Decision and Inhibitory Rule Optimization for Decision Tables with Many-valued Decisions

Thesis by
Fawaz Jaber Alsolami

In Partial Fulfillment of the Requirements
For the Degree of

Doctor of Philosophy in Computer Science

King Abdullah University of Science and Technology, Thuwal,
Kingdom of Saudi Arabia

(04, 2016)
The thesis of Fawaz Jaber Alsolami is approved by the examination committee

Committee Chairperson: Mikhail Moshkov
Committee Member: Zbigniew Suraj
Committee Member: Vladimir Bajic
Committee Member: Xiangliang Zhang
ABSTRACT

Decision and Inhibitory Rule Optimization for Decision Tables with Many-valued Decisions

Fawaz Jaber Alsolami

‘If-then’ rule sets are one of the most expressive and human-readable knowledge representations. This thesis deals with optimization and analysis of decision and inhibitory rules for decision tables with many-valued decisions. The most important areas of applications are knowledge extraction and representation.

The benefit of considering inhibitory rules is connected with the fact that in some situations they can describe more knowledge than the decision ones. Decision tables with many-valued decisions arise in combinatorial optimization, computational geometry, fault diagnosis, and especially under the processing of data sets.

In this thesis, various examples of real-life problems are considered which help to understand the motivation of the investigation. We extend relatively simple results obtained earlier for decision rules over decision tables with many-valued decisions to the case of inhibitory rules. The behavior of Shannon functions (which characterize complexity of rule systems) is studied for finite and infinite information systems, for global and local approaches, and for decision and inhibitory rules.

The extensions of dynamic programming for the study of decision rules over decision tables with single-valued decisions are generalized to the case of decision tables with many-valued decisions. These results are also extended to the case of inhibitory
rules. As a result, we have algorithms (i) for multi-stage optimization of rules relative to such criteria as length or coverage, (ii) for counting the number of optimal rules, (iii) for construction of Pareto optimal points for bi-criteria optimization problems, (iv) for construction of graphs describing relationships between two cost functions, and (v) for construction of graphs describing relationships between cost and accuracy of rules.

The applications of created tools include comparison (based on information about Pareto optimal points) of greedy heuristics for bi-criteria optimization of rules, and construction (based on multi-stage optimization of rules) of relatively short systems of rules that can be used for knowledge representation.
ACKNOWLEDGEMENTS

All praise is to Allah for giving me the ability to complete this dissertation. I would like to express my deepest and sincere gratitude to my supervisor, Professor Mikhail Moshkov for his valuable support, guidance and encouragement were vital to achieving this degree.

I would also like to extend my appreciation to my committee members, Professor Zbigniew Suraj, Professor Vladimir Bajic, and Professor Xiangliang Zhang, for their time in reading this dissertation, and for their valuable comments.

Last but not least, I want to thank my wonderful parents and family for being patient with me, and for offering words of encouragement.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examination Committee Approval</td>
<td>2</td>
</tr>
<tr>
<td>Copyright</td>
<td>3</td>
</tr>
<tr>
<td>Abstract</td>
<td>4</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>6</td>
</tr>
<tr>
<td>List of Figures</td>
<td>11</td>
</tr>
<tr>
<td>List of Tables</td>
<td>12</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>1.1 Contributions</td>
<td>16</td>
</tr>
<tr>
<td>1.2 Organization of the Thesis</td>
<td>17</td>
</tr>
<tr>
<td>2 Explaining Examples</td>
<td>19</td>
</tr>
<tr>
<td>2.1 Problems with Many-valued Decisions</td>
<td>19</td>
</tr>
<tr>
<td>2.2 Decision Tables</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Examples of Problems with Many-valued Decisions</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1 Problem of Three Post-Offices</td>
<td>23</td>
</tr>
<tr>
<td>2.3.2 Traveling Salesman Problem with Four Cities</td>
<td>26</td>
</tr>
<tr>
<td>2.3.3 Diagnosis of One-gate Circuit</td>
<td>28</td>
</tr>
<tr>
<td>2.3.4 Example of Data Table</td>
<td>30</td>
</tr>
<tr>
<td>2.4 Difference between Decision and Inhibitory</td>
<td>31</td>
</tr>
<tr>
<td>2.4.1 Prediction Problem</td>
<td>31</td>
</tr>
<tr>
<td>2.4.2 Knowledge Representation Problem</td>
<td>32</td>
</tr>
<tr>
<td>3 Preliminary Results for Decision and Inhibitory Rules and Rule Systems</td>
<td>35</td>
</tr>
<tr>
<td>3.1 Main Notions</td>
<td>35</td>
</tr>
</tbody>
</table>
6.1.1 Representation of the Set of \((U, \alpha)\)-Decision Rules \hspace{1cm} 92
6.1.2 Procedure of Optimization \hspace{1cm} 93
6.1.3 Number of Rules in Rule\((G, \Theta, r)\) \hspace{1cm} 99
6.1.4 Experimental Results for Decision Rule Optimization \hspace{1cm} 101
6.1.5 Greedy Algorithm for Construction of Decision Rule Set \hspace{1cm} 104
6.1.6 Experimental Results for Greedy Algorithm \hspace{1cm} 108
6.2 Multi-stage Optimization of Inhibitory Rules \hspace{1cm} 111
6.2.1 Experimental Results for Inhibitory Rule Optimization \hspace{1cm} 113

7 Bi-Criteria Optimization Problem: Cost vs Cost \hspace{1cm} 119
7.1 Bi-Criteria Optimization Problem for Decision Rules and Systems of Rules: Cost vs Cost \hspace{1cm} 119
7.1.1 Pareto Optimal Points for Decision Rules: Cost vs Cost \hspace{1cm} 120
7.1.2 Relationships for Decision Rules: Cost vs Cost \hspace{1cm} 125
7.1.3 Pareto Optimal Points for Systems of Decision Rules: Cost vs Cost \hspace{1cm} 126
7.1.4 Relationships for Systems of Decision Rules: Cost vs Cost \hspace{1cm} 130
7.2 Comparison of Heuristics for Decision Rule Construction \hspace{1cm} 132
7.2.1 Greedy Heuristics \hspace{1cm} 132
7.2.2 Experimental Results \hspace{1cm} 134
7.3 Bi-criteria Optimization of Inhibitory Rules: Cost vs Cost \hspace{1cm} 138
7.3.1 Pareto Optimal Points for Inhibitory Rules: Cost vs Cost \hspace{1cm} 138
7.3.2 Pareto Optimal Points for Systems of Inhibitory Rules: Cost vs Cost \hspace{1cm} 140
7.4 Experimental Results for Pareto Optimal Points for Systems of Inhibitory Rules \hspace{1cm} 142

8 Bi-Criteria Optimization Problem: Cost vs Uncertainty (Completeness) \hspace{1cm} 145
8.1 Bi-Criteria Optimization Problem for Decision Rules and Systems of Rules: Cost vs Uncertainty \hspace{1cm} 145
8.1.1 Pareto Optimal Points for Decision Rules: Cost vs Uncertainty \hspace{1cm} 146
8.1.2 Relationships for Decision Rules: Cost vs Uncertainty \hspace{1cm} 153
8.1.3 Pareto Optimal Points for Systems of Decision Rules: Cost vs Uncertainty \hspace{1cm} 154
8.1.4 Relationships for Systems of Decision Rules: Cost vs Uncertainty \hspace{1cm} 158
8.1.5 Experimental Results: Cost vs Uncertainty \hspace{1cm} 159
LIST OF FIGURES

2.1 Problem of three post-offices ........................................ 23
2.2 Distances to points $B_1$ and $B_2$ ..................................... 24
2.3 Decision table for problem of three post-offices ..................... 25
2.4 Traveling salesman problem with four cities .......................... 26
2.5 Decision table for traveling salesman problem with four cities . . . 27
2.6 One-gate circuit .......................................................... 29
2.7 Decision table for problem of diagnosis of one-gate circuit ......... 29
2.8 Data table $D$ ............................................................. 30
2.9 Decision table corresponding to data table $D$ ........................ 30
2.10 Decision table with many-valued decisions ............................ 31
2.11 Data table (information system) $I$ .................................. 32
3.1 Decision table $T_3^{IC}$ complementary to the decision table $T_3'$ .... 40
7.1 Pareto optimal points and rule heuristic solutions for breast-cancer-1 137
7.2 Pareto optimal points and rule heuristic solutions for mushroom-5 ... 137
7.3 Pareto optimal points and rule heuristic solutions for zoo-data-5 .... 138
7.4 Pareto optimal point and rule heuristic solutions for the decision table complementary to zoo-data-5' .................................. 144
8.1 Pareto optimal points for zoo-data-5: $-c$ vs $rme$ ..................... 160
8.2 Pareto optimal points for cars-1: $-c$ vs $rme$ .......................... 161
8.3 Pareto optimal points for zoo-data-5': $-c$ vs. $irme$ .................... 166
8.4 Pareto optimal points for cars-1': $-c$ vs $irme$ .......................... 166
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Modified decision tables from UCI ML Repository</td>
<td>102</td>
</tr>
<tr>
<td>6.2</td>
<td>Length of decision rules: after the procedure relative to length</td>
<td>103</td>
</tr>
<tr>
<td>6.3</td>
<td>Coverage of decision rules: after the procedure relative to coverage</td>
<td>103</td>
</tr>
<tr>
<td>6.4</td>
<td>Sequential optimization of decision rules</td>
<td>104</td>
</tr>
<tr>
<td>6.5</td>
<td>Bounds for decision tables with many-valued decisions</td>
<td>109</td>
</tr>
<tr>
<td>6.6</td>
<td>Bounds for decision table: breast-cancer-1</td>
<td>109</td>
</tr>
<tr>
<td>6.7</td>
<td>Bounds for decision table: breast-cancer-5</td>
<td>110</td>
</tr>
<tr>
<td>6.8</td>
<td>Bounds for decision table: mushroom-5</td>
<td>110</td>
</tr>
<tr>
<td>6.9</td>
<td>Bounds for decision table: nursery-1</td>
<td>111</td>
</tr>
<tr>
<td>6.10</td>
<td>Sequential optimization of decision rules for decision tables complementary to $T'$</td>
<td>114</td>
</tr>
<tr>
<td>6.11</td>
<td>Bounds for decision table complementary to breast-cancer-1'</td>
<td>115</td>
</tr>
<tr>
<td>6.12</td>
<td>Bounds for decision table complementary to breast-cancer-5'</td>
<td>115</td>
</tr>
<tr>
<td>6.13</td>
<td>Bounds for decision table complementary to cars-1'</td>
<td>116</td>
</tr>
<tr>
<td>6.14</td>
<td>Bounds for decision table complementary to mushroom-5'</td>
<td>116</td>
</tr>
<tr>
<td>6.15</td>
<td>Bounds for decision table complementary to nursery-1'</td>
<td>117</td>
</tr>
<tr>
<td>6.16</td>
<td>Bounds for decision table complementary to nursery-4'</td>
<td>117</td>
</tr>
<tr>
<td>6.17</td>
<td>Bounds for decision table complementary to teeth-1'</td>
<td>118</td>
</tr>
<tr>
<td>6.18</td>
<td>Bounds for decision table complementary to teeth-5'</td>
<td>118</td>
</tr>
<tr>
<td>7.1</td>
<td>Number of Pareto optimal points for some modified decision tables from UCI ML Repository</td>
<td>135</td>
</tr>
<tr>
<td>7.2</td>
<td>Comparing heuristics for decision tables with many-valued decisions</td>
<td>136</td>
</tr>
<tr>
<td>7.3</td>
<td>Number of Pareto optimal points for decision tables complementary to $T'$</td>
<td>143</td>
</tr>
<tr>
<td>7.4</td>
<td>Comparing heuristics on complementary decision tables $T^C$ and on $T'$</td>
<td>144</td>
</tr>
<tr>
<td>8.1</td>
<td>Number of Pareto optimal points for decision tables $T$: $-c$ vs. $rme$</td>
<td>160</td>
</tr>
<tr>
<td>8.2</td>
<td>Number of Pareto optimal points for decision tables $T$: $l$ vs. $rme$</td>
<td>160</td>
</tr>
<tr>
<td>8.3</td>
<td>Number of Pareto optimal points for decision tables $T'$: $-c$ vs. $irme$</td>
<td>165</td>
</tr>
</tbody>
</table>
8.4  Number of Pareto optimal points for decision tables $T'$: $l$ vs. $irme$ . . 165
Chapter 1

Introduction

Decision rules are widely used as a way to solve problems, as a tool to extract and represent knowledge from decision tables (data sets), and in classifiers for unseen objects. For instance, data mining community prefers using decision rule systems when the goal is to gain insight into data [1, 2]. Many methods are proposed in the literature to construct decision rule systems either directly from data sets, for example, logical analysis of data [3, 4, 5], rough set theory [6, 7], sequential covering [8, 9, 10, 11, 12, 13], or from models such as decision trees [14, 15]. In this work, we describe algorithms based on extensions of dynamic programming which allows us to optimize decision and inhibitory rules for decision tables with many-valued decisions.

Usual decision and association rules with a relation “attribute = value” on the right-hand side are well-known as a way for knowledge representation and pattern discovery [16]. However, for some information systems, it was shown in [17, 18] that usual association rules cannot describe all information (knowledge) contained in the information systems. On the other hand, inhibitory association rules describe all information for every information system [19]. Inhibitory rules have in the consequent part a relation “attribute ≠ value”. Our work is devoted to the study of both decision and inhibitory rules and rule systems.

Decision tables with many-valued decisions (or data sets with multivariate responses, or multi-label data sets) arise in important problems of combinatorial op-
timization, fault diagnosis, computational geometry, and under processing of inconsistent decision tables in rough set theory, etc. [20, 21, 6]. In real-life applications, we can encounter multi-label data sets when we study the problem of categorizing images into semantic classes [22] where an image can be annotated by more than one class, functional genomics [23] where a gene can have more than one function, text categorization [24] where a document may belong to more than one category, and music categorization [25] where a song may classify into multiple emotions. In contrast with single-label data sets, rows (objects) of multi-label data sets are associated with a set of decisions (labels). Two main approaches have been proposed for dealing with multi-label data sets in data analysis. The first approach is based on algorithm adaptation methods which means extending existing algorithms to handle multi-label data sets such as decision trees [26, 27]. The second approach is called problem transformation methods where a given multi-label data set transforms into the corresponding single-label data set(s) [28, 29]. In this thesis, we consider the former approach where we find an arbitrary decision from the associated set of decisions attached to a row.

Generally, two rule characteristics play important roles in the choice of meaningful rules. Rule coverage, the number of rows (objects) that a rule applies to and gives an appropriate answer, is crucial in order to discover major patterns in a given data set. Second, rule length, the number of conditions (constraints) on the left-hand side of a rule, is important for understanding and interoperability. Unfortunately, the problems of construction of rules with maximum coverage [30] or rules with minimum length [21] are NP-hard.

We study rules as combinatorial objects where we aim to design algorithms for optimization and analysis of rules based on extensions of dynamic programming approach. These extensions allow us to make multi-stage optimization of rules relative to different cost functions, to count the number of optimal rules, to study relationships
between two cost functions or between a cost function and uncertainty (accuracy) of rules, and to find all Pareto optimal points for bi-criteria optimization problems.

The created tools are applicable to medium-sized data sets, which allow us to use these tools for comparison of heuristics for bi-criteria optimization problems and for solving of problems connected with knowledge representation.

In this work, we used a dynamic programming solver Dagger [31] and additional software created to support experiments with decision and inhibitory rules.

The most part of the experiments were done on high performance computing system with 60 cores and 1.5 TB of RAM since the algorithms based on dynamic programming approach need large amount of shared memory.

1.1 Contributions

The main contributions of this thesis can be divided into four categories:

- **Generalization** of bounds on complexity and approximate algorithms for decision rule optimization for decision tables with many-valued decisions obtained in [21] to inhibitory rules (see Chapter 3). This chapter also contains relationships between complete systems of inhibitory rules, inhibitory tests, and inhibitory trees.

- **Generalization** of extensions of dynamic programming for decision rules obtained by Talha Amin for decision tables with single-valued decisions (these results are partially published in [32, 33]) to decision tables with many-valued decisions (see Chapters 4-8). As a result, we have tools which are applicable to decision tables with many-valued decisions (see Chapter 2 which shows the importance of such tables) and allow us to make multi-stage optimization of decision rules, count the number of optimal rules, study the relationships cost vs. cost and cost vs. accuracy.
• **Generalization** of extensions of dynamic programming for decision rules over
decision tables with many-valued decisions to inhibitory rules (see Chapters 4-8). The usefulness of inhibitory rules for knowledge representation and pre-
diction is discussed in Chapter 2. The obtained results allow us to apply to
inhibitory rules all methods created for decision rules including construction of
the set of Pareto optimal points for bi-criteria optimization problems.

• **Studying** of various types of Shannon functions for decision and inhibitory
rules for decision tables with many-valued decisions (see Chapter 9). As a
result, we described all possible types of Shannon functions behavior both for
finite and infinite information systems, and proved that Shannon functions for
decision and inhibitory rules have the same behavior.

### 1.2 Organization of the Thesis

This thesis is organized as follows. Chapter 2 contains some definitions and examples
of problems with many-valued decisions from different areas of applications: fault
diagnosis, computational geometry, combinatorial optimization, and analysis of ex-
perimental data. In Chapter 2, we also discuss two examples which explain why we
consider not only decision but also inhibitory rules.

Chapter 3 is devoted to the consideration of some results from [21] related to binary
decision tables with many-valued decisions: relationships among decision trees, rules
and tests, bounds on their complexity, greedy algorithms for construction of decision
rules and systems of rules, and dynamic programming algorithms for minimization of
tree depth and rule length. We extend the discussed results related to decision rules
and rule systems to inhibitory rules and rule systems over binary decision tables with
many-valued decisions.

Chapter 4 presents some notions connected with decision tables with many-valued

decisions (the notions of table, directed acyclic graph for this table, uncertainty and completeness measures, and restricted information system) and discusses tools for working with Pareto optimal points.

In Chapter 5, we consider different types of decision and inhibitory rules and systems of rules.

Chapter 6 dedicates to optimization of decision and inhibitory rules including multi-stage optimization relative to a sequence of cost functions. We discuss algorithms for counting the number of optimal rules. We also consider simulation of a greedy algorithm for construction of decision rule set. This chapter contains experimental results connected with knowledge representation.

We consider algorithms which construct the sets of Pareto optimal points for bi-criteria optimization problems for decision and inhibitory rules and rule systems relative two cost functions in Chapter 7. We also show how the constructed set of Pareto optimal points can be transformed into the graphs of functions which describe the relationships between the considered cost functions. This chapter contains experimental results on comparison of heuristics for bi-criteria optimization of rules.

In Chapter 8, we consider algorithms which construct the sets of Pareto optimal points for bi-criteria optimization problems for decision and inhibitory rules and rule systems relative to a cost function and an uncertainty (completeness) measure. We also show how the constructed set of Pareto optimal points can be transformed into the graphs of functions which describe the relationships between the considered cost function and uncertainty (completeness) measure.

Chapter 9 studies time complexity of decision and inhibitory rule systems over arbitrary sets of attributes represented in the form of information systems.

Finally, Chapter 10 concludes the thesis.
2.1 Problems with Many-valued Decisions

We begin with a simple model of a problem. Let $A$ be a set (set of inputs or the universe). It is possible that $A$ is an infinite set. Let $f_1, \ldots, f_n$ be attributes, each of which is a function from $A$ to $B$ where $B$ is a nonempty finite set. Attributes $f_1, \ldots, f_n$ divide the set $A$ into a number of domains in each of which values of
attributes are constant. These domains are labeled with nonempty finite subsets of the set \( \omega = \{0, 1, 2, \ldots \} \) of nonnegative integers. We will interpret these subsets as sets of decisions.

More formally, a problem is a tuple \( z = (\nu, f_1, \ldots, f_n) \) where \( \nu \) is a mapping from \( B^n \) to the set of all nonempty finite subsets of the set \( \omega \). Each domain corresponds to the nonempty set of solutions on \( A \) of a set of equations of the kind

\[
\{ f_1(x) = \delta_1, \ldots, f_n(x) = \delta_n \}
\]

where \( \delta_1, \ldots, \delta_n \in B \). Denote \( D(z) = \bigcup \nu(\delta_1, \ldots, \delta_n) \) where union is considered over all \( (\delta_1, \ldots, \delta_n) \in B^n \) for which \( \{ f_1(x) = \delta_1, \ldots, f_n(x) = \delta_n \} \) has a solution from \( A \).

For a given \( a \in A \), denote \( z(a) = \nu(f_1(a), \ldots, f_n(a)) \).

We will consider two interpretations of the problem \( z \): decision and inhibitory.

**Decision** interpretation: for a given \( a \in A \), we should find a number from the set \( z(a) \).

**Inhibitory** interpretation: for a given \( a \in A \), we should find a number from the set \( D(z) \setminus z(a) \) (we will assume here that \( D(z) \neq z(a) \) for any \( a \in A \)).

In the case of decision interpretation, we study decision rules. A decision rule \( \rho \) over \( z \) is an expression of the kind

\[
f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \rightarrow t
\]

where \( f_{i_1}, \ldots, f_{i_m} \in \{ f_1, \ldots, f_n \} \), \( b_1, \ldots, b_m \in B \), and \( t \in \omega \). The number \( m \) is called the length of the rule \( \rho \). This rule is called realizable for an element \( a \in A \) if

\[
f_{i_1}(a) = b_1, \ldots, f_{i_m}(a) = b_m.
\]

The rule \( \rho \) is called true for \( z \) if, for any \( a \in A \) such that \( \rho \) is realizable for \( a, t \in z(a) \).
A *decision rule system* $S$ over $z$ is a nonempty finite set of decision rules over $z$. A system $S$ is called a *complete* decision rule system for $z$ if each rule from $S$ is true for $z$ and, for every $a \in A$, there exists a rule from $S$ which is realizable for $a$.

We denote by $l(S)$ the maximum length of a rule from $S$ (the *length* of $S$), and by $l(z)$ we denote the minimum value of $l(S)$ among all complete decision rule systems $S$ for $z$.

In the case of inhibitory interpretation, we study inhibitory rules. An *inhibitory rule* $\rho$ over $z$ is an expression of the kind

$$f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \rightarrow \neq t$$

where $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $b_1, \ldots, b_m \in B$, and $t \in \omega$. The number $m$ is called the *length* of the rule $\rho$. This rule is called *realizable* for an element $a \in A$ if

$$f_{i_1}(a) = b_1, \ldots, f_{i_m}(a) = b_m.$$

The rule $\rho$ is called *true* for $z$ if, for any $a \in A$ such that $\rho$ is realizable for $a$, $t \in D(z) \setminus z(a)$.

An *inhibitory rule system* $S$ over $z$ is a nonempty finite set of inhibitory rules over $z$. A system $S$ is called a *complete* inhibitory rule system for $z$ if each rule from $S$ is true for $z$ and, for every $a \in A$, there exists a rule from $S$ which is realizable for $a$.

We denote by $l(S)$ the maximum length of a rule from $S$ (the *length* of $S$), and by $il(z)$ we denote the minimum value of $l(S)$ among all complete inhibitory rule systems $S$ for $z$.

### 2.2 Decision Tables

We associate a *decision table* $T = T(z)$ with the considered problem $z$. 
This table is a rectangular table with \( n \) columns labeled with attributes \( f_1, \ldots, f_n \). A tuple \( (\delta_1, \ldots, \delta_n) \in B^n \) is a row of \( T \) if and only if the system of equations

\[
\{ f_1(x) = \delta_1, \ldots, f_n(x) = \delta_n \}
\]

is compatible on the set \( A \) (has a solution on the set \( A \)). This row is labeled with the set \( \nu(\delta_1, \ldots, \delta_n) \).

We can formulate the notion of decision rule over \( T \), the notion of decision rule realizable for a row of \( T \), and the notion of decision rule true for \( T \) in a natural way. We will say that a system \( S \) of decision rules over \( T \) is a complete decision rule system for \( T \) if each rule from \( S \) is true for \( T \) and, for every row of \( T \), there exists a rule from \( S \) which is realizable for this row. We denote by \( l(T) \) the minimum value of \( l(S) \) among all complete decision rule systems \( S \) for \( T \). One can show that a decision rule system \( S \) over \( z \) is complete for \( z \) if and only if \( S \) is complete for \( T = T(z) \). So \( l(z) = l(T(z)) \).

We can formulate the notion of inhibitory rule over \( T \), the notion of inhibitory rule realizable for a row of \( T \), and the notion of inhibitory rule true for \( T \) in a natural way. We will say that a system \( S \) of inhibitory rules over \( T \) is a complete inhibitory rule system for \( T \) if each rule from \( S \) is true for \( T \) and, for every row of \( T \), there exists a rule from \( S \) which is realizable for this row. We denote by \( il(T) \) the minimum value of \( l(S) \) among all complete inhibitory rule systems \( S \) for \( T \). One can show that an inhibitory rule system \( S \) over \( z \) is complete for \( z \) if and only if \( S \) is complete for \( T = T(z) \). So \( il(z) = il(T(z)) \).

As a result, instead of the problem \( z \) we can study the decision table \( T(z) \).
2.3 Examples of Problems with Many-valued Decisions

Classes of exactly formulated problems and experimental data are the main providers of problems with many-valued decisions and corresponding decision tables. The areas from which examples of exactly formulated problems with many-valued decisions will be considered are:

- Computational geometry,
- Combinatorial optimization,
- Diagnosis of faults in combinatorial circuits.

Additionally, an example of a decision table from experimental data will be considered.

2.3.1 Problem of Three Post-Offices

We begin with an example of problems studied in computational geometry. Let us assume we have three post-offices $P_1$, $P_2$ and $P_3$ (see Figure 2.1 from [21]). A new client is served by a nearest post-office.

![Figure 2.1: Problem of three post-offices](image)

Figure 2.1: Problem of three post-offices
Let we have two points \( B_1 \) and \( B_2 \). We join those two points by a segment (of straight line) and draw the perpendicular line through the center of this segment (see Figure 2.2 from [21]). All points which lie on the left of this perpendicular line are nearer to \( B_1 \), all points that lie on the perpendicular line have the same distance to \( B_1 \) and to \( B_2 \), and all points which lie on the right of the perpendicular line are nearer to the point \( B_2 \). This reasoning allows us to construct attributes for the problem of three post-offices which will be denoted \( z_1 \).

We joint all pairs of post-offices \( P_1, P_2, P_3 \) by segments (these segments are invisible in Figure 2.1) and draw perpendicular lines through centers of these segments. These perpendicular lines correspond to three attributes \( f_1, f_2, f_3 \). Each such attribute takes value \(-1\) on the left of the considered line, takes value \(0\) on the line, and takes value \(+1\) on the right of the considered line (arrow points to the right). These three straight lines divide the plane into 13 regions. We correspond to each region the set of numbers of post-offices which are nearest to points of this region and construct the decision table \( T_1 \) for the considered problem (see Figure 2.3).

One can show that

\[
\{ f_3 = -1 \land f_2 = -1 \rightarrow 1, f_3 = +1 \land f_1 = -1 \rightarrow 2, f_2 = +1 \land f_1 = +1 \rightarrow 3, \\
 f_1 = 0 \land f_3 = +1 \rightarrow 2, f_3 = 0 \land f_1 = -1 \rightarrow 1, f_2 = 0 \land f_1 = +1 \rightarrow 1, \\
 f_1 = 0 \land f_2 = 0 \rightarrow 1 \}
\]
is a complete decision rule system with minimum length for the problem $z_1$ and for the table $T_1$.

For the consideration of inhibitory rules, we should remove the central point which is the intersection of lines corresponding to attributes $f_1, f_2, f_3$ (see Figure 2.1) since the whole set of decisions $\{1, 2, 3\}$ corresponds to this point. We denote the obtained problem $z'_1$. We should also remove from the table $T_1$ the row $(0, 0, 0)$ labeled with the whole set of decisions $\{1, 2, 3\}$. We denote the obtained decision table $T'_1$.

One can show that

$$\{f_3 = +1 \rightarrow \neq 1, f_3 = -1 \rightarrow \neq 2, f_1 = +1 \rightarrow \neq 2, f_1 = -1 \rightarrow \neq 3\}$$

is a complete inhibitory rule system with minimum length for the problem $z'_1$ and for the table $T'_1$. 

\begin{figure}
\centering
\begin{tabular}{|c|c|c|c|}
\hline
& $f_1$ & $f_2$ & $f_3$ \\
\hline
+1 & +1 & +1 & \{3\} \\
0 & +1 & +1 & \{2, 3\} \\
-1 & +1 & +1 & \{2\} \\
-1 & 0 & +1 & \{2\} \\
-1 & -1 & +1 & \{2\} \\
-1 & -1 & 0 & \{1, 2\} \\
-1 & -1 & -1 & \{1\} \\
0 & -1 & -1 & \{1\} \\
+1 & -1 & -1 & \{1\} \\
+1 & 0 & -1 & \{1, 3\} \\
+1 & +1 & -1 & \{3\} \\
+1 & +1 & 0 & \{3\} \\
0 & 0 & 0 & \{1, 2, 3\} \\
\hline
\end{tabular}
\caption{Decision table for problem of three post-offices}
\end{figure}
2.3.2 Traveling Salesman Problem with Four Cities

Now, we will present an example of *combinatorial optimization* problem. Let we have a complete unordered graph with four nodes in which each edge is labeled with a real number – the length of this edge (see Figure 2.4 from [21]).

![Figure 2.4: Traveling salesman problem with four cities](image)

A Hamiltonian circuit is a closed path which passes through each node exactly one time. We should find a Hamiltonian circuit which has minimum length. There are three Hamiltonian circuits:

- $H_1$: 12341 or, which is the same, 14321,
- $H_2$: 12431 or 13421,
- $H_3$: 13241 or 14231.

For $i = 1, 2, 3$, we denote by $L_i$ the length of $H_i$.

Then

$$L_1 = x_{12} + x_{23} + x_{34} + x_{14} = (x_{12} + x_{34}) + (x_{23} + x_{14}),$$

$$L_2 = x_{12} + x_{24} + x_{34} + x_{13} = (x_{12} + x_{34}) + (x_{24} + x_{13}),$$

$$L_3 = x_{13} + x_{23} + x_{24} + x_{14} = (x_{24} + x_{13}) + (x_{23} + x_{14}).$$

In the capacity of attributes we will use three functions $f_1 = \text{sign}(L_1 - L_2)$,
27

\[ f_2 = \text{sign}(L_1 - L_3) \text{, and } f_3 = \text{sign}(L_2 - L_3) \text{ where } \text{sign}(x) = -1 \text{ if } x < 0, \text{sign}(x) = 0 \text{ if } x = 0, \text{and } \text{sign}(x) = +1 \text{ if } x > 0. \]

The values \(L_1, L_2\) and \(L_3\) are linearly ordered. Let us show that any order is possible. It is clear that the values of \(\alpha, \beta\) and \(\gamma\) can be chosen independently.

We can construct corresponding decision table \(T_2\) (see Figure 2.5).

<table>
<thead>
<tr>
<th>(f_1)</th>
<th>(f_2)</th>
<th>(f_3)</th>
<th>(\alpha &lt; \beta &lt; \gamma)</th>
<th>(\alpha = \beta &lt; \gamma)</th>
<th>(\alpha &lt; \beta = \gamma)</th>
<th>(\alpha = \beta = \gamma)</th>
<th>(\beta &lt; \alpha &lt; \gamma)</th>
<th>(\beta &lt; \alpha = \gamma)</th>
<th>(\beta &lt; \gamma &lt; \alpha)</th>
<th>(\beta = \gamma &lt; \alpha)</th>
<th>(\gamma &lt; \alpha &lt; \beta)</th>
<th>(\gamma &lt; \alpha = \beta)</th>
<th>(\gamma &lt; \beta &lt; \alpha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1)</td>
<td>(-1)</td>
<td>(-1)</td>
<td>(L_1 &lt; L_2 &lt; L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
</tr>
<tr>
<td>(-1)</td>
<td>(-1)</td>
<td>(0)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
</tr>
<tr>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
<td>(L_2 = L_3)</td>
<td>(L_2 &lt; L_3)</td>
</tr>
</tbody>
</table>

Figure 2.5: Decision table for traveling salesman problem with four cities
One can show that

\[
\{ f_1 = -1 \land f_2 = -1 \rightarrow 1, f_1 = +1 \land f_3 = -1 \rightarrow 2, \\
f_2 = +1 \land f_3 = +1 \rightarrow 3, f_1 = 0 \land f_2 = 0 \rightarrow 1, \\
f_1 = 0 \land f_2 = -1 \rightarrow 1, f_1 = -1 \land f_2 = 0 \rightarrow 1, \\
f_2 = +1 \land f_3 = 0 \rightarrow 2 \}
\]

is a complete decision rule system with minimum length for \( T_2 \).

For the consideration of inhibitory rules, we should remove from the table \( T_2 \) the row \( (0, 0, 0) \) labeled with the whole set of decisions \( \{1, 2, 3\} \). We denote the obtained decision table \( T'_2 \).

One can show that

\[
\{ f_2 = -1 \nrightarrow \neq 3, f_2 = +1 \nrightarrow \neq 1, f_3 = -1 \nrightarrow \neq 3, f_3 = +1 \nrightarrow \neq 2 \}
\]

is a complete inhibitory rule system with minimum length for \( T'_2 \).

### 2.3.3 Diagnosis of One-gate Circuit

An example of faults diagnosis problem will be considered in this section. This example was discussed in [21]. We add here only a result about complete system of inhibitory rules. Let us consider the combinatorial circuit \( S \) depicted in Figure 2.6 from [21]. An input to the circuit \( S \) may either work correctly or can have a constant fault 1. Let us assume that we have at least one fault for \( S \). The goal is to find an input with a fault. To solve such problem, attributes from the set \( \{0, 1\}^3 \) will be considered. Then, we feed the circuit \( S \) with a set of values of attributives on inputs and check the value of the output of \( S \) which is the value of the considered attribute. Obviously, the circuit \( S \) with at least one fault 1 on an input realizes one of functions
Figure 2.6: One-gate circuit

from the set \{1, x, y, z, xy, xz, yz\} (we write $xy$ instead of $x \land y$). A corresponding decision table $T_3$ is represented in Figure 2.7 (see [21]). We do not encode decisions $x, y, z$ by numbers $1, 2, 3.$

\[
T_3 = \begin{array}{cccccccccc}
000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \{x, y, z\} \\
x & 0 & 0 & 0 & 0 & 1 & 1 & 1 & \{y, z\} \\
y & 0 & 0 & 1 & 1 & 0 & 0 & 1 & \{x, z\} \\
z & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \{x, y\} \\
xy & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \{z\} \\
xz & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \{y\} \\
xyz & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \{x\} \\
\end{array}
\]

Figure 2.7: Decision table for problem of diagnosis of one-gate circuit

One can show that

\[
\{011 = 1 \rightarrow x, 101 = 1 \rightarrow y, 110 = 1 \rightarrow z\}
\]

is a complete decision rule system with minimum length for the table $T_3$.

For the consideration of inhibitory rules, we should remove from the table $T_3$ the first row labeled with the whole set of decisions \{x, y, x\}. We denote the obtained decision table $T_3'$.

One can show that

\[
\{110 = 0 \rightarrow \neq z, 101 = 0 \rightarrow \neq y, 011 = 0 \rightarrow \neq x\}
\]
is a complete inhibitory rule system with minimum length for the table $T'_3$.

2.3.4 Example of Data Table

Decision tables with many-valued decisions are usually obtained from experimental data. Data set may contain conflicting rows or objects with the same values of conditional attributes but with different class labels (or different decisions). When we encounter such rows, we remove the duplicates and keep the most common decisions in the set of decisions associated with the row (as shown in next example). There are other techniques to construct this set as well such as including all decisions attached to duplicate rows or the top $k$ frequent class labels.

The following example was considered in [21]. Here, we only add a result for a complete system of inhibitory rules.

Suppose we have the table $D$ with discrete variables as shown in Figure 2.8 from [21]. The goal is to predict the value of $y$ given the values of attributes $f_1 = x_1$, $f_2 = x_2$, $f_3 = x_3$, $y$.

$$D = \begin{array}{cccc}
 x_1 & x_2 & x_3 & y \\
 0 & 0 & 1 & 1 \\
 0 & 0 & 1 & 2 \\
 0 & 0 & 1 & 2 \\
 0 & 0 & 1 & 3 \\
 0 & 0 & 1 & 3 \\
 1 & 0 & 0 & 1 \\
 1 & 0 & 0 & 3 \\
 1 & 0 & 0 & 3 \\
 1 & 0 & 1 & 2 \\
 1 & 0 & 1 & 3 \\
 0 & 1 & 1 & 1 \\
 0 & 1 & 1 & 2 \\
 0 & 1 & 1 & 2 \\
 1 & 1 & 0 & 1 \\
 1 & 1 & 0 & 1 \\
 1 & 1 & 0 & 3 \\
\end{array}$$

$$T_4 = \begin{array}{ccc}
 f_1 & f_2 & f_3 \\
 0 & 0 & 1 & \{2, 3\} \\
 1 & 0 & 0 & \{1, 3\} \\
 1 & 0 & 1 & \{2, 3\} \\
 0 & 1 & 1 & \{2\} \\
 1 & 1 & 0 & \{1\} \\
\end{array}$$

Figure 2.9: Decision table corresponding to data table $D$
$f_2 = x_2$, and $f_3 = x_3$. A corresponding decision table $T_4$ with many-valued decisions is shown in Figure 2.9 from [21].

One can show that

$$\{f_3 = 0 \rightarrow 1, f_3 = 1 \rightarrow 2\}$$

is a complete decision rule system with minimum length for the table $T_4$.

One can show that

$$\{f_3 = 0 \rightarrow \neq 2, f_3 = 1 \rightarrow \neq 1\}$$

is a complete inhibitory rule system with minimum length for the table $T$.

### 2.4 Difference between Decision and Inhibitory Rules

We consider here two examples which show that the use of inhibitory rules can give us additional possibilities in comparison with decisions rules.

#### 2.4.1 Prediction Problem

Let us consider the decision table $T_5$ with many-valued decisions depicted in Figure 2.10.

\[
T_5 = \begin{array}{c|c|c}
\hline
f_1 & f_2 & \{1\}, \{2, 4\}, \{2\}, \{3, 4\} \\
\hline
0 & 1 & \{1\} \\
0 & 2 & \{2, 4\} \\
1 & 0 & \{2\} \\
2 & 0 & \{3, 4\} \\
\hline
\end{array}
\]

**Figure 2.10:** Decision table with many-valued decisions

We will consider only true for $T_5$ and realizable for at least one row from $T_5$ decision and inhibitory rules. It is easy to see that decision rules can be simulated
in some sense by inhibitory rules. Let us consider, for example, the pair of decision rules $f_1 = 2 \rightarrow 3$ and $f_1 = 2 \rightarrow 4$. This pair can be simulated by the following pair of inhibitory rules: $f_1 = 2 \rightarrow \neq 1$ and $f_1 = 2 \rightarrow \neq 2$.

However, not each inhibitory rule can be simulated by decision rules. In particular, there are two inhibitory rules which are applicable to the row $(0, 0)$ which does not belong to $T_5$: $f_1 = 0 \rightarrow \neq 3$ and $f_2 = 0 \rightarrow \neq 1$, but there are no decision rules which are applicable to this row.

Let us consider the problem of prediction of decision value for a new object given by values of condition attributes $f_1 = 0$ and $f_2 = 0$. Decision rules will not give us any information about decisions corresponding to this new object. However, the inhibitory rules will restrict the set of possible decisions to $\{2, 4\}$.

### 2.4.2 Knowledge Representation Problem

Let us consider the data table (information system) $I$ depicted in Figure 2.11.

\[
I = \begin{array}{ccc}
(f_1 & f_2 \\
0 & 1 \\
1 & 0 \\
0 & 2 \\
2 & 0 \\
\end{array}
\]

**Figure 2.11:** Data table (information system) $I$

We will compare now knowledge that can be derived from $I$ by inhibitory and decision association rules which are true for $I$ and realizable for at least one row of $I$. *Decision association* rules have on the right hand side expressions of the kind $f_i = a$ where $a \in V_I(f_i)$ and $V_I(f_i)$ is the set of values of the attribute $f_i$ in the information system $I$. *Inhibitory association* rules have on the right hand side expressions of the kind $f_i \neq a$ where $a \in V_I(f_i)$.

For the information system $I$, $V_I(f_1) = V_I(f_2) = \{0, 1, 2\}$. We will consider the
set

\[ V_I(f_1) \times V_I(f_2) = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)\} \]

which contains all 2-tuples from \( I \) and some additional 2-tuples, for example, 2-tuple \((0, 0)\).

The question under consideration is about possibility to remove from the set \( V_I(f_1) \times V_I(f_2) \) all 2-tuples that do not belong to \( I \) using decision (inhibitory) association rules that are true for \( I \) and realizable for at least one 2-tuple (row) from \( I \).

One can show that the set of decision association rules that are true for \( I \) and realizable for at least one 2-tuple from \( I \) is the following:

\[
\begin{align*}
&f_1 = 1 \rightarrow f_2 = 0, \\
&f_1 = 2 \rightarrow f_2 = 0, \\
&f_2 = 1 \rightarrow f_1 = 0, \\
&f_2 = 2 \rightarrow f_1 = 0.
\end{align*}
\]

Based on these rules we can remove all 2-tuples from \( V_I(f_1) \times V_I(f_2) \) that do not belong to \( I \) with the exception of \((0, 0)\). For example the tuple \((2, 2)\) can be removed by the rule \( f_2 = 2 \rightarrow f_1 = 0 \). We cannot remove the 2-tuple \((0, 0)\) since no one rule is realizable for this 2-tuple. So, we have that the information derived from \( I \) by decision association rules is in some sense incomplete. The existence of such information systems was discovered in \([17, 18]\).

One can show that the set of inhibitory association rules that are true for \( I \) and
realizable for at least one 2-tuple from $I$ is the following:

\[
\begin{align*}
f_1 = 0 & \rightarrow f_2 \neq 0, \quad f_2 = 1 \rightarrow f_1 \neq 1, \\
f_1 = 1 & \rightarrow f_2 \neq 1, \quad f_2 = 1 \rightarrow f_1 \neq 2, \\
f_1 = 1 & \rightarrow f_2 \neq 2, \quad f_2 = 0 \rightarrow f_1 \neq 0, \\
f_1 = 2 & \rightarrow f_2 \neq 1, \quad f_2 = 2 \rightarrow f_1 \neq 1, \\
f_1 = 2 & \rightarrow f_2 \neq 2, \quad f_2 = 2 \rightarrow f_1 \neq 1.
\end{align*}
\]

Based on these rules we can remove all 2-tuples from $V_I(f_1) \times V_I(f_2)$ that do not belong to $I$. For example the 2-tuple $(2, 2)$ can be removed by the rule $f_1 = 2 \rightarrow f_2 \neq 2$, and the 2-tuple $(0, 0)$ can be removed by the rule $f_1 = 0 \rightarrow f_2 \neq 0$.

So we have that the information derived from $I$ by inhibitory association rules is in some sense complete. This fact was proven for arbitrary information systems in [34].
Chapter 3

Preliminary Results for Decision and Inhibitory Rules and Rule Systems

In the book [21], some relatively simple results were considered for binary decision tables with many-valued decisions: relationships among decision trees, rules and tests, bounds on their complexity, greedy algorithms for construction of decision trees, rules and tests, and dynamic programming algorithms for minimization of tree depth and rule length.

In this chapter, we mention results from [21] related to decision rules and rule systems without proofs and extend them to inhibitory rules and rule systems over binary decision tables with many-valued decisions. Some part of this extension connected with relationships among inhibitory trees, rules and tests, and bounds on their complexity was done jointly with Mohammad Azad.

3.1 Main Notions

Now we consider formal definitions of notions corresponding to binary decision tables with many-valued decisions including decision and inhibitory trees, tests, rules, and
rule systems.

3.1.1 Binary Decision Tables with Many-valued Decisions

A binary decision table with many-valued decisions is a rectangular table $T$ filled by numbers from the set $\{0, 1\}$. Columns of this table are labeled with attributes $f_1, \ldots, f_n$. Rows of the table are pairwise different, and each row $r$ is labeled with a nonempty finite set $D(r) = D_T(r)$ of numbers from $\omega = \{0, 1, 2, \ldots\}$ (set of decisions).

Let $\text{Row}(T)$ be the set of rows of $T$ and $D(T) = \bigcup_{r \in \text{Row}(T)} D(r)$. We denote by $N(T)$ the number of rows in the table $T$. When we consider inhibitory trees, rules and tests, we assume that $D(r) \neq D(T)$ for any $r \in \text{Row}(T)$.

