nine  $S$ -arcs going out and nine  $S^c$ -arcs going in, and therefore its score is equal to 18.

### 3.2.10 Ranking Errors

Ranking errors are written to a \*.txt file. These errors, which are values of Kendall's  $\tau$  measure, result from the comparison of the preference information (i.e., reference ranking or pairwise comparisons) with the final ranking. Moreover, if the test information table contains a decision criterion, the ranking errors are also calculated w.r.t. the so-called *true global ranking*, i.e., the weak order of all objects from the test information table determined by the order of decision classes.

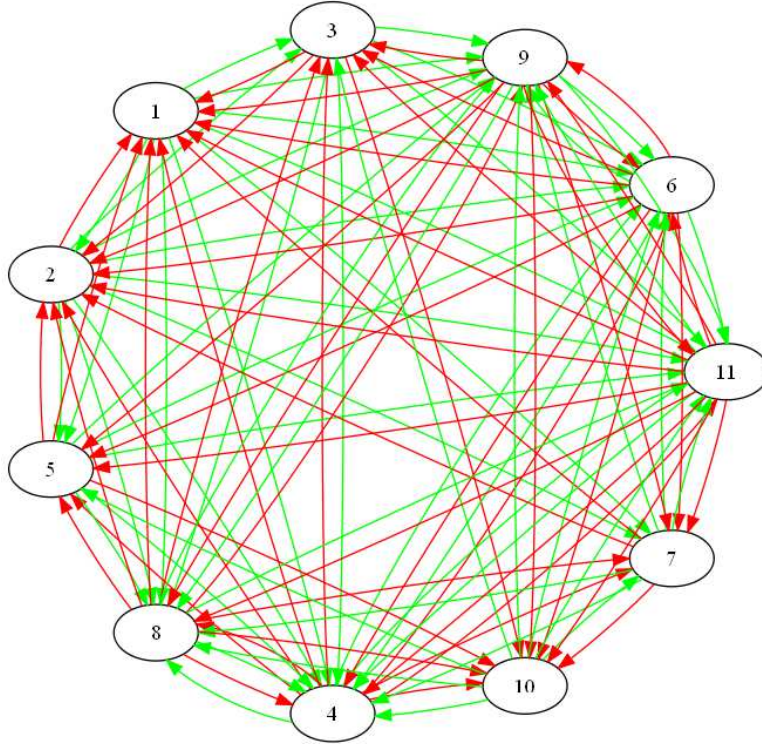


Figure 3: Preference graph for the house location problem

## 4 Exemplary Applications of Dominance-based Rough Set Approach

There are many possibilities of applying DRSA to real life problems. The non-exhaustive list of potential applications includes:

- decision support in medicine: in this area there are already many interesting applications (see, e.g., [32, 27, 28, 44]), however, they exploit the classical rough set approach; applications requiring DRSA, which handle ordered value sets of medical signs, as well as monotonic relationships between the values of signs and the degree of a disease, are in progress;
- customer satisfaction survey: theoretical foundations for application of DRSA in this field are available in [20], however, a fully documented application is still missing;
- bankruptcy risk evaluation: this is a field of many potential applications, as can be seen from promising results reported, e.g., in [40, 41, 12], however, a wider comparative study involving real data sets is needed;
- operational research problems, such as location, routing, scheduling or inventory management: these are problems formulated either in terms of classification of feasible solutions (see, e.g., [11]), or in terms of interactive multiobjective optimization, for which there is a suitable IMO-DRSA [22] procedure;
- finance: this is a domain where DRSA for decision under uncertainty has to be combined with interactive multiobjective optimization using IMO-DRSA; some promising results in this direction have been presented in [21];
- ecology: assessment of the impact of human activity on the ecosystem is a challenging problem for which the presented methodology is suitable; the up to date applications are based on the classical rough set concept (see, e.g., [9, 34]), however, it seems that DRSA handling ordinal data has a greater potential in this field.

## 5 Glossary

*Multiple attribute (or multiple criteria) decision support* aims at giving the decision maker (DM) a recommendation concerning a set of *objects*  $A$  (also called alternatives, actions, acts, solutions, options, candidates,...) evaluated from multiple points of view called *attributes* (also called features, variables, criteria,...).