A table obtained by removal of some rows of $T$ is called a subtable of $T$. Let $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $\delta_1, \ldots, \delta_m \in \{0, 1\}$. By $T(f_{i_1}, \delta_m) \ldots T(f_{i_m}, \delta_m)$ we denote a subtable of the table $T$ which contains only rows that at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$ have numbers $\delta_1, \ldots, \delta_m$, respectively. We denote by $\text{SEP}(T)$ the set of all nonempty subtables of the table $T$ including the table $T$ which can be represented in the form $T(f_{i_1}, \delta_m) \ldots T(f_{i_m}, \delta_m)$. We will call such tables separable subtables of the table $T$.

We will consider two interpretations of the decision table $T$: decision and inhibitory. In the case of decision interpretation, for a given row $r$ of $T$, we should find a decision from the set $D(r)$. In the case of inhibitory interpretation, for a given row $r$ of $T$, we should find a decision from the set $D(T) \setminus D(r)$.

3.1.2 Decision Trees, Rule Systems and Tests

We begin from the consideration of decision interpretation.

A decision tree over $T$ is a finite tree with root in which each terminal node is labeled with a decision (a number from $\omega$), each nonterminal node (such nodes will be called working) is labeled with an attribute from the set $\{f_1, \ldots, f_n\}$. Two edges
start in each working node. These edges are labeled with 0 and 1, respectively.

Let $\Gamma$ be a decision tree over $T$. For a given row $r$ of $T$ this tree works in
the following way. We begin the work in the root of $\Gamma$. If the considered node is terminal
then the result of $\Gamma$ work is the number attached to this node. Let the considered
node be working node which is labeled with an attribute $f_i$. If the value of $f_i$ in the
considered row is 0 then we pass along the edge which is labeled with 0. Otherwise,
we pass along the edge which is labeled with 1, etc.

We will say that a decision tree $\Gamma$ over the decision table $T$ is a decision tree for
$T$ if for any row $r$ of $T$ the work of $\Gamma$ finishes in a terminal node which is labeled with
a number from the set $D(r)$ attached to the row $r$.

We denote by $h(\Gamma)$ the depth of $\Gamma$ which is the maximum length of a path from
the root to a terminal node. We denote by $h(T)$ the minimum depth of a decision
tree for the table $T$.

A decision rule over $T$ is an expression of the kind

$$f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \to t$$

where $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $b_1, \ldots, b_m \in \{0, 1\}$, and $t \in \omega$. We denote this rule
$\rho$. The number $m$ is called the length of the rule $\rho$ and is denoted $l(\rho)$. The decision
t $t$ is called the right-hand side of the rule $\rho$. This rule is called realizable for a row
$r = (\delta_1, \ldots, \delta_n)$ if

$$\delta_{i_1} = b_1, \ldots, \delta_{i_m} = b_m.$$ 

The rule $\rho$ is called true for $T$ if, for any row $r$ of $T$ such that the rule $\rho$ is realizable
for row $r$, $t \in D(r)$. We denote by $l(T, r)$ the minimum length of a decision rule over
$T$ which is true for $T$ and realizable for $r$. We will say that the considered rule is a
rule for $T$ and $r$ if this rule is true for $T$ and realizable for $r$.

A nonempty finite set $S$ of decision rules over $T$ is called a complete decision rule
system for $T$ if each rule from $S$ is true for $T$ and, for every row of $T$, there exists a rule from $S$ which is realizable for this row. We denote by $l(S)$ the maximum length of a rule from $S$ (we will call $l(S)$ the length of $S$), and by $l(T)$ we denote the minimum value of $l(S)$ among all complete decision rule systems $S$ for $T$.

We will say that $T$ is a degenerate table if either $T$ has no rows, or the intersection of sets of decisions attached to rows of $T$ is nonempty (in this case, we will say that $T$ has a common decision).

A decision test for the table $T$ is a subset of columns $\{f_{i_1}, \ldots, f_{i_m}\}$ such that, for any numbers $\delta_1, \ldots, \delta_m \in \{0, 1\}$, the subtable $T(f_{i_1}, \delta_1) \ldots T(f_{i_m}, \delta_m)$ is a degenerate table. Empty set is a decision test for $T$ if and only if $T$ is a degenerate table.

A decision reduct for the table $T$ is a decision test for $T$ for which each proper subset is not a test. It is clear that each decision test has a decision reduct as a subset. We denote by $R(T)$ the minimum cardinality of a decision reduct for $T$.

### 3.1.3 Inhibitory Trees, Rule Systems and Tests

We consider now the inhibitory interpretation of the decision table $T$.

An inhibitory tree over $T$ is a finite tree with root in which each terminal node is labeled with an expression $\neq t$ where $t$ is a decision (a number from $\omega$), each nonterminal node (such nodes will be called working) is labeled with an attribute from the set $\{f_1, \ldots, f_n\}$. Two edges start in each working node. These edges are labeled with 0 and 1 respectively.

Let $\Gamma$ be an inhibitory tree over $T$. For a given row $r$ of $T$ this tree works in the following way. We begin the work in the root of $\Gamma$. If the considered node is terminal then the result of $\Gamma$ work is the expression attached to this node. Let the considered node be working node which is labeled with an attribute $f_i$. If the value of $f_i$ in the considered row is 0 then we pass along the edge which is labeled with 0. Otherwise, we pass along the edge which is labeled with 1, etc.
We will say that an inhibitory tree $\Gamma$ over the decision table $T$ is an inhibitory tree for $T$ if, for any row $r$ of $T$, the work of $\Gamma$ finishes in a terminal node which is labeled with an expression $\neq t$ where $t \in D(T) \setminus D(r)$.

We denote by $h(\Gamma)$ the depth of $\Gamma$ which is the maximum length of a path from the root to a terminal node. We denote by $ih(T)$ the minimum depth of an inhibitory tree for the table $T$.

An inhibitory rule over $T$ is an expression of the kind

$$f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \rightarrow \neq t$$

where $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, $b_1, \ldots, b_m \in \{0, 1\}$, and $t \in \omega$. We denote this rule $\rho$. The number $m$ is called the length of the rule $\rho$ and is denoted by $l(\rho)$. The expression $\neq t$ is called the right-hand side of the rule $\rho$. This rule is called realizable for a row $r = (\delta_1, \ldots, \delta_n)$ if

$$\delta_{i_1} = b_1, \ldots, \delta_{i_m} = b_m.$$

The rule $\rho$ is called true for $T$ if, for any row $r$ of $T$ such that the rule $\rho$ is realizable for row $r$, $t \in D(T) \setminus D(r)$. We denote by $il(T,r)$ the minimum length of an inhibitory rule over $T$ which is true for $T$ and realizable for $r$. We will say that the considered rule is a rule for $T$ and $r$ if this rule is true for $T$ and realizable for $r$.

A nonempty finite set $S$ of inhibitory rules over $T$ is called a complete inhibitory rule system for $T$ if each rule from $S$ is true for $T$ and, for every row of $T$, there exists a rule from $S$ which is realizable for this row. We denote by $l(S)$ the maximum length of a rule from $S$ (we will call $l(S)$ the length of $S$), and by $il(T)$ we denote the minimum value of $l(S)$ among all complete inhibitory rule systems $S$ for $T$.

Let $\Theta$ be a subtable of $T$. We will say that $\Theta$ is incomplete relative to $T$ if $D(\Theta) \subset D(T)$.
An inhibitory test for the table $T$ is a subset of columns $\{f_{i_1}, \ldots, f_{i_m}\}$ such that, for any numbers $\delta_1, \ldots, \delta_m \in \{0, 1\}$, the subtable $T(f_{i_1}, \delta_m) \ldots T(f_{i_m}, \delta_m)$ is incomplete relative to $T$.

An inhibitory reduct for the table $T$ is an inhibitory test for $T$ for which each proper subset is not an inhibitory test. It is clear that each inhibitory test has an inhibitory reduct as a subset. We denote by $iR(T)$ the minimum cardinality of an inhibitory reduct for $T$.

### 3.1.4 Complementary Decision Table

Let $T$ be a nondegenerate binary decision table with many-valued decisions. We denote by $T^C$ complementary to $T$ decision table obtained from the table $T$ by changing, for each row $r \in \text{Row}(T)$, the set $D(r)$ with the set $D(T) \setminus D(r)$, i.e., $D_{T^C}(r) = D(T) \setminus D_T(r)$ for any $r \in \text{Row}(T)$. It is clear that $\text{Row}(T) = \text{Row}(T^C)$, $D(T) = D(T^C)$, and $D_T(r) = D(T) \setminus D_{T^C}(r)$ for any $r \in \text{Row}(T)$. In particular, we have $T^{CC} = T$.

**Example 1** Let us consider the decision table $T_3$ depicted in Figure 2.7. We denoted by $T_3'$ the decision table obtained from $T_3$ by removal of the first row labeled with the whole set of decisions $\{x, y, z\}$. The table $T_3'^C$ complementary to $T_3'$ is depicted in Figure 3.1.

$$T_3'^C = \begin{array}{cccccccccc}
\text{ } & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\
\text{x} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \{x\} \\
y & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \{y\} \\
z & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \{z\} \\
xy & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \{x, y\} \\
xz & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \{x, z\} \\
yz & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \{y, z\} \\
\end{array}$$

**Figure 3.1:** Decision table $T_3'^C$ complementary to the decision table $T_3'$.
Let $\Gamma_1$ be a decision tree over $T^C$ and $\Gamma_2$ be an inhibitory tree over $T$. We denote by $\Gamma_1^-$ an inhibitory tree over $T$ obtained from $\Gamma_1$ by changing expressions attached to terminal nodes: if a terminal node in $\Gamma_1$ is labeled with $t$ then the corresponding node in $\Gamma_1^-$ is labeled with $\neq t$. We denote by $\Gamma_2^+$ a decision tree over $T^C$ obtained from $\Gamma_2$ by changing expressions attached to terminal nodes: if a terminal node in $\Gamma_2$ is labeled with $\neq t$ then the corresponding node in $\Gamma_2^+$ is labeled with $t$.

Let $\rho_1$ be a decision rule over $T^C$ and $\rho_2$ be an inhibitory rule over $T$. We denote by $\rho_1^-$ an inhibitory rule over $T$ obtained from $\rho_1$ by changing the right-hand side of $\rho_1$: if the right-hand side of $\rho_1$ is $t$ then the right-hand side of $\rho_1^-$ is $\neq t$. We denote by $\rho_2^+$ a decision rule over $T^C$ obtained from $\rho_2$ by changing the right-hand side of $\rho_2$: if the right-hand side of $\rho_2$ is $\neq t$ then the right-hand side of $\rho_2^+$ is $t$.

Let $S_1$ be a nonempty finite set of decision rules over $T^C$ and $S_2$ be a nonempty finite system of inhibitory rules over $T$. We denote $S_1^- = \{ \rho^- : \rho \in S_1 \}$ and $S_2^+ = \{ \rho^+ : \rho \in S_2 \}$.

It is not difficult to prove the following statement.

**Proposition 2** Let $T$ be a nondegenerate binary decision table with many-valued decisions containing $n$ columns labeled with attributes $f_1, \ldots, f_n$, and $T^C$ be complementary to $T$ decision table. Then

1. A decision tree $\Gamma$ over $T^C$ is a decision tree for $T^C$ if and only if $\Gamma^-$ is an inhibitory tree for $T$;

2. A decision rule $\rho$ over $T^C$ is true for $T$ if and only if the inhibitory rule $\rho^-$ is true for $T$;

3. A nonempty finite set $S$ of decision rules over $T^C$ is a complete system of decision rules for $T^C$ if and only if $S^-$ is a complete system of inhibitory rules for $T$;
4. A subset \( \{f_{i_1}, \ldots, f_{i_m}\} \) of the set \( \{f_1, \ldots, f_n\} \) is a decision test for \( T^C \) if and only if the subset \( \{f_{i_1}, \ldots, f_{i_m}\} \) is an inhibitory test for \( T \).

**Proof.** 1. Let \( \Gamma \) be a decision tree over \( T^C \). Let us assume that \( \Gamma \) is a decision tree for \( T^C \). Then, for any row \( r \) of \( T^C \), the work of \( \Gamma \) finishes in a terminal node which is labeled with a number \( t \in D_{T^C}(r) \). From here it follows that, for any row \( r \) of \( T \), the work of \( \Gamma^- \) finishes in a terminal node which is labeled with an expression \( \neq t \) where \( t \in D_{T^C}(r) = D(T) \setminus D_T(r) \). Therefore \( \Gamma^- \) is an inhibitory tree for \( T \).

Let us assume now that \( \Gamma^- \) is an inhibitory tree for \( T \). Then, for any row \( r \) of \( T \), the work of \( \Gamma^- \) finishes in a terminal node which is labeled with an expression \( \neq t \) where \( t \in D(T) \setminus D_T(r) \). From here it follows that, for any row \( r \) of \( T^C \), the work of \( \Gamma \) finishes in a terminal node which is labeled with a number \( t \) where \( t \in D(T) \setminus D_T(r) = D_{T^C}(r) \). Therefore \( \Gamma \) is a decision tree for \( T^C \).

2. Let \( \rho \) be a decision rule over \( T^C \) and \( \rho \) be equal to \( f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \to t \). Then \( \rho^- \) is equal to \( f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \to \neq t \). Denote \( T^C_\ast = T^C(f_{i_1}, b_1) \ldots (f_{i_m}, b_m) \) and \( T_\ast = T(f_{i_1}, b_1) \ldots (f_{i_m}, b_m) \). One can show that \( \rho \) is true for \( T^C \) if and only if \( t \in D_{T^C}(r) \) for any \( r \in Row(T^C_\ast) \), and \( \rho^- \) is true for \( T \) if and only if \( t \in D(T) \setminus D_T(r) \) for any \( r \in Row(T_\ast) \). We know that \( Row(T^C_\ast) = Row(T_\ast) \) and \( D_{T^C}(r) = D(T) \setminus D_T(r) \) for any \( r \in Row(T_\ast) \). From here it follows that \( \rho \) is true for \( T \) if and only if \( \rho^- \) is true for \( T \).

3. Let \( S \) be a nonempty finite set of decision rules over \( T^C \). From statement 2 it follows that each rule \( \rho \) from \( S \) is true for \( T^C \) if and only if each rule \( \rho^- \) from \( S^- \) is true for \( T \). It is clear that a rule \( \rho \) from \( S \) is realizable for a row \( r \) from \( Row(T^C) = Row(T) \) if and only if \( \rho^- \) is realizable for \( r \). From here it follows that \( S \) is a complete system of decision rules for \( T^C \) if and only if \( S^- \) is a complete system of inhibitory rules for \( T \).

4. Let \( \{f_{i_1}, \ldots, f_{i_m}\} \) be a subset of the set \( \{f_1, \ldots, f_n\} \) and \( \delta_1, \ldots, \delta_m \in \{0, 1\} \). One can show that \( T^C(f_{i_1}, \delta_m) \ldots T(f_{i_m}, \delta_m) \) is degenerate iff \( T(f_{i_1}, \delta_m) \ldots T(f_{i_m}, \delta_m) \)
is incomplete relative to $T$. Therefore the subset \( \{ f_{i_1}, \ldots, f_{i_m} \} \) is a decision test for $T^C$ if and only the subset \( \{ f_{i_1}, \ldots, f_{i_m} \} \) is an inhibitory test for $T$. ■

**Corollary 3** Let $T$ be a nondegenerate binary decision table with many-valued decisions, $T^C$ be complementary to $T$ decision table, and $r$ be a row of $T$. Then $ih(T) = h(T^C)$, $il(T, r) = l(T^C, r)$, $il(T) = l(T^C)$, and $iR(T) = R(T^C)$.

**Proof.** 1. Let $\Gamma_1$ be a decision tree for $T^C$ such that $h(\Gamma_1) = h(T^C)$. Then, by Proposition 2, $\Gamma_1^-$ is an inhibitory tree for $T$. It is clear that $h(\Gamma_1) = h(\Gamma_1^-)$. Therefore $ih(T) \leq h(T^C)$. Let $\Gamma_2$ be an inhibitory tree for $T$ such that $h(\Gamma_2) = ih(T)$. We now consider a decision tree $\Gamma_2^+$ over $T^C$. It is clear that $(\Gamma_2^+)^- = \Gamma_2$. By Proposition 2, $\Gamma_2^+$ is a decision tree for $T^C$. It is clear that $h(\Gamma_2) = h(\Gamma_2^+)$. Therefore $h(T^C) \leq ih(T)$ and $h(T^C) = ih(T)$.

2. Let $\rho_1$ be a decision rule for $T^C$ and $r$ such that $l(\rho_1) = l(T^C, r)$. It is clear that $\rho_1$ is realizable for $r$ if and only if the rule $\rho_1^-$ is realizable for $r$. Then, by Proposition 2, $\rho_1^-$ is an inhibitory rule for $T$ and $r$. It is clear that $l(\rho_1) = l(\rho_1^-)$. Therefore $il(T, r) \leq l(T^C, r)$. Let $\rho_2$ be an inhibitory rule for $T$ and $r$ such that $l(\rho_2) = il(T, r)$. We now consider a decision rule $\rho_2^+$ over $T^C$. It is clear that $(\rho_2^+)^- = \rho_2$, and $\rho_2$ is realizable for $r$ if and only if the rule $\rho_2^+$ is realizable for $r$. By Proposition 2, $\rho_2^+$ is a decision rule for $T^C$. It is clear that $l(\rho_2) = l(\rho_2^+)$. Therefore $l(T^C, r) \leq il(T, r)$ and $l(T^C, r) = il(T, r)$.

3. Let $S_1$ be a complete system of decision rules for $T^C$ such that $l(S_1) = l(T^C)$. Then, by Proposition 2, $S_1^-$ is a complete system of inhibitory rules for $T$. It is clear that $l(S_1) = l(S_1^-)$. Therefore $il(T) \leq l(T^C)$. Let $S_2$ be a complete system of inhibitory rules for $T$ such that $l(S_2) = il(T)$. We now consider a set of decision rules $S_2^+$ over $T^C$. It is clear that $(S_2^+)^- = S_2$. By Proposition 2, $S_2^+$ is a complete system of decision rules for $T^C$. It is clear that $l(S_2) = l(S_2^+)$. Therefore $l(T^C) \leq il(T)$ and $l(T^C) = il(T)$.

4. The equality $iR(T) = R(T^C)$ follows directly from Proposition 2. ■
3.2 Relationships among Trees, Rule Systems and Tests

We consider now relationships among decision (inhibitory) trees, rule systems and tests.

3.2.1 Relationships among Decision Trees, Rule Systems and Tests

Theorem 4 [21] Let $T$ be a binary decision table with many-valued decisions.

1. If $\Gamma$ is a decision tree for $T$ then the set of attributes attached to working nodes of $\Gamma$ is a decision test for the table $T$.

2. Let $\{f_{i_1}, \ldots, f_{i_m}\}$ be a decision test for $T$. Then there exists a decision tree $\Gamma$ for $T$ which uses only attributes from $\{f_{i_1}, \ldots, f_{i_m}\}$ and for which $h(\Gamma) = m$.

Corollary 5 [21] Let $T$ be a binary decision table with many-valued decisions. Then

$$h(T) \leq R(T).$$

Theorem 6 [21] Let $T$ be a binary decision table with many-valued decisions containing $n$ columns labeled with attributes $f_1, \ldots, f_n$.

1. If $S$ is a complete system of decision rules for $T$ then the set of attributes from rules in $S$ is a decision test for $T$.

2. If $F = \{f_{i_1}, \ldots, f_{i_m}\}$ is a decision test for $T$ then there exists a complete system $S$ of decision rules for $T$ which uses only attributes from $F$ and for which $l(S) = m$.

Corollary 7 [21] $l(T) \leq R(T)$. 
Let $\Gamma$ be a decision tree for $T$ and $\tau$ be a path in $\Gamma$ from the root to a terminal node in which working nodes are labeled with attributes $f_{i_1}, \ldots, f_{i_m}$, edges are labeled with numbers $b_1, \ldots, b_m$, and the terminal node of $\tau$ is labeled with the decision $t$. We correspond to $\tau$ the decision rule

$$f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \rightarrow t.$$ 

**Theorem 8** [21] Let $T$ be a binary decision table with many-valued decisions, $\Gamma$ be a decision tree for $T$, and $S$ be the set of decision rules corresponding to paths in $\Gamma$ from the root to terminal nodes. Then $S$ is a complete system of decision rules for $T$, and $l(S) = h(\Gamma)$.

**Corollary 9** $l(T) \leq h(T)$.

### 3.2.2 Relationships among Inhibitory Trees, Rules and Tests

**Theorem 10** Let $T$ be a nondegenerate binary decision table with many-valued decisions.

1. If $\Gamma$ is an inhibitory tree for $T$ then the set of attributes attached to working nodes of $\Gamma$ is an inhibitory test for the table $T$.

2. Let $\{f_{i_1}, \ldots, f_{i_m}\}$ be an inhibitory test for $T$. Then there exists an inhibitory tree for $T$ which uses only attributes from $\{f_{i_1}, \ldots, f_{i_m}\}$ and which depth is equal to $m$.

**Proof.** 1. Let $\Gamma$ be an inhibitory tree for $T$ and $F(\Gamma)$ be the set of attributes attached to working nodes of $\Gamma$. We now consider a decision tree $\Gamma^+$ over $T^C$. It is clear that $F(\Gamma)$ is the set of attributes attached to working nodes of $\Gamma^+$. By Proposition 2, $\Gamma^+$ is a decision tree for $T^C$. Using Theorem 4 we obtain that $F(\Gamma)$ is a decision test for $T^C$. By Proposition 2, $F(\Gamma)$ is an inhibitory test for $T$. 
2. Let \( \{f_{i_1}, \ldots, f_{i_m}\} \) be an inhibitory test for \( T \). By Proposition 2, \( \{f_{i_1}, \ldots, f_{i_m}\} \) is a decision test for \( T^C \). Then, by Theorem 4, there exists a decision tree \( \Gamma \) for \( T^C \) which uses only attributes from \( \{f_{i_1}, \ldots, f_{i_m}\} \) and for which \( h(\Gamma) = m \). By Proposition 2, \( \Gamma^- \) is an inhibitory tree for \( T \). It is clear that \( \Gamma^- \) uses only attributes from \( \{f_{i_1}, \ldots, f_{i_m}\} \), and \( h(\Gamma^-) = m \). ■

**Corollary 11** Let \( T \) be a nondegenerate binary decision table with many-valued decisions. Then

\[ ih(T) \leq iR(T). \]

**Theorem 12** Let \( T \) be a nondegenerate binary decision table with many-valued decisions.

1. If \( S \) is a complete system of inhibitory rules for \( T \) then the set of attributes from rules in \( S \) is an inhibitory test for \( T \).

2. If \( F \) is an inhibitory test for \( T \) then there exists a complete system of inhibitory rules for \( T \) which uses only attributes from \( F \) and for which the length is equal to \( m \).

**Proof.** 1. Let \( S \) be a complete system of inhibitory rules for \( T \) and \( F(S) \) be the set of attributes from rules in \( S \). We now consider a set of decision rules \( S^+ \) over \( T^C \). It is clear that \( F(S) \) is the set of attributes from rules in \( S^+ \). By Proposition 2, \( S^+ \) is a complete system of decision rules for \( T^C \). Using Theorem 6 we obtain that \( F(S) \) is a decision test for \( T^C \). By Proposition 2, \( F(S) \) is an inhibitory test for \( T \).

2. Let \( F \) be an inhibitory test for \( T \). By Proposition 2, \( F \) is a decision test for \( T^C \). Then, by Theorem 6, there exists a complete system \( G \) of decision rules for \( T \) which uses only attributes from \( F \) and for which \( l(G) = m \). By Proposition 2, \( G^- \) is a complete system of inhibitory rules for \( T \). It is clear that \( G^- \) uses only attributes from \( F \), and \( l(G^-) = m \). ■
Corollary 13 \( il(T) \leq iR(T) \).

Let \( \Gamma \) be an inhibitory tree for \( T \) and \( \tau \) be a path in \( \Gamma \) from the root to a terminal node in which working nodes are labeled with attributes \( f_{i_1}, \ldots, f_{i_m} \), edges are labeled with numbers \( b_1, \ldots, b_m \), and the terminal node of \( \tau \) is labeled with the expression \( \neq t \). We correspond to \( \tau \) the inhibitory rule

\[
   f_{i_1} = b_1 \land \ldots \land f_{i_m} = b_m \rightarrow \neq t.
\]

Theorem 14 Let \( T \) be a nondegenerate binary decision table with many-valued decisions, \( \Gamma \) be an inhibitory tree for \( T \), and \( S \) be the set of inhibitory rules corresponding to paths in \( \Gamma \) from the root to terminal nodes. Then \( S \) is a complete system of inhibitory rules for \( T \), and \( l(S) = h(\Gamma) \).

Proof. Let \( \Gamma \) be an inhibitory tree for \( T \), and \( S \) be the set of inhibitory rules corresponding to paths in \( \Gamma \) from the root to terminal nodes. Let us consider a decision tree \( \Gamma^+ \) over \( T^C \). By Proposition 2, \( \Gamma^+ \) is a decision tree for \( T^C \). It is clear that \( S^+ \) is the set of decision rules corresponding to paths in \( \Gamma^+ \) from the root to terminal nodes. Using Theorem 8 we obtain that \( S^+ \) is a complete system of decision rules for \( T^C \). By Proposition 2, \( (S^+)^- = S \) is a complete system of inhibitory rules for \( T \). \( \blacksquare \)

Corollary 15 \( il(T) \leq ih(T) \).

3.3 Bounds on Complexity of Trees, Rules, Rule Systems and Tests

We consider some bounds on the length of decision (inhibitory) rules and rule systems, and some upper bounds on the complexity of decision (inhibitory) trees and tests which are also upper bounds on the length of decision (inhibitory) rule systems.
3.3.1 Bounds for Decision Trees, Rules and Tests

From Corollaries 5 and 9 it follows that \( l(T) \leq h(T) \leq R(T) \). So each upper bound on \( h(T) \) or on \( R(T) \) is also an upper bound on \( l(T) \).

Let \( T \) be a binary decision table with many-valued decisions which has \( n \) columns labeled with attributes \( f_1, \ldots, f_n \), and \( \bar{\delta} = (\delta_1, \ldots, \delta_n) \in \{0,1\}^n \). We define the parameters \( M(T, \bar{\delta}) \) and \( M(T) \). If \( T \) is a degenerate table then \( M(T, \bar{\delta}) = 0 \) and \( M(T) = 0 \). Let now \( T \) be a nondegenerate table. Then \( M(T, \bar{\delta}) \) is the minimum natural \( m \) such that there exist attributes \( f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\} \) for which \( T(f_{i_1}, \delta_{i_1}) \cdot \ldots \cdot (f_{i_m}, \delta_{i_m}) \) is a degenerate table. We denote \( M(T) = \max \{M(T, \bar{\delta}) : \bar{\delta} \in \{0,1\}^n\} \).

**Theorem 16** [21] Let \( T \) be a binary decision table with many-valued decisions and \( \text{Row}(T) \) be the set of rows of \( T \). Then \( l(T, \bar{\delta}) = M(T, \bar{\delta}) \) for any row \( \bar{\delta} \in \text{Row}(T) \), and \( l(T) = \max \{M(T, \bar{\delta}) : \bar{\delta} \in \text{Row}(T)\} \).

**Theorem 17** [21] Let \( T \) be a nonempty binary decision table with many-valued decisions. Then

\[
R(T) \leq N(T) - 1.
\]

**Theorem 18** [21] Let \( T \) be a nonempty binary decision table with many-valued decisions. Then

\[
h(T) \leq M(T) \log_2 N(T).
\]

3.3.2 Bounds for Inhibitory Trees, Rules and Tests

From Corollaries 11 and 15 it follows that \( il(T) \leq ih(T) \leq iR(T) \). So each upper bound on \( ih(T) \) or on \( iR(T) \) is also an upper bound on \( il(T) \).

Let \( T \) be a nondegenerate binary decision table with many-valued decisions which has \( n \) columns labeled with attributes \( f_1, \ldots, f_n \) and \( \bar{\delta} = (\delta_1, \ldots, \delta_n) \in \{0,1\}^n \). We define now the parameters \( iM(T, \bar{\delta}) \) and \( iM(T) \). Denote \( iM(T, \bar{\delta}) \) the minimum natural \( m \) such that there exist attributes \( f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\} \) for which
subtable $T(f_{i_1}, \delta_{i_1}) \ldots (f_{i_m}, \delta_{i_m})$ is incomplete relative to $T$. We denote $iM(T) = \max \{ iM(T, \bar{\delta}) : \bar{\delta} \in \{0, 1\}^n \}$.

**Theorem 19** Let $T$ be a nondegenerate binary decision table with many-valued decisions and Row($T$) be the set of rows of $T$. Then $il(T, \bar{\delta}) = iM(T, \bar{\delta})$ for any row $\bar{\delta} \in \text{Row}(T)$, and $il(T) = \max \{ iM(T, \bar{\delta}) : \bar{\delta} \in \text{Row}(T) \}$.

**Proof.** Let $T$ have $n$ columns labeled with attributes $f_1, \ldots, f_n$, $\bar{\delta} = (\delta_1, \ldots, \delta_n) \in \text{Row}(T)$, and $T^C$ be complementary to $T$ decision table. It is clear that Row($T$) = Row($T^C$). One can show that, for any $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, the subtable $T(f_{i_1}, \delta_{i_1}) \ldots (f_{i_m}, \delta_{i_m})$ is incomplete relative to $T$ if and only if the subtable $T^C(f_{i_1}, \delta_{i_1}) \ldots (f_{i_m}, \delta_{i_m})$ is degenerate. From here it follows $iM(T, \bar{\delta}) = M(T^C, \bar{\delta})$. By Corollary 3, $il(T, \bar{\delta}) = l(T^C, \bar{\delta})$ and $il(T) = l(T^C)$. By Theorem 16, $l(T^C, \bar{\delta}) = M(T^C, \bar{\delta})$ for any row $\bar{\delta} \in \text{Row}(T^C)$. Therefore $il(T, \bar{\delta}) = iM(T, \bar{\delta})$ for any row $\bar{\delta} \in \text{Row}(T)$. By Theorem 16, $l(T^C) = \max \{ M(T^C, \bar{\delta}) : \bar{\delta} \in \text{Row}(T^C) \}$. Therefore $il(T) = \max \{ iM(T, \bar{\delta}) : \bar{\delta} \in \text{Row}(T) \}$. □

**Theorem 20** Let $T$ be a nondegenerate binary decision table with many-valued decisions. Then

$$iR(T) \leq N(T) - 1.$$  

**Proof.** Let $T^C$ be complementary to $T$ decision table. From Corollary 3 it follows that $iR(T) = R(T^C)$. It is clear that $N(T) = N(T^C)$. By Theorem 17, $R(T^C) \leq N(T^C) - 1$. Therefore $iR(T) \leq N(T) - 1$. □

**Theorem 21** Let $T$ be a nondegenerate binary decision table with many-valued decisions. Then

$$ih(T) \leq iM(T) \log_2 N(T).$$

**Proof.** Let $T$ have $n$ columns labeled with attributes $f_1, \ldots, f_n$, and $T^C$ be complementary to $T$ decision table. From Corollary 3 it follows that $ih(T) = h(T^C)$. It is
clear that \( N(T) = N(T^C) \). One can show that, for any \( f_{i_1}, \ldots, f_{i_m} \in \{ f_1, \ldots, f_n \} \) and \( \delta_1, \ldots, \delta_m \in \{0,1\}^n \), the subtable \( T(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m) \) is incomplete relative to \( T \) if and only if the subtable \( T^C(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m) \) is degenerate. From here it follows that \( i M(T) = M(T^C) \). By Theorem 18, \( h(T^C) \leq M(T^C) \log_2 N(T^C) \) Therefore \( ih(T) \leq i M(T) \log_2 N(T) \).

### 3.4 Approximate Algorithms for Optimization of Decision and Inhibitory Rules and Rule Systems

We consider approximate polynomial algorithms for problem of minimization of decision (inhibitory) rule length, and problem of minimization of decision (inhibitory) rule system length. First, we consider well known greedy algorithm for set cover problem which will be used later.

#### 3.4.1 Greedy Algorithm for Set Cover Problem

Let \( A \) be a set containing \( N > 0 \) elements, and \( F = \{ S_1, \ldots, S_p \} \) be a family of subsets of the set \( A \) such that \( A = \bigcup_{i=1}^p S_i \). A subfamily \( \{ S_{i_1}, \ldots, S_{i_t} \} \) of the family \( F \) is called a cover if \( \bigcup_{j=1}^t S_{i_j} = A \). Set cover problem: it is required to find a cover with minimum cardinality \( t \).

We now consider well-known greedy algorithm for set cover problem. During each step this algorithm chooses a set from the family \( F \) which covers the maximum number of previously uncovered elements from \( A \).

We denote by \( C_{\text{greedy}} \) the cardinality of the cover constructed by greedy algorithm and by \( C_{\text{min}} \) – the minimum cardinality of a cover. The proof of the following theorem which was discovered by different authors [35, 36] can be found in [21].
Theorem 22 \( C_{\text{greedy}} \leq C_{\text{min}} \ln N + 1. \)

3.4.2 Optimization of Decision Rules

We can apply greedy algorithm for set cover problem to construct decision rules for decision tables with many-valued decisions.

Let \( T \) be a nondegenerate binary decision table with many-valued decisions containing \( n \) columns labeled with attributes \( f_1, \ldots, f_n \). Let \( r = (b_1, \ldots, b_n) \) be a row of \( T \), \( D(r) \) be the set of decisions attached to \( r \), and \( d \in D(r) \).

We consider a set cover problem \( A(T, r, d) \), \( F(T, r, d) = \{S_1, \ldots, S_n\} \), where \( A(T, r, d) \) is the set of all rows \( r' \) of \( T \) such that \( d \notin D(r') \). For \( i = 1, \ldots, n \), the set \( S_i \) coincides with the set of all rows from \( A(T, r, d) \) which are different from \( r \) in the column \( f_i \). One can show that the decision rule

\[
  f_{i_1} = b_{i_1} \land \ldots \land f_{i_m} = b_{i_m} \rightarrow d
\]

is true for \( T \) (it is clear that this rule is realizable for \( r \)) if and only if the subfamily \( \{S_{i_1}, \ldots, S_{i_m}\} \) is a cover for the set cover problem \( A(T, r, d) \), \( F(T, r, d) \).

We denote \( P(T, r, d) = |A(T, r, d)| \) and \( l(T, r, d) \) the minimum length of a decision rule over \( T \) which is true for \( T \), realizable for \( r \) and has \( d \) on the right-hand side. It is clear that for the constructed set cover problem \( C_{\text{min}} = l(T, r, d) \) where \( C_{\text{min}} \) is the minimum cardinality of cover.

Let us apply the greedy algorithm to the set cover problem \( A(T, r, d) \), \( F(T, r, d) \). This algorithm constructs a cover which corresponds to a decision rule \( \text{rule}(T, r, d) \) which is true for \( T \), realizable for \( r \) and has \( d \) on the right-hand side. We denote by \( l_{\text{greedy}}(T, r, d) \) the length of \( \text{rule}(T, r, d) \). From Theorem 22 it follows that

\[
  l_{\text{greedy}}(T, r, d) \leq l(T, r, d) \ln P(T, r, d) + 1.
\]
We denote by $l_{\text{greedy}}(T,r)$ the length of the rule constructed by the following polynomial algorithm (we will say about this algorithm as about modified greedy algorithm). For a given binary decision table $T$ with many-valued decisions and row $r$ of $T$, for each $d \in D(r)$ we construct the set cover problem $A(T,r,d)$, $F(T,r,d)$ and then apply to this problem the greedy algorithm. We transform the constructed cover to the rule $\text{rule}(T,r,d)$. Among the rules $\text{rule}(T,r,d)$, $d \in D(r)$, we choose a rule with minimum length. This rule is the output of the considered algorithm. We have

$$l_{\text{greedy}}(T,r) = \min \{l_{\text{greedy}}(T,r,d) : d \in D(r)\}.$$ 

It is clear that

$$l(T,r) = \min \{l(T,r,d) : d \in D(r)\}.$$ 

Let $K(T,r) = \max \{P(T,r,d) : d \in D(r)\}$.

Then

$$l_{\text{greedy}}(T,r) \leq l(T,r) \ln K(T,r) + 1.$$ 

So we have the following statement.

**Theorem 23** [21] Let $T$ be a nondegenerate binary decision table with many-valued decisions and $r$ be a row of $T$. Then

$$l_{\text{greedy}}(T,r) \leq l(T,r) \ln K(T,r) + 1.$$ 

We can use the considered modified greedy algorithm to construct a complete decision rule system for the decision table $T$ with many-valued decisions. To this end, we apply this algorithm sequentially to the table $T$ and to each row $r$ of $T$. As a result, we obtain a system of rules $S$ in which each rule is true for $T$ and, for every row of $T$, there exists a rule from $S$ which is realizable for this row.
We denote $l_{\text{greedy}}(T) = l(S)$ and

$$K(T) = \max\{K(T,r) : r \in \text{Row}(T)\},$$

where $\text{Row}(T)$ is the set of rows of $T$. It is clear that $l(T) = \max\{l(T,r) : r \in \text{Row}(T)\}$. Using Theorem 23 we obtain

**Theorem 24** [21] Let $T$ be a nondegenerate binary decision table with many-valued decisions. Then

$$l_{\text{greedy}}(T) \leq l(T) \ln K(T) + 1.$$

**Proposition 25** [21] The problem of minimization of decision rule length for binary decision tables with many-valued decisions is NP-hard.

**Theorem 26** [21] If $NP \not\subseteq \text{DTIME}(n^{O(\log \log n)})$ then, for any $\varepsilon$, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given nondegenerate binary decision table $T$ with many-valued decisions and row $r$ of $T$ constructs a decision rule which is true for $T$, realizable for $r$, and which length is at most

$$(1 - \varepsilon)l(T,r) \ln K(T,r).$$

**Proposition 27** [21] The problem of minimization of complete decision rule system length for binary decision tables with many-valued decisions is NP-hard.

**Theorem 28** [21] If $NP \not\subseteq \text{DTIME}(n^{O(\log \log n)})$ then, for any $\varepsilon$, $0 < \varepsilon < 1$, there is no polynomial algorithm that, for a given nondegenerate binary decision table $T$ with many-valued decisions, constructs a complete decision rule system $S$ for $T$ such that

$$l(S) \leq (1 - \varepsilon)l(T) \ln K(T).$$
3.4.3 Optimization of Inhibitory Rules

We can apply greedy algorithm for set cover problem to construct inhibitory rules for binary decision tables with many-valued decisions.

Let $T$ be a nondegenerate binary decision table with many-valued decisions containing $n$ columns labeled with attributes $f_1, \ldots, f_n$. Let $r = (b_1, \ldots, b_n)$ be a row of $T$, $D(r)$ be the set of decisions attached to $r$, and $d \in D(T) \setminus D(r)$.

We consider a set cover problem $iA(T, r, d), iF(T, r, d) = \{iS_1, \ldots, iS_n\}$, where $iA(T, r, d)$ is the set of all rows $r'$ of $T$ such that $d \in D(r')$. For $j = 1, \ldots, n$, the set $iS_j$ coincides with the set of all rows from $iA(T, r, d)$ which are different from $r$ in the column $f_j$. One can show that the inhibitory rule

$$f_{j_1} = b_{j_1} \land \ldots \land f_{j_m} = b_{j_m} \rightarrow \neq d$$

is true for $T$ (it is clear that this rule is realizable for $r$) if and only if the subfamily $\{iS_{j_1}, \ldots, iS_{j_m}\}$ is a cover for the set cover problem $iA(T, r, d), iF(T, r, d)$.

We denote $iP(T, r, d) = |iA(T, r, d)|$ and $il(T, r, d)$ the minimum length of an inhibitory rule over $T$ which is true for $T$, realizable for $r$ and has $\neq d$ on the right-hand side. It is clear that, for the constructed set cover problem, $il(T, r, d)$ is the minimum cardinality of cover.

Let us apply the greedy algorithm to the set cover problem $iA(T, r, d), iF(T, r, d)$. This algorithm constructs a cover which corresponds to an inhibitory rule $irule(T, r, d)$ which is true for $T$, realizable for $r$ and has $\neq d$ on the right-hand side. We denote by $il_{\text{greedy}}(T, r, d)$ the length of $irule(T, r, d)$. From Theorem 22 it follows that

$$il_{\text{greedy}}(T, r, d) \leq il(T, r, d) \ln(iP(T, r, d)) + 1.$$

We denote by $il_{\text{greedy}}(T, r)$ the length of the rule constructed by the following
polynomial algorithm (we will say about this algorithm as about modified greedy algorithm). For a given nondegenerate binary decision table $T$ with many-valued decisions and row $r$ of $T$, for each $d \in D(T) \setminus D(r)$ we construct the set cover problem $iA(T, r, d)$, $iF(T, r, d)$ and then apply to this problem the greedy algorithm. We transform the constructed cover to the rule $irule(T, r, d)$. Among the rules $irule(T, r, d)$, $d \in D(T) \setminus D(r)$, we choose a rule with minimum length. This rule is the output of the considered algorithm. We have

$$il_{\text{greedy}}(T, r) = \min \{ il_{\text{greedy}}(T, r, d) : d \in D(T) \setminus D(r) \}.$$  

It is clear that

$$il(T, r) = \min \{ il(T, r, d) : d \in D(T) \setminus D(r) \}.$$  

Let $iK(T, r) = \max \{ iP(T, r, d) : d \in D(T) \setminus D(r) \}$. Then

$$il_{\text{greedy}}(T, r) \leq il(T, r) \ln(iK(T, r)) + 1.$$  

So we have the following statement.

**Theorem 29** Let $T$ be a nondegenerate binary decision table with many-valued decisions and $r$ be a row of $T$. Then

$$il_{\text{greedy}}(T, r) \leq il(T, r) \ln(iK(T, r)) + 1.$$  

We can use the considered modified greedy algorithm to construct a complete inhibitory rule system for the nondegenerate binary decision table $T$ with many-valued decisions. To this end, we apply this algorithm sequentially to the table $T$ and to each row $r$ of $T$. As a result, we obtain a system of rules $S$ in which each inhibitory rule is true for $T$ and, for every row of $T$, there exists a rule from $S$ which is realizable for this row.
We denote $i_{\text{greedy}}(T) = l(S)$ and

$$iK(T) = \max\{iK(T, r) : r \in \text{Row}(T)\},$$

where $\text{Row}(T)$ is the set of rows of $T$. It is clear that $i(T) = \max\{i(T, r) : r \in \text{Row}(T)\}$. Using Theorem 29 we obtain the following statement.

**Theorem 30** Let $T$ be a nondegenerate binary decision table with many-valued decisions. Then

$$i_{\text{greedy}}(T) \leq i(T) \ln(iK(T)) + 1.$$ 

**Proposition 31** The problem of minimization of inhibitory rule length for nondegenerate binary decision tables with many-valued decisions is NP-hard.

**Proof.** Let $T$ be a nondegenerate binary decision table with many-valued decisions and $r$ be a row of $T$. We denote $\Theta = T^C$. It is clear that $\Theta$ is nondegenerate and $\Theta^C = T$. From Proposition 2 it follows that a decision rule $\rho$ over $\Theta^C = T$ is true for $\Theta^C = T$ if and only if the inhibitory rule $\rho^-$ over $\Theta = T^C$ is true for $\Theta = T^C$. It is clear that $\rho$ is realizable for $r$ if and only if $\rho^-$ is realizable for $r$. By Corollary 3, the minimum length of a decision rule for $\Theta^C = T$ and $r$ is equal to the minimum length of an inhibitory rule for $\Theta = T^C$ and $r$. So we have a polynomial time reduction of the problem of minimization of decision rule length for nondegenerate binary decision tables with many-valued decisions to the problem of minimization of inhibitory rule length for nondegenerate binary decision tables with many-valued decisions. By Proposition 25, the problem of minimization of decision rule length for nondegenerate binary decision tables with many-valued decisions is NP-hard. Therefore the problem of minimization of inhibitory rule length for nondegenerate binary decision tables with many-valued decisions is NP-hard. ✷
Theorem 32 If $NP \not\subseteq DTIME(n^{O(\log \log n)})$ then, for any $\varepsilon$, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given nondegenerate binary decision table $T$ with many-valued decisions and row $r$ of $T$ constructs an inhibitory rule for $T$ and $r$ which length is at most $(1 - \varepsilon)il(T, r)\ln(iK(T, r))$.