Main categories of *multiple attribute (or multiple criteria) decision problems* are:

- *classification*, when the decision aims at assigning objects to predefined classes,
- *choice*, when the decision aims at selecting the best object(s),
- *ranking*, when the decision aims at ordering objects from the best to the worst.

Two kinds of *classification problems* are distinguished:

- *taxonomy*, when the value sets of attributes and the predefined classes are not preference ordered,
- *ordinal classification with monotonicity constraints* (also called *multiple criteria sorting*), when the value sets of attributes and the predefined classes are preference ordered, and there exists a monotonic relationship between evaluations of objects and their assignment to ordered decision classes.

Two kinds of *choice problems* are distinguished:

- *discrete choice*, when the set of objects is finite and reasonably small to be listed,
- *multiple objective optimization*, when the set of objects is infinite and defined by constraints of a mathematical program.

If value sets of attributes are preference-ordered, they are called *criteria* or *objectives*, otherwise they are called *regular attributes*.

*Criterion* is a real-valued function  $q_i$  defined on  $A$ , reflecting a worth of objects from a particular point of view, such that in order to compare any two objects  $x, y \in A$  from this point of view it is sufficient to compare two values:  $q_i(x)$  and  $q_i(y)$ .

*Dominance*: object  $x$  is non-dominated in set  $A$  (Pareto-optimal) if and only if there is no other object  $y$  in  $A$  such that  $y$  is not worse than  $x$  on all considered criteria, and strictly better on at least one criterion.

*Preference model* is a representation of a value system of the decision maker on the considered set of objects.

*Rough set* in universe  $A$  is an approximation of a set based on available information about objects of  $A$ . The rough approximation is composed of two ordinary sets, called *lower and upper approximation*. Lower approximation is a maximal subset of objects which, according to the available information, certainly belong to the approximated set, and upper approximation is a minimal subset of objects which, according to the available information, possibly belong to the approximated set. The difference between upper and lower approximation is called *boundary*.

*Decision rule* is a logical statement of the type “*if... then...*”, where the premise (condition part) specifies values assumed by one or more condition attributes and the conclusion (decision part) specifies an overall judgment.

## References

- [1] Błaszczyński, J., Greco, S., Słowiński, R., Szelağ, M.: On Variable Consistency Dominance-based Rough Set Approaches. In: LNAI, vol. 4259, Springer-Verlag, Berlin 2006, pp. 191-202.
- [2] Błaszczyński, J., Greco, S., Słowiński, R., Szelağ, M.: Monotonic Variable Consistency Rough Set Approaches. In: J. Yao, P. Lingras, W. Wu, M. Szczuka, N. J. Cercone, D. Ślęzak (eds.): Rough Sets and Knowledge Technology, LNAI, vol. 4481, Springer-Verlag, 2007, pp. 126-133.
- [3] Błaszczyński, J., Greco, S., Słowiński, R., Szelağ, M.: Monotonic Variable Consistency Rough Set Approaches. Research Report RA-010/07, Poznań University of Technology, 2007.