Proof. Let us assume that for some $\varepsilon$, $0 < \varepsilon < 1$, there exists a polynomial algorithm $A$ which, for a given nondegenerate binary decision table $T$ with many-valued decisions and row $r$ of $T$, constructs an inhibitory rule for $T$ and $r$ which length is at most $(1 - \varepsilon)il(T, r)\ln(iK(T, r))$.

Let $T$ be a nondegenerate binary decision table with many-valued decisions and $r$ be a row of $T$. We denote $\Theta = T^C$. It is clear that $\Theta$ is nondegenerate and $\Theta^C = T$. We apply the algorithm $A$ to the table $\Theta = T^C$ and row $r$. As a result, we obtain an inhibitory rule $\rho$ for $\Theta = T^C$ and $r$ which length is at most $(1 - \varepsilon)il(\Theta, r)\ln(iK(\Theta, r))$. Let us consider a decision rule $\rho^+$ over $\Theta^C = T$. Using Proposition 2 one can show that $\rho^+$ is a decision rule for $\Theta^C = T$ and $r$. The length of $\rho^+$ is at most $(1 - \varepsilon)il(\Theta, r)\ln(iK(\Theta, r))$. By Corollary 3, $il(\Theta, r) = l(\Theta^C, r)$. One can show that $iK(\Theta, r) = K(\Theta^C, r)$. Therefore the length of rule $\rho^+$ is at most $(1 - \varepsilon)l(T, r)\ln K(T, r)$.

As a result, there is a polynomial algorithm which, for a given nondegenerate binary decision table $T$ with many-valued decisions and row $r$ of $T$, constructs a decision rule for $T$ and $r$ which length is at most $(1 - \varepsilon)l(T, r)\ln K(T, r)$ which is impossible if $NP \not\subseteq DTIME(n^{O(\log \log n)})$. Thus, if $NP \not\subseteq DTIME(n^{O(\log \log n)})$ then, for any $\varepsilon$, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given nondegenerate binary decision table $T$ with many-valued decisions and row $r$ of $T$ constructs an inhibitory rule which is true for $T$, realizable for $r$, and which length is at most $(1 - \varepsilon)il(T, r)\ln(iK(T, r))$. ■
Proposition 33 The problem of minimization of complete inhibitory rule system length for nondegenerate binary decision tables with many-valued decisions is \( NP \)-hard.

Proof. Let \( T \) be a nondegenerate binary decision table with many-valued decisions. We denote \( \Theta = T^C \). It is clear that \( \Theta \) is nondegenerate and \( \Theta^C = T \). From Proposition 2 it follows that a finite nonempty set \( S \) of decision rules over \( \Theta^C = T \) is complete for \( \Theta^C = T \) if and only if the set \( S^- \) of inhibitory rules over \( \Theta = T^C \) is complete for \( \Theta = T^C \). By Corollary 3, \( il(\Theta) = l(\Theta^C) \). So we have a polynomial time reduction of the problem of minimization of complete decision rule system length for nondegenerate binary decision tables with many-valued decisions to the problem of minimization of complete inhibitory rule system length for nondegenerate binary decision tables with many-valued decisions. By Proposition 27, the problem of minimization of complete decision rule system length for nondegenerate binary decision tables with many-valued decisions is \( NP \)-hard. Therefore the problem of minimization of complete inhibitory rule system length for nondegenerate binary decision tables with many-valued decisions is \( NP \)-hard. ■

Theorem 34 If \( NP \not\subseteq \text{DTIME}(n^{O(\log \log n)}) \) then, for any \( \varepsilon, 0 < \varepsilon < 1 \), there is no polynomial algorithm that for a given nondegenerate binary decision table \( T \) with many-valued decisions constructs a complete inhibitory rule system \( S \) for \( T \) such that

\[
l(S) \leq (1 - \varepsilon)il(T) \ln(iK(T)).
\]

Proof. Let us assume that for some \( \varepsilon, 0 < \varepsilon < 1 \), there exists a polynomial algorithm \( \mathcal{A} \) which, for a given nondegenerate binary decision table \( T \) with many-valued decisions, constructs a complete inhibitory rule system \( S \) for \( T \) such that \( l(S) \leq (1 - \varepsilon)il(T) \ln(iK(T)) \).

Let \( T \) be a nondegenerate binary decision table with many-valued decisions. We
denote $\Theta = T^C$. It is clear that $\Theta$ is nondegenerate and $\Theta^C = T$. We apply the 
algorithm $A$ to the table $\Theta = T^C$. As a result, we obtain a complete inhibitory rule 
system $S$ for $\Theta$ such that $l(S) \leq (1 - \varepsilon)il(\Theta)\ln(iK(\Theta))$. Let us consider a system $S^+$ of decision rules over $\Theta^C = T$. From Proposition 2 it follows that $S^+$ is a complete decision rule system for $\Theta^C = T$. We have $l(S^+) \leq (1 - \varepsilon)il(\Theta, r)\ln(iK(\Theta, r))$. By Corollary 3, $il(\Theta) = l(\Theta^C)$. One can show that $iK(\Theta) = K(\Theta^C)$. Therefore $l(S^+) \leq (1 - \varepsilon)l(T)\ln K(T)$.

As a result, there is a polynomial algorithm which, for a given nondegenerate binary decision table $T$ with many-valued decisions, constructs a complete decision rule system $S$ for $T$ such that $l(S) \leq (1 - \varepsilon)l(T)\ln K(T)$ which is impossible if $NP \notin DTIME(n^{O(\log \log n)})$. Thus, if $NP \notin DTIME(n^{O(\log \log n)})$ then, for any $\varepsilon$, $0 < \varepsilon < 1$, there is no polynomial algorithm that, for a given nondegenerate binary decision table $T$ with many-valued decisions, constructs a complete inhibitory rule system $S$ for $T$ such that $l(S) \leq (1 - \varepsilon)il(T)\ln(iK(T))$. ■

3.5 Exact Algorithms for Optimization of Decision and Inhibitory Trees, Rules and Tests

A dynamic programming algorithm $V$ for minimization of decision rule length was described in [21].

**Theorem 35** [21] For any nondegenerate binary decision table $T$ with many-valued decisions and any row $r$ of $T$, the algorithm $V$ constructs a decision rule $V(T, r)$ which is true for $T$, realizable for $r$ and has minimum length $l(T, r)$. During the construction of optimal rules for rows of $T$ the algorithm $V$ makes exactly $2|\text{SEP}(T)| + 3$ steps. The time of the algorithm $V$ work is bounded from below by $|\text{SEP}(T)|$, and bounded from above by a polynomial on $|\text{SEP}(T)|$ and on the number of columns in the table $T$. 
Similar algorithm for minimization of decision tree depth was considered in [21].

The situation with decision test minimization is another.

**Theorem 36** [21] If \( P \neq NP \) then there is no algorithm which, for a given binary decision table \( T \) with many-valued decisions, constructs a decision test for \( T \) with minimum cardinality, and for which the time of work is bounded from above by a polynomial depending on the number of columns in \( T \) and the number of separable subtables of \( T \).

In the next chapters, we consider dynamic programming algorithms for optimization of decision and inhibitory rules relative to different cost functions.
Chapter 4

Decision Tables and Tools for the Work with Pareto Optimal Points

In this chapter, we consider some notions connected with decision tables with many-valued decisions (the notions of table, directed acyclic graph for this table, uncertainty and completeness measures, and restricted information system) and discuss tools for the work with Pareto optimal points. The most part of the definitions and results considered in this section were obtained jointly with Hassan AbouEisha, Talha Amin, Mohammad Azad, and Shahid Hussain.

4.1 Decision Tables

A decision table with many-valued decisions is a rectangular table $T$ with $n \geq 1$ columns filled with numbers from the set $\omega = \{0, 1, 2, \ldots\}$ of nonnegative integers. Columns of the table are labeled with conditional attributes $f_1, \ldots, f_n$. Rows of the table are pairwise different, and each row $r$ is labeled with a nonempty finite subset $D(r)$ of $\omega$ which is interpreted as a set of decisions. Rows of the table are interpreted as tuples of values of conditional attributes. We denote by $\text{Row}(T)$ the set of rows of $T$. Let $D(T) = \bigcup_{r \in \text{Row}(T)} D(r)$.

We denote by $T^C$ complementary to $T$ decision table obtained from the table $T$
by changing, for each row \( r \in \text{Row}(T) \), the set \( D(r) \) with the set \( D(T) \setminus D(r) \). When we consider complementary to \( T \) table \( T^C \) or when we study inhibitory rules for \( T \), we will assume that, for any row \( r \) of \( T \), \( D(r) \neq D(T) \). All results for decision rules continue to be true if in the considered decision table \( T \) there are rows \( r \) such that \( D(r) = D(T) \).

A decision table can be represented by a word over the alphabet \( \{0, 1, ;, |\} \) in which numbers from \( \omega \) are in binary representation (are represented by words over the alphabet \( \{0, 1\} \)), the symbol “;” is used to separate two numbers from \( \omega \), and the symbol “|” is used to separate two rows (we add numbers from \( D(r) \) at the end of each row \( r \) and separate these numbers from \( r \) by the symbol “;”). The length of this word will be called the size of the decision table.

A decision table is called empty if it has no rows. We denote by \( T \) the set of all decision tables with many-valued decisions and by \( T^+ \) – the set of nonempty decision tables with many-valued decisions. Let \( T \in T \). The table \( T \) is called degenerate if it is empty or has a common decision – a decision \( d \in D(T) \) such that \( d \in D(r) \) for any row \( r \) of \( T \). We denote by \( \dim(T) \) the number of columns (conditional attributes) in \( T \). We denote by \( N(T) \) the number of rows in the table \( T \) and, for any \( d \in \omega \), we denote by \( N_d(T) \) the number of rows \( r \) of \( T \) such that \( d \in D(r) \). By \( mcd(T) \) we denote the most common decision for \( T \) which is the minimum decision \( d_0 \) from \( D(T) \) such that \( N_{d_0}(T) = \max\{N_d(T) : d \in D(T)\} \). If \( T \) is empty then \( mcd(T) = 0 \). A nonempty decision table is called diagnostic if \( D(r_1) \cap D(r_2) = \emptyset \) for any rows \( r_1, r_2 \) of \( T \) such that \( r_1 \neq r_2 \).

For any conditional attribute \( f_i \in \{f_1, \ldots, f_n\} \), we denote by \( E(T, f_i) \) the set of values of the attribute \( f_i \) in the table \( T \). We denote by \( E(T) \) the set of conditional attributes for which \( |E(T, f_i)| \geq 2 \). Let \( \text{range}(T) = \max\{|E(T, f_i)| : i = 1, \ldots, n\} \).

Let \( T \) be a nonempty decision table. A subtable of \( T \) is a table obtained from \( T \) by removal of some rows. We denote by \( \text{Word}(T) \) the set of all finite words
over the alphabet \( \{(f_i, a) : f_i \in \{f_1, \ldots, f_n\}, a \in \omega\} \) including the empty word \( \lambda \).

Let \( \alpha \in \text{Word}(T) \). We define now a subtable \( T\alpha \) of the table \( T \). If \( \alpha = \lambda \) then \( T\alpha = T \). If \( \alpha \neq \lambda \) and \( \alpha = (f_{i_1}, a_1) \ldots (f_{i_m}, a_m) \) then \( T\alpha = T(f_{i_1}, a_1) \ldots (f_{i_m}, a_m) \) is the subtable of the table \( T \) containing the rows from \( T \) which at the intersection with the columns \( f_{i_1}, \ldots, f_{i_m} \) have numbers \( a_1, \ldots, a_m \), respectively. Such nonempty subtables, including the table \( T \), are called separable subtables of \( T \). We denote by \( \text{SEP}(T) \) the set of separable subtables of the table \( T \). Note that \( N(T) \leq |\text{SEP}(T)| \).

It is clear that the size of each subtable of \( T \) is at most the size of \( T \).

Let \( \Theta \) be a subtable of \( T \). The subtable \( \Theta \) is called incomplete relative to \( T \) if \( D(\Theta) \subset D(T) \). By \( \text{lcd}(T, \Theta) \) we denote the least common decision for \( \Theta \) relative to \( T \) which is the minimum decision \( d_0 \) from \( D(T) \) such that \( N_{d_0}(\Theta) = \min\{N_d(\Theta) : d \in D(T)\} \).

### 4.2 Uncertainty Measures

Let \( \mathbb{R} \) be the set of real numbers and \( \mathbb{R}_+ \) be the set of all nonnegative real numbers. An uncertainty measure is a function \( U : \mathcal{T} \rightarrow \mathbb{R} \) such that \( U(T) \geq 0 \) for any \( T \in \mathcal{T} \), and \( U(T) = 0 \) if and only if \( T \) is a degenerate table. One can show that the following functions (we assume that, for any empty table, the value of each of the considered functions is equal to 0) are uncertainty measures:

- **Misclassification error**: \( \text{me}(T) = N(T) - N_{\text{med}(T)}(T) \).

- **Relative misclassification error**: \( \text{rme}(T) = (N(T) - N_{\text{med}(T)}(T))/N(T) \).

- **Absence**: \( \text{abs}(T) = \prod_{d \in D(T)} (N(T) - N_d(T)) \).

We assume that each of the following numerical operations (we call these operations basic) has time complexity \( O(1) \): \( \max(x, y) \), \( x + y \), \( x \times y \), \( x \div y \), \( \log_2 x \). This assumption is reasonable for computations with floating-point numbers. Under this
assumption, each of the considered three uncertainty measures has polynomial time complexity depending on the size of decision tables.

4.3 Completeness Measures

A completeness measure is a function $W$ defined on the pairs decision table $T$ and its subtable $\Theta$ such that $W(T, \Theta) \geq 0$ for any $T \in \mathcal{T}$ and its subtable $\Theta$, and $W(T, \Theta) = 0$ if and only if $\Theta$ is incomplete relative to $T$. One can show that the following functions (we assume that, for any empty table, the value of each of the considered functions is equal to 0) are completeness measures:

- **Inhibitory misclassification error:** $ime(T, \Theta) = N_{\text{lcd}(T, \Theta)}(\Theta)$.

- **Inhibitory relative misclassification error:**

  $$irme(T, \Theta) = \frac{N_{\text{lcd}(T, \Theta)}(T)}{N(\Theta)}.$$ 

- **Inhibitory absence:** $iabs(T, \Theta) = \prod_{d \in D(T)} N_d(\Theta)$.

Each of the considered three completeness measures has polynomial time complexity depending on the size of decision tables.

It is not difficult to prove the following statement.

**Lemma 37** Let $T$ be a nondegenerate decision table and $\alpha \in \text{Word}(T)$. Then

1. $D(T) = D(T^C)$;
2. $N(T\alpha) = N(T^C\alpha)$;
3. $T\alpha$ is incomplete relative to $T$ if and only if $T^C\alpha$ is degenerate;
4. For any $d \in D(T)$, $N_d(T\alpha) = N(T^C\alpha) - N_d(T^C\alpha)$;
5. \( \text{lcd}(T, T\alpha) = \text{mcd}(T^C\alpha) \).

Let \( U \) be an uncertainty measure and \( W \) be a completeness measure. We will say that \( W \) and \( U \) are dual if, for any nondegenerate \( T \in \mathcal{T} \) and any \( \alpha \in \text{Word}(T) \), \( W(T, T\alpha) = U(T^C\alpha) \).

**Proposition 38** The following pairs of completeness and uncertainty measures are dual: ime and me, irme and rme, iabs and abs.

**Proof.** Let \( T \) be a nondegenerate decision table and \( \alpha \in \text{Word}(T) \). From Lemma 37 it follows that

\[
\text{ime}(T, T\alpha) = N_{\text{lcd}(T,T\alpha)}(T\alpha) = N(T^C\alpha) - N_{\text{mcd}(T^C\alpha)}(T^C\alpha) = \text{me}(T^C\alpha),
\]

\[
\text{irme}(T, T\alpha) = N_{\text{lcd}(T,T\alpha)}(T\alpha)/N(T\alpha) = (N(T^C\alpha) - N_{\text{mcd}(T^C\alpha)}(T^C\alpha))/N(T^C\alpha) = \text{rme}(T^C\alpha),
\]

\[
\text{iabs}(T, T\alpha) = \prod_{d \in D(T)} N_d(T\alpha) = \prod_{d \in D(T^C)} (N(T^C\alpha) - N_d(T^C\alpha)) = \text{abs}(T^C\alpha).
\]

\[\blacksquare\]

### 4.4 Directed Acyclic Graph \( \Delta_{U,\alpha}(T) \)

Let \( T \) be a nonempty decision table with \( n \) conditional attributes \( f_1, \ldots, f_n \), \( U \) be an uncertainty measure, and \( \alpha \in \mathbb{R}_+ \). We now consider an algorithm \( A_1 \) for the construction of a directed acyclic graph \( \Delta_{U,\alpha}(T) \) which will be used for the description and study of decision rules and decision trees. Nodes of this graph are some separable subtables of the table \( T \). During each iteration we process one node. We start with the graph that consists of one node \( T \) which is not processed and finish when all nodes of the graph are processed.
Algorithm \( \mathcal{A}_1 \)

Input: A nonempty decision table \( T \) with \( n \) conditional attributes \( f_1, \ldots, f_n \), an uncertainty measure \( U \), and a number \( \alpha \in \mathbb{R}_+ \).

Output: Directed acyclic graph \( \Delta_{U, \alpha}(T) \).

1. Construct the graph that consists of one node \( T \) which is not marked as processed.

2. If all nodes of the graph are processed then the work of the algorithm is finished.
   Return the resulting graph as \( \Delta_{U, \alpha}(T) \). Otherwise, choose a node (table) \( \Theta \) that has not been processed yet.

3. If \( U(\Theta) \leq \alpha \) mark the node \( \Theta \) as processed and proceed to step 2.

4. If \( U(\Theta) > \alpha \) then, for each \( f_i \in E(\Theta) \), draw a bundle of edges from the node \( \Theta \) (this bundle of edges will be called \( f_i \)-bundle). Let \( E(\Theta, f_i) = \{a_1, \ldots, a_k\} \).
   Then draw \( k \) edges from \( \Theta \) and label these edges by the pairs \((f_i, a_1), \ldots, (f_i, a_k)\). These edges enter nodes \( \Theta(f_i, a_1), \ldots, \Theta(f_i, a_k) \), respectively. If some of the nodes \( \Theta(f_i, a_1), \ldots, \Theta(f_i, a_k) \) are not present in the graph then add these nodes to the graph. Mark the node \( \Theta \) as processed and return to step 2.

We now analyze time complexity of the algorithm \( \mathcal{A}_1 \). By \( L(\Delta_{U, \alpha}(T)) \) we denote the number of nodes in the graph \( \Delta_{U, \alpha}(T) \).

Proposition 39 Let the algorithm \( \mathcal{A}_1 \) use an uncertainty measure \( U \) which has polynomial time complexity depending on the size of the input decision table \( T \). Then the time complexity of the algorithm \( \mathcal{A}_1 \) is bounded from above by a polynomial on the size of the input table \( T \) and the number \( |\text{SEP}(T)| \) of different separable subtables of \( T \).

Proof. Since the uncertainty measure \( U \) has polynomial time complexity depending on the size of decision tables, each step of the algorithm \( \mathcal{A}_1 \) has polynomial time
complexity depending on the size of the table $T$ and the number $L(\Delta_{U,\alpha}(T))$. The number of steps is $O(L(\Delta_{U,\alpha}(T)))$. Therefore the time complexity of the algorithm $A_1$ is bounded from above by a polynomial on the size of the input table $T$ and the number $L(\Delta_{U,\alpha}(T))$. The number $L(\Delta_{U,\alpha}(T))$ is bounded from above by the number $|SEP(T)|$ of different separable subtables of $T$. ■

In Section 4.5, we describe classes of decision tables such that the number of separable subtables in tables from the class is bounded from above by a polynomial on the number of conditional attributes in the table, and the size of tables is bounded from above by a polynomial on the number of conditional attributes. For each such class, time complexity of the algorithm $A_1$ is polynomial depending on the number of conditional attributes in decision tables.

**Remark 40** Note that, for any decision table $T$, the graph $\Delta_{U,0}(T)$ does not depend on the uncertainty measure $U$. We denote this graph $\Delta(T)$. Note also that $L(\Delta(T)) = |SEP(T)|$ for any diagnostic decision table $T$.

A node of directed graph is called *terminal* if there are no edges starting in this node. A *proper subgraph* of the graph $\Delta_{U,\alpha}(T)$ is a graph $G$ obtained from $\Delta_{U,\alpha}(T)$ by removal of some bundles of edges such that each nonterminal node of $\Delta_{U,\alpha}(T)$ keeps at least one bundle of edges starting in this node. By definition, $\Delta_{U,\alpha}(T)$ is a proper subgraph of $\Delta_{U,\alpha}(T)$. A node $\Theta$ of the graph $G$ is terminal if and only if $U(\Theta) \leq \alpha$. We denote by $L(G)$ the number of nodes in the graph $G$.

**4.5 Restricted Information Systems**

In this section, we describe classes of decision tables for which algorithms that deal with the graphs $\Delta_{U,\alpha}(T)$ have polynomial time complexity depending on the number of conditional attributes in the input table $T$.
Let $A$ be a nonempty set and $F$ be a nonempty set of non-constant functions from $A$ to $E_k = \{0, \ldots, k-1\}$, $k \geq 2$. Functions from $F$ are called attributes, and the pair $\mathcal{U} = (A, F)$ is called a $k$-valued information system.

For arbitrary attributes $f_1, \ldots, f_n \in F$ and a mapping $\nu$ which corresponds to each tuple $(\delta_1, \ldots, \delta_n) \in E_k^n$ a nonempty subset $\nu(\delta_1, \ldots, \delta_n)$ of the set $\{0, \ldots, k^n - 1\}$ which cardinality is at most $n^k$, we denote by $T_\nu(f_1, \ldots, f_n)$ the decision table with $n$ conditional attributes $f_1, \ldots, f_n$ which contains the row $(\delta_1, \ldots, \delta_n) \in E_k^n$ if and only if the system of equations

$$\{f_1(x) = \delta_1, \ldots, f_n(x) = \delta_n\}$$

is compatible (has a solution from the set $A$). This row is labeled with the set of decisions $\nu(\delta_1, \ldots, \delta_n)$. The table $T_\nu(f_1, \ldots, f_n)$ is called a decision table over the information system $\mathcal{U}$. We denote by $\mathcal{T}(\mathcal{U})$ the set of decision tables over $\mathcal{U}$.

Let us consider the function

$$SEP_\mathcal{U}(n) = \max\{|SEP(T)| : T \in \mathcal{T}(\mathcal{U}), \text{dim}(T) \leq n\},$$

where $\text{dim}(T)$ is the number of conditional attributes in $T$, which characterizes the maximum number of separable subtables depending on the number of conditional attributes in decision tables over $\mathcal{U}$.

A system of equations of the kind (4.1) is called a system of equations over $\mathcal{U}$. Two systems of equations are called equivalent if they have the same set of solutions from $A$. A compatible system of equations will be called irreducible if each of its proper subsystems is not equivalent to the system. Let $r$ be a natural number. An information system $\mathcal{U}$ will be called $r$-restricted if each irreducible system of equations over $\mathcal{U}$ consists of at most $r$ equations. An information system $\mathcal{U}$ will be called restricted if it is $r$-restricted for some natural $r$.

**Theorem 41** [37] Let $\mathcal{U} = (A, F)$ be a $k$-valued information system. Then the fol-
lowing statements hold:

1. If \( U \) is an \( r \)-restricted information system then \( SEP_U(n) \leq (nk)^r + 1 \) for any natural \( n \).

2. If \( U \) is not a restricted information system then \( SEP_U(n) \geq 2^n \) for any natural \( n \).

We now evaluate the time complexity of the algorithm \( A_1 \) for decision tables over a restricted information system \( U \) under the assumption that the uncertainty measure \( U \) used by \( A_1 \) has polynomial time complexity.

Lemma 42 Let \( U \) be a restricted information system. Then, for decision tables from \( T(U) \), both the size and the number of separable subtables are bounded from above by polynomials on the number of conditional attributes.

Proof. Let \( U \) be a \( k \)-valued information system which is \( r \)-restricted. For any decision table \( T \in T(U) \), each value of each conditional attribute is at most \( k \), the value of each decision is at most \( k^{\dim(T)} \), the cardinality of each set of decisions attached to rows of \( T \) is at most \( \dim(T)^k \) and, by Theorem 41, \( N(T) \leq |SEP(T)| \leq (\dim(T)k)^r+1 \). From here it follows that the size of decision tables from \( T(U) \) is bounded from above by a polynomial on the number of conditional attributes in decision tables. By Theorem 41, the number of separable subtables for decision tables from \( T(U) \) is bounded from above by a polynomial on the number of conditional attributes in decision tables.

Proposition 43 Let \( U \) be a restricted information system, and the uncertainty measure \( U \) used by the algorithm \( A_1 \) have polynomial time complexity depending on the size of decision tables. Then the algorithm \( A_1 \) has polynomial time complexity for decision tables from \( T(U) \) depending on the number of conditional attributes.
Proof. By Proposition 39, the time complexity of the algorithm $A_1$ is bounded from above by a polynomial on the size of the input table $T$ and the number $|SEP(T)|$ of different separable subtables of $T$. From Lemma 42 it follows that, for decision tables from $T(U)$, both the size and the number of separable subtables are bounded from above by polynomials on the number of conditional attributes. ■

Remark 44 Let $U$ be an information system which is not restricted. Using Remark 40 and Theorem 41 one can show that there is no an algorithm which constructs the graph $\Delta(T)$ for decision tables $T \in T(U)$ and which time complexity is bounded from above by a polynomial on the number of conditional attributes in the considered decision tables.

Let us consider a family of restricted information systems. Let $d$ and $t$ be natural numbers, $f_1, \ldots, f_t$ be functions from $\mathbb{R}^d$ to $\mathbb{R}$, and $s$ be a function from $\mathbb{R}$ to $\{0, 1\}$ such that $s(x) = 0$ if $x < 0$ and $s(x) = 1$ if $x \geq 0$. Then the 2-valued information system $U = (\mathbb{R}^d, F)$ where $F = \{s(f_i + c) : i = 1, \ldots, t, c \in \mathbb{R}\}$ is a $2t$-restricted information system.

If $f_1, \ldots, f_t$ are linear functions then we deal with attributes corresponding to $t$ families of parallel hyperplanes in $\mathbb{R}^d$ which is usual for decision trees for datasets with $t$ numerical attributes only [38].

We consider now a class of so-called linear information systems for which all restricted systems are known. Let $P$ be the set of all points in the plane and $l$ be a straight line (line in short) in the plane. This line divides the plane into two open half-planes $H_1$ and $H_2$ and the line $l$. Two attributes correspond to the line $l$. The first attribute takes value 0 on points from $H_1$, and value 1 on points from $H_2$ and $l$. The second one takes value 0 on points from $H_2$, and value 1 on points from $H_1$ and $l$. We denote by $\mathcal{L}$ the set of all attributes corresponding to lines in the plane. Information systems of the kind $(P, F)$ where $F \subseteq \mathcal{L}$, will be called linear information systems. We describe all restricted linear information systems.
Let \( l \) be a line in the plane. Let us denote by \( \mathcal{L}(l) \) the set of all attributes corresponding to lines which are parallel to \( l \). Let \( p \) be a point in the plane. We denote by \( \mathcal{L}(p) \) the set of all attributes corresponding to lines which pass through \( p \). A set \( C \) of attributes from \( \mathcal{L} \) is called a clone if \( C \subseteq \mathcal{L}(l) \) for some line \( l \) or \( C \subseteq \mathcal{L}(p) \) for some point \( p \).

**Theorem 45** [39] A linear information system \((P,F)\) is restricted if and only if \( F \) is the union of a finite number of clones.

### 4.6 Time Complexity of Algorithms on \( \Delta_{U,\alpha}(T) \)

In this thesis, we consider a number of algorithms which deal with the graph \( \Delta_{U,\alpha}(T) \). To evaluate time complexity of these algorithms, we will count the number of elementary operations made by the algorithms. These operations can either be basic numerical operations or computations of numerical parameters of decision tables. We assume, as we already mentioned, that each basic numerical operation \((\max(x, y), x + y, x - y, x \times y, x \div y, \log_2 x)\) has time complexity \( O(1) \). This assumption is reasonable for computations with floating-point numbers. Furthermore, computing the considered parameters of decision tables usually has polynomial time complexity depending on the size of the decision table.

**Proposition 46** Let, for some algorithm \( \mathcal{A} \) working with decision tables, the number of elementary operations (basic numerical operations and computations of numerical parameters of decision tables) be polynomial depending on the size of the input table \( T \) and on the number of separable subtables of \( T \), and the computations of parameters of decision tables used by the algorithm \( \mathcal{A} \) have polynomial time complexity depending on the size of decision tables. Then, for any restricted information system \( U \), the algorithm \( \mathcal{A} \) has polynomial time complexity for decision tables from \( \mathcal{T}(U) \) depending on the number of conditional attributes.
Proof. Let $\mathcal{U}$ be a restricted information system. From Lemma 42 it follows that the size and the number of separable subtables for decision tables from $\mathcal{T}(\mathcal{U})$ are bounded from above by polynomials on the number of conditional attributes in the tables. From here it follows that the algorithm $\mathcal{A}$ has polynomial time complexity for decision tables from $\mathcal{T}(\mathcal{U})$ depending on the number of conditional attributes. ■

4.7 Tools for Study of Pareto Optimal Points

In this section, we consider tools (statements and algorithms) which are used for the study of Pareto optimal points for bi-criteria optimization problems.

Let $\mathbb{R}^2$ be the set of pairs of real numbers (points). We consider a partial order $\leq$ on the set $\mathbb{R}^2$: $(c, d) \leq (a, b)$ if $c \leq a$ and $d \leq b$. Two points $\alpha$ and $\beta$ are comparable if $\alpha \leq \beta$ or $\beta \leq \alpha$. A subset of $\mathbb{R}^2$ in which no two different points are comparable is called an antichain. We will write $\alpha < \beta$ if $\alpha \leq \beta$ and $\alpha \neq \beta$. If $\alpha$ and $\beta$ are comparable then $\min(\alpha, \beta) = \alpha$ if $\alpha \leq \beta$ and $\min(\alpha, \beta) = \beta$ if $\alpha > \beta$.

Let $A$ be a nonempty finite subset of $\mathbb{R}^2$. A point $\alpha \in A$ is called a Pareto optimal point for $A$ if there is no a point $\beta \in A$ such that $\beta < \alpha$. We denote by $\text{Par}(A)$ the set of Pareto optimal points for $A$. It is clear that $\text{Par}(A)$ is an antichain.

Lemma 47 Let $A$ be a nonempty finite subset of the set $\mathbb{R}^2$. Then, for any point $\alpha \in A$, there is a point $\beta \in \text{Par}(A)$ such that $\beta \leq \alpha$.

Proof. Let $\beta = (a, b)$ be a point from $A$ such that $(a, b) \leq \alpha$ and $a + b = \min\{c + d : (c, d) \in A, (c, d) \leq \alpha\}$. Then $(a, b) \in \text{Par}(A)$. ■

Lemma 48 Let $A, B$ be nonempty finite subsets of the set $\mathbb{R}^2$, $A \subseteq B$, and, for any $\beta \in B$, there exists $\alpha \in A$ such that $\alpha \leq \beta$. Then $\text{Par}(A) = \text{Par}(B)$.

Proof. Let $\beta \in \text{Par}(B)$. Then there exists $\alpha \in A$ such that $\alpha \leq \beta$. By Lemma 47, there exists $\gamma \in \text{Par}(A)$ such that $\gamma \leq \alpha$. Therefore $\gamma \leq \beta$ and $\gamma = \beta$ since $\beta \in \text{Par}(B)$. Hence $\text{Par}(B) \subseteq \text{Par}(A)$. ■
Let \( \alpha \in Par(A) \). By Lemma 47, there exists \( \beta \in Par(B) \) such that \( \beta \leq \alpha \). We know that there exists \( \gamma \in A \) such that \( \gamma \leq \beta \). Therefore \( \gamma \leq \alpha \) and \( \gamma = \alpha \) since \( \alpha \in Par(A) \). As a result, we have \( \beta = \alpha \) and \( Par(A) \subseteq Par(B) \). Hence \( Par(A) = Par(B) \). \( \blacksquare \)

**Lemma 49** Let \( A \) be a nonempty finite subset of \( \mathbb{R}^2 \). Then

\[
|Par(A)| \leq \min \left( |A^{(1)}|, |A^{(2)}| \right)
\]

where \( A^{(1)} = \{a : (a, b) \in A\} \) and \( A^{(2)} = \{b : (a, b) \in A\} \).

**Proof.** Let \( (a, b), (c, d) \in Par(A) \) and \( (a, b) \neq (c, d) \). Then \( a \neq c \) and \( b \neq d \) (otherwise, \( (a, b) \) and \( (c, d) \) are comparable). Therefore \( |Par(A)| \leq \min \left( |A^{(1)}|, |A^{(2)}| \right) \).

\( \blacksquare \)

Points from \( Par(A) \) can be ordered in the following way: \( (a_1, b_1), \ldots, (a_t, b_t) \) where \( a_1 < \ldots < a_t \). Since points from \( Par(A) \) are incomparable, \( b_1 > \ldots > b_t \). We will refer to the sequence \( (a_1, b_1), \ldots, (a_t, b_t) \) as the normal representation of the set \( Par(A) \).

We now describe an algorithm which, for a given nonempty finite subset \( A \) of the set \( \mathbb{R}^2 \), constructs the normal representation of the set \( Par(A) \). We assume that \( A \) is a multiset containing, possibly, repeating elements. The cardinality \( |A| \) of \( A \) is the total number of elements in \( A \).

**Algorithm \( A_2 \)**

**Input:** A nonempty finite subset \( A \) of the set \( \mathbb{R}^2 \) containing, possibly, repeating elements (multiset).

**Output:** Normal representation \( P \) of the set \( Par(A) \) of Pareto optimal points for \( A \).

1. Set \( P \) equal to the empty sequence.
2. Construct a sequence $B$ of all points from $A$ ordered according to the first coordinate in the ascending order.

3. If there is only one point in the sequence $B$, then add this point to the end of the sequence $P$, return $P$, and finish the work of the algorithm. Otherwise, choose the first $\alpha = (\alpha_1, \alpha_2)$ and the second $\beta = (\beta_1, \beta_2)$ points from $B$.

4. If $\alpha$ and $\beta$ are comparable then remove $\alpha$ and $\beta$ from $B$, add the point $\min(\alpha, \beta)$ to the beginning of $B$, and proceed to step 3.

5. If $\alpha$ and $\beta$ are not comparable (in this case $\alpha_1 < \beta_1$ and $\alpha_2 > \beta_2$) then remove $\alpha$ from $B$, add the point $\alpha$ to the end of $P$, and proceed to step 3.

**Proposition 50** Let $A$ be a nonempty finite subset of the set $\mathbb{R}^2$ containing, possibly, repeating elements (multiset). Then the algorithm $\mathcal{A}_2$ returns the normal representation of the set $\text{Par}(A)$ of Pareto optimal points for $A$ and makes $O(|A| \log |A|)$ comparisons.

**Proof.** The step 2 of the algorithm requires $O(|A| \log |A|)$ comparisons. Each call to step 3 (with the exception of the last one) leads to two comparisons. The number of calls to step 3 is at most $|A|$. Therefore the algorithm $\mathcal{A}_2$ makes $O(|A| \log |A|)$ comparisons.

Let the output sequence $P$ be equal to $(a_1, b_1), \ldots, (a_t, b_t)$ and let us set $Q = \{(a_1, b_1), \ldots, (a_t, b_t)\}$. It is clear that $a_1 < \ldots < a_t$, $b_1 > \ldots > b_t$ and, for any $\alpha \in A$, $\alpha \notin Q$, there exists $\beta \in Q$ such that $\beta < \alpha$. From here it follows that $\text{Par}(A) \subseteq Q$ and $Q$ is an antichain. Let us assume that there exists $\gamma \in Q$ which does not belong to $\text{Par}(A)$. Then there exists $\alpha \in A$ such that $\alpha < \gamma$. Since $Q$ is an antichain, $\alpha \notin Q$. We know that there exists $\beta \in Q$ such that $\beta \leq \alpha$. This results in two different points $\beta$ and $\gamma$ from $Q$ being comparable, which is impossible. Therefore $Q = \text{Par}(A)$ and $P$ is the normal representation of the set $\text{Par}(A)$. \blacksquare
Remark 51 Let $A$ be a nonempty finite subset of $\mathbb{R}^2$, $(a_1, b_1), \ldots, (a_t, b_t)$ be the normal representation of the set $\text{Par}(A)$, and $\text{rev}(A) = \{(b, a) : (a, b) ∈ A\}$. Then $\text{Par}(\text{rev}(A)) = \text{rev}(\text{Par}(A))$ and $(b_t, a_t), \ldots, (b_1, a_1)$ is the normal representation of the set $\text{Par}(\text{rev}(A))$.

Lemma 52 Let $A$ be a nonempty finite subset of $\mathbb{R}^2$, $B ⊆ A$, and $\text{Par}(A) ⊆ B$. Then $\text{Par}(B) = \text{Par}(A)$.

Proof. It is clear that $\text{Par}(A) ⊆ \text{Par}(B)$. Let us assume that, for some $β$, $β ∈ \text{Par}(B)$ and $β ∉ \text{Par}(A)$. Then there exists $α ∈ A$ such that $α < β$. By Lemma 47, there exists $γ ∈ \text{Par}(A) ⊆ B$ such that $γ ≤ α$. Therefore $γ < β$ and $β ∉ \text{Par}(B)$. Hence $\text{Par}(B) = \text{Par}(A)$. ■

Lemma 53 Let $A_1, \ldots, A_k$ be nonempty finite subsets of $\mathbb{R}^2$. Then $\text{Par}(A_1 ∪ \ldots ∪ A_k) ⊆ \text{Par}(A_1) ∪ \ldots ∪ \text{Par}(A_k)$.

Proof. Let $α ∈ (A_1 ∪ \ldots ∪ A_k) \setminus (\text{Par}(A_1) ∪ \ldots ∪ \text{Par}(A_k))$. Then there is $i ∈ \{1, \ldots, k\}$ such that $α ∈ A_i$ but $α ∉ \text{Par}(A_i)$. Therefore there is $β ∈ A_i$ such that $β < α$. Hence $α ∉ \text{Par}(A_1 ∪ \ldots ∪ A_k)$, and $\text{Par}(A_1 ∪ \ldots ∪ A_k) ⊆ \text{Par}(A_1) ∪ \ldots ∪ \text{Par}(A_k)$. ■

A function $f : \mathbb{R}^2 → \mathbb{R}$ is called increasing if $f(x_1, y_1) ≤ f(x_2, y_2)$ for any $x_1, x_2, y_1, y_2 ∈ \mathbb{R}$ such that $x_1 ≤ x_2$ and $y_1 ≤ y_2$. For example, $\text{sum}(x, y) = x + y$ and $\text{max}(x, y)$ are increasing functions.

Let $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$, and $A, B$ be nonempty finite subsets of the set $\mathbb{R}^2$. We denote by $A \langle fg \rangle B$ the set $\{(f(a, c), g(b, d)) : (a, b) ∈ A, (c, d) ∈ B\}$.

Lemma 54 Let $A, B$ be nonempty finite subsets of $\mathbb{R}^2$, and $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$. Then $\text{Par}(A \langle fg \rangle B) ⊆ \text{Par}(A) \langle fg \rangle \text{Par}(B)$.

Proof. Let $β ∈ \text{Par}(A \langle fg \rangle B)$ and $β = (f(a, c), g(b, d))$ where $(a, b) ∈ A$ and $(c, d) ∈ B$. Then, by Lemma 47, there exist $(a', b') ∈ \text{Par}(A)$ and $(c', d') ∈ \text{Par}(B)$
such that \((a', b') \leq (a, b)\) and \((c', d') \leq (c, d)\). It is clear that \(\alpha = (f(a', c'), g(b', d')) \leq (f(a, c), g(b, d)) = \beta\), and \(\alpha \in \text{Par}(A)\langle fg \rangle \text{Par}(B)\). Since \(\beta \in \text{Par}(A\langle fg \rangle B)\), we have \(\beta = \alpha\). Therefore

\[
\text{Par}(A\langle fg \rangle B) \subseteq \text{Par}(A\langle fg \rangle \text{Par}(B)).
\]

Let \(f, g\) be increasing functions from \(\mathbb{R}^2\) to \(\mathbb{R}\), \(P_1, \ldots, P_t\) be nonempty finite subsets of \(\mathbb{R}^2\), \(Q_1 = P_1\), and, for \(i = 2, \ldots, t\), \(Q_i = Q_{i-1}\langle fg \rangle P_i\). We assume that, for \(i = 1, \ldots, t\), the sets \(\text{Par}(P_1), \ldots, \text{Par}(P_t)\) are already constructed. We now describe an algorithm that constructs the sets \(\text{Par}(Q_1), \ldots, \text{Par}(Q_t)\) and returns \(\text{Par}(Q_t)\).

**Algorithm \(A_3\)**

**Input:** Increasing functions \(f, g\) from \(\mathbb{R}^2\) to \(\mathbb{R}\), and sets \(\text{Par}(P_1), \ldots, \text{Par}(P_t)\) for some nonempty finite subsets \(P_1, \ldots, P_t\) of \(\mathbb{R}^2\).

**Output:** The set \(\text{Par}(Q_t)\) where \(Q_1 = P_1\), and, for \(i = 2, \ldots, t\), \(Q_i = Q_{i-1}\langle fg \rangle P_i\).

1. Set \(B_1 = \text{Par}(P_1)\) and \(i = 2\).

2. Construct the multiset \(A_i = B_{i-1}\langle fg \rangle \text{Par}(P_i) = \{(f(a, c), g(b, d)) : (a, b) \in B_{i-1}, (c, d) \in \text{Par}(P_i)\}\) - we will not remove equal pairs from the constructed set.

3. Using algorithm \(A_2\), construct the set \(B_i = \text{Par}(A_i)\).

4. If \(i = t\) then return \(B_i\) and finish the work of the algorithm. Otherwise, set \(i = i + 1\) and proceed to step 2.

**Proposition 55** Let \(f, g\) be increasing functions from \(\mathbb{R}^2\) to \(\mathbb{R}\), \(P_1, \ldots, P_t\) be nonempty finite subsets of \(\mathbb{R}^2\), \(Q_1 = P_1\), and, for \(i = 2, \ldots, t\), \(Q_i = Q_{i-1}\langle fg \rangle P_i\). Then the algorithm \(A_3\) returns the set \(\text{Par}(Q_t)\).
Proof. We will prove by induction on $i$ that, for $i = 1, \ldots, t$, the set $B_i$ (see the description of the algorithm $A_3$) is equal to the set $Par(Q_i)$. Since $B_1 = Par(P_1)$ and $Q_1 = P_1$, we have $B_1 = Par(Q_1)$. Let for some $i - 1, 2 \leq i \leq t$, the considered statement hold, i.e., $B_{i - 1} = Par(Q_{i - 1})$. Then $B_i = Par(B_{i - 1} \langle fg \rangle Par(P_i)) = Par(Par(Q_{i - 1}) \langle fg \rangle Par(P_i))$. We know that $Q_i = Q_{i - 1} \langle fg \rangle P_i$. By Lemma 54, $Par(Q_i) \subset Par(Q_{i - 1}) \langle fg \rangle Par(P_i)$. By Lemma 52, $Par(Q_i) = Par(Par(Q_{i - 1}) \langle fg \rangle Par(P_i))$.

Therefore $B_i = Par(Q_i)$. So we have $B_t = Par(Q_t)$, and the algorithm $A_3$ returns the set $Par(Q_t)$. ■

Proposition 56 Let $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$, $f \in \{x + y, \max(x, y)\}$, $P_1, \ldots, P_t$ be nonempty finite subsets of $\mathbb{R}^2$, $Q_1 = P_1$, and for $i = 2, \ldots, t$, $Q_i = Q_{i - 1} \langle fg \rangle P_i$. Let $P_i^{(1)} = \{a : (a, b) \in P_i\}$ for $i = 1, \ldots, t$, $m \in \omega$, and $P_i^{(1)} \subseteq \{0, 1, \ldots, m\}$ for $i = 1, \ldots, t$, or $P_i^{(1)} \subseteq \{0, -1, \ldots, -m\}$ for $i = 1, \ldots, t$. Then, during the construction of the set $Par(Q_t)$, the algorithm $A_3$ makes

$$O(t^2m^2 \log(tm))$$

elementary operations (computations of $f, g$ and comparisons) if $f = x + y$, and

$$O(tm^2 \log m)$$

elementary operations (computations of $f, g$ and comparisons) if $f = \max(x, y)$. If $f = x + y$ then $|Par(Q_t)| \leq tm + 1$, and if $f = \max(x, y)$ then $|Par(Q_t)| \leq m + 1$.

Proof. For $i = 1, \ldots, t$, we denote $p_i = |Par(P_i)|$ and $q_i = |Par(Q_i)|$. Let $i \in \{2, \ldots, t\}$. To construct the multiset $A_i = Par(Q_{i - 1}) \langle fg \rangle Par(P_i)$, we need to compute the values of $f$ and $g$ a number of times equal to $q_{i - 1}p_i$. The cardinality of $A_i$
is equal to \(q_{i-1}p_i\). We apply to \(A_i\) the algorithm \(A_2\) which makes \(O(q_{i-1}p_i \log(q_{i-1}p_i))\) comparisons. As a result, we find the set \(\text{Par}(A_i) = \text{Par}(Q_i)\). To construct the sets \(\text{Par}(Q_1), \ldots, \text{Par}(Q_t)\), the algorithm \(A_3\) makes \(\sum_{i=2}^{t} q_{i-1} p_i\) computations of \(f\), \(\sum_{i=2}^{t} q_{i-1} p_i\) computations of \(g\), and \(O(\sum_{i=2}^{t} q_{i-1} p_i \log(q_{i-1}p_i))\) comparisons.

We know that \(P^{(1)}_i \subseteq \{0, 1, \ldots, m\}\) for \(i = 1, \ldots, t\), or \(P^{(1)}_i \subseteq \{0, -1, \ldots, -m\}\) for \(i = 1, \ldots, t\). Then, by Lemma 49, \(p_i \leq m + 1\) for \(i = 1, \ldots, t\).

Let \(f = x + y\). Then, for \(i = 1, \ldots, t\), \(Q^{(1)}_i = \{a : (a, b) \in Q_i\} \subseteq \{0, 1, \ldots, m\}\) or, for \(i = 1, \ldots, t\), \(Q^{(1)}_i \subseteq \{0, -1, \ldots, -m\}\) and, by Lemma 49, \(q_i \leq im + 1\). In this case, to construct the sets \(\text{Par}(Q_1), \ldots, \text{Par}(Q_t)\) the algorithm \(A_3\) makes \(O(t^2 m^2)\) computations of \(f\), \(O(t^2 m^2)\) computations of \(g\), and \(O(t^2 m^2 \log(tm))\) comparisons, i.e.,

\[
O(t^2 m^2 \log(tm))
\]

elementary operations (computations of \(f\), \(g\), and comparisons). Since \(q_t \leq tm + 1\),

\[
|\text{Par}(Q_t)| \leq tm + 1.
\]

Let \(f = \max(x, y)\). Then, for \(i = 1, \ldots, t\), \(Q^{(1)}_i = \{a : (a, b) \in Q_i\} \subseteq \{0, 1, \ldots, m\}\) or, for \(i = 1, \ldots, t\), \(Q^{(1)}_i \subseteq \{0, -1, \ldots, -m\}\) and, by Lemma 49, \(q_i \leq m + 1\). In this case, to construct the sets \(\text{Par}(Q_1), \ldots, \text{Par}(Q_t)\) the algorithm \(A_3\) makes \(O(tm^2)\) computations of \(f\), \(O(tm^2)\) computations of \(g\), and \(O(tm^2 \log m)\) comparisons, i.e.,

\[
O(tm^2 \log m)
\]

elementary operations (computations of \(f\), \(g\), and comparisons). Since \(q_t \leq m + 1\),

\[
|\text{Par}(Q_t)| \leq m + 1.
\]

Similar analysis can be done for the sets \(P^{(2)}_i = \{b : (a, b) \in P_i\}\), \(Q^{(2)}_i = \{b : (a, b) \in Q_i\}\), and the function \(g\).

A function \(p : \mathbb{R} \to \mathbb{R}\) is called strictly increasing if \(p(x) < p(y)\) for any \(x, y \in \mathbb{R}\) such that \(x < y\). Let \(p\) and \(q\) be strictly increasing functions from \(\mathbb{R}\) to \(\mathbb{R}\). For
$(a, b) \in \mathbb{R}^2$, we denote by $(a, b)^pq$ the pair $(p(a), q(b))$. For a nonempty finite subset $A$ of $\mathbb{R}^2$, we denote $A^pq = \{(a, b)^pq : (a, b) \in A\}$.

**Lemma 57** Let $A$ be a nonempty finite subset of $\mathbb{R}^2$, and $p$, $q$ be strictly increasing functions from $\mathbb{R}$ to $\mathbb{R}$. Then $\text{Par}(A^pq) = \text{Par}(A)^pq$.

**Proof.** Let $(a, b), (c, d) \in A$. It is clear that $(c, d) = (a, b)$ if and only if $(c, d)^pq = (a, b)^pq$, and $(c, d) < (a, b)$ if and only if $(c, d)^pq < (a, b)^pq$. From here it follows that $\text{Par}(A^pq) = \text{Par}(A)^pq$. ■

Let $A$ be a nonempty finite subset of $\mathbb{R}^2$. We correspond to $A$ a partial function $\mathcal{F}_A : \mathbb{R} \to \mathbb{R}$ defined in the following way: $\mathcal{F}_A(x) = \min\{b : (a, b) \in A, a \leq x\}$.

**Lemma 58** Let $A$ be a nonempty finite subset of $\mathbb{R}^2$, and $(a_1, b_1), \ldots, (a_t, b_t)$ be the normal representation of the set $\text{Par}(A)$. Then, for any $x \in \mathbb{R}$, $\mathcal{F}_A(x) = \mathcal{F}(x)$ where

$$
\mathcal{F}(x) = \begin{cases} 
\text{undefined,} & x < a_1 \\
b_1, & a_1 \leq x < a_2 \\
\vdots & \vdots \\
b_{t-1}, & a_{t-1} \leq x < a_t \\
b_t, & a_t \leq x 
\end{cases}
$$

**Proof.** One can show that $a_1 = \min\{a : (a, b) \in A\}$. Therefore the value $\mathcal{F}_A(x)$ is undefined if $x < a_1$. Let $x \geq a_1$. Then both values $\mathcal{F}(x)$ and $\mathcal{F}_A(x)$ are defined. It is easy to check that $\mathcal{F}(x) = \mathcal{F}_{\text{Par}(A)}(x)$. Since $\text{Par}(A) \subseteq A$, we have $\mathcal{F}_A(x) \leq \mathcal{F}(x)$. By Lemma 47, for any point $(a, b) \in A$, there is a point $(a_i, b_i) \in \text{Par}(A)$ such that $(a_i, b_i) \leq (a, b)$. Therefore $\mathcal{F}(x) \leq \mathcal{F}_A(x)$ and $\mathcal{F}_A(x) = \mathcal{F}(x)$. ■

**Remark 59** We can consider not only function $\mathcal{F}_A$ but also function $\mathcal{F}_{\text{rev}(A)} : \mathbb{R} \to \mathbb{R}$ defined in the following way:

$$
\mathcal{F}_{\text{rev}(A)}(x) = \min\{a : (b, a) \in \text{rev}(A), b \leq x\} = \min\{a : (a, b) \in A, b \leq x\}.
$$
From Remark 51 and Lemma 58 it follows that

\[ \mathcal{F}_{\text{rev}(A)}(x) = \begin{cases} 
\text{undefined}, & x < b_t \\
\alpha_t, & b_t \leq x < b_{t-1} \\
\ldots & \ldots \\
\alpha_2, & b_2 \leq x < b_1 \\
\alpha_1, & b_1 \leq x 
\end{cases} \]
Chapter 5

Decision and Inhibitory Rules and Systems of Rules

In this chapter, we consider various types of decision and inhibitory rules and systems of rules. We discuss the notion of cost function for rules, the notion of decision rule uncertainty, and the notion of inhibitory rule completeness.

5.1 Decision Rules and Systems of Rules

5.1.1 Decision Rules

Let $T$ be a decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $r = (b_1, \ldots, b_n)$ be a row of $T$. A decision rule over $T$ is an expression of the kind

$$f_{i_1} = a_1 \land \ldots \land f_{i_m} = a_m \rightarrow t$$

(5.1)

where $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$, and $a_1, \ldots, a_m, t$ are numbers from $\omega$. It is possible that $m = 0$. For the considered rule, we denote $\beta_0 = \lambda$, and if $m > 0$ we denote $\beta_j = (f_{i_1}, a_1) \ldots (f_{i_j}, a_j)$ for $j = 1, \ldots, m$. We will say that the decision rule (5.1) covers the row $r$ if $r$ belongs to $T_{\beta_m}$, i.e., $b_{i_1} = a_1, \ldots, b_{i_m} = a_m$.

A decision rule (5.1) over $T$ is called a decision rule for $T$ if $t = \text{mcd}(T_{\beta_m})$. 
and either $m = 0$, or $m > 0$ and, for $j = 1, \ldots, m$, $T \beta_{j-1}$ is not degenerate, and $f_{ij} \in E(T \beta_{j-1})$. A decision rule (5.1) for $T$ is called a decision rule for $T$ and $r$ if it covers $r$.

We denote by $DR(T)$ the set of decision rules for $T$. By $DR(T, r)$ we denote the set of decision rules for $T$ and $r$.

Let $U$ be an uncertainty measure and $\alpha \in \mathbb{R}_+$. A decision rule (5.1) for $T$ is called a $(U, \alpha)$-decision rule for $T$ if $U(T \beta_m) \leq \alpha$ and, if $m > 0$, then $U(T \beta_j) > \alpha$ for $j = 0, \ldots, m - 1$. A $(U, \alpha)$-decision rule (5.1) for $T$ is called a $(U, \alpha)$-decision rule for $T$ and $r$ if it covers $r$.

We denote by $DR_{U, \alpha}(T)$ the set of $(U, \alpha)$-decision rules for $T$, and we denote by $DR_{U, \alpha}(T, r)$ the set of $(U, \alpha)$-decision rules for $T$ and $r$.

A decision rule (5.1) for $T$ is called a $U$-decision rule for $T$ if there exists a non-negative real number $\alpha$ such that (5.1) is a $(U, \alpha)$-decision rule for $T$. A decision rule (5.1) for $T$ and $r$ is called a $U$-decision rule for $T$ and $r$ if there exists a nonnegative real number $\alpha$ such that (5.1) is a $(U, \alpha)$-decision rule for $T$ and $r$.

We denote by $DR_U(T)$ the set of $U$-decision rules for $T$. By $DR_U(T, r)$ we denote the set of $U$-decision rules for $T$ and $r$.

We define uncertainty $U(T, \rho)$ of a decision rule $\rho$ for $T$ relative to the table $T$ in the following way. Let $\rho$ be equal to (5.1). Then $U(T, \rho) = U(T \beta_m)$.

We now consider a notion of cost function for decision rules. This is a function $\psi(T, \rho)$ which is defined on pairs $T, \rho$, where $T$ is a nonempty decision table and $\rho$ is a decision rule for $T$, and has values from the set $\mathbb{R}$ of real numbers. This cost function is given by pairs of functions $\psi^0 : T^+ \to \mathbb{R}$ and $F : \mathbb{R} \to \mathbb{R}$ where $T^+$ is the set of nonempty decision tables. The value $\psi(T, \rho)$ is defined by induction:

- If $\rho$ is equal to $\rightarrow mcd(T)$ then $\psi(T, \rightarrow mcd(T)) = \psi^0(T)$. 
If $\rho$ is equal to $f_i = a \land \gamma \rightarrow t$ then

$$
\psi(T, f_i = a \land \gamma \rightarrow t) = F(\psi(T(f_i, a), \gamma \rightarrow t)).
$$

The cost function $\psi$ is called \textit{strictly increasing} if $F(x_1) > F(x_2)$ for any $x_1, x_2 \in \mathbb{R}$ such that $x_1 > x_2$.

Let us consider examples of strictly increasing cost functions for decision rules:

- The \textit{length} $l(T, \rho) = l(\rho)$ for which $\psi^0(T) = 0$ and $F(x) = x + 1$. The length of the rule (5.1) is equal to $m$.

- The \textit{coverage} $c(T, \rho)$ for which $\psi^0(T) = N_{med(T)}(T)$ and $F(x) = x$. The coverage of the rule (5.1) for table $T$ is equal to $N_{med(T\beta_m)}(T\beta_m)$.

- The \textit{relative coverage} $rc(T, \rho)$ for which $\psi^0(T) = N_{med(T)}(T)/N(T)$ and $F(x) = x$. The relative coverage of the rule (5.1) for table $T$ is equal to $N_{med(T^m)}(T\beta_m)/N(T\beta_m)$.

- The \textit{modified coverage} $c_M(T, \rho)$ for which $\psi^0(T) = N^M(T)$ and $F(x) = x$. Here $M$ is a set of rows of $T$ and, for any subtable $\Theta$ of $T$, $N^M(\Theta)$ is the number of rows of $\Theta$ which do not belong to $M$. The modified coverage of the rule (5.1) for table $T$ is equal to $N^M(T\beta_m)$.

- The \textit{miscoverage} $mc(T, \rho)$ for which $\psi^0(T) = N(T) - N_{med(T)}(T)$ and $F(x) = x$. The miscoverage of the rule (5.1) for table $T$ is equal to $N(T\beta_m) - N_{med(T\beta_m)}(T\beta_m)$.

- The \textit{relative miscoverage} $rmc(T, \rho)$ for which

$$
\psi^0(T) = (N(T) - N_{med(T)}(T))/N(T)
$$
and $F(x) = x$. The relative miscoverage of the rule (5.1) for table $T$ is equal to 

$$
(N(T\beta_m) - N_{\text{mod}}(T\beta_m))(T\beta_m)) / N(T\beta_m).
$$

We need to minimize length, miscoverage and relative miscoverage, and maximize coverage, relative coverage, and modified coverage. However, we will consider only algorithms for the minimization of cost functions. Therefore, instead of maximization of coverage $c$ we will minimize the negation of coverage $-c$ given by pair of functions $\psi^0(T) = -N_{\text{mod}}(T)(T)$ and $F(x) = x$. Similarly, instead of maximization of relative coverage $rc$ we will minimize the negation of relative coverage $-rc$ given by pair of functions $\psi^0(T) = -N_{\text{mod}}(T) / N(T)$ and $F(x) = x$. Instead of maximization of modified coverage $c_M$ we will minimize the negation of modified coverage $-c_M$ given by pair of functions $\psi^0(T) = -N^M(T)$ and $F(x) = x$. The cost functions $-c$, $-rc$, and $-c_M$ are strictly increasing cost functions.

For a given cost function $\psi$ and decision table $T$, we denote

$$
\text{Range}_{\psi}(T) = \{\psi(\Theta, \rho) : \Theta \in SE\text{P}(T), \rho \in DR(\Theta)\}.
$$

By $q_\psi(T)$ we denote the cardinality of the set $\text{Range}_{\psi}(T)$. It is easy to prove the following statement:

**Lemma 60** Let $T$ be a decision table with $n$ conditional attributes. Then

- $\text{Range}_l(T) \subseteq \{0, 1, \ldots, n\}$, $\text{Range}_{-c}(T) \subseteq \{0, -1, \ldots, -N(T)\}$,
- $\text{Range}_{mc}(T) \subseteq \{0, 1, \ldots, N(T)\}$, $\text{Range}_{-rc}(T) \subseteq \{-a/b : a, b \in \{0, 1, \ldots, N(T)\}, b > 0\}$,
- $\text{Range}_{-cm}(T) \subseteq \{0, -1, \ldots, -N(T)\}$, $\text{Range}_{rmc}(T) \subseteq \{a/b : a, b \in \{0, 1, \ldots, N(T)\}, b > 0\}$, and $q_l(T) \leq n + 1$, $q_{-c}(T) \leq N(T) + 1$, $q_{-rc}(T) \leq N(T)(N(T) + 1)$, $q_{-cm}(T) \leq N(T) + 1$, $q_{mc}(T) \leq N(T) + 1$, $q_{rmc}(T) \leq N(T)(N(T) + 1)$.  


5.1.2 Systems of Decision Rules

Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1,\ldots,f_n$ and $N(T)$ rows $r_1,\ldots,r_{N(T)}$, and $U$ be an uncertainty measure.

A system of decision rules for $T$ is an $N(T)$-tuple $S = (\rho_1,\ldots,\rho_{N(T)})$ where $\rho_1 \in DR(T,r_1),\ldots,\rho_{N(T)} \in DR(T,r_{N(T)})$. Let $\alpha \in \mathbb{R}_+$. The considered system is called a $(U,\alpha)$-system of decision rules for $T$ if $\rho_i \in DR_{U,\alpha}(T,r_i)$ for $i = 1,\ldots,N(T)$. This system is called a $U$-system of decision rules for $T$ if $\rho_i \in DR_U(T,r_i)$ for $i = 1,\ldots,N(T)$.

We now consider a notion of cost function for systems of decision rules. This is a function $f(T,S)$ which is defined on pairs $T,S$, where $T$ is a nonempty decision table and $S = (\rho_1,\ldots,\rho_{N(T)})$ is a system of decision rules for $T$, and has values from the set $\mathbb{R}$. This function is given by cost function for decision rules $\psi$ and a function $f : \mathbb{R}^2 \to \mathbb{R}$. The value $f(T,S)$ is equal to $f(\psi(T,\rho_1),\ldots,\psi(T,\rho_{N(T)}))$ where the value $f(x_1,\ldots,x_k)$, for any natural $k$, is defined by induction: $f(x_1) = x_1$ and, for $k > 2$, $f(x_1,\ldots,x_k) = f(f(x_1,\ldots,x_{k-1}),x_k)$.

The cost function for systems of decision rules $\psi_f$ is called strictly increasing if $\psi$ is strictly increasing cost function for decision rules and $f$ is an increasing function from $\mathbb{R}^2$ to $\mathbb{R}$, i.e., $f(x_1,y_1) \leq f(x_2,y_2)$ for any $x_1,x_2,y_1,y_2 \in \mathbb{R}$ such that $x_1 \leq x_2$ and $y_1 \leq y_2$.

For example, if $\psi \in \{l,-c,-rc,-c_M,mc,rmc\}$ and $f \in \{\text{sum}(x,y) = x + y,\text{max}(x,y)\}$ then $\psi_f$ is a strictly increasing cost function for systems of decision rules.

We now consider a notion of uncertainty for systems of decision rules. This is a function $U_g(T,S)$ which is defined on pairs $T,S$, where $T$ is a nonempty decision table and $S = (\rho_1,\ldots,\rho_{N(T)})$ is a system of decision rules for $T$, and has values from the set $\mathbb{R}$. This function is given by uncertainty measure $U$ and a function $g : \mathbb{R}^2 \to \mathbb{R}$. The value $U_g(T,S)$ is equal to $g(U(T,\rho_1),\ldots,U(T,\rho_{N(T)}))$ where the
value \( g(x_1, \ldots, x_k) \), for any natural \( k \), is defined by induction: \( g(x_1) = x_1 \) and, for \( k > 2 \), \( g(x_1, \ldots, x_k) = g(g(x_1, \ldots, x_{k-1}), x_k) \).

## 5.2 Inhibitory Rules and Systems of Rules

### 5.2.1 Inhibitory Rules

Let \( T \) be a nondegenerate decision table with \( n \) conditional attributes \( f_1, \ldots, f_n \) and \( r = (b_1, \ldots, b_n) \) be a row of \( T \). An *inhibitory rule over \( T \)* is an expression of the kind

\[
  f_{i_1} = a_1 \land \ldots \land f_{i_m} = a_m \rightarrow \neq t \tag{5.2}
\]

where \( f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\} \), and \( a_1, \ldots, a_m, t \) are numbers from \( \omega \). It is possible that \( m = 0 \). For the considered rule, we denote \( \beta_0 = \lambda \), and if \( m > 0 \) we denote \( \beta_j = (f_{i_1}, a_1) \ldots (f_{i_j}, a_j) \) for \( j = 1, \ldots, m \). We will say that the inhibitory rule (5.2) covers the row \( r \) if \( r \) belongs to \( T_{\beta_m} \), i.e., \( b_{i_1} = a_1, \ldots, b_{i_m} = a_m \).

An inhibitory rule (5.2) over \( T \) is called an *inhibitory rule for \( T \)* if \( t = \text{lcd}(T_{\beta_m}) \), and either \( m = 0 \), or \( m > 0 \) and, for \( j = 1, \ldots, m \), \( T_{\beta_{j-1}} \) is not incomplete relative to \( T \), and \( f_{i_j} \in E(T_{\beta_{j-1}}) \). An inhibitory rule (5.2) for \( T \) is called an *inhibitory rule for \( T \) and \( r \)* if it covers \( r \).

We denote by \( IR(T) \) the set of inhibitory rules for \( T \). By \( IR(T, r) \) we denote the set of inhibitory rules for \( T \) and \( r \).

Let \( W \) be a completeness measure and \( \alpha \in \mathbb{R}_+ \). An inhibitory rule (5.2) for \( T \) is called a \((W, \alpha)\)-inhibitory rule for \( T \) if \( W(T, T_{\beta_m}) \leq \alpha \) and, if \( m > 0 \), then \( W(T, T_{\beta_j}) > \alpha \) for \( j = 0, \ldots, m - 1 \). A \((W, \alpha)\)-inhibitory rule (5.2) for \( T \) is called a \((W, \alpha)\)-inhibitory rule for \( T \) and \( r \) if it covers \( r \).

We denote by \( IR_{W, \alpha}(T) \) the set of \((W, \alpha)\)-inhibitory rules for \( T \), and we denote by \( IR_{W, \alpha}(T, r) \) the set of \((W, \alpha)\)-inhibitory rules for \( T \) and \( r \).
An inhibitory rule (5.2) for $T$ is called a $W$-inhibitory rule for $T$ if there exists a nonnegative real number $\alpha$ such that (5.2) is a $(W, \alpha)$-inhibitory rule for $T$. An inhibitory rule (5.2) for $T$ and $r$ is called a $W$-inhibitory rule for $T$ and $r$ if there exists a nonnegative real number $\alpha$ such that (5.2) is a $(W, \alpha)$-inhibitory rule for $T$ and $r$.

We denote by $IR_W(T)$ the set of $W$-inhibitory rules for $T$. By $IR_W(T, r)$ we denote the set of $W$-inhibitory rules for $T$ and $r$.

We define completeness $W(T, \rho)$ of an inhibitory rule $\rho$ for $T$ relative to the table $T$ in the following way. Let $\rho$ be equal to (5.2). Then $W(T, \rho) = W(T, T\beta_m)$.

We now consider a notion of cost function for inhibitory rules. This is a function $\psi(T, \rho)$ which is defined on pairs $T, \rho$, where $T$ is a nonempty decision table and $\rho$ is an inhibitory rule for $T$, and has values from the set $\mathbb{R}$ of real numbers. Let us consider examples of cost functions for inhibitory rules. Let $\rho$ be the rule (5.2).

- The length $l(T, \rho) = l(\rho)$ which is equal to $m$.
- The coverage $c(T, \rho)$ which is equal to $N(T\beta_m) - N_{gcd(T,T\beta_m)}(T\beta_m)$.
- The relative coverage $rc(T, \rho)$ which is equal to
  \[
  \frac{(N(T\beta_m) - N_{gcd(T,T\beta_m)}(T\beta_m))}{N(T\beta_m)}.
  \]
- The miscoverage $mc(T, \rho)$ which is equal to $N_{gcd(T,T\beta_m)}(T\beta_m)$.
- The relative miscoverage $rmc(T, \rho)$ which is equal to
  \[
  \frac{N_{gcd(T,T\beta_m)}(T\beta_m)}{N(T\beta_m)}.
  \]

We need to minimize length, miscoverage and relative miscoverage, and maximize coverage and relative coverage. However, we will consider only algorithms for the
minimization of cost functions. Therefore, instead of maximization of coverage $c$ we will minimize the negation of coverage $-c$. Similarly, instead of maximization of relative coverage $rc$ we will minimize the negation of relative coverage $-rc$.

### 5.2.2 Systems of Inhibitory Rules

Let $T$ be a nondegenerate decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, and $W$ be a completeness measure.

A system of inhibitory rules for $T$ is an $N(T)$-tuple $S = (\rho_1, \ldots, \rho_{N(T)})$ where $\rho_1 \in IR(T,r_1), \ldots, \rho_{N(T)} \in IR(T,r_{N(T)})$. Let $\alpha \in \mathbb{R}_+$. The considered system is called a $(W,\alpha)$-system of inhibitory rules for $T$ if $\rho_i \in IR_{W,\alpha}(T,r_i)$ for $i = 1, \ldots, N(T)$. This system is called a $W$-system of inhibitory rules for $T$ if $\rho_i \in IR_W(T,r_i)$ for $i = 1, \ldots, N(T)$.

We now consider a notion of cost function for systems of inhibitory rules. This is a function $\psi_f(T,S)$ which is defined on pairs $T,S$, where $T$ is a nonempty decision table and $S = (\rho_1, \ldots, \rho_{N(T)})$ is a system of inhibitory rules for $T$, and has values from the set $\mathbb{R}$. This function is given by cost function for decision rules $\psi$ and a function $f : \mathbb{R}^2 \to \mathbb{R}$. The value $\psi_f(T,S)$ is equal to $f(\psi(T,\rho_1), \ldots, \psi(T,\rho_{N(T)}))$ where the value $f(x_1, \ldots, x_k)$, for any natural $k$, is defined by induction: $f(x_1) = x_1$ and, for $k > 2$, $f(x_1, \ldots, x_k) = f(f(x_1, \ldots, x_{k-1}), x_k)$. Later we consider only cases when $\psi \in \{l,-c,-rc,mc,rmc\}$ and $f \in \{\text{sum}(x,y) = x + y, \text{max}(x,y)\}$.

We now consider a notion of completeness for systems of inhibitory rules. This is a function $W_g(T,S)$ which is defined on pairs $T,S$, where $T$ is a nonempty decision table and $S = (\rho_1, \ldots, \rho_{N(T)})$ is a system of inhibitory rules for $T$, and has values from the set $\mathbb{R}$. This function is given by completeness measure $W$ and a function $g : \mathbb{R}^2 \to \mathbb{R}$. The value $W_g(T,S)$ is equal to

$$g(W(T,\rho_1), \ldots, W(T,\rho_{N(T)}))$$
where the value $g(x_1, \ldots, x_k)$, for any natural $k$, is defined by induction: $g(x_1) = x_1$ and, for $k > 2$, $g(x_1, \ldots, x_k) = g(g(x_1, \ldots, x_{k-1}), x_k)$. Later we consider only cases when $g \in \{\text{sum}(x, y) = x + y, \text{max}(x, y)\}$.

Let $\rho_1$ be an inhibitory rule over $T$ and $\rho_2$ be a decision rule over $T^C$. We denote by $\rho_1^+$ a decision rule over $T^C$ obtained from $\rho_1$ by changing the right hand side of $\rho_1$: if the right hand side of $\rho_1$ is $\neq t$ than the right hand side of $\rho_1^+$ is $t$. We denote by $\rho_2^-$ an inhibitory rule over $T$ obtained from $\rho_2$ by changing the right hand side of $\rho_2$: if the right hand side of $\rho_2$ is $t$ than the right hand side of $\rho_2^-$ is $\neq t$. It is clear that $(\rho_1^-)^+ = \rho_1$ and $(\rho_2^-)^+ = \rho_2$. Let $A$ be a set of inhibitory rules over $T$. We denote $A^+ = \{\rho^+ : \rho \in A\}$. Let $B$ be a set of decision rules over $T^C$. We denote $B^- = \{\rho^- : \rho \in B\}$. It is clear that $(A^+)^- = A$ and $(B^-)^+ = B$.

**Proposition 61** Let $T$ be a nondegenerate decision table with attributes $f_1, \ldots, f_n$, $r$ be a row of $T$, $\rho$ be a decision rule over $T^C$, $W$ be a completeness measure, $U$ be an uncertainty measure, $W$ and $U$ are dual, and $\alpha \in \mathbb{R}_+$. Then

1. $\rho \in \text{DR}(T^C)$ if and only if $\rho^- \in \text{IR}(T)$;
2. $\rho \in \text{DR}_{U,\alpha}(T^C)$ if and only if $\rho^- \in \text{IR}_{W,\alpha}(T)$;
3. $\rho \in \text{DR}_U(T^C)$ if and only if $\rho^- \in \text{IR}_W(T)$;
4. $\rho \in \text{DR}(T^C, r)$ if and only if $\rho^- \in \text{IR}(T, r)$;
5. $\rho \in \text{DR}_{U,\alpha}(T^C, r)$ if and only if $\rho^- \in \text{IR}_{W,\alpha}(T, r)$;
6. $\rho \in \text{DR}_U(T^C, r)$ if and only if $\rho^- \in \text{IR}_W(T, r)$;
7. $W(T, \rho^-) = U(T^C, \rho)$;
8. If $\rho \in \text{DR}(T^C)$ then $\psi(T, \rho^-) = \psi(T^C, \rho)$ for any $\psi \in \{l, -c, -rc, mc, rmc\}$. 
Proof. Let \( \rho^- \) be the rule (5.2). Since \( W \) and \( U \) are dual, \( W(T, T\beta_j) = U(T^C\beta_j) \) for \( j = 0, \ldots, m \). From Lemma 37 it follows that \( T\beta_j \) is incomplete relative to \( T \) if and only if \( T^C\beta_j \) is degenerate for \( j = 0, \ldots, m \). It is clear that \( E(T\beta_j) = E(T^C\beta_j) \) for \( j = 0, \ldots, m \). By Lemma 37, \( \text{mcd}(T\beta_m) = \text{lcd}(T, T\beta_m) \). It is clear that \( \rho \) covers \( r \) if and only if \( \rho^- \) covers \( r \). Using these facts it is not difficult to show that the statements 1-7 of the considered proposition hold.

Let \( \rho \in DR(T^C) \). It is clear that \( l(T, \rho^-) = l(T^C, \rho) \). Set \( \beta = \beta_m \) and \( d = \text{mcd}(T^C\beta) = \text{lcd}(T, T\beta) \). By Lemma 37, \( N(T\beta) = N(T^C\beta) \) and \( N_d(T\beta) = N(T^C\beta) - N_d(T^C\beta) \). Therefore \( -N(T\beta) + N_d(T\beta) = -N_d(T^C\beta) \). As a result, we have

\[
-c(T, \rho^-) = -N(T\beta) + N_d(T\beta) = -N_d(T^C\beta) = -c(T^C, \rho),
\]

\[
-rc(T, \rho^-) = (-N(T\beta) + N_d(T\beta))/N(T\beta) = -N_d(T^C\beta)/N(T^C\beta) = -rc(T^C, \rho),
\]

\[
mc(T, \rho^-) = N_d(T\beta) = N(T^C\beta) - N_d(T^C\beta) = mc(T^C, \rho),
\]

\[
rmc(T, \rho^-) = N_d(T\beta)/N(T\beta) = (N(T^C\beta) - N_d(T^C\beta))/N(T^C\beta) = rmc(T^C, \rho).
\]

Therefore the statement 8 of the considered proposition holds. ■

Corollary 62 Let \( T \) be a nondegenerate decision table, \( r \) be a row of \( T \), \( W \) be a completeness measure, \( U \) be an uncertainty measure, \( W \) and \( U \) are dual, and \( \alpha \in \mathbb{R}_+ \). Then

1. \( IR(T) = DR(T^C)^-; \)

2. \( IR_{W,\alpha}(T) = DR_{U,\alpha}(T^C)^-; \)

3. \( IR_W(T) = DR_U(T^C)^-. \)
4. $IR(T, r) = DR(T^C, r)^-$;

5. $IR_{W, \alpha}(T, r) = DR_{U, \alpha}(T^C, r)^-$;

6. $IR_W(T, r) = DR_{U}(T^C, r)^-.$

**Proof.** Let $\rho_1 \in DR(T^C)$. Then, by the statement 1 of Proposition 61, $\rho_1^- \in IR(T)$. Therefore $DR(T^C)^- \subseteq IR(T)$. Let $\rho_2 \in IR(T)$. Then $\rho_2^+$ is a decision rule over $T^C$ and $\rho_2 = (\rho_2^+)^-$. By the statement 1 of Proposition 61, $\rho_2^+ \in DR(T^C)$. Therefore $\rho_2 \in DR(T^C)^-$ and $DR(T^C)^- \supseteq IR(T)$. Hence the statement 1 of the corollary holds. The statements 2-6 can be proven in similar way. $\blacksquare$
Chapter 6

Multi-stage Optimization of Decision and Inhibitory Rules

In this chapter, we consider optimization of decision and inhibitory rules including multi-stage optimization relative to a sequence of cost functions supported by experimental results. We discuss algorithms for counting the number of optimal rules. We also consider simulation of a greedy algorithm for construction of decision rule set, and applications of this approach to knowledge representation.

6.1 Multi-stage Optimization of Decision Rules

In this section, we concentrate on optimization of decision rules.

6.1.1 Representation of the Set of \((U, \alpha)\)-Decision Rules

Let \(T\) be a nonempty decision table with \(n\) conditional attributes \(f_1, \ldots, f_n\), \(U\) be an uncertainty measure, \(\alpha \in \mathbb{R}_+\), and \(G\) be a proper subgraph of \(\Delta_{U,\alpha}(T)\) (it is possible that \(G = \Delta_{U,\alpha}(T)\)).

Let \(\tau\) be a directed path from a node \(\Theta\) of \(G\) to a terminal node \(\Theta'\) in which edges (in the order from \(\Theta\) to \(\Theta'\)) are labeled with pairs \((f_{i_1}, c_{i_1}), \ldots, (f_{i_m}, c_{i_m})\), and
$t = mcd(\Theta')$. We denote by $rule(\tau)$ the decision rule over $T$

$$f_{i_1} = c_{i_1} \land \ldots \land f_{i_m} = c_{i_m} \rightarrow t.$$ 

If $m = 0$ (if $\Theta = \Theta'$) then the rule $rule(\tau)$ is equal to $\rightarrow t$.

Let $r = (b_1, \ldots, b_n)$ be a row of $T$, and $\Theta$ be a node of $G$ (subtable of $T$) containing the row $r$. We denote by $Rule(G, \Theta, r)$ the set of rules $rule(\tau)$ corresponding to all directed paths $\tau$ from $\Theta$ to terminal nodes $\Theta'$ containing $r$.

**Proposition 63** Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, and $\Theta$ be a node of the graph $\Delta_{U,\alpha}(T)$ containing $r$. Then the set $Rule(\Delta_{U,\alpha}(T), \Theta, r)$ coincides with the set of all $(U, \alpha)$-decision rules for $\Theta$ and $r$, i.e., $Rule(\Delta_{U,\alpha}(T), \Theta, r) = DR_{U,\alpha}(\Theta, r)$.

**Proof.** From the definition of the graph $\Delta_{U,\alpha}(T)$ it follows that each rule from $Rule(\Delta_{U,\alpha}, \Theta, r)$ is a $(U, \alpha)$-decision rule for $\Theta$ and $r$.

Let us consider an arbitrary $(U, \alpha)$-decision rule $\rho$ for $\Theta$ and $r$:

$$f_{i_1} = b_{i_1} \land \ldots \land f_{i_m} = b_{i_m} \rightarrow t.$$ 

It is easy to show that there is a directed path $\Theta_0 = \Theta, \Theta_1, \ldots, \Theta_m$ in $\Delta_{U,\alpha}(T)$ such that, for $j = 1, \ldots, m$, $\Theta_j = \Theta(f_{i_1}, b_{i_1}) \ldots (f_{i_j}, b_{i_j})$, there is an edge from $\Theta_{j-1}$ to $\Theta_j$ labeled with $(f_{i_j}, b_{i_j})$, and $\Theta_m$ is a terminal node in $\Delta_{U,\alpha}(T)$. Therefore $\rho \in Rule(\Delta_{U,\alpha}, \Theta, r)$. 

### 6.1.2 Procedure of Optimization

We describe now a procedure of optimization (minimization of cost) of rules for row $r = (b_1, \ldots, b_n)$ relative to a strictly increasing cost function $\psi$ given by pair of
functions $\psi^0$ and $F$. We will move from terminal nodes of the graph $G$ to the node $T$. We will attach to each node $\Theta$ of the graph $G$ containing $r$ the minimum cost $c(\Theta, r)$ of a rule from $\text{Rule}(G, \Theta, r)$ and, probably, we will remove some bundles of edges starting in nonterminal nodes. As a result we obtain a proper subgraph $G^\psi = G^\psi(r)$ of the graph $G$.

Algorithm $\mathcal{A}_4$

Input: A proper subgraph $G$ of the graph $\Delta_{U,\alpha}(T)$ for some decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$, uncertainty measure $U$, and a number $\alpha \in \mathbb{R}_+$, a row $r = (b_1, \ldots, b_n)$ of $T$, and a strictly increasing cost function $\psi$ for decision rules given by pair of functions $\psi^0$ and $F$.

Output: The proper subgraph $G^\psi = G^\psi(r)$ of the graph $G$.

1. If all nodes of the graph $G$ containing $r$ are processed then return the obtained graph as $G^\psi$ and finish the work of the algorithm. Otherwise, choose a node $\Theta$ of the graph $G$ containing $r$ which is not processed yet and which is either a terminal node of $G$ or a nonterminal node of $G$ for which all children containing $r$ are processed.

2. If $\Theta$ is a terminal node then set $c(\Theta, r) = \psi^0(\Theta)$, mark node $\Theta$ as processed and proceed to step 1.

3. If $\Theta$ is a nonterminal node then, for each $f_i \in E_G(\Theta)$, compute the value $c(\Theta, r, f_i) = F(c(\Theta(f_i, b_i), r))$ and set $c(\Theta, r) = \min\{c(\Theta, r, f_i) : f_i \in E_G(\Theta)\}$. Remove all $f_i$-bundles of edges starting from $\Theta$ for which $c(\Theta, r) < c(\Theta, r, f_i)$. Mark the node $\Theta$ as processed and proceed to step 1.

**Proposition 64** Let $G$ be a proper subgraph of the graph $\Delta_{U,\alpha}(T)$ for some decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$, uncertainty measure $U$, and a number $\alpha \in \mathbb{R}_+$, $r = (b_1, \ldots, b_n)$ be a row of $T$, and $\psi$ be a strictly increasing cost function
for decision rules given by pair of functions $\psi^0$ and $F$. Then, to construct the graph $G^\psi = G^\psi(r)$, the algorithm $A_4$ makes

$$O(nL(G))$$

elementary operations (computations of $\psi^0$, $F$, and comparisons).

**Proof.** In each terminal node of the graph $G$, the algorithm $A_4$ computes the value of $\psi^0$. In each nonterminal node of $G$, the algorithm $A_4$ computes the value of $F$ at most $n$ times and makes at most $2n$ comparisons. Therefore the algorithm $A_4$ makes

$$O(nL(G))$$

elementary operations. ■

**Proposition 65** Let $\mathcal{U}$ be a restricted information system and $\psi \in \{l, -c, -rc, -c_M, mc, rmc\}$. Then the algorithm $A_4$ has polynomial time complexity for decision tables from $\mathcal{T}(\mathcal{U})$ depending on the number of conditional attributes in these tables.

**Proof.** Since $\psi \in \{l, -c, -rc, -c_M, mc, rmc\}$, $\psi^0$ is one of the functions 0, $-N_{\text{med}(T)}(T)$, $-N_{\text{med}(T)}(T)/N(T)$, $-N^M(T)$, $N(T) - N_{\text{med}(T)}(T)$, and $(N(T) - N_{\text{med}(T)}(T))/N(T)$, and $F$ is either $x$ or $x + 1$. Therefore the elementary operations used by the algorithm $A_4$ are either basic numerical operations or computations of numerical parameters of decision tables which have polynomial time complexity depending on the size of decision tables. From Proposition 64 it follows that the number of elementary operations is bounded from above by a polynomial depending on the size of input table $T$ and on the number of separable subtables of $T$.

According to Proposition 46, the algorithm $A_4$ has polynomial time complexity for decision tables from $\mathcal{T}(\mathcal{U})$ depending on the number of conditional attributes in these tables. ■
Theorem 66 Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, $G$ be a proper subgraph of the graph $\Delta_{V,\alpha}(T)$, and $\psi$ be a strictly increasing cost function given by pair of functions $\psi^0$ and $F$. Then, for any node $\Theta$ of the graph $G^\psi = G^\psi(r)$ containing the row $r$, $c(\Theta, r) = \min\{\psi(\Theta, \rho) : \rho \in \text{Rule}(G, \Theta, r)\}$ and the set $\text{Rule}(G^\psi, \Theta, r)$ coincides with set of rules from $\text{Rule}(G, \Theta, r)$ that have minimum cost relative to $\psi$.

Proof. We prove this theorem by induction on nodes of $G^\psi$ containing $r$. If $\Theta$ is a terminal node containing $r$ then $\text{Rule}(G^\psi, \Theta, r) = \text{Rule}(G, \Theta, r) = \{ \rightarrow \text{mcd}(\Theta) \}$ and $c(\Theta, r) = \psi^0(\Theta) = \psi(\Theta, \rightarrow \text{mcd}(\Theta))$. Therefore the statement of theorem holds for $\Theta$.

Let $\Theta$ be a nonterminal node containing $r$ such that, for each child of $\Theta$ containing $r$, the statement of theorem holds. It is clear that

$$\text{Rule}(G, \Theta, r) = \bigcup_{f_i \in E_G(\Theta)} \text{Rule}(G, \Theta, r, f_i)$$

where, for $f_i \in E_G(\Theta)$,

$$\text{Rule}(G, \Theta, r, f_i) = \{ f_i = b_i \land \gamma \rightarrow t : \gamma \rightarrow t \in \text{Rule}(G, \Theta(f_i, b_i), r) \}.$$

By induction hypothesis, for any $f_i \in E_G(\Theta)$, the minimum cost of rule from $\text{Rule}(G, \Theta(f_i, b_i), r)$ is equal to $c(\Theta(f_i, b_i), r)$. Since $\psi$ is a strictly increasing cost function, the minimum cost of a rule from the set $\text{Rule}(G, \Theta, r, f_i)$ is equal to $F(c(\Theta(f_i, b_i), r) = c(\Theta, r, f_i)$. Therefore $c(\Theta, r) = \min\{c(\Theta, r, f_i) : f_i \in E_G(\Theta)\}$ is the minimum cost of rule from $\text{Rule}(G, \Theta, r)$. Set $q = c(\Theta, r)$.

Let $f_i = b_i \land \gamma \rightarrow t$ be a rule from $\text{Rule}(G, \Theta, r)$ which cost is equal to $q$. It is clear that $G$ contains the node $\Theta(f_i, b_i)$ and the edge $e$ which starts in $\Theta$, en-
ters $\Theta(f_i, b_i)$, and is labeled with $(f_i, b_i)$. Let $p$ be the minimum cost of a rule from $\text{Rule}(G, \Theta(f_i, b_i), r)$, i.e., $p = c(\Theta(f_i, b_i), r)$. The rule $\gamma \rightarrow t$ belongs to the set $\text{Rule}(G, \Theta(f_i, b_i), r)$ and, since $\psi$ is strictly increasing, the cost of $\gamma \rightarrow t$ is equal to $p$ (otherwise, the minimum cost of a rule from $\text{Rule}(G, \Theta, r)$ is less than $q$). Therefore $F(p) = q$, and the edge $e$ belongs to the graph $G^\psi$. By the induction hypothesis, the set $\text{Rule}(G^\psi, \Theta(f_i, b_i), r)$ coincides with the set of rules from $\text{Rule}(G, \Theta(f_i, b_i), r)$ which cost is equal to $p$. From here it follows that $\gamma \rightarrow t$ belongs to $\text{Rule}(G^\psi, \Theta(f_i, b_i), r)$ and $f_i = b_i \land \gamma \rightarrow t$ belongs to $\text{Rule}(G^\psi, \Theta, r)$.