- [4] Błaszczczyński, J., Greco, S., Słowiński, R., Szelaż, M.: Monotonic Variable Consistency Rough Set Approaches. *International Journal of Approximate Reasoning*, 50(7), 2009, pp. 979-999.
- [5] Błaszczczyński, J., Słowiński, R., Szelaż, M.: Sequential Covering Rule Induction Algorithm for Variable Consistency Rough Set Approaches. *Information Sciences* (to appear).
- [6] Błaszczczyński, J., Słowiński, R., Szelaż, M.: VC-DomLEM: Rule induction algorithm for variable consistency rough set approaches. Research Report RA-07/09, Poznań University of Technology, 2009.
- [7] Bouyssou, D., Vincke, P.: Ranking Alternatives on the Basis of Preference Relations: A Progress Report with Special Emphasis on Outranking Relations. *Journal of Multi-Criteria Decision Analysis*, vol. 6, 1997, pp. 77-85.
- [8] Dembczyński, K., Pindur, R., Susmaga, R.: Dominance-based Rough Set Classifier without Induction of Decision Rules. *Electronic Notes in Theoretical Computer Science*, 82(4), 2003, pp. 84-95.
- [9] Flinkman, M., Michałowski, W., Nilsson, S., Słowiński, R., Susmaga, R., Wilk, S.: Use of rough sets analysis to classify Siberian forest ecosystem according to net primary production of phytomass, *INFOR*, 38 (2000) 145-161.
- [10] Fortemps, P., Greco, S., Słowiński, R.: Multicriteria decision support using rules that represent rough-graded preference relations. *European Journal of Operational Research*, 188(1), 2008, pp. 206-223.
- [11] Gorsevski, P.V., Jankowski, P.: Discerning landslide susceptibility using rough sets. *Computers, Environment and Urban Systems*, 32(2008) 53-65.
- [12] Greco, S., Matarazzo, B., Słowiński, R.: A new rough set approach to evaluation of bankruptcy risk, in C. Zopounidis (ed.), *Operational Tools in the Management of Financial Risks*, Kluwer, Dordrecht, 1998, pp. 121-136.
- [13] Greco, S., Matarazzo, B., Słowiński, R.: The use of rough sets and fuzzy sets in MCDM, chapter 14 in T. Gal, T. Stewart, T. Hanne (eds.), *Advances in Multiple Criteria Decision Making*, Kluwer, Boston, 1999, pp. 14.1-14.59.
- [14] Greco S., Matarazzo, B., Słowiński, R.: Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research*, 117 (1999) 63-83.
- [15] Greco, S., Matarazzo, B., Słowiński R.: Rough sets theory for multicriteria decision analysis, *European Journal of Operational Research*, 129(2001) 1-47.
- [16] Greco, S., Matarazzo, B., Słowiński, R.: Multicriteria classification. [In]: W. Kloesgen, J. Zytkow (eds.), *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, 2002, chapter 16.1.9, pp. 318-328.
- [17] Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach to Knowledge Discovery, (I) - General Perspective, (II) - Extensions and Applications, chapters 20 and 21 in N. Zhong, J. Liu, *Intelligent Technologies for Information Analysis*, Springer, Berlin, 2004, pp. 513-612.
- [18] Greco, S., Matarazzo, B., Słowiński, R.: Decision rule approach, chapter 13 in J. Figueira, S. Greco, M. Ehrgott (eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, Berlin, 2005, pp. 507-563.
- [19] Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach as a proper way of handling graduality in rough set theory. *Transactions on Rough Sets VII*, LNCS 4400, Springer, Berlin, 2007, pp. 36-52.
- [20] Greco, S., Matarazzo, B., Słowiński, R.: Customer satisfaction analysis based on rough set approach. *Zeitschrift für Betriebswirtschaft*, 16 (2007) no. 3, 325-339.