Let $f_i = b_i \land \gamma \rightarrow t$ belong to $\text{Rule}(G^\psi, \Theta, r)$. Then $\Theta(f_i, b_i)$ is a child of $\Theta$ in the graph $G^\psi$, $\gamma \rightarrow t$ belongs to $\text{Rule}(G^\psi, \Theta(f_i, b_i), r)$ and, by the induction hypothesis, the set $\text{Rule}(G^\psi, \Theta(f_i, b_i), r)$ coincides with the set of rules from $\text{Rule}(G, \Theta(f_i, b_i), r)$ which cost is equal to $p$ – the minimum cost of a rule from $\text{Rule}(G, \Theta(f_i, b_i), r)$. From the description of the procedure of optimization and from the fact that $\Theta(f_i, b_i)$ is a child of $\Theta$ in the graph $G^\psi$ it follows that $F(p) = q$. Therefore, the cost of rule $f_i = b_i \land \gamma \rightarrow t$ is equal to $q$. ❑

We can make sequential optimization of $(U, \alpha)$-rules for $T$ and $r$ relative to a sequence of strictly increasing cost functions $\psi_1, \psi_2, \ldots$ for decision rules. We begin from the graph $G = \Delta_{U, \alpha}(T)$ and apply to it the procedure of optimization (algorithm $\mathcal{A}_i$) relative to the cost function $\psi_1$. As a result, we obtain a proper subgraph $G^\psi_1 = G^\psi_1(r)$ of the graph $G$. By Proposition 63, the set $\text{Rule}(G, T, r)$ is equal to the set of all $(U, \alpha)$-rules for $T$ and $r$. From here and from Theorem 66 it follows that the set $\text{Rule}(G^\psi_1, T, r)$ is equal to the set of all $(U, \alpha)$-rules for $T$ and $r$ which have minimum cost relative to $\psi_1$. If we apply to the graph $G^\psi_1$ the procedure of optimization relative to the cost function $\psi_2$ we obtain a proper subgraph $G^{\psi_1, \psi_2} = G^{\psi_1, \psi_2}(r)$ of the graph $G^{\psi_1}$. The set $\text{Rule}(G^{\psi_1, \psi_2}, T, r)$ is equal to the set of all rules from the set $\text{Rule}(G^{\psi_1, \psi_2}, T, r)$ which have minimum cost relative to $\psi_2$, etc.

We described the work of optimization procedure for one row. If we would like
to work with all rows in parallel, then instead of removal of bundles of edges we will change the list of bundles attached to row. We begin from the graph $G = \Delta_{U,\alpha}(T)$. In this graph, for each nonterminal node $\Theta$, each row $r$ of $\Theta$ is labeled with the set of attributes $E(G, \Theta, r) = E(\Theta)$. It means that, for the row $r$, we consider only $f_i$-bundles of edges starting from $\Theta$ such that $f_i \in E(G, \Theta, r)$. During the work of the procedure of optimization relative to a cost function $\psi$ we will not change the “topology” of the graph $G$ but will change sets $E(G, \Theta, r)$ attached to rows $r$ of nonterminal nodes $\Theta$. In a new graph $G^\psi$ (we will say about this graph as about labeled proper subgraph of the graph $G$), for each nonterminal node $\Theta$, each row $r$ of $\Theta$ is labeled with a subset $E(G^\psi, \Theta, r)$ of the set $E(G, \Theta, r)$ containing only attributes for which corresponding bundles were not removed during the optimization relative to $\psi$ for the row $r$.

We can study also totally optimal decision rules relative to various combinations of cost functions. For a cost function $\psi$, we denote $\psi^{U,\alpha}(T, r) = \min\{\psi(T, \rho) : \rho \in DR_{U,\alpha}(T, r)\}$, i.e., $\psi^{U,\alpha}(T, r)$ is the minimum cost of a $(U, \alpha)$-decision rule for $T$ and $r$ relative to the cost function $\psi$. Let $\psi_1, \ldots, \psi_m$ be cost functions and $m \geq 2$. A $(U, \alpha)$-decision rule $\rho$ for $T$ and $r$ is called a totally optimal $(U, \alpha)$-decision rule for $T$ and $r$ relative to the cost functions $\psi_1, \ldots, \psi_m$ if $\psi_1(T, \rho) = \psi_1^{U,\alpha}(T, r), \ldots, \psi_m(T, \rho) = \psi_m^{U,\alpha}(T, r)$, i.e., $\rho$ is optimal relative to $\psi_1, \ldots, \psi_m$ simultaneously.

Assume that $\psi_1, \ldots, \psi_m$ are strictly increasing cost functions for decision rules. We now describe how to recognize the existence of a $(U, \alpha)$-decision rule for $T$ and $r$ which is a totally optimal $(U, \alpha)$-decision rule for $T$ and $r$ relative to the cost functions $\psi_1, \ldots, \psi_m$.

First, we construct the graph $G = \Delta_{U,\alpha}(T)$ using the Algorithm $A_1$. For $i = 1, \ldots, m$, we apply to $G$ and $r$ the procedure of optimization relative to $\psi_i$ (the Algorithm $A_4$). As a result, we obtain, for $i = 1, \ldots, m$, the graph $G^{\psi_i}(r)$ and the number $\psi_i^{U,\alpha}(T, r)$ attached to the node $T$ of $G^{\psi_i}(r)$. Next, we apply to $G$ sequentially the
procedures of optimization relative to the cost functions $\psi_1, \ldots, \psi_m$. As a result, we obtain graphs $G^{\psi_1}(r), G^{\psi_1, \psi_2}(r), \ldots, G^{\psi_1, \ldots, \psi_m}(r)$ and numbers $\varphi_1, \varphi_2, \ldots, \varphi_m$ attached to the node $T$ of these graphs. One can show that a totally optimal $(U, \alpha)$-decision rule for $T$ and $r$ relative to the cost functions $\psi_1, \ldots, \psi_m$ exists if and only if $\varphi_i = \psi_i^{U, \alpha}(T, r)$ for $i = 1, \ldots, m$.

6.1.3 Number of Rules in $\text{Rule}(G, \Theta, r)$

Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, and $G$ be a proper subgraph of the graph $\Delta_{U, \alpha}(T)$. We describe now an algorithm which counts, for each node $\Theta$ of the graph $G$ containing $r$, the cardinality $C(\Theta, r)$ of the set $\text{Rule}(G, \Theta, r)$, and returns the number $C(T, r) = |\text{Rule}(G, T, r)|$.

Algorithm $A_5$

Input: A proper subgraph $G$ of the graph $\Delta_{U, \alpha}(T)$ for some decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$, uncertainty measure $U$, and number $\alpha \in \mathbb{R}_+$, and row $r = (b_1, \ldots, b_n)$ of the table $T$.

Output: The number $|\text{Rule}(G, T, r)|$.

1. If all nodes of the graph $G$ containing $r$ are processed then return the number $C(T, r)$ and finish the work of the algorithm. Otherwise, choose a node $\Theta$ of the graph $G$ containing $r$ which is not processed yet and which is either a terminal node of $G$ or a nonterminal node of $G$ such that, for each $f_i \in E_G(T)$ and $a_j \in E(\Theta, f_i)$, the node $\Theta(f_i, a_j)$ is processed.

2. If $\Theta$ is a terminal node then set $C(\Theta, r) = 1$, mark the node $\Theta$ as processed, and proceed to step 1.
3. If $\Theta$ is a nonterminal node then set
\[
C(\Theta, r) = \sum_{f_i \in E_G(\Theta)} C(\Theta(f_i, b_i), r),
\]
mark the node $\Theta$ as processed, and proceed to step 1.

**Proposition 67** Let $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, $T$ be a decision table with $n$ attributes $f_1, \ldots, f_n$, $G$ be a proper subgraph of the graph $\Delta_{U,\alpha}(T)$, and $r = (b_1, \ldots, b_n)$ be a row of the table $T$. Then the algorithm $A_5$ returns the number $|\text{Rule}(G, T, r)|$ and makes at most $nL(G)$ operations of addition.

**Proof.** We prove by induction on the nodes of $G$ that $C(\Theta, r) = |\text{Rule}(G, \Theta, r)|$ for each node $\Theta$ of $G$ containing $r$. Let $\Theta$ be a terminal node of $G$. Then $\text{Rule}(G, \Theta, r) = \{\rightarrow \text{mcd}(\Theta)\}$ and $|\text{Rule}(G, \Theta, r)| = 1$. Therefore the considered statement holds for $\Theta$. Let now $\Theta$ be a nonterminal node of $G$ such that the considered statement holds for its children containing $r$. It is clear that
\[
\text{Rule}(G, \Theta, r) = \bigcup_{f_i \in E_G(\Theta)} \text{Rule}(G, \Theta, r, f_i)
\]
where, for each $f_i \in E_G(\Theta)$,
\[
\text{Rule}(G, \Theta, r, f_i) = \{f_i = b_i \land \gamma \rightarrow t : \gamma \rightarrow t \in \text{Rule}(G, \Theta(f_i, b_i), r)\}
\]
and $|\text{Rule}(G, \Theta, r, f_i)| = |\text{Rule}(G, \Theta(f_i, b_i), r)|$. Therefore
\[
|\text{Rule}(G, \Theta, r)| = \sum_{f_i \in E_G(\Theta)} |\text{Rule}(G, \Theta(f_i, b_i), r)|.
\]
By the induction hypothesis, $C(\Theta(f_i, b_i), r) = |\text{Rule}(G, \Theta(f_i, b_i), r)|$ for any $f_i \in E_G(\Theta)$. Therefore $C(\Theta, r) = |\text{Rule}(G, \Theta, r)|$. Hence, the considered statement holds.
From here it follows that $C(T, r) = |\text{Rule}(G, T, r)|$, and the algorithm $A_5$ returns the cardinality of the set $\text{Rule}(G, T, r)$.

It is easy to see that the considered algorithm makes at most $nL(G)$ operations of addition where $L(G)$ is the number of nodes in the graph $G$. ■

**Proposition 68** Let $U$ be a restricted information system. Then the algorithm $A_5$ has polynomial time complexity for decision tables from $\mathcal{T}(U)$ depending on the number of conditional attributes in these tables.

**Proof.** All operations made by the algorithm $A_5$ are basic numerical operations (additions). From Proposition 67 it follows that the number of these operations is bounded from above by a polynomial depending on the size of input table $T$ and on the number of separable subtables of $T$.

According to Proposition 46, the algorithm $A_5$ has polynomial time complexity for decision tables from $\mathcal{T}(U)$ depending on the number of conditional attributes in these tables. ■

### 6.1.4 Experimental Results for Decision Rule Optimization

In this section, we consider results of experiments involving decision tables with many-valued decisions. We consider 9 decision tables with many-valued decisions (See Table 6.1) derived from usual decision tables from the UCI ML Repository [40]. For example, the “breast-cancer-5” decision table is obtained from the “breast-cancer” data set by the removal of 5 conditional attributes. The positions of these attributes in the original dataset can be found under the column “Removed Attributes” of Table 6.1. The resulting table contains groups of equal rows possibly with different decisions.

We keep a single row from each group and label it with the set of all decisions attached to the rows in its group. As a result, we obtain a decision table $T$ with many-valued decisions. We will not remove from $T$ rows $r$ such that $D(r) = D(T)$. The number
of rows and attributes in these tables can be found in the columns “Rows” and “Attributes”, respectively.

**Table 6.1:** Modified decision tables from UCI ML Repository

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Rows</th>
<th>Attributes</th>
<th>Removed Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>4,5,6,8,9</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>4078</td>
<td>17</td>
<td>5,8,11,13,22</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
</tr>
</tbody>
</table>

For each decision table $T$ in Table 6.1, we construct by the algorithm $A_1$ the directed acyclic graph $\Delta(T) = \Delta_{U,0}(T)$ for some uncertainty measure $U$. Table 6.2 shows minimum, average and maximum (column “Dynamic programming”) length of rules (among all rows of $T$) obtained as a result of applying to $\Delta(T)$ the procedure of optimization relative to the length (see the algorithm $A_1$). One can see that we obtained short on average rules for each data set. For example, “nursery-1” has 7 conditional attributes whereas on average the system of rules has 2 conditions per rule after applying the procedure of optimization relative to the length. In similar way, we apply to $\Delta(T)$ the procedure of optimization relative to the coverage. Minimum, average and maximum coverage of obtained rules (among all rows of $T$) can be found in Table 6.3 (column “Dynamic programming”). We should mention good results obtained for decision tables “cars-1”, “mushroom-5”, “nursery-1” and “nursery-4”.

We applied also to the directed acyclic graph $\Delta(T)$ sequentially the procedure of optimization relative to the coverage, and then the procedure of optimization relative to the length. Average length and coverage of obtained rules (among all rows of $T$) can be found in Table 6.4 (column “coverage+length”). In the same way, we also present in Table 6.4 results of sequential optimization relative to the length, and then
Table 6.2: Length of decision rules: after the procedure relative to length

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Rows</th>
<th>Attributes</th>
<th>Dynamic programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>min</td>
</tr>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>4078</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.3: Coverage of decision rules: after the procedure relative to coverage

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Rows</th>
<th>Attr</th>
<th>Dynamic programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>min</td>
</tr>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>4078</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

relative to the coverage (column “length+coverage”). We used such results to count the number of rows for which there exist totally optimal decision rule relative to length and coverage (column “Rows tot”). For decision tables in bold in Table 6.4, the order of optimization relative to length and coverage do not matter since there exists a totally optimal decision rule for each row in those tables.
Table 6.4: Sequential optimization of decision rules

<table>
<thead>
<tr>
<th>Table Name</th>
<th>coverage+length</th>
<th>length+coverage</th>
<th>Rows</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>coverage</td>
<td>length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>breast-cancer-1</td>
<td>9.27</td>
<td>3.73</td>
<td>2.83</td>
<td>6.56</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>7.74</td>
<td>1.87</td>
<td>1.71</td>
<td>7.02</td>
</tr>
<tr>
<td>cars-1</td>
<td>110.19</td>
<td>1.37</td>
<td>1.37</td>
<td>110.19</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>741.32</td>
<td>2.28</td>
<td>1.49</td>
<td>512.95</td>
</tr>
<tr>
<td>nursery-1</td>
<td>620.84</td>
<td>2.03</td>
<td>2.03</td>
<td>620.84</td>
</tr>
<tr>
<td>nursery-4</td>
<td>59.92</td>
<td>1.33</td>
<td>1.33</td>
<td>59.92</td>
</tr>
<tr>
<td>teeth-1</td>
<td>1.0</td>
<td>2.23</td>
<td>2.23</td>
<td>1.0</td>
</tr>
<tr>
<td>teeth-5</td>
<td>1.0</td>
<td>1.93</td>
<td>1.93</td>
<td>1.0</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>7.17</td>
<td>2.26</td>
<td>2.1</td>
<td>7.02</td>
</tr>
</tbody>
</table>

6.1.5 Greedy Algorithm for Construction of Decision Rule Set

Let $U$ be an uncertainty measure, $\alpha, \beta \in \mathbb{R}_+, 0 \leq \beta \leq 1$, and $T$ be a decision table with $n$ conditional attributes $f_1, \ldots, f_n$. Let $S$ be a finite nonempty set of $(U, \alpha)$-decision rules for $T$. We will say that $S$ is a \( \beta \)-system of $(U, \alpha)$-decision rules for $T$ if rules from $S$ cover at least $(1 - \beta)N(T)$ rows of $T$.

We would like to construct a $\beta$-system of $(U, \alpha)$-decision rules for $T$ with minimum cardinality. Unfortunately, our approach does not allow us to do this. However, we can simulate the work of a greedy algorithm for the set cover problem (algorithm $A_6$).

The algorithm $A_6$ works with the graph $G = \Delta_{U, \alpha}(T)$. During each step, this algorithm constructs (based on algorithm $A_4$) a $(U, \alpha)$-rule for $T$ with minimum length among all $(U, \alpha)$-rules for $T$ which cover maximum number of uncovered previously rows. The algorithm finishes the work when at least $(1 - \beta)N(T)$ rows of $T$ are covered. We will call the constructed set a $(U, \alpha, \beta)$-greedy set of decision rules for $T$. Of course, the constructed set is a $\beta$-system of $(U, \alpha)$-decision rules for $T$. Using results of Slavik for the set cover problem [41, 42, 43] we obtain that the cardinality
of the constructed set of rules \( S \) is less than \( C_{\min}(U, \alpha, \beta, T)(\ln[(1 - \beta)N(T)] - \ln\ln[(1 - \beta)N(T)] + 0.78) \) where \( C_{\min}(U, \alpha, \beta, T) \) is the minimum cardinality of a \( \beta \)-system of \((U, \alpha)\)-decision rules for \( T \).

The study of algorithm \( A_6 \) was done jointly with Talha Amin who considered decision tables with single-valued decisions.

Let us recall that the cost function \(-c_M\) is given by the pair of functions \( \psi^0(T) = -N^M(T) \) and \( F(x) = x \), and the cost function \( l \) is given by the pair of functions \( \varphi^0(T) = 0 \) and \( H(x) = x + 1 \).

\textit{Algorithm \( A_6 \)}

\textit{Input:} A decision table \( T \) with \( n \) conditional attributes \( f_1, \ldots, f_n \), uncertainty measure \( U \), and numbers \( \alpha, \beta \in \mathbb{R}_+, 0 \leq \beta \leq 1 \).

\textit{Output:} A \((U, \alpha, \beta)\)-greedy set of decision rules for \( T \).

1. Set \( M = \emptyset \) and \( S = \emptyset \).

2. If \( |M| \geq (1 - \beta)N(T) \) then return \( S \) and finish the algorithm.

3. Apply the algorithm \( A_4 \) to each row of \( T \) two times: first, as the procedure of optimization of rules relative to the cost function \(-c_M\), and after that, as the procedure of optimization of rules relative to the cost function \( l \).

4. As a result, for each row \( r \) of \( T \), we obtain two numbers \(-c_M(r)\) which is the minimum cost of a rule from \( \text{Rule}(G, T, r) \) relative to the cost function \(-c_M\), and \( l(r) \) which is the minimum length among rules from \( \text{Rule}(G, T, r) \) which have minimum cost relative to the cost function \(-c_M\).

5. Choose a row \( r \) of \( T \) for which the value of \(-c_M(r)\) is minimum among all rows of \( T \), and the value of \( l(r) \) is minimum among all rows of \( T \) with minimum value of \(-c_M(r)\). In the graph \( G^{c_M,l} \), which is the result of the bi-stage procedure...
of optimization of rules for the row \( r \) relative to the cost functions \(-c_M\) and \( l\),
choose a directed path \( \tau \) from \( T \) to a terminal node of \( G^{-c_M,l} \) containing \( r \).

6. Add the rule \( \text{rule}(\tau) \) to the set \( S \), and add all rows covered by \( \text{rule}(\tau) \), which
do not belong to \( M \), to the set \( M \). Proceed to step 2.

**Proposition 69** Let \( T \) be a decision table with \( n \) conditional attributes, \( U \) be an
uncertainty measure, and \( \alpha, \beta \in \mathbb{R}_+, \, 0 \leq \beta \leq 1 \). Then the algorithm \( A_6 \) returns a
\((U, \alpha, \beta)\)-greedy set of decision rules for \( T \) and makes

\[
O(N(T)^2nL(G))
\]

elementary operations (computations of \( \psi^0, F, \varphi^0, H \), and comparisons).

**Proof.** Let us analyze one iteration of the algorithm \( A_6 \) (steps 3-6) and rule \( \text{rule}(\tau) \)
added at the step 6 to the set \( S \). Using Proposition 63 and Theorem 66 we obtain
that the rule \( \text{rule}(\tau) \) is a \((U, \alpha)\)-rule for \( T \) which covers the maximum number of
uncovered rows (rows which does not belong to \( M \)) and has minimum length among
such rules. Therefore the algorithm \( A_6 \) returns a \((U, \alpha, \beta)\)-greedy set of decision rules
for \( T \).

Let us analyze the number of elementary operations (computations of \( \psi^0, F, \varphi^0, H \),
and comparisons) which algorithm \( A_6 \) makes during one iteration. We know that
the algorithm \( A_4 \), under the bi-stage optimization of rules for one row, makes

\[
O(nL(G))
\]

elementary operations (computations of \( \psi^0, F, \varphi^0, H \), and comparisons). The number
of rows is equal to \( N(T) \). To choose a row \( r \) of \( T \) for which the value of \(-c_M(r)\) is
minimum among all rows of \( T \), and the value of \( l(r) \) is minimum among all rows of \( T \)
with minimum value of \(-c_M(r)\), the algorithm \( A_6 \) makes at most \( 2N(T) \) comparisons.
Therefore the number of elementary operations which algorithm $A_6$ makes during one iteration is $O(N(T)nL(G))$.

The number of iterations is at most $N(T)$. Therefore, during the construction of a $(U, \alpha, \beta)$-greedy set of decision rules for $T$, the algorithm $A_6$ makes

$$O(N(T)^2nL(G))$$

elementary operations (computations of $\psi^0$, $F$, $\varphi^0$, $H$, and comparisons). ■

**Proposition 70** Let $U$ be a restricted information system. Then the algorithm $A_6$ has polynomial time complexity for decision tables from $T(U)$ depending on the number of conditional attributes in these tables.

**Proof.** From Proposition 69 it follows that, for the algorithm $A_6$, the number of elementary operations (computations of $\psi^0$, $F$, $\varphi^0$, $H$, and comparisons) is bounded from above by a polynomial depending on the size of input table $T$ and on the number of separable subtables of $T$. The computations of numerical parameters of decision tables used by the algorithm $A_6$ (constant 0 and $-N^M(T)$) have polynomial time complexity depending on the size of decision tables. All operations with numbers are basic ones ($x$, $x + 1$, comparisons).

According to Proposition 46, the algorithm $A_6$ has polynomial time complexity for decision tables from $T(U)$ depending on the number of conditional attributes in these tables. ■

Using information based on the work of algorithm $A_6$, we can obtain lower bound on the parameter $C_{\min}(U, \alpha, \beta, T)$ which is the minimum cardinality of a $\beta$-system of $(U, \alpha)$-decision rules for $T$. During the construction of $(U, \alpha, \beta)$-greedy set of decision rules for $T$, let the algorithm $A_6$ choose consequently rules $\rho_1, \ldots, \rho_t$. Let $B_1, \ldots, B_t$ be sets of rows covered by rules $\rho_1, \ldots, \rho_t$, respectively. Set $B_0 = \emptyset$, $\delta_0 = 0$ and, for $i = 1, \ldots, t$, set $\delta_i = |B_i \setminus (B_0 \cup \ldots \cup B_{i-1})|$. The information derived from the
algorithm's $\mathcal{A}_6$ work consists of the tuple $(\delta_1, \ldots, \delta_t)$ and the numbers $N(T)$ and $\beta$.

From the results obtained in [44] regarding a greedy algorithm for the set cover problem it follows that

$$C_{\min}(U, \alpha, \beta, T) \geq \max \left\{ \left[ \frac{(1 - \beta)N(T)}{\delta_{i+1}} - (\delta_0 + \ldots + \delta_i) \right] : i = 0, \ldots, t - 1 \right\}.$$ 

### 6.1.6 Experimental Results for Greedy Algorithm

Let $T$ be a decision table with many-valued decisions, $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$. We denote by $Rule_{\alpha}^{\beta}(T)$ the $(rme, \alpha, \beta)$-greedy set of decision rules for $T$ constructed by the algorithm $\mathcal{A}_6$.

Table 6.5 presents information regarding set $Rule_{0}^{0}(T)$ for 9 decision tables $T$. The column “Bounds” contains the number of rules in $Rule_{0}^{0}(T)$ in the upper left and a lower bound on the cardinality of a 0-system of $(rme, 0)$-decision rules for $T$ obtained based on equation from Section 6.1.5 in the lower right. The column “Len” contains the average and maximum length of rules from $Rule_{0}^{0}(T)$ in the upper left and lower right sides, respectively. Finally, “Cov” contains the maximum support of rules from $Rule_{0}^{0}(T)$.

We consider a threshold of 30 as a reasonable bound on the number of rules for a system used for knowledge representation. In the cases where this threshold is exceeded, we consider $\beta$-systems $Rule_{\alpha}^{\beta}(T)$ of $(rme, \alpha)$-decision rules for $\beta \in \{0, 0.01, 0.05, 0.1, 0.15, 0.2\}$ and $\alpha \in \{0, 0.1, 0.3, 0.5\}$. The value $\alpha = 0.5$ is given only for completeness since $(rme, 0.5)$-decision rules are likely to be too inaccurate for practical use. Results of experiments show that for all four tables with many-valued decisions that exceeded the threshold, we can find combinations of $\beta$ and $\alpha$ such that the set of rules $Rule_{\alpha}^{\beta}(T)$ contains at most 30 rules. In Tables 6.6-6.9 we present, for each $\beta$ and $\alpha$, the cardinality of the system $Rule_{\alpha}^{\beta}(T)$ and a lower bound on the card-
Table 6.5: Bounds for decision tables with many-valued decisions

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Rows</th>
<th>Attr</th>
<th>Removed Attr</th>
<th>Bounds</th>
<th>Len</th>
<th>Cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>3</td>
<td>54</td>
<td>24</td>
<td>3.26 16</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>4,5,6,8,9</td>
<td>31</td>
<td>14</td>
<td>2.03 17</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
<td>17</td>
<td>7</td>
<td>2.53 144</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>4078</td>
<td>17</td>
<td>5,8,11,13,22</td>
<td>36</td>
<td>10</td>
<td>2.89 1080</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>1</td>
<td>71</td>
<td>17</td>
<td>3.87 1440</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
<td>9</td>
<td>4</td>
<td>1.89 80</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>2.23 1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
<td>14</td>
<td>14</td>
<td>1.93 3</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
<td>9</td>
<td>5</td>
<td>2.56 12</td>
</tr>
</tbody>
</table>

nality of a \( \beta \)-system of \( (rme,\alpha) \)-decision rules for \( T \), where \( T \) is one of the considered four decision tables. Similar results for association rules were obtained in [45].

Table 6.6: Bounds for decision table: breast-cancer-1

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>54</td>
<td>54</td>
<td>38</td>
<td>18</td>
</tr>
<tr>
<td>0.01</td>
<td>53</td>
<td>53</td>
<td>37</td>
<td>17</td>
</tr>
<tr>
<td>0.05</td>
<td>45</td>
<td>45</td>
<td>29</td>
<td>13</td>
</tr>
<tr>
<td>0.1</td>
<td>35</td>
<td>35</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>0.15</td>
<td>31</td>
<td>31</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>0.2</td>
<td>26</td>
<td>26</td>
<td>13</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 6.7: Bounds for decision table: breast-cancer-5

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>0.01</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>0.05</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>0.1</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>0.15</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>0.2</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 6.8: Bounds for decision table: mushroom-5

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>36</td>
<td>27</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>0.01</td>
<td>25</td>
<td>20</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>0.05</td>
<td>15</td>
<td>13</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>0.1</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>0.15</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>0.2</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
6.2 Multi-stage Optimization of Inhibitory Rules

In this section, we focus on optimization of inhibitory rules. Let $T$ be a nondegenerate nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r$ be a row of $T$, $T^C$ be the decision table complementary to $T$, $U$ be an uncertainty measure, $W$ be a completeness measure, $U$ is dual to $W$, and $\alpha \in \mathbb{R}_+$. 

Let $G$ be a proper subgraph of the graph $\Delta_{U,\alpha}(T^C)$. We correspond to the node $T^C$ of $G$ and row $r$ a set $\text{Rule}(G,T^C,r)$ of $(U,\alpha)$-decision rules for $T^C$ and $r$. If $G = \Delta_{U,\alpha}(T^C)$ then, by Proposition 63, $\text{Rule}(G,T^C,r) = \text{DR}_{U,\alpha}(T^C,r)$. In general case, $\text{Rule}(G,T^C,r) \subseteq \text{DR}_{U,\alpha}(T^C,r)$. Let us consider the set $\text{Rule}(G,T^C,r)^-$. By Corollary 62, $\text{IR}_{W,\alpha}(T,r) = \text{DR}_{U,\alpha}(T^C,r)^-$. Therefore, if $G = \Delta_{U,\alpha}(T^C)$ then

$$\text{Rule}(G,T^C,r)^- = \text{IR}_{W,\alpha}(T,r).$$

In general case, $\text{Rule}(G,T^C,r)^- \subseteq \text{DR}_{U,\alpha}(T^C,r)^- = \text{IR}_{W,\alpha}(T,r)$

In Section 6.1.2, the algorithm $A_4$ is considered which, for a strictly increasing cost function $\psi$ for decision rules, constructs the proper subgraph $G^\psi = G^\psi(r)$ of the graph
G. Let \( \psi \in \{l, -c, -rc, mc, rmc\} \). Then, by Theorem 66, the set \( \text{Rule}(G^\psi, T^C, r) \) is equal to the set of all rules from \( \text{Rule}(G, T^C, r) \) which have minimum cost relative to \( \psi \) among all decision rules from the set \( \text{Rule}(G, T^C, r) \). From Proposition 61 it follows that, for any \( \rho \in \text{Rule}(G, T^C, r) \), \( \psi(T^C, \rho) = \psi(T, \rho^-) \). Therefore, the set \( \text{Rule}(G^\psi, T^C, r)^- \) is equal to the set of all rules from \( \text{Rule}(G, T^C, r)^- \) which have minimum cost relative to \( \psi \) among all inhibitory rules from the set \( \text{Rule}(G, T^C, r)^- \).

We can make sequential optimization of \((W, \alpha)\)-inhibitory rules for \( T \) and \( r \) relative to a sequence of cost functions \( \psi_1, \psi_2, \ldots \) from \( \{l, -c, -rc, mc, rmc\} \). We begin from the graph \( G = \Delta_U(T^C) \) and apply to it the procedure of optimization (algorithm \( A_i \)) relative to the cost function \( \psi_1 \). As a result, we obtain a proper subgraph \( G^{\psi_1} = G^{\psi_1}(r) \) of the graph \( G \). The set \( \text{Rule}(G, T^C, r)^- \) is equal to the set of all \((W, \alpha)\)-inhibitory rules for \( T \) and \( r \). The set \( \text{Rule}(G^{\psi_1}, T^C, r)^- \) is equal to the set of all \((W, \alpha)\)-inhibitory rules for \( T \) and \( r \) which have minimum cost relative to \( \psi_1 \). If we apply to the graph \( G^{\psi_1} \) the procedure of optimization relative to the cost function \( \psi_2 \) we obtain a proper subgraph \( G^{\psi_1, \psi_2} = G^{\psi_1, \psi_2}(r) \) of the graph \( G^{\psi_1} \). The set \( \text{Rule}(G^{\psi_1, \psi_2}, T^C, r)^- \) is equal to the set of all rules from the set \( \text{Rule}(G^{\psi_1}, T, r)^- \) which have minimum cost relative to \( \psi_2 \) among all inhibitory rules from \( \text{Rule}(G^{\psi_1}, T, r)^- \), etc.

We can study also totally optimal inhibitory rules relative to various combinations of cost functions. For a cost function \( \psi \), we denote \( \psi^{W, \alpha}(T, r) = \min\{\psi(T, \rho) : \rho \in IR_{W, \alpha}(T, r)\} \), i.e., \( \psi^{W, \alpha}(T, r) \) is the minimum cost of a \((W, \alpha)\)-inhibitory rule for \( T \) and \( r \) relative to the cost function \( \psi \). Let \( \psi_1, \ldots, \psi_m \) be cost functions and \( m \geq 2 \). A \((W, \alpha)\)-inhibitory rule \( \rho \) for \( T \) and \( r \) is called a totally optimal \((W, \alpha)\)-inhibitory rule for \( T \) and \( r \) relative to the cost functions \( \psi_1, \ldots, \psi_m \) if \( \psi_1(T, \rho) = \psi^{W, \alpha}_1(T, r), \ldots, \psi_m(T, \rho) = \psi^{W, \alpha}_m(T, r) \), i.e., \( \rho \) is optimal relative to \( \psi_1, \ldots, \psi_m \) simultaneously.

Assume that \( \psi_1, \ldots, \psi_m \in \{l, -c, -rc, mc, rmc\} \). We now describe how to recog-
nize the existence of a \((W, \alpha)\)-inhibitory rule for \(T\) and \(r\) which is a totally optimal \((W, \alpha)\)-inhibitory rule for \(T\) and \(r\) relative to the cost functions \(\psi_1, \ldots, \psi_m\).

First, we construct the graph \(G = \Delta_{U, \alpha}(T^C)\) using the Algorithm \(\mathcal{A}_1\). For \(i = 1, \ldots, m\), we apply to \(G\) and \(r\) the procedure of optimization relative to \(\psi_i\) (the Algorithm \(\mathcal{A}_4\)). As a result, we obtain, for \(i = 1, \ldots, m\), the graph \(G^{\psi_i}(r)\) and the number \(\psi^U_{i, \alpha}(T^C, r)\) attached to the node \(T^C\) of \(G^{\psi_i}(r)\). Next, we apply to \(G\) sequentially the procedures of optimization relative to the cost functions \(\psi_1, \ldots, \psi_m\). As a result, we obtain graphs \(G^{\psi_1}(r), G^{\psi_1, \psi_2}(r), \ldots, G^{\psi_1, \ldots, \psi_m}(r)\) and numbers \(\varphi_1, \varphi_2, \ldots, \varphi_m\) attached to the node \(T^C\) of these graphs. We know (see Section 6.1.2) that a totally optimal \((U, \alpha)\)-decision rule for \(T^C\) and \(r\) relative to the cost functions \(\psi_1, \ldots, \psi_m\) exists if and only if \(\varphi_i = \psi^U_{i, \alpha}(T^C, r)\) for \(i = 1, \ldots, m\). Using Proposition 61 one can show that a totally optimal \((W, \alpha)\)-inhibitory rule for \(T\) and \(r\) relative to the cost functions \(\psi_1, \ldots, \psi_m\) exists if and only if a totally optimal \((U, \alpha)\)-decision rule for \(T^C\) and \(r\) relative to the cost functions \(\psi_1, \ldots, \psi_m\) exists.

Let \(G\) be a proper subgraph of the graph \(\Delta_{U, \alpha}(T^C)\) and \(r\) be a row of \(T^C\). The algorithm \(\mathcal{A}_5\) from Section 6.1.3 allows us to find the cardinality of the set \(\text{Rule}(G, T^C, r)\) containing some \((U, \alpha)\)-decision rules for \(T^C\) and \(r\), and in the same time, the cardinality of the set \(\text{Rule}(G, T^C, r)^-\) containing some \((W, \alpha)\)-inhibitory rules for \(T\) and \(r\). It can be, for example, the set of all \((W, \alpha)\)-inhibitory rules for \(T\) and \(r\) with minimum length, if \(\text{Rule}(G, T^C, r)\) is the set of all \((U, \alpha)\)-decision rules for \(T^C\) and \(r\) with minimum length.

### 6.2.1 Experimental Results for Inhibitory Rule Optimization

For some decision tables \(T\) in Table 6.1, the assumption that, for any row \(r\) of \(T\), \(D(r) \neq D(T)\) is invalid. We cannot study inhibitory rules for such tables. Thus, we performed preprocessing of each table \(T\) in Table 6.1 to a table \(T'\) by removing rows labeled with \(D(T)\). Then, we converted \(T'\) to the corresponding complementary
decision table $T^rC$ by changing, for each row $r \in Row(T')$, the set $D(r)$ with the set $D(T') \setminus D(r)$. The results considered here for decision rules for $T^rC$ will be in the same time the results for inhibitory rules for $T'$.

We construct the directed acyclic graph $\Delta(T^rC)$. We apply to the directed acyclic graph $\Delta(T^rC)$ sequentially the procedure of optimization (algorithm $A_4$) relative to the coverage, and then the procedure of optimization relative to the length. Average length and coverage of obtained rules (among all rows of $T^rC$) can be found in Table 6.10 (column “coverage+length”). Similarly, the output of sequential optimization relative to the length, and then relative to the coverage is shown in Table 6.10 (column “length+coverage”). We compared the number of rows for which there exists a totally optimal decision rule relative to length and coverage for $T^rC$ (column “Rows tot $T^rC$”) with the number of rows in $T^rC$ (column “Rows”). For decision tables in bold, each row has a totally optimal decision rule.

<table>
<thead>
<tr>
<th>Table Name T</th>
<th>coverage+length</th>
<th>length+coverage</th>
<th>Rows tot $T^rC$</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>7.50</td>
<td>3.64</td>
<td>2.80</td>
<td>5.16</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>5.17</td>
<td>1.81</td>
<td>1.69</td>
<td>4.71</td>
</tr>
<tr>
<td>cars-1</td>
<td>133.07</td>
<td>1.11</td>
<td>1.11</td>
<td>133.07</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>746.62</td>
<td>2.26</td>
<td>1.48</td>
<td>516.54</td>
</tr>
<tr>
<td>nursery-1</td>
<td>1800.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1800.00</td>
</tr>
<tr>
<td>nursery-4</td>
<td>80.00</td>
<td>1.00</td>
<td>1.00</td>
<td>80.00</td>
</tr>
<tr>
<td>teeth-1</td>
<td>15.18</td>
<td>1.00</td>
<td>1.00</td>
<td>15.18</td>
</tr>
<tr>
<td>teeth-5</td>
<td>7.36</td>
<td>1.00</td>
<td>1.00</td>
<td>7.36</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>33.79</td>
<td>1.00</td>
<td>1.00</td>
<td>33.79</td>
</tr>
</tbody>
</table>

Tables 6.11-6.18 contain, for each $\beta \in \{0, 0.01, 0.05, 0.1, 0.15, 0.2\}$ and $\alpha \in \{0, 0.1, 0.3\}$, the cardinality of the system $Rule^\beta_\alpha(T^rC)$ constructed by the algorithm $A_6$ and a lower bound on the cardinality of a $\beta$-system of (rme, $\alpha$)-decision rules for $T^rC$, where $T^rC$ is one of the considered complementary decision tables. Again, such results considered here for decision rules for $T^rC$ will be in the same time the results for
inhibitory rules for $T'$. Note that the constructed systems of inhibitory rules are usually short.

Results for multi-stage optimization of inhibitory rules over decision tables with single-valued decisions can be found in [46, 47, 48, 49].

**Table 6.11**: Bounds for decision table complementary to breast-cancer-1'

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>53</td>
<td>53</td>
<td>38</td>
</tr>
<tr>
<td>0.01</td>
<td>52</td>
<td>52</td>
<td>37</td>
</tr>
<tr>
<td>0.05</td>
<td>45</td>
<td>45</td>
<td>30</td>
</tr>
<tr>
<td>0.1</td>
<td>37</td>
<td>37</td>
<td>24</td>
</tr>
<tr>
<td>0.15</td>
<td>32</td>
<td>32</td>
<td>19</td>
</tr>
<tr>
<td>0.2</td>
<td>28</td>
<td>28</td>
<td>16</td>
</tr>
</tbody>
</table>

**Table 6.12**: Bounds for decision table complementary to breast-cancer-5'

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>0.01</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>0.05</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>0.1</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>0.15</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>0.2</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>
Table 6.13: Bounds for decision table complementary to cars-1'

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\alpha = 0$</th>
<th>$\alpha = 0.1$</th>
<th>$\alpha = 0.3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>12/4</td>
<td>12/4</td>
<td>12/4</td>
</tr>
<tr>
<td>0.01</td>
<td>10/4</td>
<td>10/4</td>
<td>10/4</td>
</tr>
<tr>
<td>0.05</td>
<td>7/3</td>
<td>7/3</td>
<td>7/3</td>
</tr>
<tr>
<td>0.1</td>
<td>5/3</td>
<td>5/3</td>
<td>5/3</td>
</tr>
<tr>
<td>0.15</td>
<td>4/3</td>
<td>4/3</td>
<td>4/3</td>
</tr>
<tr>
<td>0.2</td>
<td>4/3</td>
<td>4/3</td>
<td>4/3</td>
</tr>
</tbody>
</table>

Table 6.14: Bounds for decision table complementary to mushroom-5'

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>$\alpha = 0$</th>
<th>$\alpha = 0.1$</th>
<th>$\alpha = 0.3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>30/9</td>
<td>25/8</td>
<td>10/4</td>
</tr>
<tr>
<td>0.01</td>
<td>23/8</td>
<td>20/7</td>
<td>9/4</td>
</tr>
<tr>
<td>0.05</td>
<td>14/5</td>
<td>12/5</td>
<td>7/4</td>
</tr>
<tr>
<td>0.1</td>
<td>9/4</td>
<td>8/4</td>
<td>5/3</td>
</tr>
<tr>
<td>0.15</td>
<td>7/4</td>
<td>6/3</td>
<td>4/3</td>
</tr>
<tr>
<td>0.2</td>
<td>5/3</td>
<td>5/3</td>
<td>4/2</td>
</tr>
</tbody>
</table>
Table 6.15: Bounds for decision table complementary to nursery-1

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.01</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.05</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.1</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.15</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
<td>4 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.2</td>
<td>3 ( \uparrow ) 2</td>
<td>3 ( \uparrow ) 2</td>
<td>3 ( \uparrow ) 2</td>
</tr>
</tbody>
</table>

Table 6.16: Bounds for decision table complementary to nursery-4

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.01</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.05</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.1</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.15</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
</tr>
<tr>
<td>0.2</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
<td>3 ( \uparrow ) 3</td>
</tr>
</tbody>
</table>
### Table 6.17: Bounds for decision table complementary to teeth-1’

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.01</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.05</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.15</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 6.18: Bounds for decision table complementary to teeth-5’

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>0</th>
<th>0.1</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.01</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.05</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.15</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0.2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Chapter 7

Bi-Criteria Optimization Problem for Rules and Rule Systems: Cost vs Cost

In this chapter, we consider algorithms which construct the sets of Pareto optimal points for bi-criteria optimization problems for decision and inhibitory rules and rule systems relative two cost functions. We also show how the constructed set of Pareto optimal points can be transformed into the graphs of functions which describe the relationships between the considered cost functions. The considered tools are used to compare heuristics for rule construction.

7.1 Bi-Criteria Optimization Problem for Decision Rules and Systems of Rules: Cost vs Cost

In this section, we study bi-criteria cost vs cost optimization problem for decision rules and systems of decision rules, and consider an application of the created tools.
7.1.1 Pareto Optimal Points for Decision Rules: Cost vs Cost

Let $\psi$ and $\varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively. Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, and $G$ be a proper subgraph of the graph $\Delta_{U, \alpha}(T)$ (it is possible that $G = \Delta_{U, \alpha}(T)$).

For each node $\Theta$ of the graph $G$ containing $r$, we denote

$$p_{\psi, \varphi}(G, \Theta, r) = \{ (\psi(\Theta, \rho), \varphi(\Theta, \rho)) : \rho \in \text{Rule}(G, \Theta, r) \}.$$  

We denote by $\text{Par}(p_{\psi, \varphi}(G, \Theta, r))$ the set of Pareto optimal points for $p_{\psi, \varphi}(G, \Theta, r)$. Note that, by Proposition 63, if $G = \Delta_{U, \alpha}(T)$ then the set $\text{Rule}(G, \Theta, r)$ is equal to the set of $(U, \alpha)$-decision rules for $\Theta$ and $r$. Another interesting case is when $G$ is the result of application of procedure of optimization of rules for $r$ (algorithm $A_4$) relative to cost functions different from $\psi$ and $\varphi$ to the graph $\Delta_{U, \alpha}(T)$.

We now describe an algorithm $A_7$ constructing the set $\text{Par}(p_{\psi, \varphi}(G, T, r))$. In fact, this algorithm constructs, for each node $\Theta$ of the graph $G$, the set $B(\Theta, r) = \text{Par}(p_{\psi, \varphi}(G, \Theta, r))$.

Algorithm $A_7$.

Input: Strictly increasing cost functions $\psi$ and $\varphi$ for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, a nonempty decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$, a row $r = (b_1, \ldots, b_n)$ of $T$, and a proper subgraph $G$ of the graph $\Delta_{U, \alpha}(T)$ where $U$ is an uncertainty measure and $\alpha \in \mathbb{R}_+$.

Output: The set $\text{Par}(p_{\psi, \varphi}(G, T, r))$ of Pareto optimal points for the set of pairs $p_{\psi, \varphi}(G, T, r) = \{ (\psi(T, \rho), \varphi(T, \rho)) : \rho \in \text{Rule}(G, T, r) \}$.

1. If all nodes in $G$ containing $r$ are processed, then return the set $B(T, r)$. Otherwise, choose in the graph $G$ a node $\Theta$ containing $r$ which is not processed.
yet and which is either a terminal node of $G$ or a nonterminal node of $G$ such that, for any $f_i \in E_G(\Theta)$, the node $\Theta(f_i, b_i)$ is already processed, i.e., the set $B(\Theta(f_i, b_i), r)$ is already constructed.

2. If $\Theta$ is a terminal node, then set $B(\Theta, r) = \{(\psi^0(\Theta), \varphi^0(\Theta))\}$. Mark the node $\Theta$ as processed and proceed to step 1.

3. If $\Theta$ is a nonterminal node then, for each $f_i \in E_G(\Theta)$, construct the set $B(\Theta(f_i, b_i), r)^{FH}$, and construct the multiset

$$A(\Theta, r) = \bigcup_{f_i \in E_G(\Theta)} B(\Theta(f_i, b_i), r)^{FH}$$

by simple transcription of elements from the sets $B(\Theta(f_i, b_i), r)^{FH}, f_i \in E_G(\Theta)$.

4. Apply to the multiset $A(\Theta, r)$ the algorithm $A_2$ which constructs the set $Par(A(\Theta, r))$. Set $B(\Theta, r) = Par(A(\Theta, r))$. Mark the node $\Theta$ as processed and proceed to step 1.

**Proposition 71** Let $\psi$ and $\varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, and $G$ be a proper subgraph of the graph $\Delta_{U, \alpha}(T)$. Then, for each node $\Theta$ of the graph $G$ containing $r$, the algorithm $A_7$ constructs the set $B(\Theta, r) = Par(p_{\psi, \varphi}(G, \Theta, r)).$

**Proof.** We prove the considered statement by induction on nodes of $G$. Let $\Theta$ be a terminal node of $G$ containing $r$. Then $Rule(G, \Theta, r) = \{\rightarrow mcld(\Theta)\}$, $p_{\psi, \varphi}(G, \Theta, r) = Par(p_{\psi, \varphi}(G, \Theta, r)) = \{\psi^0(\Theta), \varphi^0(\Theta)\} \}$, and

$$B(\Theta, r) = Par(p_{\psi, \varphi}(G, \Theta, r)).$$
Let \( \Theta \) be a nonterminal node of \( G \) containing \( r \) such that, for any \( f_i \in E_G(\Theta) \), the considered statement holds for the node \( \Theta(f_i, b_i) \), i.e.,

\[
B(\Theta(f_i, b_i), r) = \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r)).
\]

It is clear that

\[
p_{\psi, \varphi}(G, \Theta, r) = \bigcup_{f_i \in E_G(\Theta)} p_{\psi, \varphi}(G, \Theta(f_i, b_i), r)^{FH}.
\]

From Lemma 53 it follows that

\[
\text{Par}(p_{\psi, \varphi}(G, \Theta, r)) \subseteq \bigcup_{f_i \in E_G(\Theta)} \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))^{FH}.
\]

By Lemma 57, \( \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r)^{FH}) = \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))^{FH} \) for any \( f_i \in E_G(\Theta) \). Therefore

\[
\text{Par}(p_{\psi, \varphi}(G, \Theta, r)) \subseteq \bigcup_{f_i \in E_G(\Theta)} \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))^{FH} \subseteq p_{\psi, \varphi}(G, \Theta, r).
\]

Using Lemma 52 we obtain

\[
\text{Par}(p_{\psi, \varphi}(G, \Theta, r)) = \text{Par} \left( \bigcup_{f_i \in E_G(\Theta)} \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))^{FH} \right).
\]

Since \( B(\Theta, r) = \text{Par} \left( \bigcup_{f_i \in E_G(\Theta)} B(\Theta(f_i, b_i), r)^{FH} \right) \) and

\[
B(\Theta(f_i, b_i), r) = \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))
\]

for any \( f_i \in E_G(\Theta) \), we have \( B(\Theta, r) = \text{Par}(p_{\psi, \varphi}(G, \Theta, r)) \). ■

We now evaluate the number of elementary operations (computations of \( F, H, \psi^0, \varphi^0 \), and comparisons) made by the algorithm \( A_7 \). Let us recall that, for a given
cost function $\psi$ for decision rules and decision table $T$,

$$q_\psi(T) = |\{\psi(\Theta, \rho) : \Theta \in SEP(T), \rho \in DR(\Theta)\}|.$$ 

In particular, by Lemma 60,

$$q_l(T) \leq n + 1, q_c(T) \leq N(T) + 1, q_{rc}(T) \leq N(T)(N(T) + 1), q_{mc}(T) \leq N(T) + 1, q_{rmc}(T) \leq N(T)(N(T) + 1).$$

**Proposition 72** Let $\psi$ and $\varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, and $G$ be a proper subgraph of the graph $\Delta_{U, \alpha}(T)$. Then, to construct the set $\text{Par}(p_{\psi, \varphi}(G, T, r))$, the algorithm $A_7$ makes

$$O(L(G) \min(q_\psi(T), q_\varphi(T)) n \log(\min(q_\psi(T), q_\varphi(T)))n)$$

elementary operations (computations of $F$, $H$, $\psi^0$, $\varphi^0$, and comparisons).

**Proof.** To process a terminal node, the algorithm $A_7$ makes two elementary operations – computes $\psi^0$ and $\varphi^0$. We now evaluate the number of elementary operations under the processing of a nonterminal node $\Theta$. From Lemma 49 it follows that $|\text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))| \leq \min(q_\psi(T), q_\varphi(T))$ for any $f_i \in E_G(\Theta)$. It is clear that $|E_G(\Theta)| \leq n$, $|\text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))| = |\text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))|$ for any $f_i \in E_G(\Theta)$. From Proposition 71 it follows that $B(\Theta(f_i, b_i), r) = \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))$ and $B(\Theta(f_i, b_i), r)^{FH} = \text{Par}(p_{\psi, \varphi}(G, \Theta(f_i, b_i), r))^{FH}$ for any $f_i \in E_G(\Theta)$. Hence

$$|A(\Theta, r)| \leq \min(q_\psi(T), q_\varphi(T))n.$$ 

Therefore to construct the sets $B(\Theta(f_i, b_i), r)^{FH}, f_i \in E_G(\Theta)$, from the sets $B(\Theta(f_i, b_i), r), f_i \in E_G(\Theta)$, the algorithm $A_7$ makes $O(\min(q_\psi(T), q_\varphi(T))n)$ compu-
tations of $F$ and $H$, and to construct the set

$$Par(A(\Theta, r)) = Par(p_{\psi, \varphi}(G, \Theta, r))$$

from the set $A(\Theta, r)$, the algorithm $A_7$ makes

$$O(\min(q_\psi(T), q_\varphi(T))n \log(\min(q_\psi(T), q_\varphi(T))n))$$

comparisons (see Proposition 50). Hence, to treat a nonterminal node $\Theta$, the algorithm makes

$$O(\min(q_\psi(T), q_\varphi(T))n \log(\min(q_\psi(T), q_\varphi(T))n))$$

computations of $F$, $H$, and comparisons.

To construct the set $Par(p_{\psi, \varphi}(G, T, r))$ the algorithm $A_7$ makes

$$O(L(G) \min(q_\psi(T), q_\varphi(T))n \log(\min(q_\psi(T), q_\varphi(T))n))$$

elementary operations (computations of $F$, $H$, $\psi^0$, $\varphi^0$, and comparisons). ■

**Proposition 73** Let $\psi$ and $\varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0$, $F$ and $\varphi^0$, $H$, respectively, $\psi, \varphi \in \{l, -c, -rc, -c_M, mc, rmc\}$, and $\mathcal{U}$ be a restricted information system. Then the algorithm $A_7$ has polynomial time complexity for decision tables from $\mathcal{T}(\mathcal{U})$ depending on the number of conditional attributes in these tables.

**Proof.** Since $\psi, \varphi \in \{l, -c, -rc, -c_M, mc, rmc\}$,

$$\psi^0, \varphi^0 \in \{0, -N_{med(T)}(T), -N_{med(T)}(T)/N(T), -N^M(T),$$

$$N(T) - N_{med(T)}(T), (N(T) - N_{med(T)}(T))/N(T)\},$$
and \( F, H \in \{x, x+1\} \). From Lemma 60 and Proposition 72 it follows that, for the algorithm \( \mathcal{A}_7 \), the number of elementary operations (computations of \( F, H, \psi^0, \varphi^0 \), and comparisons) is bounded from above by a polynomial depending on the size of input table \( T \) and on the number of separable subtables of \( T \). All operations with numbers are basic ones. The computations of numerical parameters of decision tables used by the algorithm \( \mathcal{A}_7 \) have polynomial time complexity depending on the size of decision tables.

According to Proposition 46, the algorithm \( \mathcal{A}_7 \) has polynomial time complexity for decision tables from \( \mathcal{T}(\mathcal{U}) \) depending on the number of conditional attributes in these tables.

\subsection{7.1.2 Relationships for Decision Rules: Cost vs Cost}

Let \( \psi \) and \( \varphi \) be strictly increasing cost functions for decision rules, \( T \) be a nonempty decision table with \( n \) conditional attributes \( f_1, \ldots, f_n \), \( r = (b_1, \ldots, b_n) \) be a row of \( T \), \( U \) be an uncertainty measure, \( \alpha \in \mathbb{R}_+ \), and \( G \) be a proper subgraph of the graph \( \Delta_{U,\alpha}(T) \) (it is possible that \( G = \Delta_{U,\alpha}(T) \)).

To study relationships between cost functions \( \psi \) and \( \varphi \) on the set of rules \( \text{Rule}(G, T, r) \), we consider partial functions \( \mathcal{R}^{\psi,\varphi}_{G,T,r} : \mathbb{R} \to \mathbb{R} \) and \( \mathcal{R}^{\varphi,\psi}_{G,T,r} : \mathbb{R} \to \mathbb{R} \) defined in the following way:

\[
\mathcal{R}^{\psi,\varphi}_{G,T,r}(x) = \min \{ \varphi(T, \rho) : \rho \in \text{Rule}(G, T, r), \psi(T, \rho) \leq x \},
\]

\[
\mathcal{R}^{\varphi,\psi}_{G,T,r}(x) = \min \{ \psi(T, \rho) : \rho \in \text{Rule}(G, T, r), \varphi(T, \rho) \leq x \}.
\]

Let \( p_{\psi,\varphi}(G, T, r) = \{ (\psi(T, \rho), \varphi(T, \rho)) : \rho \in \text{Rule}(G, T, r) \} \) and \( (a_1, b_1), \ldots, (a_k, b_k) \) be the normal representation of the set \( \text{Par}(p_{\psi,\varphi}(G, T, r)) \) where \( a_1 < \ldots < a_k \)
and $b_1 > \ldots > b_k$. By Lemma 58 and Remark 59, for any $x \in \mathbb{R}$,

$$R_{G,T,r}^\psi(x) = \begin{cases} 
\text{undefined,} & x < a_1 \\
b_1, & a_1 \leq x < a_2 \\
\ldots & \ldots \\
b_{k-1}, & a_{k-1} \leq x < a_k \\
b_k, & a_k \leq x 
\end{cases}$$

$$R_{G,T,r}^\varphi(x) = \begin{cases} 
\text{undefined,} & x < b_k \\
a_k, & b_k \leq x < b_{k-1} \\
\ldots & \ldots \\
a_2, & b_2 \leq x < b_1 \\
a_1, & b_1 \leq x 
\end{cases}$$

7.1.3 Pareto Optimal Points for Systems of Decision Rules:

Cost vs Cost

Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, and $G = (G_1, \ldots, G_{N(T)})$ be an $N(T)$-tuple of proper subgraphs of the graph $\Delta_{U,\alpha}(T)$. Let $G = \Delta_{U,\alpha}(T)$ and $\xi$ be a cost function for decision rules. Then two interesting examples of such $N(T)$-tuples are $(G, \ldots, G)$ and $(G^\xi(r_1), \ldots, G^\xi(r_{N(T)}))$.

We denote by $S(G, T)$ the set $\text{Rule}(G_1, T, r_1) \times \ldots \times \text{Rule}(G_{N(T)}, T, r_{N(T)})$ of $(U, \alpha)$-systems of decision rules for $T$. Let $\psi, \varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, and $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$. It is clear that $\psi_f$ and $\varphi_g$ are strictly increasing cost functions for systems of decision rules.

We describe now an algorithm which constructs the set of Pareto optimal points for the set of pairs $f^{\psi,\varphi}(G, T) = \{ (\psi_f(T, S), \varphi_g(T, S)) : S \in S(G, T) \}$.
Algorithm $A_8$.

Input: Strictly increasing cost functions for decision rules $\psi$ and $\varphi$ given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, increasing functions $f, g$ from $\mathbb{R}^2$ to $\mathbb{R}$, a nonempty decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, and an $N(T)$-tuple $G = (G_1, \ldots, G_{N(T)})$ of proper subgraphs of the graph $\Delta_{U,\alpha}(T)$ where $U$ is an uncertainty measure and $\alpha \in \mathbb{R}_+$. 

Output: The set $\text{Par}(p_{\psi,\varphi}^f(G,T))$ of Pareto optimal points for the set of pairs $p_{\psi,\varphi}^{f,g}(G,T) = \{(\psi(T,S), (\varphi(T,S)) : S \in \mathcal{S}(G,T)\}$.

1. Using the algorithm $A_7$ construct, for $i = 1, \ldots, N(T)$, the set $\text{Par}(P_i)$ where

   $$P_i = p_{\psi,\varphi}(G_i, T, r_i) = \{((\psi(T,\rho), \varphi(T,\rho)) : \rho \in \text{Rule}(G_i, T, r_i)\}.$$

2. Apply the algorithm $A_3$ to the functions $f, g$ and the sets $\text{Par}(P_1), \ldots, \text{Par}(P_{N(T)})$. Set $C(G,T)$ the output of the algorithm $A_3$ and return it.

Proposition 74 Let $\psi, \varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, $T$ be a decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, and $G = (G_1, \ldots, G_{N(T)})$ be an $N(T)$-tuple of proper subgraphs of the graph $\Delta_{U,\alpha}(T)$. Then the algorithm $A_8$ constructs the set $C(G,T) = \text{Par}(p_{\psi,\varphi}^{f,g}(G,T))$.

Proof. For $i = 1, \ldots, N(T)$, denote $P_i = p_{\psi,\varphi}(G_i, T, r_i)$. During the first step, the algorithm $A_8$ constructs (using the algorithm $A_7$) the sets $\text{Par}(P_1), \ldots, \text{Par}(P_{N(T)})$ (see Proposition 71). During the second step, the algorithm $A_8$ constructs (using the algorithm $A_3$) the set $C(G,T) = \text{Par}(Q_{N(T)})$ where $Q_1 = P_1$, and, for $i = 2, \ldots, N(T)$, $Q_i = Q_{i-1} \{fg\} P_i$ (see Proposition 55). One can show that $Q_{N(T)} = p_{\psi,\varphi}^{f,g}(G,T)$. Therefore $C(G,T) = \text{Par}(p_{\psi,\varphi}^{f,g}(G,T))$. $\blacksquare$
Let us recall that, for a given cost function $\psi$ and a decision table $T$, $q_\psi(T) = |\{\psi(\Theta, \rho) : \Theta \in SEP(T), \rho \in DR(\Theta)\}|$. In particular, by Lemma 60, $q(T) \leq n + 1$, $q_{-c}(T) \leq N(T) + 1$, $q_{-rc}(T) \leq N(T)(N(T) + 1)$, $q_{-cM}(T) \leq N(T) + 1$, $q_{mc}(T) \leq N(T) + 1$, and $q_{rmc}(T) \leq N(T)(N(T) + 1)$.

Let us recall also that, for a given cost function $\psi$ for decision rules and a decision table $T$, $Range_\psi(T) = \{\psi(\Theta, \rho) : \Theta \in SEP(T), \rho \in DR(\Theta)\}$. By Lemma 60, $Range_{tl}(T) \subseteq \{0, 1, \ldots, n\}$, $Range_{-c}(T) \subseteq \{0, -1, \ldots, -N(T)\}$, $Range_{-cM}(T) \subseteq \{0, -1, \ldots, -N(T)\}$, and $Range_{mc}(T) \subseteq \{0, 1, \ldots, N(T)\}$. Let $t_l(T) = n$, $t_{-c}(T) = N(T)$, $t_{-cM}(T) = N(T)$, and $t_{mc}(T) = N(T)$.

**Proposition 75** Let $\psi, \varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, $\psi \in \{l, -c, -cM, mc\}$, $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$, $f \in \{x+y, \max(x, y)\}$, $U$ be an uncertainty measure, $\alpha \in \mathbb{R}_+$, $T$ be a decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, $G = (G_1, \ldots, G_{N(T)})$ be an $N(T)$-tuple of proper subgraphs of the graph $\Delta_{U, \alpha}(T)$, and $L(G) = L(\Delta_{U, \alpha}(T))$. Then, to construct the set $Par(p_{\psi, \varphi}^f[g](G, T))$, the algorithm $A_8$ makes

$$O(N(T)L(G) \min(q_\psi(T), q_\varphi(T))n \log(\min(q_\psi(T), q_\varphi(T)))n))$$
$$+ O(N(T)t_\psi(T)^2 \log(t_\psi(T)))$$

elementary operations (computations of $F$, $H$, $\psi^0$, $\varphi^0$, $f$, $g$ and comparisons) if $f = \max(x, y)$, and

$$O(N(T)L(G) \min(q_\psi(T), q_\varphi(T))n \log(\min(q_\psi(T), q_\varphi(T)))n))$$
$$+ O(N(T)^2t_\psi(T)^2 \log(N(T)t_\psi(T))))$$

elementary operations (computations of $F$, $H$, $\psi^0$, $\varphi^0$, $f$, $g$ and comparisons) if
\[ f = x + y. \]

**Proof.** To construct the sets \( Par(P_i) = Par(p_{\psi,\varphi}(G_i, T, r_i)), \ i = 1, \ldots, N(T), \) the algorithm \( \mathcal{A}_7 \) makes

\[
O(N(T)L(G) \min(q_{\psi}(T), q_{\varphi}(T))n \log(\min(q_{\psi}(T), q_{\varphi}(T))n))
\]
elementary operations (computations of \( F, H,\psi^0,\varphi^0, \) and comparisons) – see Proposition 72.

We now evaluate the number of elementary operations (computations of \( f, g, \) and comparisons) made by the algorithm \( \mathcal{A}_3 \) during the construction of the set \( C(G, T) = Par(Q_{N(T)}) = Par(p^{f,g}_{\psi,\varphi}(G, T)). \) We know that \( \psi \in \{l, -c, -c_M, mc\} \) and \( f \in \{x + y, \max(x, y)\}. \)

For \( i = 1, \ldots, N(T), \) let \( P_i^1 = \{a : (a, b) \in P_i\}. \) Since \( \psi \in \{l, -c, -c_M, mc\}, \) we have \( P_i^1 \subseteq \{0, 1, \ldots, t_{\psi}(T)\} \) for \( i = 1, \ldots, N(T) \) or \( P_i^1 \subseteq \{0, -1, \ldots, -t_{\psi}(T)\} \) for \( i = 1, \ldots, N(T). \)

Using Proposition 56 we obtain the following.

If \( f = x + y, \) then to construct the set \( Par(Q_{N(T)}) \) the algorithm \( \mathcal{A}_3 \) makes

\[
O(N(T)^2t_{\psi}(T)^2 \log(N(T)t_{\psi}(T)))
\]
elementary operations (computations of \( f, g, \) and comparisons).

If \( f = \max(x, y), \) then to construct the sets \( Par(Q_{N(T)}) \) the algorithm \( \mathcal{A}_3 \) makes

\[
O(N(T)t_{\psi}(T)^2 \log(t_{\psi}(T)))
\]
elementary operations (computations of \( f, g, \) and comparisons). \( \blacksquare \)

Similar analysis can be done for the case when \( \varphi \in \{l, -c, -c_M, mc\} \) and \( g \in \{x + y, \max(x, y)\}. \)
Proposition 76 Let $\psi, \varphi$ be strictly increasing cost functions for decision rules given by pairs of functions $\psi^0, F$ and $\varphi^0, H$, respectively, $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$, $\psi \in \{l, -c, -c_M, mc\}$, $\varphi \in \{l, -c, -rc, -c_M, mc, rmc\}$, $f, g \in \{x+y, \max(x, y)\}$, and $\mathcal{U}$ be a restricted information system. Then the algorithm $\mathcal{A}_8$ has polynomial time complexity for decision tables from $\mathcal{T}(\mathcal{U})$ depending on the number of conditional attributes in these tables.

Proof. Since $\psi \in \{l, -c, -c_M, mc\}$ and $\varphi \in \{l, -c, -rc, -c_M, mc, rmc\}$,

$$
\psi^0, \varphi^0 \in \{0, -N_{\text{mod}(T)}(T), -N_{\text{mod}(T)}(T)/N(T), -N^M(T), N(T) - N_{\text{mod}(T)}(T), (N(T) - N_{\text{mod}(T)}(T))/N(T)\},
$$

and $F, H \in \{x, x+1\}$. From Lemma 60 and Proposition 75 it follows that, for the algorithm $\mathcal{A}_8$, the number of elementary operations (computations of $F, H, \psi^0, \varphi^0, f, g$, and comparisons) is bounded from above by a polynomial depending on the size of input table $T$ and on the number of separable subtables of $T$. All operations with numbers are basic ones. The computations of numerical parameters of decision tables used by the algorithm $\mathcal{A}_8$ $(0, -N_{\text{mod}(T)}(T), -N_{\text{mod}(T)}(T)/N(T), -N^M(T), N(T) - N_{\text{mod}(T)}(T), (N(T) - N_{\text{mod}(T)}(T))/N(T))$ have polynomial time complexity depending on the size of decision tables.

According to Proposition 46, the algorithm $\mathcal{A}_8$ has polynomial time complexity for decision tables from $\mathcal{T}(\mathcal{U})$ depending on the number of conditional attributes in these tables. ■

7.1.4 Relationships for Systems of Decision Rules: Cost vs Cost

Let $\psi$ and $\varphi$ be strictly increasing cost functions for decision rules, and $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$. Let $T$ be a nonempty decision table, $U$ be an uncertainty
measure, \( \alpha \in \mathbb{R}_{+} \), and \( G = (G_1, \ldots, G_{N(T)}) \) be an \( N(T) \)-tuple of proper subgraphs of the graph \( \Delta_{U,\alpha}(T) \).

To study relationships between cost functions \( \psi_f \) and \( \varphi_g \) on the set of systems of rules \( S(G, T) \), we consider two partial functions \( R_{G,T}^{\psi_f,\varphi_g} : \mathbb{R} \to \mathbb{R} \) and \( R_{G,T}^{\varphi_g,\psi_f} : \mathbb{R} \to \mathbb{R} \) defined in the following way:

\[
R_{G,T}^{\psi_f,\varphi_g}(x) = \min \{ \varphi_g(T, S) : S \in S(G, T), \psi_f(T, S) \leq x \},
\]
\[
R_{G,T}^{\varphi_g,\psi_f}(x) = \min \{ \psi_f(T, S) : S \in S(G, T), \varphi_g(T, S) \leq x \}.
\]

Let \( p_{\psi_f,\varphi_g}(G, T) = \{(\psi_f(T, S), \varphi_g(T, S)) : S \in S(G, T)\} \), and \((a_1, b_1), \ldots, (a_k, b_k)\) be the normal representation of the set \( \text{Par}(p_{\psi_f,\varphi_g}(G, T)) \) where \( a_1 < \ldots < a_k \) and \( b_1 > \ldots > b_k \). By Lemma 58 and Remark 59, for any \( x \in \mathbb{R} \),

\[
R_{G,T}^{\psi_f,\varphi_g}(x) = \begin{cases} 
\text{undefined,} & x < a_1 \\
\quad b_1, & a_1 \leq x < a_2 \\
\quad \ldots, & \quad \ldots \\
b_{k-1}, & a_{k-1} \leq x < a_k \\
b_k, & a_k \leq x 
\end{cases}
\]

\[
R_{G,T}^{\varphi_g,\psi_f}(x) = \begin{cases} 
\text{undefined,} & x < b_k \\
\quad a_k, & b_k \leq x < b_{k-1} \\
\quad \ldots, & \quad \ldots \\
a_2, & b_2 \leq x < b_1 \\
a_1, & b_1 \leq x 
\end{cases}
\]
7.2 Comparison of Heuristics for Decision Rule Construction

In this section, we compare 13 greedy heuristics for construction of decision rules from the point of view of single-criterion optimization (relative to length or coverage) and bi-criteria optimization (relative to length and coverage).

7.2.1 Greedy Heuristics

We consider greedy heuristics each of which, for a given decision table $T$, row $r$ of $T$, and decision $t \in D(r)$, constructs an exact decision rule for $T$ and $r$ (a $(U,0)$-decision rule for $T$ and $r$ for an uncertainty measure $U$).

Each heuristic is described by the algorithm $A_{\text{greedy}}$ which uses specific for this heuristic attribute selection function $F(T, r, t)$ which is called the heuristic function (these functions are defined after the description if $A_{\text{greedy}}$).

Algorithm $A_{\text{greedy}}$

Input: A decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$, a row $r = (b_1, \ldots, b_n)$ from $T$, a decision $t$ from $D(r)$, and a heuristic function $F(T, r, t)$.

Output: An exact decision rule for $T$ and $r$.

1. Create a rule $\rho_0$ of the form $\rightarrow t$.

2. For $k \geq 0$, assume $\rho_k$ is already defined as $f_{i_1} = b_{i_1} \land \cdots \land f_{i_k} = b_{i_k} \rightarrow t$ (Step 1 defines for $k = 0$.)

3. Define a subtable $T^k$ such that $T^k = T(f_{i_1}, b_{i_1}) \ldots (f_{i_k}, b_{i_k})$.

4. If $T^k$ has a common decision $t'$ (it is possible that $t \neq t'$), then replace the right-hand side of $\rho_k$ with $t'$ and end the algorithm with $\rho_k$ as the constructed rule.
5. Select an attribute $f_{i_{k+1}} \in E(T^k)$ such that $F(T^k, r, t) = f_{i_{k+1}}$.

6. Define $\rho_{k+1}$ by adding the term $f_{i_{k+1}} = b_{i_{k+1}}$ to the left-hand side of $\rho_k$.

7. Repeat from step 2 with $k = k + 1$.

Now we define the heuristic functions $F(T, r, t)$ used by the algorithm $A_{\text{greedy}}$. Given that $T$ is a decision table with conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ is a row $T$, and $t \in D(r)$, let $N_\ell(T)$ be the number of rows $r'$ in $T$ such that $t \in D(r')$. Let $M(T, t) = N(T) - N_\ell(T)$ and $P(T, t) = N_\ell(T)/N(T)$. For any attribute $f_j \in E(T)$, we define $a(T, r, f_j, t) = N_\ell(T) - N_\ell(T(f_j, b_j))$ and $b(T, r, f_j, t) = M(T, t) - M(T(f_j, b_j), t)$. By $mcd(T)$, we denote the most common decision for $T$ which is the minimum decision $t_0$ from $D(T)$ such that $N_{t_0}(T) = \max\{N_\ell(T) : t \in D(T)\}$.

In this case we can define the following heuristics that will be used in the experiments:

1. poly: $F(T, r, t) = f_j$ such that $f_j \in E(T)$ and $\frac{b(T, r, f_j, t)}{a(T, r, f_j, t) + 1}$ is maximized.

2. log: $F(T, r, t) = f_j$ such that $f_j \in E(T)$ and $\frac{b(T, r, f_j, t)}{\log_2(a(T, r, f_j, t) + 2)}$ is maximized.

3. maxCov: $F(T, r, t) = f_j$ such that $f_j \in E(T)$, $b(T, r, f_j, t) > 0$ and $a(T, r, f_j, t)$ is minimized.

4. M: $F(T, r, t) = f_j$ such that $f_j \in E(T)$, $T' = T(f_j, b_j)$ and $M(T', t)$ is minimized.

5. RM: $F(T, r, t) = f_j$ such that $f_j \in E(T)$, $T' = T(f_j, b_j)$ and $\frac{M(T', t)}{N(T')}$ is minimized.

6. me: $F(T, r, t) = f_j$ such that $f_j \in E(T)$, $T' = T(f_j, b_j)$ and $M(T', mcd(T'))$ is minimized.

7. mep: $F(T, r, t) = f_j$ such that $f_j \in E(T)$, $T' = T(f_j, b_j)$ and $\frac{M(T', mcd(T'))}{N(T')}$ is minimized.
8. ABS: $F(T, r, t) = f_j$ such that $f_j \in E(T)$, $T' = T(f_j, b_j)$ and $\prod_{q \in D(T')} (1 - P(T', q))$ is minimized.

For the first five heuristic functions (poly, log, maxCov, M, and RM), we apply the algorithm $A_{\text{greedy}}$ using each decision $t \in D(r)$. As a result, we obtain $|D(r)|$ rules. Depending on our aim, we either choose among these rules a single rule with minimum length (we denote this algorithm by $\langle$name of heuristic$\rangle_{\text{L}}$) or a single rule with maximum coverage (we denote this algorithm by $\langle$name of heuristic$\rangle_{\text{C}}$). For the last three heuristics, the algorithm $A_{\text{greedy}}$ is only applied to each row $r$ once using any given $t \in D(r)$ since the resulting rule will be the same regardless of the input decision. We apply each of the considered heuristics to each row of $T$. Overall, this results in 13 systems of decision rules for the decision table $T$ with many-valued decisions (constructed by heuristics: poly$_C$, poly$_L$, log$_C$, log$_L$, maxCov$_C$, maxCov$_L$, M$_C$, M$_L$, RM$_C$, RM$_L$, me, mep, and ABS).

### 7.2.2 Experimental Results

We carried out the experiments on 9 decision tables with many-valued decisions derived from usual data sets from the UCI ML Repository [40] as shown in Table 7.1. For instance, the “nursery-4” decision table is obtained from the “nursery” data set by the removal of 4 conditional attributes. The positions of these attributes in the original data set can be found in the column “Removed Attributes” of Table 7.1. The resulting table contains groups of equal rows possibly with different decisions. We keep a single row from each group and label it with the set of all decisions attached to the rows in its group. As a result, we obtain a decision table $T$ with many-valued decisions. We do not remove from $T$ rows $r$ such that $D(r) = D(T)$. The number of rows and attributes in these tables can be found in the columns “Rows” and “Attributes”, respectively. The last column “# POPs” shows the number of Pareto optimal points constructed by the algorithm $A_8$ for each table. We apply the algo-
Table 7.1: Number of Pareto optimal points for some modified decision tables from UCI ML Repository

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Rows</th>
<th>Attributes</th>
<th>Removed Attributes</th>
<th># POPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>3</td>
<td>169</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>4,5,6,8,9</td>
<td>16</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>4078</td>
<td>17</td>
<td>5,8,11,13,22</td>
<td>3203</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
<td>1</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
<td>1</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
<td>4</td>
</tr>
</tbody>
</table>

Algorithm $\mathcal{A}_8$ to the cost functions $l$ and $-c$, functions $f = g = x + y$ (it means that we consider average length of rules from systems and negative average coverage of rules from systems), decision table $T$ and $N(T)$-tuple $G = (\Delta(T), \ldots, \Delta(T))$ where $\Delta = \Delta_{U,0}(T)$ for some uncertainty measure $U$. More than half of the tables have only a single Pareto optimal point that means, for each row, there exists a rule which simultaneously has both minimum length and maximum coverage (totally optimal rule relative to coverage and length).

We measured how well heuristics perform for single-criterion and bi-criteria optimizations for length and coverage. For single-criterion optimization, we used ranking to avoid bias introduced from different table sizes and to penalize heuristics for performing very badly on a few decision tables. In case of equal values for some rule heuristics, we break ties by averaging their ranks. Table 7.2 presents ranking of greedy heuristics averaged across all 9 decision tables with many-valued decisions. The greedy heuristics that perform well for one cost function do not necessarily perform well for another cost function. By looking at Table 7.2, for example, we can see that “poly_C” and “poly_L” perform very well for coverage, but quite badly for length. Similarly, “ML” and “RML” are the best heuristics for length but they do not give good results for coverage. For bi-criteria optimization, we utilized Pareto
Table 7.2: Comparing heuristics for decision tables with many-valued decisions

<table>
<thead>
<tr>
<th>Heuristic Name</th>
<th>Average Rank</th>
<th>Overall Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cov</td>
<td>Len</td>
</tr>
<tr>
<td>poly_C</td>
<td>2.61</td>
<td>7.5</td>
</tr>
<tr>
<td>poly_L</td>
<td>3.72</td>
<td>7.17</td>
</tr>
<tr>
<td>log_C</td>
<td>4.06</td>
<td>5.94</td>
</tr>
<tr>
<td>log_L</td>
<td>5.17</td>
<td>5.44</td>
</tr>
<tr>
<td>maxCov_C</td>
<td>8.17</td>
<td>12.44</td>
</tr>
<tr>
<td>maxCov_L</td>
<td>8.61</td>
<td>12.33</td>
</tr>
<tr>
<td>M_C</td>
<td>8.5</td>
<td>4.61</td>
</tr>
<tr>
<td>M_L</td>
<td>9.61</td>
<td>4.5</td>
</tr>
<tr>
<td>RM_C</td>
<td>5.78</td>
<td>4.83</td>
</tr>
<tr>
<td>RM_L</td>
<td>6.94</td>
<td>4.5</td>
</tr>
<tr>
<td>ABS</td>
<td>8.75</td>
<td>8.56</td>
</tr>
<tr>
<td>me</td>
<td>10.56</td>
<td>5.83</td>
</tr>
<tr>
<td>mep</td>
<td>8.5</td>
<td>7.33</td>
</tr>
</tbody>
</table>

optimal points. We measured the distance between a greedy solution and the nearest Pareto optimal point to get a good idea of its performance. In order to remove any bias due to differing values for different cost functions (length only varies from 0 to the number of attributes while coverage varies from 0 to the number of rows) we normalized the distance along each axis first before calculating the distances. Another observation is that the heuristic that performed best for bi-criteria optimization (closest to being Pareto optimal) is “log_C”. This heuristic may not be the best for either length or coverage, but it gives reasonable results for both at the same time.

Figures 7.1-7.3 depict the Pareto optimal points and greedy heuristic results for three decision tables “breast-cancer-1”, “mushroom-5” and “zoo-data-5”, respectively, before normalization.

Comparison of heuristics was done jointly with Talha Amin who considered decision tables with single-valued decisions.

The work in [50] is devoted to the comparison of heuristics for optimization of association rules. Comparisons of heuristics from the point of view of accuracy of classifiers based on rules created by the heuristics can be found in [51].

Some heuristics for construction of rules over decision tables with many-valued decisions were considered and compared [52, 53]. However, at least 10 heuristics considered in this section are different from ones studied in [52].
**Figure 7.1:** Pareto optimal points and rule heuristic solutions for breast-cancer-1

**Figure 7.2:** Pareto optimal points and rule heuristic solutions for mushroom-5
7.3 Bi-criteria Optimization of Inhibitory Rules: Cost vs Cost

In this section, we study bi-criteria cost vs cost optimization problem for inhibitory rules and systems of inhibitory rules, and consider an application of the created tools.

7.3.1 Pareto Optimal Points for Inhibitory Rules: Cost vs Cost

Let \( T \) be a nondegenerate decision table with \( n \) conditional attributes \( f_1, \ldots, f_n \), \( r \) be a row of \( T \), \( T^C \) be the decision table complementary to \( T \), \( U \) be an uncertainty measure, \( W \) be a completeness measure, \( U \) and \( W \) are dual, \( \alpha \in \mathbb{R}_+ \), and \( \psi, \varphi \in \{ l, -c, -rc, mc, rmc \} \).

Let \( G \) be a proper subgraph of the graph \( \Delta_{U, \alpha}(T^C) \) and

\[
p_{\psi, \varphi}(G, T^C, r) = \{ (\psi(T^C, \rho), \varphi(T^C, \rho)) : \rho \in \text{Rule}(G, T^C, r) \}.
\]
In Section 7.1.1, the algorithm $\mathcal{A}_7$ is described which constructs the set $\Par(p_\psi,\varphi(G, T^C, r))$ of Pareto optimal points for the set of pairs $p_\psi,\varphi(G, T^C, r)$.

In Section 6.2, we show that $\Rule(G, T^C, r)^- \subseteq IR_{W,\alpha}(T, r)$. Let $G = \Delta_{U,\alpha}(T^C)$. Then $\Rule(G, T^C, r)^- = IR_{W,\alpha}(T, r)$. If $\eta \in \{l, -c, -rc, mc, rmc\}$ then $\Rule(G^n(r), T^C, r)^-$ is equal to the set of all rules from $IR_{W,\alpha}(T, r)$ which have minimum cost relative to $\eta$ among all inhibitory rules from $IR_{W,\alpha}(T, r)$.

Denote $ip_\psi,\varphi(G, T, r) = \{(\psi(T, r^-), \varphi(T, r^-)) : \rho \in \Rule(G, T^C, r)^\}$.

From Proposition 61 it follows that $(\psi(T, r^-), \varphi(T, r^-)) = (\psi(T^C, \rho), \varphi(T^C, \rho))$ for any $\rho \in \Rule(G, T^C, r)$. Therefore $p_\psi,\varphi(G, T^C, r) = ip_\psi,\varphi(G, T, r)$ and

$$\Par(p_\psi,\varphi(G, T^C, r)) = \Par(ip_\psi,\varphi(G, T, r)).$$

To study relationships between cost functions $\psi$ and $\varphi$ on the set of inhibitory rules $\Rule(G, T^C, r)^-$, we consider partial functions $\mathcal{IR}^{\psi,\varphi}_{G, T, r} : \mathbb{R} \to \mathbb{R}$ and $\mathcal{IR}^{\varphi,\psi}_{G, T, r} : \mathbb{R} \to \mathbb{R}$ defined in the following way:

$$\mathcal{IR}^{\psi,\varphi}_{G, T, r}(x) = \min \{\varphi(T, r^-) : r^- \in \Rule(G, T^C, r)^-, \psi(T, r^-) \leq x\},$$

$$\mathcal{IR}^{\varphi,\psi}_{G, T, r}(x) = \min \{\psi(T, r^-) : r^- \in \Rule(G, T^C, r)^-, \varphi(T, r^-) \leq x\}.$$
where $a_1 < \ldots < a_k$ and $b_1 > \ldots > b_k$. By Lemma 58 and Remark 59, for any $x \in \mathbb{R}$,

$$
\mathcal{I} \mathcal{R}_{G,T,r}^{\psi,\varphi}(x) = \begin{cases} 
\text{undefined,} & x < a_1 \\
b_1, & a_1 \leq x < a_2 \\
\ldots & \ldots \\
b_{k-1}, & a_{k-1} \leq x < a_k \\
b_k, & a_k \leq x 
\end{cases}
$$

$$
\mathcal{I} \mathcal{R}_{G,T,r}^{\varphi,\psi}(x) = \begin{cases} 
\text{undefined,} & x < b_k \\
a_k, & b_k \leq x < b_{k-1} \\
\ldots & \ldots \\
a_2, & b_2 \leq x < b_1 \\
a_1, & b_1 \leq x 
\end{cases}
$$

7.3.2 Pareto Optimal Points for Systems of Inhibitory Rules: Cost vs Cost

Let $T$ be a nondegenerate decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$. $T^C$ be the decision table complementary to $T$, $U$ be an uncertainty measure, $W$ be a completeness measure, $U$ and $W$ are dual, $\alpha \in \mathbb{R}_+$, $\psi, \varphi \in \{l, -c, -rc, mc, rmc\}$, and $f, g \in \{\text{sum}(x, y), \text{max}(x, y)\}$.

Let $G = (G_1, \ldots, G_{N(T)})$ be an $N(T)$-tuple of proper subgraphs of the graph $\Delta_{U,\alpha}(T^C)$. Let $G = \Delta_{U,\alpha}(T^C)$ and $\xi \in \{l, -c, -rc, mc, rmc\} \setminus \{\psi, \varphi\}$. Then, as we mentioned, two interesting examples of such $N(T)$-tuples are $(G, \ldots, G)$ and $(G^\xi(r_1), \ldots, G^\xi(r_{N(T)}))$.

We denote by $S(G,T^C)$ the set

$$
\text{Rule}(G_1, T^C, r_1) \times \ldots \times \text{Rule}(G_{N(T)}, T^C, r_{N(T)})
$$
of \((U, \alpha)\)-systems of decision rules for \(T^C\). It is clear that \(\psi_f\) and \(\varphi_g\) are strictly increasing cost functions for systems of decision rules.

In Section 7.1.3, we described the algorithm \(A_{8}\) which constructs the set of Pareto optimal points for the set of pairs

\[
p_{\psi, \varphi}^{f, g}(G, T^C) = \{(\psi_f(T^C, S), \varphi_g(T^C, S)) : S \in \mathcal{S}(G, T^C)\}.
\]

Let \(S = (\rho_1, \ldots, \rho_{N(T)}) \in \mathcal{S}(G, T^C)\). We denote \(S^\sim = (\rho_{1}^\sim, \ldots, \rho_{N(T)}^\sim)\). From Proposition 61 it follows that \(S^\sim\) is a \((W, \alpha)\)-system of inhibitory rules for \(T\), and \((\psi_f(T^C, S), \varphi_g(T^C, S)) = (\psi_f(T, S^\sim), \varphi_g(T, S^\sim))\).

We denote by \(\mathcal{S}(G, T^C)^\sim\) the set

\[
\text{Rule}(G_1, T^C, r_1)^\sim \times \ldots \times \text{Rule}(G_{N(T)}, T^C, r_{N(T)})^\sim.
\]

One can show that \(\mathcal{S}(G, T^C)^\sim = \{S^\sim : S \in \mathcal{S}(G, T^C)\}\). Let

\[
\text{ip}^{f, g}_{\psi, \varphi}(G, T) = \{(\psi_f(T, S^\sim), \varphi_g(T, S^\sim)) : S^\sim \in \mathcal{S}(G, T^C)^\sim\}.
\]

It is clear that \(p_{\psi, \varphi}^{f, g}(G, T^C) = \text{ip}^{f, g}_{\psi, \varphi}(G, T)\) and \(\text{Par}(p_{\psi, \varphi}^{f, g}(G, T^C)) = \text{Par}(\text{ip}^{f, g}_{\psi, \varphi}(G, T))\).

From results obtained in Section 6.2 it follows that

\[
\text{Rule}(G_i, T^C, r_i)^\sim \subseteq IR_{W, \alpha}(T, r_i).
\]

Let \(G = \Delta_{W, \alpha}(T^C)\). If, for \(i = 1, \ldots, N(T)\), \(G_i = G\) then \(\text{Rule}(G_i, T^C, r_i)^\sim = IR_{W, \alpha}(T, r_i)\). If \(\xi \in \{l, -c, -rc, mc, rmc\}\) and \(G_i = G^\xi(r_i)\) for \(i = 1, \ldots, N(T)\), then \(\text{Rule}(G_i, T^C, r_i)^\sim\) is equal to the set of all rules from \(IR_{W, \alpha}(T, r_i)\) which have minimum cost relative to \(\xi\) among all inhibitory rules from the set \(IR_{W, \alpha}(T, r_i)\).

To study relationships between cost functions \(\psi_f\) and \(\varphi_g\) on the set of systems of
inhibitory rules \(S(G, T^C)^-\), we consider two partial functions \(\mathcal{IR}_{G,T}^{\psi, f, \psi, g} : \mathbb{R} \rightarrow \mathbb{R}\) and \(\mathcal{IR}_{G,T}^{\varphi, g, \psi, f} : \mathbb{R} \rightarrow \mathbb{R}\) defined in the following way:

\[
\begin{align*}
\mathcal{IR}_{G,T}^{\psi, f, \psi, g}(x) &= \min\{\varphi_g(T, S^-) : S^- \in S(G, T^C)^-, \psi_f(T, S^-) \leq x\}, \\
\mathcal{IR}_{G,T}^{\varphi, g, \psi, f}(x) &= \min\{\psi_f(T, S^-) : S^- \in S(G, T^C)^-, \varphi_g(T, S^-) \leq x\}.
\end{align*}
\]

Let \((a_1, b_1), \ldots, (a_k, b_k)\) be the normal representation of the set

\[
Par(p_{\psi, \varphi}^{f,g}(G, T^C)) = Par(ip_{\psi, \varphi}^{f,g}(G, T))
\]

where \(a_1 < \ldots < a_k\) and \(b_1 > \ldots > b_k\). By Lemma 58 and Remark 59, for any \(x \in \mathbb{R}\),

\[
\mathcal{IR}_{G,T}^{\psi, f, \psi, g}(x) = \begin{cases} 
\text{undefined,} & x < a_1 \\
b_1, & a_1 \leq x < a_2 \\
\vdots & \vdots \\
b_{k-1}, & a_{k-1} \leq x < a_k \\
b_k, & a_k \leq x
\end{cases}
\]

\[
\mathcal{IR}_{G,T}^{\varphi, g, \psi, f}(x) = \begin{cases} 
\text{undefined,} & x < b_k \\
a_k, & b_k \leq x < b_{k-1} \\
\vdots & \vdots \\
a_2, & b_2 \leq x < b_1 \\
a_1, & b_1 \leq x
\end{cases}
\]

### 7.4 Experimental Results for Pareto Optimal Points for Systems of Inhibitory Rules

To work with inhibitory rules, we transform each table \(T\) in Table 7.1 to a table \(T'\) by removing rows labeled with \(D(T)\). Then, we convert \(T'\) to the corresponding
Table 7.3: Number of Pareto optimal points for decision tables complementary to $T'$

<table>
<thead>
<tr>
<th>Table Name</th>
<th># POPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>136</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>8</td>
</tr>
<tr>
<td>cars-1</td>
<td>1</td>
</tr>
<tr>
<td>mushroom-5</td>
<td>3154</td>
</tr>
<tr>
<td>nursery-1</td>
<td>1</td>
</tr>
<tr>
<td>nursery-4,</td>
<td>1</td>
</tr>
<tr>
<td>teeth-1</td>
<td>1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>1</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>1</td>
</tr>
</tbody>
</table>

complementary decision table $T'^C$ by changing, for each row $r \in Row(T')$, the set $D(r)$ with the set $D(T') \setminus D(r)$. Results obtained for decision rules for $T'^C$ are in the same time results for inhibitory rules for $T'$.