- [21] Greco, S., Matarazzo, B., Słowiński, R.: Financial portfolio decision analysis using Dominance-based Rough Set Approach. Invited paper at the 22nd European Conference on Operational Research (EURO XXII), Prague, 08-11.07.2007.
- [22] Greco, S., Matarazzo, B., Słowiński, R.: Dominance-based Rough Set Approach to Interactive Multiobjective Optimization, chapter 5 in J.Branke, K.Deb, K.Miettinen, R.Slowinski (eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, Berlin, 2008.
- [23] Greco, S., Matarazzo, B., Słowiński, R., Stefanowski, J.: An algorithm for induction of decision rules consistent with dominance principle, in W. Ziarko, Y. Yao (eds.): *Rough Sets and Current Trends in Computing*, LNAI 2005, Springer, Berlin, 2001, pp. 304-313.
- [24] Greco, S., Matarazzo, B., Słowiński R., Stefanowski, J.: Variable Consistency Model of Dominance-based Rough Sets Approach. In: W. Ziarko, Y. Yao (eds.): *Rough Sets and Current Trends in Computing*, LNAI, vol. 2005, Springer-Verlag, Berlin 2001, pp. 170-181.
- [25] Greco, S., Matarazzo, B., Słowiński R., Tsoukias, A.: Exploitation of a Rough Approximation of the Outranking Relation in Multicriteria Choice and Ranking. [In]: Theodor J. Stewart, Robin C. van den Honert, *Trends in Multicriteria Decision Making. Lecture Notes in Economics and Mathematical Systems*, vol. 465, Springer-Verlag, Berlin, 1998, pp. 45-60.
- [26] Greco, S., Pawlak, Z., Słowiński, R.: Can Bayesian confirmation measures be useful for rough set decision rules? *Engineering Applications of Artificial Intelligence*, 17 (2004) 345-361.
- [27] Michałowski, W., Rubin, S., Słowiński, R., Wilk, S.: Mobile clinical support system for pediatric emergencies. *Journal of Decision Support Systems*, 36 (2003) 161-176.
- [28] Michałowski, W., Wilk, S., Farion, K., Pike, J., Rubin, S., Słowiński, R.: Development of a decision algorithm to support emergency triage of scrotal pain and its implementation in the MET system. *INFOR*, 43 (2005) 287-301.
- [29] Pawlak, Z.: Rough Sets, *International Journal of Computer and Information Sciences*, 11 (1982) 341-356.
- [30] Pawlak, Z.: *Rough Sets*, Kluwer, Dordrecht, 1991.
- [31] Pawlak, Z., Skowron, A., Rough Membership Functions. In: R. R. Yager, M. Fedrizzi and J. Kacprzyk (eds.): *Advances in the Dempster-Shafer Theory of Evidence*, Wiley, New York 1994, pp. 251-271.
- [32] Pawlak, Z., Słowiński, K., Słowiński, R.: Rough classification of patients after highly selective vagotomy for duodenal ulcer. *International Journal of Man-Machine Studies*, 24 (1986) 413-433.
- [33] Pawlak, Z., Słowiński, R.: Rough set approach to multi-attribute decision analysis. *European J. of Operational Research*, 72 (1994) 443-459.
- [34] Rossi, L., Słowiński, R., Susmaga, R.: Rough set approach to evaluation of stormwater pollution. *International Journal of Environment and Pollution*, 12 (1999) 232-250.
- [35] Słowiński, R.: Rough set learning of preferential attitude in multi-criteria decision making, in J. Komorowski, Z. W. Ras (eds.), *Methodologies for Intelligent Systems*, LNAI 689, Springer, Berlin, 1993, pp. 642-651.
- [36] Słowiński, R., Greco, S.: Inducing Robust Decision Rules from Rough Approximations of a Preference Relation. [In]: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L. A. Zadeh (eds.), *Artificial Intelligence and Soft Computing. Lecture Notes in Artificial Intelligence*, vol. 3070, Springer-Verlag, Berlin Heidelberg, 2004, pp. 118-132.
- [37] Słowiński, R., Greco, S., Matarazzo, B.: Rough set analysis of preference-ordered data, in J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (eds.), *Rough Sets and Current Trends in Computing*, LNAI 2475, Springer, Berlin, 2002, pp. 44-59.

- [38] Słowiński, R., Greco, S., Matarazzo, B.: Rough Set Based Decision Support. Chapter 16 [in]: E. K. Burke, G. Kendall (eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer-Verlag, New York, 2005, pp. 475-527.
- [39] Słowiński, R., Greco, S., Matarazzo, B.: Dominance-based rough set approach to reasoning about ordinal data, keynote lecture in M.Kryszkiewicz, J.F.Peters, H.Rybiński, A.Skowron (eds.), *Rough Sets and Intelligent Systems Paradigms*. LNAI 4585, Springer, Berlin, 2007, pp. 5-11.
- [40] Słowiński, R., Zopounidis, C.: Application of the rough set approach to evaluation of bankruptcy risk. *International Journal of Intelligent Systems in Accounting, Finance and Management* 4 (1995) 27-41.
- [41] Słowiński, R., Zopounidis, C., Dimitras, A.I.: Prediction of company acquisition in Greece by means of the rough set approach. *European Journal of Operational Research*, 100 (1997) 1-15.
- [42] Szeląg, M., Greco, S., Słowiński, R.: Rule-Based Approach to Multicriteria Ranking. [In]: M. Doumpos, E. Grigoroudis (eds.), *Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications*. Wiley, 2013, pp. 127-160.
- [43] Vincke, P.: Exploitation of a crisp relation in a ranking problem. *Theory and Decision*, vol. 32, 1992, pp. 221-240.
- [44] Wilk, S., Słowiński, R., Michałowski, W., Greco, S.: Supporting triage of children with abdominal pain in the emergency room. *European Journal of Operational Research*, 160 (2005) 696-709.
- [45] Wong, S. K. M., Ziarko, W.: Comparison of the probabilistic approximate classification and the fuzzy set model. *Fuzzy Sets and Systems*, vol. 21, 1987, pp. 357-362.