Table 7.3 presents the number of Pareto optimal points constructed by the algorithm $A_8$ for each complementary table $T'^C$ in the same framework as it was described in Section 7.2.2. Figure 7.4 depicts the unique Pareto optimal point and greedy heuristic solutions for the decision table complementary to “zoo-data-5” and 13 heuristics described in Section 7.2.1. We compare these heuristics as algorithms for construction of inhibitory rules for tables $T'$.

Note that some results about comparison of heuristics for single-criterion optimization of inhibitory rules over decision tables with single-valued decisions can be found in [54].

Table 7.4 presents three ways of comparing 13 greedy heuristics across all 9 decision tables $T'^C$ complementary to $T'$ and in the same way across all 9 tables $T'$. The first two ways are based on comparing the heuristics using ranking for averaged length or coverage of the obtained inhibitory rule systems. In case of equal values for some rule heuristics, we break ties by averaging their ranks. One can see that “poly.C” and “log.C” perform well for coverage compared to the other greedy heuristics. Similarly, “M.C” and “M.L” are the best heuristics for length, however, they
Table 7.4: Comparing heuristics on complementary decision tables $T^{cC}$ and on $T'$

<table>
<thead>
<tr>
<th>Heuristic Name</th>
<th>Average Rank Cov</th>
<th>Len</th>
<th>Bi-criteria</th>
<th>Overall Rank Cov</th>
<th>Len</th>
<th>Bi-criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>poly$_C$</td>
<td>1.94</td>
<td>7.44</td>
<td>4.17</td>
<td>1</td>
<td>8.5</td>
<td>2</td>
</tr>
<tr>
<td>poly$_L$</td>
<td>4.61</td>
<td>7.44</td>
<td>6.61</td>
<td>3</td>
<td>8.5</td>
<td>6</td>
</tr>
<tr>
<td>log$_C$</td>
<td>3.06</td>
<td>5.94</td>
<td>1.94</td>
<td>2</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>log$_L$</td>
<td>6.28</td>
<td>5.94</td>
<td>4.94</td>
<td>6</td>
<td>3.5</td>
<td>3</td>
</tr>
<tr>
<td>maxCov$_C$</td>
<td>5.72</td>
<td>10.67</td>
<td>7.06</td>
<td>4</td>
<td>12.5</td>
<td>7.5</td>
</tr>
<tr>
<td>maxCov$_L$</td>
<td>7.17</td>
<td>10.67</td>
<td>8.50</td>
<td>7</td>
<td>12.5</td>
<td>9</td>
</tr>
<tr>
<td>M$_C$</td>
<td>7.83</td>
<td>4.83</td>
<td>6.50</td>
<td>8.5</td>
<td>1.5</td>
<td>5</td>
</tr>
<tr>
<td>M$_L$</td>
<td>10.33</td>
<td>4.83</td>
<td>9.00</td>
<td>12</td>
<td>1.5</td>
<td>10</td>
</tr>
<tr>
<td>RM$_C$</td>
<td>5.78</td>
<td>6.00</td>
<td>5.00</td>
<td>5</td>
<td>5.5</td>
<td>4</td>
</tr>
<tr>
<td>RM$_L$</td>
<td>7.83</td>
<td>6.00</td>
<td>7.86</td>
<td>8.5</td>
<td>5.5</td>
<td>7.5</td>
</tr>
<tr>
<td>ABS</td>
<td>9.5</td>
<td>7.50</td>
<td>10.06</td>
<td>10</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>me</td>
<td>11.33</td>
<td>6.11</td>
<td>10.00</td>
<td>13</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>mep</td>
<td>9.61</td>
<td>7.61</td>
<td>10.17</td>
<td>11</td>
<td>11</td>
<td>13</td>
</tr>
</tbody>
</table>

do not give good results for coverage. The last way of comparing greedy heuristics is based on bi-criteria optimizations for length and coverage where the computed distances between greedy solutions and nearest Pareto optimal point were ranked. The heuristic algorithm that performed best for bi-criteria optimization is “log$_C$”.

Figure 7.4: Pareto optimal point and rule heuristic solutions for the decision table complementary to zoo-data-5'
Chapter 8

Bi-Criteria Optimization Problem for Rules and Rule Systems: Cost vs Uncertainty (Completeness)

In this chapter, we consider algorithms which construct the sets of Pareto optimal points for bi-criteria optimization problems for decision (inhibitory) rules and rule systems relative a cost function and an uncertainty (completeness) measure supported by experimental results. We also show how the constructed set of Pareto optimal points can be transformed into the graphs of functions which describe the relationships between the considered cost function and uncertainty (completeness) measure.

8.1 Bi-Criteria Optimization Problem for Decision Rules and Systems of Rules: Cost vs Uncertainty

In this section, we study bi-criteria cost vs uncertainty optimization problem for decision rules and systems of decision rules, and consider experimental results.
8.1.1 Pareto Optimal Points for Decision Rules: Cost vs Uncertainty

Let \( \psi \) be a cost function for decision rules given by pair of functions \( \psi^0, F, U \) be an uncertainty measure, \( T \) be a nondegenerate decision table with \( n \) conditional attributes \( f_1, \ldots, f_n \), and \( r = (b_1, \ldots, b_n) \) be a row of \( T \).

Let \( \Theta \) be a node of the graph \( \Delta(T) = \Delta_{U,0}(T) \) containing \( r \). Let us recall that a rule over \( \Theta \) is called a \( U \)-decision rule for \( \Theta \) and \( r \) if there exists a nonnegative real number \( \alpha \) such that the considered rule is a \( (U, \alpha) \)-decision rule for \( \Theta \) and \( r \). Let \( p_{\psi, U}(\Theta, r) = \{((\psi(\Theta, \rho), U(\Theta, \rho)) : \rho \in DR_U(\Theta, r) \) where \( DR_U(\Theta, r) \) is the set of \( U \)-decision rules for \( \Theta \) and \( r \). Our aim is to find the set \( Par(p_{\psi, U}(T, r)) \). In fact, we will find the set \( Par(p_{\psi, U}(\Theta, r)) \) for each node \( \Theta \) of the graph \( \Delta(T) \) containing \( r \).

To this end we consider an auxiliary set of rules \( Path(\Theta, r) \) for each node \( \Theta \) of \( \Delta(T) \) containing \( r \). Let \( \tau \) be a directed path from the node \( \Theta \) to a node \( \Theta' \) of \( \Delta(T) \) containing \( r \) in which edges are labeled with pairs \( (f_i, b_i) \). We denote by \( rule(\tau) \) the decision rule over \( \Theta \)

\[
f_{i_1} = b_{i_1} \land \ldots \land f_{i_m} = b_{i_m} \rightarrow mcd(\Theta').
\]

If \( m = 0 \) (if \( \Theta = \Theta' \)) then the rule \( rule(\tau) \) is equal to \( \rightarrow mcd(\Theta) \). We denote by \( Path(\Theta, r) \) the set of rules \( rule(\tau) \) corresponding to all directed paths \( \tau \) from \( \Theta \) to a node \( \Theta' \) of \( \Delta(T) \) containing \( r \) (we consider also the case when \( \Theta = \Theta' \)). We correspond to the set of rules \( Path(\Theta, r) \) the set of pairs \( p_{\psi, U}^{\text{path}}(\Theta, r) = \{((\psi(\Theta, \rho), U(\Theta, \rho)) : \rho \in Path(\Theta, r) \} \).

**Lemma 77** Let \( U \) be an uncertainty measure, \( T \) be a nondegenerate decision table, \( r \) be a row of \( T \), and \( \Theta \) be a node of \( \Delta(T) \) containing \( r \). Then \( DR_U(\Theta, r) \subseteq Path(\Theta, r) \).

**Proof.** Let \( \alpha \in \mathbb{R}_+ \). Then either \( \Delta_{U,\alpha}(T) = \Delta_{U,0}(T) = \Delta(T) \) or the graph \( \Delta_{U,\alpha}(T) \) is
obtained from the graph $\Delta(T)$ by removal some nodes and edges. From here and from the definition of the set of rules $\text{Rule}(\Delta_{U, \alpha}(T), \Theta, r)$ it follows that this set is a subset of the set $\text{Path}(\Theta, r)$. From Proposition 63 it follows that the set $\text{Rule}(\Delta_{U, \alpha}(T), \Theta, r)$ coincides with the set of all $(U, \alpha)$-decision rules for $\Theta$ and $r$. Since $\alpha$ is an arbitrary nonnegative real number, we have $\text{DR}_U(\Theta, r) \subseteq \text{Path}(\Theta, r)$. ■

Lemma 78 Let $\psi \in \{l, -c, -c_M\}$, $U$ be an uncertainty measure, $T$ be a nondegenerate decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r = (b_1, \ldots, b_n)$ be a row of $T$, and $\Theta$ be a node of $\Delta(T)$ containing $r$. Then $\text{Par}(p_{\psi, U}^\text{path}(\Theta, r)) = \text{Par}(p_{\psi, U}(\Theta, r))$.

Proof. From Lemma 77 it follows that $p_{\psi, U}(\Theta, r) \subseteq p_{\psi, U}^\text{path}(\Theta, r)$. Let us show that, for any pair $\alpha \in p_{\psi, U}^\text{path}(\Theta, r)$ there exists a pair $\beta \in p_{\psi, U}(\Theta, r)$ such that $\beta \leq \alpha$. Let $\alpha = (\psi(\Theta, \rho), U(\Theta, \rho))$ where $\rho \in \text{Path}(\Theta, r)$. If $\rho \in \text{DR}_U(\Theta, r)$ then $\alpha \in p_{\psi, U}(\Theta, r)$, and the considered statement holds.

Let $\rho \notin \text{DR}_U(\Theta, r)$ and $\rho = \text{rule}(\tau)$ where $\tau$ is a directed path from $\Theta$ to a node $\Theta'$ containing $r$ which edges are labeled with pairs $(f_{i_1}, b_{i_1}), \ldots, (f_{i_m}, b_{i_m})$. Then $\rho = \text{rule}(\tau)$ is equal to

$$f_{i_1} = b_{i_1} \land \ldots \land f_{i_m} = b_{i_m} \rightarrow \text{mcd}(\Theta').$$

Since $\rho \notin \text{DR}_U(\Theta, r)$, $m > 0$. Let

$$\Theta^0 = \Theta, \Theta^1 = \Theta(f_{i_1}, b_{i_1}), \ldots, \Theta^m = \Theta(f_{i_1}, b_{i_1}), \ldots, (f_{i_m}, b_{i_m}) = \Theta'.$$

By definition of the graph $\Delta(T)$, $f_{ij} \in E(\Theta^{j-1})$ for $j = 1, \ldots, m$. Since $\rho \notin \text{DR}_U(\Theta, r)$, there exists $j \in \{1, \ldots, m - 1\}$ such that $U(\Theta^j) \leq U(\Theta^m)$. Let $j_0$ be the minimum number from $\{1, \ldots, m - 1\}$ for which $U(\Theta^{j_0}) \leq U(\Theta^m)$. We denote by $\rho'$ the rule

$$f_{i_1} = b_{i_1} \land \ldots \land f_{j_0} = b_{j_0} \rightarrow \text{mcd}(\Theta^{j_0}).$$
It is clear that $\rho'$ is a $(U, U(\Theta^0))$-decision rule for $\Theta$ and $r$. Therefore $\rho' \in DR_U(\Theta, r)$ and $\beta = ((\psi(\Theta, \rho')), U(\Theta, \rho')) \in p_{\psi, U}(\Theta, r)$. We have $U(\Theta, \rho') = U(\Theta^0) \leq U(\Theta^m) = U(\Theta, \rho)$. If $\psi = l$ then $\psi(\Theta, \rho') = j_0 < m = \psi(\Theta, \rho)$. Let $\psi = -c$. Then

$$\psi(\Theta, \rho') = -N_{med(\Theta^0)}(\Theta^0) \leq -N_{med(\Theta^m)}(\Theta^m) \leq -N_{med(\Theta^m)}(\Theta^m) = \psi(\Theta, \rho).$$

Let $\psi = -c_M$. Then $\psi(\Theta, \rho') = -N^M(\Theta^0) \leq -N^M(\Theta^m) = \psi(\Theta, \rho)$. Therefore $\beta \leq \alpha$.

Using Lemma 48 we obtain $Par(p_{\psi, U}^path(\Theta, r)) = Par(p_{\psi, U}(\Theta, r))$. ■

So in some cases, when $\psi \in \{l, -c, -c_M\}$, we have

$$Par(p_{\psi, U}(T, r)) = Par(p_{\psi, U}^path(T, r)).$$

In these cases we can concentrate on the construction of the set $Par(p_{\psi, U}^path(T, r))$.

Let $\psi$ be a strictly increasing cost function for decision rules given by pair of functions $\psi^0, F$, $U$ be an uncertainty measure, $T$ be a decision table with $n$ conditional attributes $f_1, \ldots, f_n$, and $r = (b_1, \ldots, b_n)$ be a row of $T$. We now describe an algorithm $A_9$ which constructs the set $Par(p_{\psi, U}^path(T, r))$. In fact, this algorithm constructs, for each node $\Theta$ of the graph $G$, the set $B(\Theta, r) = Par(p_{\psi, U}^path(\Theta, r))$.

**Algorithm $A_9$.**

**Input:** Strictly increasing cost function $\psi$ for decision rules given by pair of functions $\psi^0, F$, an uncertainty measure $U$, a nonempty decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$, a row $r = (b_1, \ldots, b_n)$ of $T$, and the graph $\Delta(T)$.

**Output:** The set $Par(p_{\psi, U}^path(T, r))$ of Pareto optimal points for the set of pairs $p_{\psi, U}^path(T, r) = \{((\psi(T, r), U(T, r)) : r \in Path(T, r))\}$.

1. If all nodes in $\Delta(T)$ containing $r$ are processed, then return the set $B(T, r)$.

   Otherwise, choose in the graph $\Delta(T)$ a node $\Theta$ containing $r$ which is not pro-
cessed yet and which is either a terminal node of $\Delta(T)$ or a nonterminal node of $\Delta(T)$ such that, for any $f_i \in E(\Theta)$, the node $\Theta(f_i, b_i)$ is already processed, i.e., the set $B(\Theta(f_i, b_i), r)$ is already constructed.

2. If $\Theta$ is a terminal node, then set $B(\Theta, r) = \{((\psi^0(\Theta), 0))\}$. Mark the node $\Theta$ as processed and proceed to step 1.

3. If $\Theta$ is a nonterminal node then construct $(\psi^0(\Theta), U(\Theta))$, for each $f_i \in E(\Theta)$, construct the set $B(\Theta(f_i, b_i), r)^{FH}$, where $H(x) = x$, and construct the multiset $A(\Theta, r) = \{((\psi^0(\Theta), U(\Theta))) \cup \bigcup_{f_i \in E(\Theta)} B(\Theta(f_i, b_i), r)^{FH}\}$ by simple transcription of elements from the sets $B(\Theta(f_i, b_i), r)^{FH}$, $f_i \in E(\Theta)$, and $(\psi^0(\Theta), U(\Theta))$.

4. Apply to the multiset $A(\Theta, r)$ the algorithm $A_2$ which constructs the set $Par(A(\Theta, r))$. Set $B(\Theta, r) = Par(A(\Theta, r))$. Mark the node $\Theta$ as processed and proceed to step 1.

**Proposition 79** Let $\psi$ be strictly increasing cost function for decision rules given by pair of functions $\psi^0, F, U$ be an uncertainty measure, $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, and $r = (b_1, \ldots, b_n)$ be a row of $T$. Then, for each node $\Theta$ of the graph $\Delta(T)$ containing $r$, the algorithm $A_9$ constructs the set $B(\Theta, r) = Par(p^\text{path}_{\psi, U}(\Theta, r))$.

**Proof.** We prove the considered statement by induction on nodes of $G$. Let $\Theta$ be a terminal node of $\Delta(T)$ containing $r$. Then $U(\Theta) = 0$, $Path(\Theta, r) = \{\rightarrow \text{mcd}(\Theta)\}$, $p^\text{path}_{\psi, U}(\Theta, r) = Par(p^\text{path}_{\psi, U}(\Theta, r)) = \{(\psi^0(\Theta), 0)\}$, and $B(\Theta, r) = Par(p^\text{path}_{\psi, U}(\Theta, r))$.

Let $\Theta$ be a nonterminal node of $\Delta(T)$ containing $r$ such that, for any $f_i \in E(\Theta)$, the considered statement holds for the node $\Theta(f_i, b_i)$, i.e., $B(\Theta(f_i, b_i), r) = Par(p^\text{path}_{\psi, U}(\Theta(f_i, b_i), r))$. 

$B(\Theta(f_i, b_i), r) = Par(p^\text{path}_{\psi, U}(\Theta(f_i, b_i), r))$. 

$B(\Theta, r) = \{((\psi^0(\Theta), 0))\}$.
One can show that

\[ p_{\psi, \varphi}^{\text{path}}(\Theta, r) = \{(\psi^0(\Theta), U(\Theta))\} \cup \bigcup_{f_i \in E(\Theta)} p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r)^{FH} \]

where \( H \) is a function from \( \mathbb{R} \) to \( \mathbb{R} \) such that \( H(x) = x \). It is clear that

\[ \text{Par}(\{(\psi^0(\Theta), U(\Theta))\}) = \{(\psi^0(\Theta), U(\Theta))\}. \]

From Lemma 53 it follows that

\[ \text{Par}(p_{\psi, U}^{\text{path}}(\Theta, r)) \subseteq \{(\psi^0(\Theta), U(\Theta))\} \cup \bigcup_{f_i \in E(\Theta)} \text{Par}(p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r)^{FH}). \]

By Lemma 57, \( \text{Par}(p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r)^{FH}) = \text{Par}(p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r))^{FH} \) for any \( f_i \in E(\Theta) \).

Therefore

\[ \text{Par}(p_{\psi, U}^{\text{path}}(\Theta, r)) \subseteq \{(\psi^0(\Theta), U(\Theta))\} \cup \bigcup_{f_i \in E(\Theta)} \text{Par}(p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r))^{FH} \subseteq p_{\psi, U}^{\text{path}}(\Theta, r). \]

Using Lemma 52 we obtain

\[ \text{Par}(p_{\psi, U}^{\text{path}}(\Theta, r)) = \text{Par} \left( \{(\psi^0(\Theta), U(\Theta))\} \cup \bigcup_{f_i \in E(\Theta)} \text{Par}(p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r))^{FH} \right). \]

Since \( B(\Theta, r) = \text{Par} \left( \{(\psi^0(\Theta), U(\Theta))\} \cup \bigcup_{f_i \in E(\Theta)} B(\Theta(f_i, b_i), r)^{FH} \right) \) and

\[ B(\Theta(f_i, b_i), r) = \text{Par}(p_{\psi, U}^{\text{path}}(\Theta(f_i, b_i), r)) \]
for any $f_i \in E(\Theta)$, we have $B(\Theta, r) = Par(p_{\psi,U}^{\text{path}}(\Theta, r))$. ■

We now evaluate the number of elementary operations (computations of $F$, $H$, $\psi^0$, $U$, and comparisons) made by the algorithm $A_9$. Let us recall that, for a given cost function $\psi$ for decision rules and decision table $T$,

$$q_\psi(T) = |\{\psi(\Theta, \rho) : \Theta \in SEP(T), \rho \in DR(\Theta)\}|.$$

In particular, by Lemma 60, $q_l(T) \leq n + 1$, $q_{-c}(T) \leq N(T) + 1$, $q_{-rc}(T) \leq N(T)(N(T) + 1)$, $q_{mc}(T) \leq N(T) + 1$, $q_{-cm}(T) \leq N(T) + 1$, and $q_{rmc}(T) \leq N(T)(N(T) + 1)$.

**Proposition 80** Let $\psi$ be strictly increasing cost function for decision rules given by pair of functions $\psi^0, F, H$ be a function from $\mathbb{R}$ to $\mathbb{R}$ such that $H(x) = x$, $U$ be an uncertainty measure, $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$, and $r = (b_1, \ldots, b_n)$ be a row of $T$. Then, to construct the set $Par(p_{\psi,U}^{\text{path}}(\Theta, r))$, the algorithm $A_9$ makes

$$O(L(\Delta(T))q_\psi(T)n \log(q_\psi(T)n))$$

elementary operations (computations of $F$, $H$, $\psi^0$, $U$, and comparisons).

**Proof.** To process a terminal node, the algorithm $A_9$ makes one elementary operation – computes $\psi^0$. We now evaluate the number of elementary operations under the processing of a nonterminal node $\Theta$.

From Proposition 79 it follows that $B(\Theta, r) = Par(p_{\psi,U}^{\text{path}}(\Theta, r))$ and

$$B(\Theta(f_i, b_i), r) = Par(p_{\psi,U}^{\text{path}}(\Theta(f_i, b_i), r))$$

for any $f_i \in E(\Theta)$. From Lemma 49 it follows that $|B(\Theta(f_i, b_i), r)| \leq q_\psi(T)$ for any
\[ f_i \in E(\Theta). \text{ It is clear that } |E(\Theta)| \leq n, \]

\[ |B(\Theta(f_i, b_i), r)^{FH}| = |B(\Theta(\Theta(f_i, b_i), r)| \]

for any \( f_i \in E(\Theta), \) and \( |A(\Theta, r)| \leq q(\psi(T)n). \) Therefore to construct the sets

\[ B(\Theta(f_i, b_i), r)^{FH}, f_i \in E(\Theta), \]

from the sets \( B(\Theta(f_i, b_i), r), f_i \in E(\Theta), \) the algorithm \( A_9 \) makes \( O(q(\psi(T)n) \) computations of \( F \) and \( H. \) To construct the pair \( (\psi^0(\Theta), U(\Theta)), \)

the algorithm \( A_9 \) makes two operations – computes \( \psi^0 \) and \( U. \) To construct the set

\[ Par(A(\Theta, r)) = B(\Theta, r) \]

from the set \( A(\Theta, r), \) the algorithm \( A_9 \) makes

\[ O(q(\psi(T)n \log(q(\psi(T)n))) \) comparisons (see Proposition 50). Hence, to process a non-

terminal node \( \Theta, \) the algorithm makes

\[ O(q(\psi(T)n \log(q(\psi(T)n))) \]

elementary operations.

To construct the set \( Par(p^\text{path}_{\psi, U}(T, r)), \) the algorithm \( A_9 \) makes

\[ O(L(\Delta(T))q(\psi(T)n \log(q(\psi(T)n))) \]

elementary operations (computations of \( F, H, \psi^0, U, \) and comparisons). ■

**Proposition 81** Let \( \psi \) be strictly increasing cost function for decision rules given by pair of functions \( \psi^0, F, H \) be a function from \( \mathbb{R} \) to \( \mathbb{R} \) such that \( H(x) = x, \) \( U \) be an uncertainty measure, \( \psi \in \{l, -c, -rc, -c_M, mc, rmc\}, U \in \{me, rme, abs\}, \) and \( \mathcal{U} \) be a restricted information system. Then the algorithm \( A_9 \) has polynomial time complexity for decision tables from \( \mathcal{T}(\mathcal{U}) \) depending on the number of conditional attributes in these tables.
Proof. Since $\psi \in \{l, -c, -rc, -c_M, mc, rmc\}$,

$$\psi^0 \in \{0, -N_{med}(T), -N_{med}(T)/N(T), -N^M(T), N(T) - N_{med}(T)(T), (N(T) - N_{med}(T)(T))/N(T)\},$$

and $F, H \in \{x, x + 1\}$. From Lemma 60 and Proposition 80 it follows that, for the algorithm $A_9$, the number of elementary operations (computations of $F$, $H$, $\psi^0$, $U$, and comparisons) is bounded from above by a polynomial depending on the size of input table $T$ and on the number of separable subtables of $T$. All operations with numbers are basic ones. The computations of numerical parameters of decision tables used by the algorithm $A_9 (0, -N_{med}(T)(T), -N_{med}(T)(T)/N(T), -N^M(T), N(T) - N_{med}(T)(T), (N(T) - N_{med}(T)(T))/N(T)$ and $U \in \{me, rme, abs\}$) have polynomial time complexity depending on the size of decision tables.

According to Proposition 46, the algorithm $A_9$ has polynomial time complexity for decision tables from $T(U)$ depending on the number of conditional attributes in these tables. □

8.1.2 Relationships for Decision Rules: Cost vs Uncertainty

Let $\psi \in \{l, -c, -c_M\}$. Then the set $Par(p_{\psi,U}^\text{path}(T, r))$ constructed by the algorithm $A_9$ is equal to the set $Par(p_{\psi,U}(T, r))$. Using this set we can construct two partial functions $R_{T,r}^{\psi,U} : \mathbb{R} \rightarrow \mathbb{R}$ and $R_{T,r}^{U,\psi} : \mathbb{R} \rightarrow \mathbb{R}$ which describe relationships between cost function $\psi$ and uncertainty measure $U$ on the set $DR_U(T, r)$ of $U$-decision rules for $T$ and $r$, and are defined in the following way:

$$R_{T,r}^{\psi,U}(x) = \min\{U(T, \rho) : \rho \in DR_U(T, r), \psi(T, \rho) \leq x\},$$

$$R_{T,r}^{U,\psi}(x) = \min\{\psi(T, \rho) : \rho \in DR_U(T, r), U(T, \rho) \leq x\}.$$
Let $p_{\psi,U}(T,r) = \{(\psi(T,\rho),U(T,\rho)) : \rho \in DR_U(T,r)\}$ and $(a_1, b_1), \ldots, (a_k, b_k)$ be the normal representation of the set $Par(p_{\psi,U}(T,r))$ where $a_1 < \ldots < a_k$ and $b_1 > \ldots > b_k$. By Lemma 58 and Remark 59, for any $x \in \mathbb{R}$,

$$
\mathcal{R}_{T,r}^{\psi,U}(x) = \begin{cases}
\text{undefined,} & x < a_1 \\
 b_1, & a_1 \leq x < a_2 \\
 \vdots & \vdots \\
 b_{k-1}, & a_{k-1} \leq x < a_k \\
 b_k, & a_k \leq x
\end{cases}
$$

and

$$
\mathcal{R}_{T,r}^{U,\psi}(x) = \begin{cases}
\text{undefined,} & x < b_k \\
 a_k, & b_k \leq x < b_{k-1} \\
 \vdots & \vdots \\
 a_2, & b_2 \leq x < b_1 \\
 a_1, & b_1 \leq x
\end{cases}
$$

### 8.1.3 Pareto Optimal Points for Systems of Decision Rules: Cost vs Uncertainty

Let $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, and $U$ be an uncertainty measure.

We denote by $\mathcal{I}_U(T)$ the set $DR_U(T,r_1) \times \ldots \times DR_U(T,r_{N(T)})$ of $U$-systems of decision rules for $T$. Let $\psi$ be a strictly increasing cost function for decision rules given by pair of functions $\psi^0$, $F$, and $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$. We consider two parameters of decision rule systems: cost $\psi_f(T,S)$ and uncertainty $U_g(T,S)$ which are defined on pairs $T, S$ where $T \in \mathcal{T}^+$ and $S \in \mathcal{I}_U(T)$. Let $S = (\rho_1, \ldots, \rho_{N(T)})$. Then $\psi_f(T,S) = f(\psi(T,\rho_1), \ldots, \psi(T,\rho_{N(T)}))$ and $U_g(T,S) = g(U(T,\rho_1), \ldots, U(T,\rho_{N(T)}))$ where $f(x_1) = x_1$, $g(x_1) = x_1$, and, for $k > 2$, $f(x_1, \ldots, x_k) = f(f(x_1, \ldots, x_{k-1}), x_k)$ and $g(x_1, \ldots, x_k) = g(g(x_1, \ldots, x_{k-1}), x_k)$. 
We assume that $\psi \in \{l, -c, -c_M\}$. According to Lemma 78, in this case $\text{Par}(p_{\psi, U}^{\text{path}}(T, r_i)) = \text{Par}(p_{\psi, U}(T, r_i))$ for $i = 1, \ldots, N(T)$. We describe now an algorithm which constructs the set of Pareto optimal points for the set of pairs $p_{\psi, U}^{f, g}(T) = \{(\psi_f(T, S), U_g(T, S)) : S \in \mathcal{I}_U(T)\}$.

Algorithm $\mathcal{A}_{10}$.

\textbf{Input}: Strictly increasing cost function for decision rules $\psi$ given by pair of functions $\psi^0, F$, $\psi \in \{l, -c, -c_M\}$, an uncertainty measure $U$, increasing functions $f, g$ from $\mathbb{R}^2$ to $\mathbb{R}$, a nonempty decision table $T$ with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, and the graph $\Delta(T)$.

\textbf{Output}: The set $\text{Par}(p_{\psi, U}^{f, g}(T))$ of Pareto optimal points for the set of pairs $p_{\psi, U}^{f, g}(T) = \{(\psi_f(T, S), U_g(T, S)) : S \in \mathcal{I}_U(T)\}$.

1. Using the algorithm $\mathcal{A}_{9}$ construct, for $i = 1, \ldots, N(T)$, the set $\text{Par}(P_i)$ where

$$P_i = p_{\psi, U}^{\text{path}}(T, r_i) = \{((\psi(T, \rho), U(T, \rho)) : \rho \in \text{Path}(T, r_i)\}.$$

2. Apply the algorithm $\mathcal{A}_3$ to the functions $f, g$ and the sets $\text{Par}(P_1), \ldots, \text{Par}(P_{N(T)})$. Set $C(T)$ the output of the algorithm $\mathcal{A}_3$ and return it.

\textbf{Proposition 82} Let $\psi$ be a strictly increasing cost function for decision rules given by pair of functions $\psi^0, F$, $\psi \in \{l, -c, -c_M\}$, $U$ be an uncertainty measure, $f, g$ be increasing functions from $\mathbb{R}^2$ to $\mathbb{R}$, and $T$ be a nonempty decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$. Then the algorithm $\mathcal{A}_{10}$ constructs the set $C(T) = \text{Par}(p_{\psi, U}^{f, g}(T))$.

\textbf{Proof}. For $i = 1, \ldots, N(T)$, denote $P_i = p_{\psi, U}^{\text{path}}(T, r_i)$ and $R_i = p_{\psi, U}(T, r_i)$. During the first step, the algorithm $\mathcal{A}_{10}$ constructs (using the algorithm $\mathcal{A}_9$) the sets $\text{Par}(P_1), \ldots, \text{Par}(P_{N(T)})$ (see Proposition 71). From Lemma 78 it follows that
\[ \text{Par}(P_1) = \text{Par}(R_1), \ldots, \text{Par}(P_{N(T)}) = \text{Par}(R_{N(T)}). \]

During the second step of the algorithm \( \mathcal{A}_{10} \) we apply the algorithm \( \mathcal{A}_3 \) to the functions \( f, g \) and the sets \( \text{Par}(R_1), \ldots, \text{Par}(R_{N(T)}) \). The algorithm \( \mathcal{A}_3 \) constructs the set \( C(T) = \text{Par}(Q_{N(T)}) \) where \( Q_1 = R_1 \), and, for \( i = 2, \ldots, N(T), Q_i = Q_{i-1} \{fg\} R_i \) (see Proposition 55). One can show that \( Q_{N(T)} = p_{f,g}^{l,U}(T) \). Therefore \( C(T) = \text{Par}(p_{f,g}^{l,U}(T)) \).

Let us recall that, for a given cost function \( \psi \) and a decision table \( T \), \( q_\psi(T) = |\{\psi(\Theta, \rho) : \Theta \in \text{SEP}(T), \rho \in \text{DR}(\Theta)\}|. \) In particular, by Lemma 60, \( q_l(T) \leq n + 1, q_{-c}(T) \leq N(T) + 1, \) and \( q_{-c_M}(T) \leq N(T) + 1. \)

Let us recall also that, for a given cost function \( \psi \) for decision rules and a decision table \( T \), \( \text{Range}_\psi(T) = \{\psi(\Theta, \rho) : \Theta \in \text{SEP}(T), \rho \in \text{DR}(\Theta)\} \). By Lemma 60, \( \text{Range}_l(T) \subseteq \{0, 1, \ldots, n\}, \text{Range}_{-c}(T) \subseteq \{0, -1, \ldots, -N(T)\}, \) and \( \text{Range}_{-c_M}(T) \subseteq \{0, -1, \ldots, -N(T)\}. \) Let \( t_l(T) = n, t_{-c}(T) = N(T), \) and \( t_{-c_M}(T) = N(T). \)

**Proposition 83** Let \( \psi \) be a strictly increasing cost function for decision rules given by pair of functions \( \psi^0, F, \psi \in \{l, -c, -c_M\}, U \) be an uncertainty measure, \( f, g \) be increasing functions from \( \mathbb{R}^2 \) to \( \mathbb{R} \), \( F \in \{x + y, \max(x, y)\}, H \) be a function from \( \mathbb{R} \) to \( \mathbb{R} \) such that \( H(x) = x \), and \( T \) be a nonempty decision table with \( n \) conditional attributes \( f_1, \ldots, f_n \) and \( N(T) \) rows \( r_1, \ldots, r_{N(T)} \). Then, to construct the set \( \text{Par}(p_{f,g}^{l,U}(T)) \), the algorithm \( \mathcal{A}_{10} \) makes

\[
O(N(T)L(\Delta(T))q_\psi(T)n \log(q_\psi(T)n)) + O(N(T)t_\psi(T)^2 \log(t_\psi(T)))
\]

elementary operations (computations of \( F, H, \psi^0, U, f, g \) and comparisons) if \( f = \max(x, y), \) and

\[
O(N(T)L(\Delta(T))q_\psi(T)n \log(q_\psi(T)n)) + O(N(T)^2t_\psi(T)^2 \log(N(T)t_\psi(T)))
\]

elementary operations (computations of \( F, H, \psi^0, U, f, g \) and comparisons) if \( f = x + y. \)
Proof. For $i = 1, \ldots, N(T)$, denote $P_i = p^\text{path}_{\psi, U}(T, r)$ and $R_i = p_{\psi, U}(T, r)$. To construct the sets $\text{Par}(P_i) = \text{Par}(R_i)$, $i = 1, \ldots, N(T)$, the algorithm $A_0$ makes

$$O(N(T)L(\Delta(T))q_\psi(T)n \log(q_\psi(T)n))$$

elementary operations (computations of $F$, $H$, $\psi^0$, $U$, and comparisons) – see Proposition 80.

We now evaluate the number of elementary operations (computations of $f$, $g$, and comparisons) made by the algorithm $A_3$ during the construction of the set $C(T) = \text{Par}(p^f_{\psi, U}(T))$ from the sets $\text{Par}(R_i)$, $i = 1, \ldots, N(T)$. We know that $\psi \in \{l, -c, -c_M\}$ and $f \in \{x + y, \max(x, y)\}$.

For $i = 1, \ldots, N(T)$, let $R_i^1 = \{a : (a, b) \in R_i\}$. Since $\psi \in \{l, -c, -c_M\}$, we have $R_i^1 \subseteq \{0, 1, \ldots, t_\psi(T)\}$ for $i = 1, \ldots, N(T)$ or $R_i^1 \subseteq \{0, -1, \ldots, -t_\psi(T)\}$ for $i = 1, \ldots, N(T)$.

Using Proposition 56 we obtain the following.

If $f = x + y$, then to construct the set $C(T)$ the algorithm $A_3$ makes

$$O(N(T)^2t_\psi(T)^2 \log(N(T)t_\psi(T)))$$

elementary operations (computations of $f$, $g$, and comparisons).

If $f = \max(x, y)$, then to construct the sets $\text{Par}(Q_{N(T)})$ the algorithm $A_3$ makes

$$O(N(T)t_\psi(T)^2 \log(t_\psi(T)))$$

elementary operations (computations of $f$, $g$, and comparisons).

Proposition 84 Let $\psi$ be a strictly increasing cost function for decision rules given by pair of functions $\psi^0, F$, $\psi \in \{l, -c, -c_M\}$, $U$ be an uncertainty measure, $U \in \{\text{me, rme, abs}\}$, $f, g \in \{\max(x, y), x + y\}$, $H$ be a function from $\mathbb{R}$ to $\mathbb{R}$ such that
H(x) = x, and \( \mathcal{U} \) be a restricted information system. Then the algorithm \( \mathcal{A}_{10} \) has polynomial time complexity for decision tables from \( \mathcal{T}(\mathcal{U}) \) depending on the number of conditional attributes in these tables.

**Proof.** We have \( \psi^0 \in \{0, -N_{\text{med}(T)}(T), -N^M(T)\} \), and \( F, H \in \{x, x + 1\} \). From Lemma 60 and Proposition 83 it follows that, for the algorithm \( \mathcal{A}_{10} \), the number of elementary operations (computations of \( F, H, \psi^0, U, f, g \), and comparisons) is bounded from above by a polynomial depending on the size of input table \( T \) and on the number of separable subtables of \( T \). All operations with numbers are basic ones. The computations of numerical parameters of decision tables used by the algorithm \( \mathcal{A}_{10} \) (0, \( -N_{\text{med}(T)}(T), -N^M(T) \), and \( U \in \{\text{me, rme, abs}\} \)) have polynomial time complexity depending on the size of decision tables.

According to Proposition 46, the algorithm \( \mathcal{A}_{10} \) has polynomial time complexity for decision tables from \( \mathcal{T}(\mathcal{U}) \) depending on the number of conditional attributes in these tables. ■

### 8.1.4 Relationships for Systems of Decision Rules: Cost vs Uncertainty

Let \( \psi \) be a strictly increasing cost function for decision rules, \( \mathcal{U} \) be an uncertainty measure, \( f, g \) be increasing functions from \( \mathbb{R}^2 \) to \( \mathbb{R} \), and \( T \) be a nonempty decision table.

To study relationships between functions \( \psi_f \) and \( U_g \) which characterize cost and uncertainty of systems of rules from \( \mathcal{I}_U(T) \) we can consider two partial functions from \( \mathbb{R} \) to \( \mathbb{R} \):

\[
\mathcal{R}_T^{\psi_f, U_g}(x) = \min \{ U_g(T, S) : S \in \mathcal{I}_U(T), \psi_f(T, S) \leq x \},
\]

\[
\mathcal{R}_T^{U_g, \psi_f}(x) = \min \{ \psi_f(T, S) : S \in \mathcal{I}_U(T), U_g(T, S) \leq x \}.
\]
Let \( p_{\psi,f,g}^f(T) = \{(\psi_f(T,S),U_g(T,S)) : S \in \mathcal{I}_U(T)\} \), and \((a_1, b_1), \ldots, (a_k, b_k)\) be the normal representation of the set \( \text{Par}(p_{\psi,f,g}^f(T)) \) where \( a_1 < \ldots < a_k \) and \( b_1 > \ldots > b_k \).

By Lemma 58 and Remark 59, for any \( x \in \mathbb{R} \),

\[
\mathcal{R}_{T}^{\psi,f,U,g}(x) = \begin{cases} 
\text{undefined,} & x < a_1 \\
b_1, & a_1 \leq x < a_2 \\
\ldots & \ldots \\
b_{k-1}, & a_{k-1} \leq x < a_k \\
b_k, & a_k \leq x 
\end{cases}
\]

\[
\mathcal{R}_{T}^{U,g,\psi,f}(x) = \begin{cases} 
\text{undefined,} & x < b_k \\
a_k, & b_k \leq x < b_{k-1} \\
\ldots & \ldots \\
a_2, & b_2 \leq x < b_1 \\
a_1, & b_1 \leq x 
\end{cases}
\]

### 8.1.5 Experimental Results: Cost vs Uncertainty

In this section, we consider Pareto optimal points for bi-criteria optimization problem for decision rule systems relative to negation of coverage (\(-c\)) or length (\(l\)) and relative misclassification error (\(rme\)). For some tables \( T \) from Table 7.1, we constructed the set of Pareto optimal points by applying the algorithm \( \mathcal{A}_{10} \). Number of constructed points can be found in Tables 8.1 and 8.2. Figures 8.1 and 8.2 depict the Pareto optimal points constructed for tables zoo-data-5 and cars-1, respectively.
Table 8.1: Number of Pareto optimal points for decision tables $T$: $-c$ vs. $rme$

<table>
<thead>
<tr>
<th>Table Name $T$</th>
<th>Rows</th>
<th>Attributes</th>
<th>Removed Attributes</th>
<th># POPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>3</td>
<td>9186</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>4,5,6,8,9</td>
<td>455</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
<td>2533</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
<td>401</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
<td>1</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
<td>35</td>
</tr>
</tbody>
</table>

Table 8.2: Number of Pareto optimal points for decision tables $T$: $l$ vs. $rme$

<table>
<thead>
<tr>
<th>Table Name $T$</th>
<th>Rows</th>
<th>Attributes</th>
<th>Removed Attributes</th>
<th># POPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>193</td>
<td>8</td>
<td>3</td>
<td>547</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>98</td>
<td>4</td>
<td>4,5,6,8,9</td>
<td>169</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
<td>591</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
<td>321</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
<td>28</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
<td>89</td>
</tr>
</tbody>
</table>

Figure 8.1: Pareto optimal points for zoo-data-5: $-c$ vs $rme$
8.2 Bi-criteria Optimization of Inhibitory Rules: Cost vs Completeness

In this section, we study bi-criteria cost vs completeness optimization problem for inhibitory rules and systems of inhibitory rules, and consider experimental results.

8.2.1 Pareto Optimal Points for Inhibitory Rules: Cost vs Completeness

Let $T$ be a nondegenerate decision table with $n$ conditional attributes $f_1, \ldots, f_n$, $r$ be a row of $T$, $T^C$ be the decision table complementary to $T$, $U$ be an uncertainty measure, $W$ be a completeness measure, $U$ and $W$ are dual, and $\psi \in \{l, -c\}$. In Section 8.1.1, the algorithm $A_9$ is described which constructs the set $\text{Par}(p_{\psi,U}(T^C, r))$ of Pareto optimal points for the set of points $p_{\psi,U}(T^C, r) = \{(\psi(T^C, \rho), U(T^C, \rho)) : \rho \in DR_U(T^C, r)\}$ (see Lemma 78).

We denote $ip_{\psi,W}(T, r) = \{(\psi(T, \rho), W(T, \rho)) : \rho \in IR_W(T, r)\}$. From Corollary 62 it follows that $DR_U(T^C, r)^- = IR_W(T, r)$. From Proposition 61 it follows that,
for any \( \rho \in DR_U(T^C, r) \),

\[
(\psi(T^C, \rho), U(T^C, \rho)) = (\psi(T, \rho^-), W(T, \rho^-)).
\]

Therefore \( p_{\psi, U}(T^C, r) = ip_{\psi, W}(T, r) \). Hence

\[
Par(p_{\psi, U}(T^C, r)) = Par(ip_{\psi, W}(T, r)).
\]

To study relationships between \( \psi \) and \( W \) on the set \( IR_W(T, r) \) of inhibitory rules for \( T \) and \( r \) we will consider two partial functions \( IR_{W,T,r}^\psi : \mathbb{R} \to \mathbb{R} \) and \( IR_{W,T,r}^{W,\psi} : \mathbb{R} \to \mathbb{R} \) which are defined in the following way:

\[
IR_{T,r}^{\psi,W}(x) = \min\{W(T, \rho) : \rho \in IR_W(T, r), \psi(T, \rho) \leq x\},
\]

\[
IR_{T,r}^{W,\psi}(x) = \min\{\psi(T, \rho) : \rho \in IR_W(T, r), W(T, \rho) \leq x\}.
\]

Let \((a_1, b_1), \ldots, (a_k, b_k)\) be the normal representation of the set

\[
Par(p_{\psi, U}(T^C, r)) = Par(ip_{\psi, W}(T, r))
\]

where \( a_1 < \ldots < a_k \) and \( b_1 > \ldots > b_k \). By Lemma 58 and Remark 59, for any \( x \in \mathbb{R} \),

\[
IR_{T,r}^{\psi,W}(x) = \begin{cases} 
  \text{undefined}, & x < a_1 \\
  b_1, & a_1 \leq x < a_2 \\
  \ldots & \ldots \\
  b_{k-1}, & a_{k-1} \leq x < a_k \\
  b_k, & a_k \leq x
\end{cases}
\]
8.2.2 Pareto Optimal Points for Systems of Inhibitory Rules: Cost vs Completeness

Let $T$ be a nondegenerate decision table with $n$ conditional attributes $f_1, \ldots, f_n$ and $N(T)$ rows $r_1, \ldots, r_{N(T)}$, $T^C$ be the decision table complementary to $T$, $U$ be an uncertainty measure, $W$ be a completeness measure, $U$ and $W$ are dual, $\psi \in \{l, -c\}$, and $f, g \in \{\text{sum}(x, y), \text{max}(x, y)\}$.

We denote by $I_U(T^C)$ the set $DR_U(T^C, r_1) \times \ldots \times DR_U(T^C, r_{N(T)})$ of $U$-systems of decision rules for $T^C$, and consider two parameters of decision rule systems: cost $\psi_f(T, S)$ and uncertainty $U_g(T, S)$.

In Section 8.1.3, the algorithm $A_{10}$ is described which constructs the set of Pareto optimal points for the set of pairs $p_{\psi, U}^{f, g}(T^C) = \{(\psi_f(T^C, S), U_g(T^C, S)) : S \in I_U(T^C)\}$.

Let $S = (\rho_1, \ldots, \rho_{N(T)}) \in I_U(T^C)$. We denote $S^- = (\rho_1^-, \ldots, \rho_{N(T)}^-)$. From Proposition 61 it follows that $S^-$ is a $W$-system of inhibitory rules for $T$, and $(\psi_f(T^C, S), U_g(T^C, S)) = (\psi_f(T, S^-), W_g(T, S^-))$.

We denote by $I_U(T^C)^-$ the set $DR_U(T^C, r_1)^- \times \ldots \times DR_U(T^C, r_{N(T)})^-$. One can show that $I_U(T^C)^- = \{S^- : S \in I_U(T^C)\}$. Let

$$ip_{\psi, W}^{f, g}(T) = \{(\psi_f(T, S^-), W_g(T, S^-)) : S^- \in I_U(T^C)^-\}.$$

It is clear that $p_{\psi, U}^{f, g}(T^C) = ip_{\psi, W}^{f, g}(T)$ and $\text{Par}(p_{\psi, U}^{f, g}(T^C)) = \text{Par}(ip_{\psi, W}^{f, g}(T))$.

From Corollary 62 it follows that $IR_W(T, r_i) = DR_U(T^C, r_i)^-$ for $i = 1, \ldots, N(T)$. 

\[
\mathcal{I}R_{W, r}^{W, \psi}(x) = \begin{cases} 
\text{undefined}, & x < b_k \\
 a_k, & b_k \leq x < b_{k-1} \\
 \vdots & \vdots \\
 a_2, & b_2 \leq x < b_1 \\
 a_1, & b_1 \leq x 
\end{cases}
\]
Therefore $\mathcal{I}_U(T^C)^- = \mathcal{I}_W(T, r_1) \times \ldots \times \mathcal{I}_W(T, r_{N(T)})$.

To study relationships between functions $\psi_f$ and $W_g$ on the set $\mathcal{I}_U(T^C)^-$ of $W$-systems of inhibitory rules for $T$ we can consider two partial functions from $\mathbb{R}$ to $\mathbb{R}$:

\[
\mathcal{I} \mathcal{R}_T^{\psi, f, W, g}(x) = \min\{W_g(T, S^-) : S^- \in \mathcal{I}_U(T)^-, \psi_f(T, S^-) \leq x\},
\]

\[
\mathcal{I} \mathcal{R}_T^{W, g, \psi, f}(x) = \min\{\psi_f(T, S^-) : S^- \in \mathcal{I}_U(T)^-, W_g(T, S^-) \leq x\}.
\]

Let $(a_1, b_1), \ldots, (a_k, b_k)$ be the normal representation of the set

\[
\text{Par}(p_{\psi,T}^{f,g}(T^C)) = \text{Par}(ip_{\psi,W}^{f,g}(T))
\]

where $a_1 < \ldots < a_k$ and $b_1 > \ldots > b_k$. By Lemma 58 and Remark 59, for any $x \in \mathbb{R}$,

\[
\mathcal{I} \mathcal{R}_T^{\psi, f, W, g}(x) = \begin{cases} 
  \text{undefined}, & x < a_1 \\
  b_1, & a_1 \leq x < a_2 \\
  \ldots & \ldots \\
  b_{k-1}, & a_{k-1} \leq x < a_k \\
  b_k, & a_k \leq x 
\end{cases}
\]

\[
\mathcal{I} \mathcal{R}_T^{W, g, \psi, f}(x) = \begin{cases} 
  \text{undefined}, & x < b_k \\
  a_k, & b_k \leq x < b_{k-1} \\
  \ldots & \ldots \\
  a_2, & b_2 \leq x < b_1 \\
  a_1, & b_1 \leq x 
\end{cases}
\]
8.2.3 Experimental Results: Cost vs Completeness

In this section, we consider Pareto optimal points for bi-criteria optimization problem for inhibitory rule systems relative to negation of coverage \((-c)\) or length \((l)\) and inhibitory relative misclassification error \((irme)\). We transformed each table \(T\) from Table 7.1 to a decision table \(T'\) by removal from \(T\) rows with \(D(r) = D(T)\). We constructed for \(T'^C\), using algorithm \(A_{10}\), the set of Pareto optimal points for bi-criteria optimization problem for decision rule systems relative to \(-c\) (or \(l\)) and relative misclassification error \((rme)\). This set coincides with the set of Pareto optimal points for inhibitory rule system for \(T'\) relative to \(-c\) (or \(l\)) and \(irme\). Number of constructed points can be found in Tables 8.3 and 8.4. Figures 8.3 and 8.4 depict the Pareto optimal points for bi-criteria optimization problem \(-c\) versus \(irme\) constructed for tables zoo-data-5' and cars-1'.

**Table 8.3:** Number of Pareto optimal points for decision tables \(T'\): \(-c\) vs. \(irme\)

<table>
<thead>
<tr>
<th>Table Name (T)</th>
<th>Rows</th>
<th>Attributes</th>
<th>Removed Attributes</th>
<th># POPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>169</td>
<td>8</td>
<td>3</td>
<td>7203</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>58</td>
<td>4</td>
<td>4,5,6,8,9</td>
<td>266</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
<td>3236</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>1</td>
<td>4321</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
<td>241</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
<td>15</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
<td>53</td>
</tr>
</tbody>
</table>

**Table 8.4:** Number of Pareto optimal points for decision tables \(T'\): \(l\) vs. \(irme\)

<table>
<thead>
<tr>
<th>Table Name (T)</th>
<th>Rows</th>
<th>Attributes</th>
<th>Removed Attributes</th>
<th># POPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast-cancer-1</td>
<td>169</td>
<td>8</td>
<td>3</td>
<td>475</td>
</tr>
<tr>
<td>breast-cancer-5</td>
<td>58</td>
<td>4</td>
<td>4,5,6,8,9</td>
<td>99</td>
</tr>
<tr>
<td>cars-1</td>
<td>432</td>
<td>5</td>
<td>1</td>
<td>479</td>
</tr>
<tr>
<td>nursery-1</td>
<td>4320</td>
<td>7</td>
<td>1</td>
<td>4321</td>
</tr>
<tr>
<td>nursery-4</td>
<td>240</td>
<td>4</td>
<td>1,5,6,7</td>
<td>241</td>
</tr>
<tr>
<td>teeth-1</td>
<td>22</td>
<td>7</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>teeth-5</td>
<td>14</td>
<td>3</td>
<td>2,3,4,5,8</td>
<td>15</td>
</tr>
<tr>
<td>zoo-data-5</td>
<td>42</td>
<td>11</td>
<td>2,9,10,13,14</td>
<td>43</td>
</tr>
</tbody>
</table>
Figure 8.3: Pareto optimal points for zoo-data-5': $-c$ vs. irme

Figure 8.4: Pareto optimal points for cars-1': $-c$ vs irme
Chapter 9

Local and Global Approaches to Study of Decision and Inhibitory Rule Systems

This chapter is devoted to the study of time complexity of decision and inhibitory rule systems over arbitrary sets of attributes represented by information systems.

An information system $U = (A, B, F)$ consists of a set $A$ of objects and a set of attributes $F$ which are defined on $A$ and have values from a finite set $B$. We will assume that $F$ does not contain constant attributes. This information system is called finite if $F$ is a finite set, and infinite, otherwise.

The notion of a problem with many-valued decisions over the information system $U$ is defined as follows. Take finite number of attributes $f_1, \ldots, f_n$ from $F$. These attributes create a partition of the set $A$ into classes (for each class, values of the attributes are constant on elements from the class). A nonempty finite set of decisions is attached to each class. The number $n$ is called the dimension of the considered problem.

There are two interpretations of each problem: decision and inhibitory. For the decision interpretation, for a given element $a$ from $A$, it is required to recognize a
decision from the set attached to the class which contains \( a \). For the inhibitory interpretation, for a given element \( a \) from \( A \), it is required to recognize a decision which does not belong to the set attached to the class which contains \( a \).

As algorithms for problem solving decision rule systems are considered for decision interpretation, and inhibitory rule systems are considered for inhibitory interpretation. The length of a rule system (the maximum length of a rule from the system) is considered as time complexity measure.

We consider two approaches to the study of decision and inhibitory rule systems: local and global. Local approach assumes that rules can use only attributes \( f_1, \ldots, f_n \) from the problem description. Global approach allows us to use arbitrary attributes from the set \( F \) in rules. We use indexes \( l \) and \( g \) to distinguish local and global approaches, respectively.

For each information system and for each approach (local and global), we investigate the behavior of two Shannon functions which characterize the growth in the worst case of (i) minimum length of decision rule systems, (ii) minimum length of inhibitory rule systems with the growth of problem dimension.

### 9.1 Various Types of Shannon Functions

Let \( \omega = \{0, 1, 2, \ldots\} \) be the set of nonnegative integers including all natural numbers and the number 0, \( A \) be a nonempty set of objects, \( B \) be a finite nonempty subset of \( \omega \) containing at least two numbers, and \( F \) be a nonempty set of functions from \( A \) to \( B \). Functions from \( F \) are called attributes and the triple \( U = (A, B, F) \) is called an information system. This information system is called finite if \( F \) is a finite set, and infinite otherwise. We assume that \( f \neq \text{const} \) for any \( f \in F \).

An equation system over \( U \) is an arbitrary system of the kind

\[
\{f_1(x) = \delta_1, \ldots, f_m(x) = \delta_m\}
\]
where \( f_1, \ldots, f_m \in F \) and \( \delta_1, \ldots, \delta_m \in B \). It is possible that the considered system does not have equations. Such system is called *empty*. The set of solutions of the empty system coincides with the set \( A \). There is one-to-one correspondence between equation systems over \( U \) and words from the set \( \Omega_{F,B} = \{(f, \delta) : f \in F, \delta \in B\}^* \): the word \((f_1, \delta_1) \ldots (f_m, \delta_m)\) corresponds to the considered equation system, the empty word \( \lambda \) corresponds to the empty equation system. For any \( \alpha \in \Omega_{F,B} \), denote by \( \text{Sol}_U(\alpha) \) the set of solutions on \( A \) of the equation system corresponding to the word \( \alpha \).


A *problem over \( U \)* with single-valued decisions is an arbitrary \((n + 1)\)-tuple \( z = (\nu, f_1, \ldots, f_n) \) where \( \nu : B^n \to \omega \) and \( f_1, \ldots, f_n \in F \). The number \( \dim z = n \) is called the *dimension* of the problem \( z \). Denote \( \text{At}(z) = \{f_1, \ldots, f_n\} \). The problem \( z \) may be interpreted as a problem of searching for the value \( z(a) = \nu(f_1(a), \ldots, f_n(a)) \) for an arbitrary \( a \in A \). We say about this interpretation as about decision one. Note that one can interpret \( f_1, \ldots, f_n \) as conditional attributes and \( z \) as a decision attribute.

Different problems of pattern recognition, combinatorial optimization, fault diagnosis, and computational geometry can be represented in such form. Denote \( \text{Probl}_U \) the set of problems over \( U \) with single-valued decisions.

Decision rule systems are considered as algorithms for problem solving when we study its decision interpretation.

A *decision rule over \( U \)* is an arbitrary expression \( \rho \) of the kind

\[
f_1 = \delta_1 \land \ldots \land f_m = \delta_m \Rightarrow t
\]

where \( f_1, \ldots, f_m \in F, \delta_1, \ldots, \delta_m \in B \) and \( t \in \omega \). The number \( m \) is called the *length*
of the rule $\rho$. Denote $\text{At}(\rho) = \{f_1, \ldots, f_m\}$. Let us define a word $\pi(\rho)$ from the set $\Omega_{F,B}(\rho) = \{(f, \delta) : f \in \text{At}(\rho), \delta \in B\}$ associated with $\rho$. If $m = 0$ then $\pi(\xi) = \lambda$. Note that in this case the set $\text{Sol}_U(\pi(\rho))$ coincides with the set $A$. Let $m > 0$. Then $\pi(\rho) = (f_1, \delta_1) \cdots (f_m, \delta_m)$. Note that in this case the set $\text{Sol}_U(\pi(\rho))$ coincides with the set of solutions on $A$ of the equation system $\{f_1(a) = \delta_1, \ldots, f_m(a) = \delta_m\}$. The number $t$ is called the right-hand side of the rule $\rho$.

A decision rule system over $U$ is a nonempty finite set of decision rules over $U$. Let $\Delta$ be a decision rule system over $U$ and $z$ be a problem over $U$. Denote $\text{At}(\Delta) = \bigcup_{\rho \in \Delta} \text{At}(\rho)$. The decision rule system $\Delta$ is complete for the problem $z$ if, for any $a \in A$, there exists a rule $\rho \in \Delta$ such that $a \in \text{Sol}_U(\pi(\rho))$ and, for each rule $\rho \in \Delta$ such that $a \in \text{Sol}_U(\pi(\rho))$, the right-hand side of $\rho$ is equal to $z(a)$.

For decision rule systems, as time complexity measure the length of a decision rule system is considered which is the maximum length of a rule from the system. Denote $l(\Delta)$ the length of a decision rule system $\Delta$. For a problem $z$ over $U$, denote $l^0_U(z)$ the minimum length of a decision rule system over $U$ which is complete for the problem $z$. For a problem $z$ over $U$, denote $l^l_U(z)$ the minimum length of a decision rule system $\Delta$ over $U$ which is complete for the problem $z$ and for which $\text{At}(\Delta) \subseteq \text{At}(z)$. The considered parameters correspond to global and local approaches to the study of decision rule systems, respectively. One can show that $l^0_U(z) \leq l^l_U(z) \leq \dim z$ for each problem $z$ over $U$.

Define two functions of Shannon type $L^0_U(n)$ and $L^l_U(n)$. Let $n \in \omega \setminus \{0\}$. Then

$$L^0_U(n) = \max\{l^0_U(z) : z \in \text{Probl}_U, \dim z \leq n\},$$
$$L^l_U(n) = \max\{l^l_U(z) : z \in \text{Probl}_U, \dim z \leq n\}.$$

It is clear that $L^0_U(n) \leq L^l_U(n) \leq n$ for any $n \in \omega \setminus \{0\}$.

Let $U$ be a finite information system. A problem $z$ over $U$ is rule-stable if $l^0_U(z) = \dim z$. Denote $\text{rs}(U)$ the maximum dimension of a rule-stable problem from $\text{Probl}_U$. 

9.1.2 Shannon Functions for Decision Rule Systems. Problems with Many-valued Decisions

Let us consider an information system $U = (A, B, F)$. A problem with many-valued decisions over the information system $U$ is an arbitrary $(n+1)$-tuple $z = (\nu, f_1, \ldots, f_n)$ where $\nu : B^n \rightarrow \text{Fin}(\omega)$, $\text{Fin}(\omega)$ is the set of all nonempty finite subsets of $\omega$, and $f_1, \ldots, f_n \in F$. The number $\dim z = n$ is called the dimension of the problem $z$. Denote $\text{At}(z) = \{f_1, \ldots, f_n\}$. The problem $z$ may be interpreted as a problem of searching for a value from the set $z(a) = \nu(f_1(a), \ldots, f_n(a))$ for an arbitrary $a \in A$.

We say about this interpretation as about decision one. Denote $\text{Probl}^\infty_U$ the set of problems with many-valued decisions over $U$.

Decision rule systems are considered as algorithms for problem solving when we study its decision interpretation.

A decision rule system $\Delta$ over $U$ is complete for the problem $z \in \text{Probl}^\infty_U$ if, for any $a \in A$, there exists a rule $\rho \in \Delta$ such that $a \in \text{Sol}_U(\pi(\rho))$ and, for each rule $\rho \in \Delta$ such that $a \in \text{Sol}_U(\pi(\rho))$, the right-hand side of $\rho$ belongs to $z(a)$. Denote $l^g_U(z)$ the minimum length of a decision rule system $\Delta$ over $U$ which is complete for the problem $z$. Denote $l^l_U(z)$ the minimum length of a decision rule system $\Delta$ over $U$ which is complete for the problem $z$ and for which $\text{At}(\Delta) \subseteq \text{At}(z)$. The considered two parameters correspond to global and local approaches, respectively. One can show that $l^g_U(z) \leq l^l_U(z) \leq \dim z$.

Define two functions of Shannon type $L^g_{U,\infty}(n)$ and $L^l_{U,\infty}(n)$. Let $n \in \omega \setminus \{0\}$. Then

$$L^g_{U,\infty}(n) = \max\{l^g_U(z) : z \in \text{Probl}^\infty_U, \dim z \leq n\},$$

$$L^l_{U,\infty}(n) = \max\{l^l_U(z) : z \in \text{Probl}^\infty_U, \dim z \leq n\}.$$

It is clear that $L^g_{U,\infty}(n) \leq L^l_{U,\infty}(n) \leq n$ for any $n \in \omega \setminus \{0\}$.

Let $U$ be a finite information system. A problem $z \in \text{Probl}^\infty_U$ is rule-stable if $l^g_U(z) = \dim z$. Denote $\text{rs}^\infty(U)$ the maximum dimension of a rule-stable problem from
Proposition 85 \ Let \( U = (A, B, F) \) be an information system. Then, for any \( n \in \omega \setminus \{0\} \), \( L^0_{U, \infty}(n) = L^0_U(n) \) and \( L^1_{U, \infty}(n) = L^1_U(n) \). If \( U \) is a finite information system then \( r^\infty_s(U) = r_s(U) \). \\

**Proof.** Let \( f_1, \ldots, f_n \in F \). Denote by \( \nu_n \) a mapping of the set \( B^n \) into the set \( \omega \) such that \( \nu_n(\vec{\delta}_1) \neq \nu_n(\vec{\delta}_2) \) for any \( \vec{\delta}_1, \vec{\delta}_2 \in B^n \), \( \vec{\delta}_1 \neq \vec{\delta}_2 \). Let \( z_n = (\nu_n, f_1, \ldots, f_n) \) and \( z \) be an arbitrary problem of the kind \((\nu, f_1, \ldots, f_n)\) where \( \nu : B^n \to \omega \). One can show that \( l^1_U(z) \leq l^1_U(z_n) \) and \( l^0_U(z) \leq l^0_U(z_n) \).

Denote by \( \nu^\infty_n \) a mapping of the set \( B^n \) into the set \( \text{Fin}(\omega) \) such that \( \nu^\infty_n(\vec{\delta}_1) \cap \nu^\infty_n(\vec{\delta}_2) = \emptyset \) for any \( \vec{\delta}_1, \vec{\delta}_2 \in B^n \), \( \vec{\delta}_1 \neq \vec{\delta}_2 \). Let \( z^\infty_n = (\nu^\infty_n, f_1, \ldots, f_n) \) and \( z^\infty \) be an arbitrary problem of the kind \((\nu^\infty, f_1, \ldots, f_n)\) where \( \nu^\infty : B^n \to \text{Fin}(\omega) \). One can show that \( l^0_U(z^\infty) \leq l^0_U(z^\infty_n) \) and \( l^1_U(z^\infty) \leq l^1_U(z^\infty_n) \).

It is easy to see that \( l^0_U(z_n) = l^0_U(z^\infty_n) \) and \( l^1_U(z_n) = l^1_U(z^\infty_n) \). Using these facts it is not difficult to show that, for any \( n \in \omega \setminus \{0\} \), \( L^0_{U, \infty}(n) = L^0_U(n) \) and \( L^1_{U, \infty}(n) = L^1_U(n) \), and if \( U \) is a finite information system, then \( r^\infty_s(U) = r_s(U) \). \\

\[ \]  

9.1.3 Shannon Functions for Inhibitory Rule Systems. Problems with Many-valued Decisions

Let \( U = (A, B, F) \) be an information system and \( z = (\nu, f_1, \ldots, f_n) \in \text{Probl}^\infty_U \). Denote \( \text{Row}_U(z) \) the set of \( n \)-tuples \((\delta_1, \ldots, \delta_n) \in B^n \) such that the equation system \( \{f_1(x) = \delta_1, \ldots, f_n(x) = \delta_n\} \) has a solution from \( A \), and \( D_U(z) = \bigcup_{(\delta_1, \ldots, \delta_n) \in \text{Row}_U(z)} \nu(\delta_1, \ldots, \delta_n) \). We will say that the problem \( z \) is *inhibitory-correct* if \( D_U(z) \setminus \nu(\delta_1, \ldots, \delta_n) \neq \emptyset \) for any tuple \((\delta_1, \ldots, \delta_n) \in \text{Row}_U(z) \). Denote \( \text{IProbl}^\infty_U \) the set of inhibitory-correct problems from \( \text{Probl}^\infty_U \).

Let \( z = (\nu, f_1, \ldots, f_n) \in \text{IProbl}^\infty_U \). The problem \( z \) may be interpreted as a problem of searching for a value from the set \( D_U(z) \setminus z(a) \) for an arbitrary \( a \in A \) where
$z(a) = \nu(f_1(a), \ldots, f_n(a))$. We say about this interpretation as about inhibitory one. Inhibitory rule systems are considered as algorithms for problem solving when we study its inhibitory interpretation.

An inhibitory rule over $U$ is an arbitrary expression $\rho$ of the kind

$$f_1 \delta_1 \land \ldots \land f_m = \delta_m \Rightarrow t$$

where $f_1, \ldots, f_m \in F$, $\delta_1, \ldots, \delta_m \in B$ and $t \in \omega$. The number $m$ is called the length of the rule $\rho$. Denote $\text{At}(\rho) = \{f_1, \ldots, f_m\}$. Let us define a word $\pi(\rho)$ from the set $\Omega_{F,B}(\rho) = \{(f, \delta) : f \in \text{At}(\rho), \delta \in B\}^*$ associated with $\rho$. If $m = 0$ then $\pi(\xi) = \lambda$. Note that in this case the set $\text{Sol}_U(\pi(\rho))$ coincides with the set $A$. Let $m > 0$. Then $\pi(\rho) = (f_1, \delta_1) \ldots (f_m, \delta_m)$. Note that in this case the set $\text{Sol}_U(\pi(\rho))$ coincides with the set of solutions on $A$ of the equation system $\{f_1(a) = \delta_1, \ldots, f_m(a) = \delta_m\}$. The expression $\neq t$ is called the right-hand side of the rule $\rho$.

An inhibitory rule system over $U$ is a nonempty finite set of inhibitory rules over $U$. Let $\Delta$ be an inhibitory rule system over $U$ and $z$ be a problem from IProbl$^\infty_U$. Denote $\text{At}(\Delta) = \bigcup_{\rho \in \Delta} \text{At}(\rho)$. The inhibitory rule system $\Delta$ is complete for the problem $z$ if, for any $a \in A$, there exists a rule $\rho \in \Delta$ such that $a \in \text{Sol}_U(\pi(\rho))$ and, for each rule $\rho \in \Delta$ such that $a \in \text{Sol}_U(\pi(\rho))$, the right-hand side $\neq t$ of $\rho$ is such that $t \in D_U(z)$ and $t /\in z(a)$.

For inhibitory rule systems, as time complexity measure the length of an inhibitory rule system is considered which is the maximum length of a rule from the system. Denote $l(\Delta)$ the length of an inhibitory rule system $\Delta$. For a problem $z$ from IProbl$^\infty_U$, denote $il_U^q(z)$ the minimum length of an inhibitory rule system over $U$ which is complete for the problem $z$. For a problem $z$ from IProbl$^\infty_U$, denote $il_U^l(z)$ the minimum length of an inhibitory rule system $\Delta$ over $U$ which is complete for the problem $z$ and for which $\text{At}(\Delta) \subseteq \text{At}(z)$. The considered two parameters correspond to global and
local approaches, respectively. One can show that \( il_U^g(z) \leq il_U^l(z) \leq \dim z \) for each problem \( z \) from \( \text{IProbl}_U^\infty \).

Define two functions of Shannon type \( IL_{U,\infty}^g(n) \) and \( IL_{U,n}^l(n) \). Let \( n \in \omega \setminus \{0\} \). Then

\[
IL_{U,\infty}^g(n) = \max \{ il_U^g(z) : z \in \text{IProbl}_U^\infty, \dim z \leq n \},
\]

\[
IL_{U,n}^l(n) = \max \{ il_U^l(z) : z \in \text{IProbl}_U^\infty, \dim z \leq n \}.
\]

It is clear that \( IL_{U,\infty}^g(n) \leq IL_{U,n}^l(n) \leq n \) for any \( n \in \omega \setminus \{0\} \).

**Proposition 86** Let \( U = (A, B, F) \) be an information system. Then, for any \( n \in \omega \setminus \{0\} \), \( IL_{U,\infty}^g(n) = L_{U,\infty}^g(n) \) and \( IL_{U,n}^l(n) = L_{U,n}^l(n) \).

**Proof.** Let \( z = (\nu, f_1, \ldots, f_n) \in \text{IProbl}_U^\infty \). The problem \( z^C = (\nu^C, f_1, \ldots, f_n) \) where, for any \( (\delta_1, \ldots, \delta_n) \in B^n, \nu^C(\delta_1, \ldots, \delta_n) = D_U(z) \setminus \nu(\delta_1, \ldots, \delta_n) \) is called the complementary problem for \( z \).

Let \( \Delta \) be a decision rule system over \( U \). We denote by \( \Delta^- \) an inhibitory rule system over \( U \) obtained from \( \Delta \) by changing right-hand sides of rules: if a right-hand side of a rule from \( \Delta \) is equal to \( t \) then the right-hand side of the corresponding rule in \( \Delta^- \) is equal to \( \neq t \).

One can show that the decision rule system \( \Delta \) is complete for the problem \( z^C \) if and only if the inhibitory rule system \( \Delta^- \) is complete for the problem \( z \). From here it follows that \( il_U^g(z) = il_U^g(z^C) \) and \( il_U^l(z) = il_U^l(z^C) \).

Let \( n \in \omega \setminus \{0\} \). We now show that \( IL_{U,\infty}^g(n) = L_{U,\infty}^g(n) \). Let \( z \in \text{IProbl}_U^\infty \) and \( \dim z \leq n \). We know that \( \dim z = \dim z^C \) and \( il_U^g(z) = il_U^g(z^C) \). Therefore \( IL_{U,\infty}^g(n) \leq L_{U,\infty}^g(n) \).

Let \( z \in \text{Probl}_U^\infty \), \( \dim z \leq n \), and \( il_U^g(z) = L_{U,\infty}^g(n) \). From the proof of Proposition 85 it follows that the problem \( z \) can be chosen such that \( z = (\nu, f_1, \ldots, f_m) \) where \( m \leq n \) and \( \nu(\bar{\delta}_1) \cap \nu(\bar{\delta}_2) = \emptyset \) for any \( \bar{\delta}_1, \bar{\delta}_2 \in \text{Row}_U(z), \bar{\delta}_1 \neq \bar{\delta}_2 \). Since \( F \) does not contain constant attributes, \( |\text{Row}_U(z)| \geq 2 \). From here it follows that \( z^C \in \text{IProbl}_U^\infty \).
and \( l^0_U(z) = l^0_U(z^{CC}) \). Denote \( q = z^C \). We have \( il^0_U(q) = l^0_U(q^C) = l^0_U(z) = L^0_{U,\infty}(n) \)

Therefore \( IL^0_{U,\infty}(n) \geq L^0_{U,\infty}(n) \) and \( IL^0_{U,\infty}(n) = L^0_{U,\infty}(n) \).

We can prove in similar way that, for any \( n \in \omega \setminus \{0\} \), \( IL^1_{U,\infty}(n) = L^1_{U,\infty}(n) \). ■

9.2 Finite Information Systems. Global Approach

In this section, we study the behavior of global Shannon functions for decision rule systems for problems with single-valued decisions over finite information systems.

A problem \( z \) over \( U \) is rule-stable if \( l^0_U(z) = \dim z \). Denote \( \text{rs}(U) \) the maximum dimension of a rule-stable problem over \( U \).

Consider now two examples.

**Example 87** Let \( n \) be a natural number and \( n \geq 3 \). Consider a finite information system \( U_1 = (A_1, \{0, 1\}, F_1) \) such that \( A_1 \) is plane and \( F_1 \) is the set of attributes corresponding to \( n \) straight lines in the plane that form a convex polygon \( P \) with \( n \) sides. Each straight line divides the plane into open and closed half-planes on which the corresponding attribute has values 0 and 1, respectively. One can show that \( \text{rs}(U_1) = n \).

**Example 88** Let \( n \) be a natural number. Consider a finite information system \( U_2 = (A_2, \{0, 1\}, F_2) \) where the set \( A_2 \) contains \( 2^n \) elements and \( F_2 \) contains all possible nonconstant attributes defined on \( A_2 \) and with values from \( \{0, 1\} \). One can show that \( \text{rs}(U_2) = 1 \).

Consider a function of Shannon type \( L^0_U(n) \). Let \( n \in \omega \setminus \{0\} \). Then

\[
L^0_U(n) = \max \{l^0_U(z) : z \in \text{Probl}_U, \dim z \leq n \}.
\]
The aim is to study the behavior of function $L_U^g(n)$ for an arbitrary finite information system $U$.

**Theorem 89** Let $U = (A,B,F)$ be a finite information system. Then, for any $n \in \omega \setminus \{0\}$, the following statements hold:

a) if $n \leq \text{rs}(U)$ then $L_U^g(n) = n$;

b) if $n \geq \text{rs}(U)$ then $L_U^g(n) = \text{rs}(U)$.

**Proof.** a). Let $n \leq \text{rs}(U)$, and $z = (\nu,f_1,\ldots,f_{\text{rs}(U)})$ be a problem over $U$ such that $l_U^g(z) = \text{rs}(U)$. Consider the problem $z' = (\nu_n,f_1,\ldots,f_n)$ over $U$. Assume that $l_U^g(z') < n$. One can show that in this case there exists a decision rule system $\Delta'$ over $U$ such that $\Delta'$ is complete for the problem $z'$, $l(\Delta') < n$, and $\text{Sol}_U(\pi(\rho)) \neq \emptyset$ for any rule $\rho \in \Delta'$. It is clear that, for any rule $\rho$ from $\Delta'$, the solution of $z'$ from the right-hand side of $\rho$ allows one to restore the values of the attributes $f_1,\ldots,f_n$. Therefore, by adding different groups of conditions of the kind $f_{n+1} = \delta_{n+1},\ldots,f_{\text{rs}(U)} = \delta_{\text{rs}(U)}$, where $\delta_{n+1},\ldots,\delta_{\text{rs}(U)} \in B$, to the left-hand sides of rules and changing the right-hand sides, we can transform $\Delta'$ into a decision rule system $\Delta$ over $U$ which is complete for the problem $z$ and for which $l(\Delta) < n + \text{rs}(U) - n = \text{rs}(U)$. In this case $l_U^g(z) < \text{rs}(U)$ which is impossible. Therefore $l_U^g(z') = n$ and $L_U^g(n) \geq n$. It is clear that $L_U^g(n) \leq n$.

Thus, $L_U^g(n) = n$.

b). Let us prove by induction on $n$ that, for any $n \geq \text{rs}(U)$, $L_U^g(n) = \text{rs}(U)$. It is clear that $L_U^g(\text{rs}(U)) = \text{rs}(U)$. Let us assume that, for some $n \geq \text{rs}(U)$, $L_U^g(m) = \text{rs}(U)$ for any $m$, $\text{rs}(U) \leq m \leq n$. Let us show that $L_U^g(n + 1) = \text{rs}(U)$. Since $n + 1 > \text{rs}(U)$, $L_U^g(n + 1) < n + 1$. Let $z$ be a problem over $U$ such that $\dim z \leq n + 1$.

One can show that there exists a decision rule system $\Delta'$ over $U$ such that $\Delta'$ is complete for the problem $z$, $l(\Delta') \leq n$, and $\text{Sol}_U(\pi(\rho)) \neq \emptyset$ for any rule $\rho \in \Delta'$. Treat each rule $\rho \in \Delta'$ in the following way. If $l(\rho) \leq \text{rs}(U)$, keep the rule $\rho$ untouched. Let $\text{rs}(U) < l(\rho) \leq n$ and $\rho$ be equal to $f_1 = \delta_1 \land \ldots \land f_{l(\rho)} = \delta_{l(\rho)} \Rightarrow \sigma$. Consider the
problem \( z_\rho = (\nu_\rho, f_1, \ldots, f_{l(\rho)}) \) where, for any \((b_1, \ldots, b_{l(\rho)}) \in B^{l(\rho)}, \nu_\rho(b_1, \ldots, b_{l(\rho)}) = \sigma \) if \((b_1, \ldots, b_{l(\rho)}) = (\delta_1, \ldots, \delta_{l(\rho)}) \) and \( \nu_\rho(b_1, \ldots, b_{l(\rho)}) = \sigma + 1 \), otherwise. According to the induction hypothesis, there exists a decision rule system \( \Delta_\rho \) over \( U \) such that \( \Delta_\rho \) is complete for the problem \( z_\rho \) and \( l(\Delta_\rho) \leq \text{rs}(U) \). Remove \( \rho \) from the rule system \( \Delta' \) and add to \( \Delta' \) all rules from \( \Delta_\rho \) that have \( \sigma \) on the right-hand side. As a result, obtain a rule system \( \Delta \) over \( U \). One can prove that \( \Delta \) is complete for the problem \( z \) and \( l(\Delta) \leq \text{rs}(U) \). Therefore, \( L^0_U(z) \leq \text{rs}(U) \). Since \( z \) is an arbitrary problem over \( U \) such that \( \dim z \leq n + 1 \), \( L^0_U(n + 1) \leq \text{rs}(U) \). It is clear that \( L^0_U(n + 1) \geq \text{rs}(U) \). Therefore \( L^0_U(n + 1) \geq \text{rs}(U) \). Thus \( L^0_U(n) = \text{rs}(U) \) for any \( n \geq \text{rs}(U) \). 

9.3 Behavior of Shannon Functions

For reader’s convenience we will repeat some definitions.

9.3.1 Local Approach. Infinite Information Systems

We will say that an information system \( U = (A, B, F) \) satisfies the condition of reduction if there exists a number \( m \in \omega \setminus \{0\} \) such that, for each compatible on \( A \) system of equations

\[ \{ f_1(x) = \delta_1, \ldots, f_r(x) = \delta_r \} \]

where \( r \in \omega \setminus \{0\}, f_1, \ldots, f_r \in F \) and \( \delta_1, \ldots, \delta_r \in B \), there exists a subsystem of this system which has the same set of solutions and contains at most \( m \) equations.

In [21], the behavior of Shannon function \( L_U^1(n) \) was studied for infinite information systems. The next theorem follows from the obtained results (see Theorem 8.2 from [21]) and Propositions 85 and 86.

**Theorem 90** Let \( U = (A, B, F) \) be an infinite information system. Then the following statements hold:
a) if $U$ satisfies the condition of reduction then $IL_{U,\infty}^1(n) = L_{U,\infty}^1(n) = O(1)$; 

b) if $U$ does not satisfy the condition of reduction then 

$$IL_{U,\infty}^1(n) = L_{U,\infty}^1(n) = n$$

for each $n \in \omega \setminus \{0\}$.

Now we extend Example 8.3 from [21] to the case of problems with many-valued decisions.

**Example 91** Let $m, t \in \omega \setminus \{0\}$. We denote by $Pol(m)$ the set of all polynomials which have integer coefficients and depend on variables $x_1, \ldots, x_m$. We denote by $Pol(m, t)$ the set of all polynomials from $Pol(m)$ such that the degree of each polynomial is at most $t$. We define information systems $U(m)$ and $U(m, t)$ as follows: $U(m) = (\mathbb{R}^m, E, F(m))$ and $U(m, t) = (\mathbb{R}^m, E, F(m, t))$ where $E = \{-1, 0, +1\}$, $F(m) = \{\text{sign}(p) : p \in Pol(m), \text{sign}(p) \neq \text{const}\}$ and $F(m, t) = \{\text{sign}(p) : p \in Pol(m, t), \text{sign}(p) \neq \text{const}\}$. Here $\text{sign}(x) = -1$ if $x < 0$, $\text{sign}(x) = 0$ if $x = 0$, and $\text{sign}(x) = +1$ if $x > 0$. One can prove that $IL_{U(m),\infty}^1(n) = L_{U(m),\infty}^1(n) = n$ for each $n \in \omega \setminus \{0\}$, $IL_{U(1,1),\infty}^1(n) = L_{U(1,1),\infty}^1(n) = O(1)$, and if $m > 1$ or $t > 1$ then

$$IL_{U(m,t),\infty}^1(n) = L_{U(m,t),\infty}^1(n) = n$$

for each $n \in \omega \setminus \{0\}$.

### 9.3.2 Local Approach. Finite Information Systems

Let $U = (A, B, F)$ be a finite information system.

A system of equations over $U$ is an arbitrary system

$$\{f_1(x) = \delta_1, \ldots, f_n(x) = \delta_n\} \quad (9.1)$$
such that $n \in \omega \setminus \{0\}$, $f_1, \ldots, f_n \in F$ and $\delta_1, \ldots, \delta_n \in B$. The system (9.1) will be called cancelable if $n \geq 2$ and there exists a number $i \in \{1, \ldots, n\}$ such that the system

$$\{f_1(x) = \delta_1, \ldots, f_{i-1}(x) = \delta_{i-1}, f_{i+1}(x) = \delta_{i+1}, \ldots, f_n(x) = \delta_n\}$$

has the same set of solutions just as the system (9.1). If the system (9.1) is not cancelable then it will be called uncancelable. We denote by $\text{un}(U)$ the maximum number of equations in an uncancelable compatible system over $U$.

In [55], the behavior of Shannon function $L_{U,\infty}^L(n)$ for finite information systems was studied. The next theorem follows from the obtained results (see Theorem 4 from [55]) and Proposition 86.

**Theorem 92** Let $U = (A, B, F)$ be a finite information system, and $n \in \omega \setminus \{0\}$.

Then the following statements hold:

a) if $n \leq \text{un}(U)$ then $IL_{U,\infty}^L(n) = L_{U,\infty}^L(n) = n$;

b) if $n \geq \text{un}(U)$ then $IL_{U,\infty}^L(n) = L_{U,\infty}^L(n) = \text{un}(U)$.

We now consider Example 8.7 from [21].

**Example 93** Denote by $P$ the set of all points in the plane. Consider an arbitrary straight line $l$, which divides the plane into positive and negative open half-planes, and the line $l$ itself. Assign a function $f : P \to \{0,1\}$ to the line $l$. The function $f$ takes the value 1 if a point is situated on the positive half-plane, and $f$ takes the value 0 if a point is situated on the negative half-plane or on the line $l$. Denote by $F$ the set of functions which correspond to certain $r$ mutually disjoint finite classes of parallel straight lines. Consider a finite information system $U = (P, \{0,1\}, F)$. One can show that $\text{un}(U) \leq 2r$. 
9.3.3 Global Approach. Infinite Information Systems

In [56], the behavior of Shannon function $L^9_U(n)$ for infinite information systems was studied. The next theorem follows from the obtained results (see Lemma 3.7 from [56]) and Propositions 85 and 86.

**Theorem 94** Let $U = (A, B, F)$ be an infinite information system. Then either $IL^9_{U,\infty}(n) = L^9_{U,\infty}(n) = O(1)$ or $IL^9_{U,\infty}(n) = L^9_{U,\infty}(n) = n$ for any $n \in \omega \setminus \{0\}$.

9.3.4 Global Approach. Finite Information Systems

Let $U = (A, B, F)$ be a finite information system. A problem $z$ from $\text{Probl}_U^\infty$ is rule-stable if $l^9_U(z) = \dim z$. Denote $rs^\infty(U)$ the maximum dimension of a rule-stable problem from $\text{Probl}_U^\infty$.

Examples 87 and 88 show different behavior of parameter $rs^\infty(U)$. Note that, according to Proposition 85, $rs^\infty(U) = rs(U)$.

The next Theorem follows from Theorem 89, and Propositions 85 and 86.

**Theorem 95** Let $U = (A, B, F)$ be a finite information system. Then, for any $n \in \omega \setminus \{0\}$, the following statements hold:

a) if $n \leq rs^\infty(U)$ then $IL^9_{U,\infty}(n) = L^9_{U,\infty}(n) = n$;

b) if $n \geq rs^\infty(U)$ then $IL^9_{U,\infty}(n) = L^9_{U,\infty}(n) = rs^\infty(U)$. 
Chapter 10

Concluding Remarks

Rules are one of the traditional and practical methods used by data analysis community. This work was devoted mainly to extensions of dynamic programming for design and analysis of decision and inhibitory rules and rule systems for decision tables with many-valued decisions. These extensions allowed us to make sequential optimization of exact and approximate rules relative to different cost functions, to count the number of optimal rules, to construct the set of Pareto optimal points for bi-criteria optimization problems, and to study relationships between two cost functions and between cost and accuracy for rules and rule systems. Two applications included in this thesis which are comparison of heuristics for bi-criteria optimization of rules, and construction of relatively small systems of rules for knowledge representation purposes. This thesis also studied behavior of four types of Shannon functions which characterize the growth in the worst case of minimum length of complete decision and inhibitory rules systems with the growth of number of attributes in problem description.

The obtained results can be useful for knowledge representation, for studying the trade-offs between complexity and accuracy for systems of decision and inhibitory rules, and , we believe, for construction of rule classifiers. The most serious restriction of this work is connected with scalability of dynamic programming algorithms, where it is required to devise more efficient algorithms.
Future directions of research include (i) studying of more applications of the obtained results to classification and knowledge representation problems, (ii) extensions of obtained results to various types of multi-label data sets including hierarchical multi-label data sets, and (iii) attempts to improve the scalability of dynamic programming algorithms.
REFERENCES


APPENDICES

A Published Papers


- Fawaz Alsolami, Igor Chikalov, Mikhail Moshkov, Beata Zielosko, “Optimization of Approximate Inhibitory Rules Relative to Number of Misclassifications”, 17th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2013

- Fawaz Alsolami, Igor Chikalov, Mikhail Moshkov, Beata Zielosko, “Sequential Optimization of Approximate Inhibitory Rules Relative to the Length, Coverage and Number of Misclassifications”, 8th International Conference on Rough Sets and Knowledge Technology (RSKT), 2013

- Fawaz Alsolami, Igor Chikalov, Mikhail Moshkov, “Comparison of Heuristics for Inhibitory Rule Optimization”, 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, 2014


B  Papers Submitted and Under Preparation


- Fawaz Alsolami, Talha Amin, Mikhail Moshkov, Igor Chikalov, “Optimization of Decision Rules for Knowledge Discovery and Representation”, *Submitted to Information Sciences*.

- Fawaz Alsolami, Talha Amin, Igor Chikalov, Mikhail Moshkov, Beata Zielosko, “Dynamic Programming Approach for Construction of Association Rule Systems”, *Submitted to Fundamenta Informaticae*.

- Fawaz Alsolami, Talha Amin, Igor Chikalov, Mikhail Moshkov, Beata Zielosko, “Comparison of Heuristics for Optimization of Association Rules”, *Submitted to Fundamenta Informaticae*.